

Introduction to Network Analysis, Spring 2019

Seminar 2 Answers

Dmitry Zaytsev, PhD, Valentina Kuskova, PhD

(with deep appreciation to all used sources; references available in text and upon request)

This document contains answers to Seminar 2.

Please note: For some interpretive answers (where I ask you to provide an opinion of what a result may represent) my answer may differ from yours, and as we progress through the course, the differences in opinions might become more pronounced. Therefore, keep in mind the following:

1. I will grade your understanding of network methods and interpretation of network results, *not* your theoretical reasoning. If an answer seems logical and reasonable, I will give you full credit even if your answer differs from mine.
2. Please do not argue with me over *my* substantive interpretations. I am not a sociologist and provide an opinion based on common sense, not deep knowledge of theoretical material. My answers provide *one* of the ways to add theoretical spin to numerical results, nothing more.
3. Do pay attention at the amount of effort I put into my answers. Unless I explain a network or other mathematical concept, which you may not otherwise know, the amount of work I put into this assignment represents, roughly, the amount I expect from you. So if I write three paragraphs to answer an interpretative question, one sentence from you for the same question will not be, most likely, sufficient. Therefore, I will not reduce points for differences in opinion, but I will reduce points for differences in effort.

Onto the answers. The task was to reproduce the R code in this document and answer the questions that are marked as *Assignment questions*. My answers are included right after each question. Please note that I reproduce only the code that is essential for answering questions; all other code is excluded.

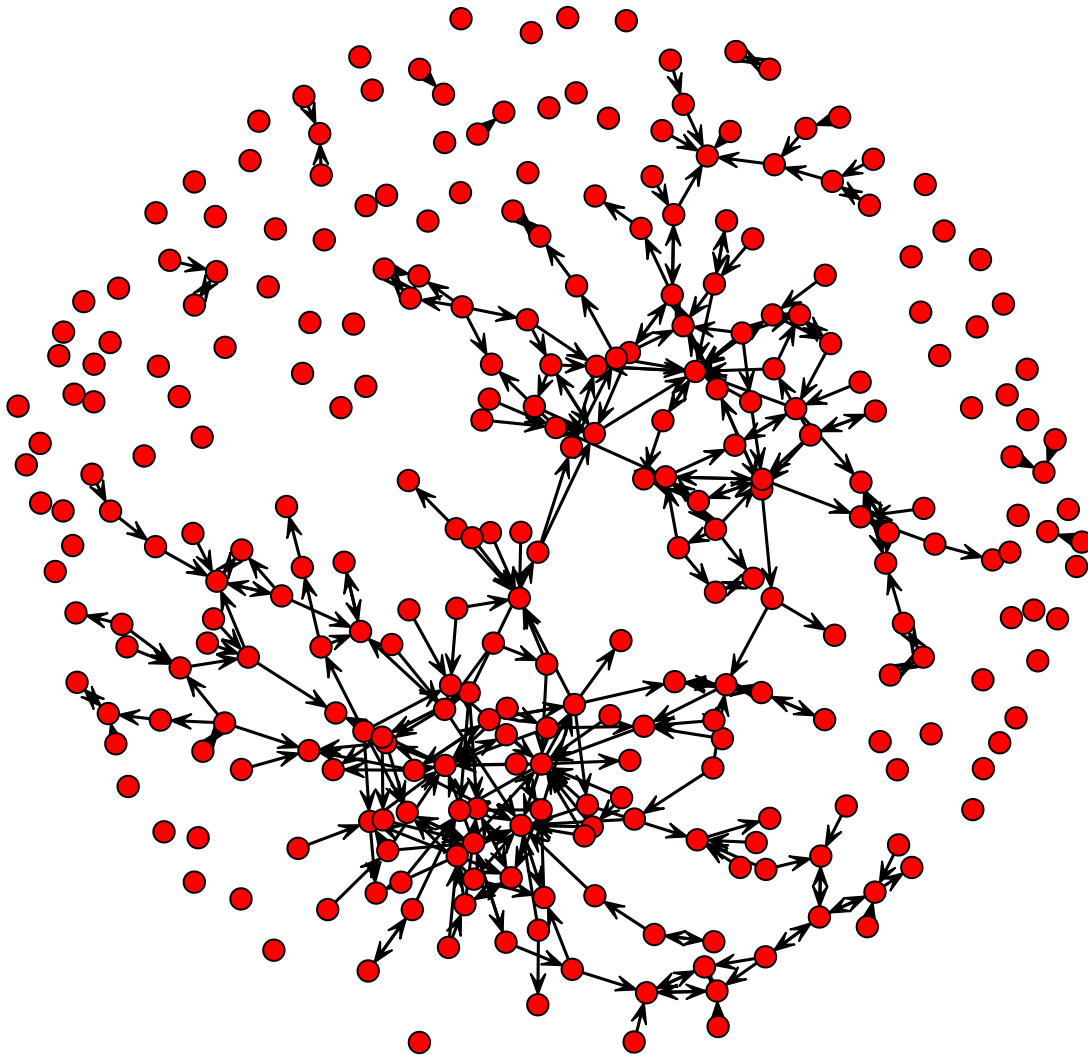
```
# I load the package here so I have it for all commands.
```

```
library(network)
```

```
## network: Classes for Relational Data
## Version 1.13.0.1 created on 2015-08-31.
## copyright (c) 2005, Carter T. Butts, University of California-Irvine
##                               Mark S. Handcock, University of California -- Los Angeles
##                               David R. Hunter, Penn State University
##                               Martina Morris, University of Washington
##                               Skye Bender-deMoll, University of Washington
## For citation information, type citation("network").
## Type help("network-package") to get started.
```

