

SaSR 8 Tutorial 1.2

[Code ▼](#)

Using the Knecht data

Rense Corten

Preamble

This document was written as an R Markdown (<http://rmarkdown.rstudio.com>) Notebook. If you are reading a html or pdf version of this, you can just keep reading, and if you want to run any of the code yourself, you need to copy and paste the code to your own R-script.

If you are reading this in the .rmd format: Try executing this chunk by clicking the **Run** button within the chunk or by placing your cursor inside it and pressing **Ctrl+Shift+Enter**. When you save the notebook, an HTML file containing the code and output will be saved alongside it (click the **Preview** button or press **Ctrl+Shift+K** to preview the HTML file).

Introduction

In this tutorial, we take some real network data that we will use throughout the course. The goal is to import the data, and perform some data management tasks as we've done in the previous tutorials, but now with real data instead of "toy data".

In doing so, we face a number of challenges:

1. The data are supplied in Stata's .dta file format, which is not a format that xUCINET or other R packages for SNA readily recognize.
2. The data themselves have a format that is also somewhat different from what for example xUCINET typically expects.

We'll solve these issues step by step. First we need to load some packages:

[Hide](#)

```
library(haven) #For reading Stata files
library(network) #For data handling of network data
library(xUCINET)
library(tidyverse) # because Tidyverse!
```

Importing the data into R

The first step is to import the data into R. The data as we get them are stored in Stata's ".dta" format. We use the Haven package to work with Stata files (we've loaded it above).

[Hide](#)

```
PupilsWaveV <- read_dta(file = "PupilsWaveV.dta")
```

Question: The variable "schoolnr" indicates the school (the number) and the class with the school (a letter); that is, "1A" means class A in school 1. How many school classes in total do the data contain? (HINT: use

Subsequently, to keep things simple, we keep only one class and one type of network (the data contain various networks, measured with name generator questions.)

```
m <- PupilsWaveV %>%
  filter(schoolnr == "12b") %>% # keep only one class
  select(namenr, friend1:friend12) # select only the "best friends" network
```

Have a look at the resulting data object. What we have now is an “adjacency list”. Although this is a common format for data produced by name generator questionnaires, SNA packages in R don’t like it. In order to work with it in xUCINET or other network packages, we first need to transform the data.

```
m <- as.matrix(m) # so that the next line works as intended; cbind() works in specific ways
                    # on matrix objects, which we exploit here.

m<- cbind(m[,1],c(m[,-1])) %>%
  as_tibble(m) %>% # So that we can easily filter
  filter(!is.na(V2)) # Now it is an edge list
```

```
g <- as.network.matrix(m,matrix.