Answers in section "Reading a Pajek file"

```
drugpaj <- read.paj('drugnet2.paj')
drug <- drugpaj$networks[[1]] # extract network
par(mar=c(0,0,0,0))
plot(drug)
```



Assignment question: what looks strange about this network? Why?

Answer:

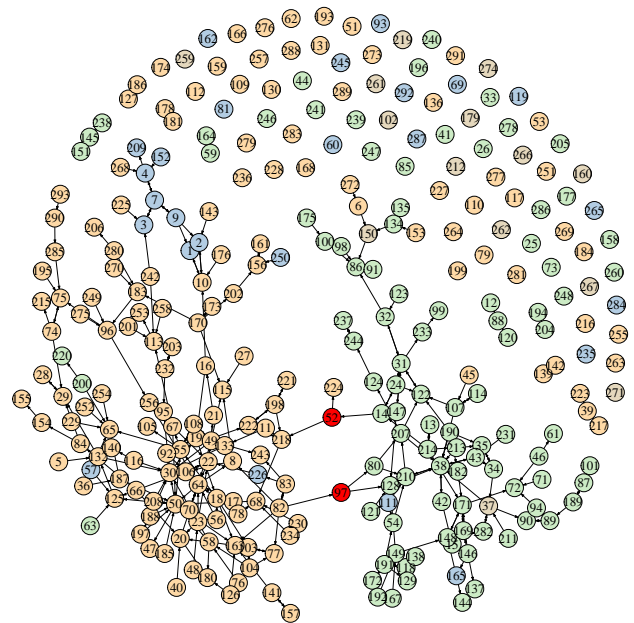
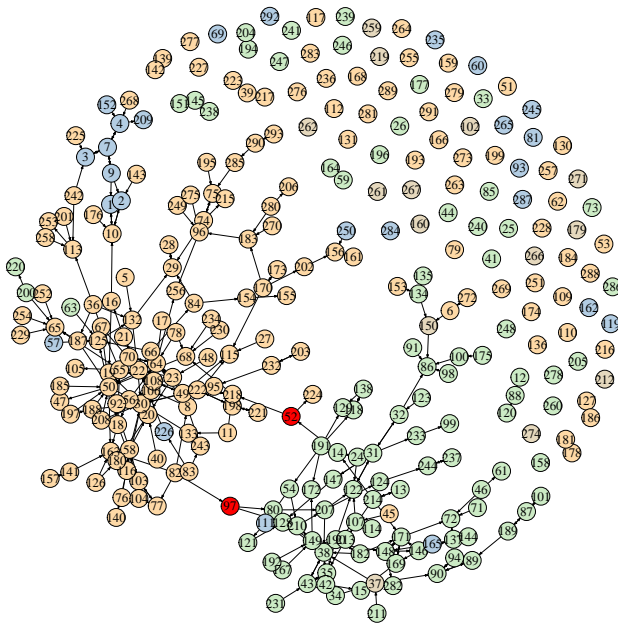
1. The immediately noticeable feature of this network is its low density. There are a lot of disconnected nodes and small disconnected components. This network structure is uncommon in social settings, but given the understanding of what this network represents, it is even more alarming: this is a network of drug users. Individual users and small groups can't be disconnected, unless they make their own narcotics (which is highly unlikely). Therefore, the following seems possible:
 - We are missing data. There may be key nodes (such as suppliers), who were not named, because drug users did not want to turn in their sources of supply. Why is it bad? Not only we can't fully evaluate the network we are interested in from the theoretical perspective, but even the "sub-network" we have here is unlikely to have accurate information.
 - However, there is a giant *connected* component. In fact, other than a few dyads and triads, there are no

other components on the network - we have one giant component and the rest are individuals or triads, at a max. What does that mean? It could mean a number of different things, but it is reasonable to assume that this plot shows a well-organized and well-running operation, reaching out to many network members.

- Also, it appears that the giant component consists of two denser components, connected by only a few links - in fact, the smallest number of links you have to remove to disconnect the graph is two links. It is not possible to disconnect the graph by removing just one, but it may be possible to generate another pair that cuts the graph in a different spot.

I have identified the two people who connect the subcomponents together - they are, most likely, dealers. The graph can be cut at any point; why have I chosen these two? Look carefully at nodes 52 and 98 - I have drawn the image a few different ways so you can see it better - but if you cut ties between these nodes and other large components, you will create two separate large, disconnected components. These two components are unique, because they mostly contain people of the same ethnicity. So while you may cut the graph differently by disconnecting at different nodes, here it makes the most sense theoretically - by cutting lines *to* these two nodes, you disconnect the component by ethnicity.

As you know, you may not be able to generate the same image, so I made a couple and include them here.



You can further examine the network:

```
network.size(drug) # how many nodes?
```

```
## [1] 293
```

```
network.edgcount(drug) # how many edges?
```

```
## [1] 337
```

```
network.dyadcount(drug) # how many dyads?
```

```
## [1] 85556
```

Assignment question: What do the numbers above represent?

Answer:

I did tell you in the comments that we are finding the number of nodes (the size of our full network), the number of edges (the number of available arcs, and the number of possible dyads (remember, if we have n nodes, we will have $n(n - 1)$ possible dyads, which is exactly 85,556 for 293 nodes). It is obvious that these numbers indicate a very low density network, even if we do not ask the program to calculate it. Now that we know what network density is, we can calculate it by hand:

```
337/85556
```

```
## [1] 0.003938941
```

Well, if anything, it confirms our original intuition that the network is very disconnected. With a density of less than half a percent, there is no doubt. However, density alone as a measure could be misleading. Remember, that we have a giant connected component inside this network, and this is a drug network - should one of the users become infected with, for example, AIDS, a lot of people will be infected.

Answers in section "Reading a native R data file"

Here, I leave only the most relevant code that can help me complete the assignments.

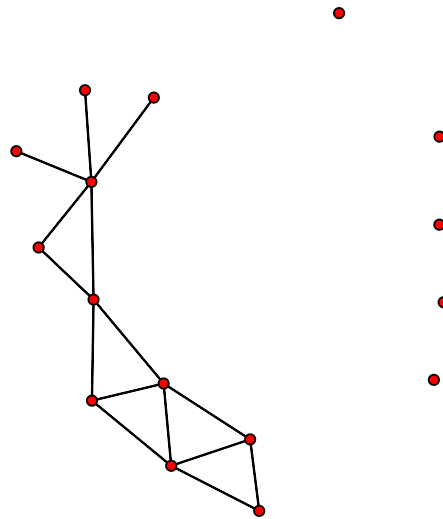
```
load('flo.Rdata')
flo.marriage<-as.network(flo.marriage, directed=FALSE)
flo.biz <- as.network(flo.biz, directed=FALSE)
# Add attributes
set.vertex.attribute(flo.marriage, 'wealth', flo.att[,2])
set.vertex.attribute(flo.biz, 'wealth', flo.att[,2])
```

Assignment task: Please plot the biz network with node attributes that you've set above.

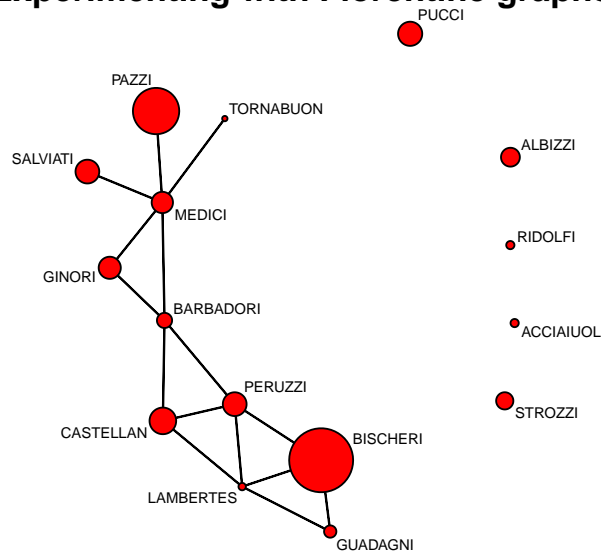
Here is the code for the plot

```
# Look at networks with node sizes proportional to wealth
# Notice that one of the options in this command is to set coordinates, so that pictures are equal.
# Otherwise, plots with attributes may differ from the original plot (compare with plot above).

par(mar=c(1,1,1,1))
plot(flo.biz,
      vertex.cex=(get.vertex.attribute(flo.biz, 'wealth')/25 +.4),
      displaylabels=TRUE,
      label.cex=.5,
      label.pos=0,
      main='Experimenting with Florentine graphs',
      coord=plot(flo.biz))
```



Experimenting with Florentine graphs

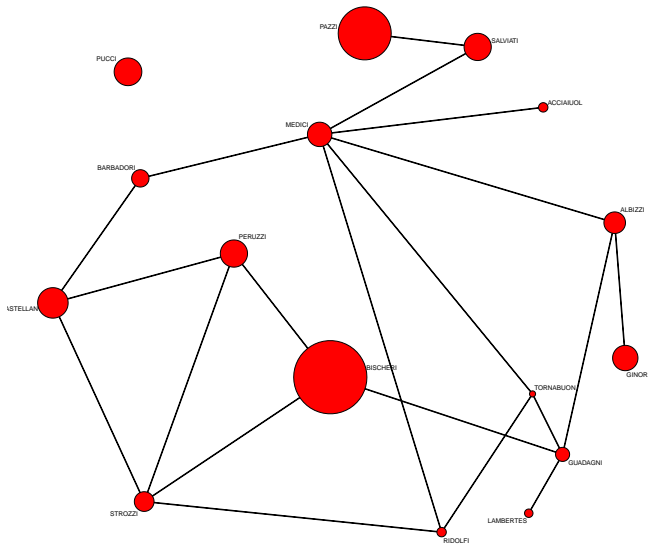


Answers in section "Importing UCINET files directly into R"

Assignment task: Using the code already shown, plot both of the new networks. Add attributes if you wish.

Answer. Well, with the *plot* option, there is very little left that I can show you - for more substantive options you needed to know other commands and other packages, which we have already seen. However, check out this picture:

Experimenting with Florentine graphs



No matter how hard you try, you will not be able to generate this plot by repeating the “plot” command. Why? Because I manually moved the nodes! Yes, R allows you to interactively change the location of the nodes, move them around until you are happy and until all the nodes are exactly where you want them. There is an option, *interactive*, which you set to TRUE, as in the code below. Unfortunately, RMarkdown chokes on this option, so for interactive imaging, you will have to work in the Console - but it’s something you can experiment with!

The code below is commented out, because the interactive option will not run in RMarkdown.

```
## plot(flo.marriage,  
##       vertex.cex=(get.vertex.attribute(flo.marriage, 'wealth')/25 +.4),  
##       displaylabels=TRUE,  
##       label.cex=.5,  
##       label.pos=0,  
##       main='Experimenting with Florentine graphs',  
##       coord=plot(flo.marriage),  
##       interactive=TRUE) #
```

Additional (optional) assignment for those who find the tasks above too easy:

For the network “drug” that we created and loaded with attributes, create several different network plots, adding gender and ethnicity to the graph as node attributes. Consider using a variety of colors to make your plot more informative.

Of course, many of you have figured out how to do it! Good for you. Here is my option at this assignment, without using any heavy artillery from other packages you may not have known. Little options (such as colors and shapes) you could have easily figured out. Of course, options are endless, and the one I show below, of course, is not the best - but it demonstrates graph shapes. :)

It seemed the most intuitive to me to color gender as pink and blue, and turn ethnicity into different shapes:

```
gender<-drugpaj$partitions[[1]]
ethnicity <- drugpaj$partitions[[2]]

#Set vectors based on attributes.

#Number of node sides allows to create different shapes (3=triangle, 4=square, etc.)
sides<-ifelse(ethnicity==1,12, ifelse(ethnicity==2, 3, ifelse(ethnicity==3, 4, 6)))

#Set colors by gender, including gray for undecided:
colors<-ifelse(gender==2,"palevioletred",ifelse(gender==1,"royalblue2","gray8"))

par(mar=c(0,0,0,0)) # And the plot itself:
plot(drug, vertex.col=colors, vertex.sides=sides, vertex.cex=1.5)
```

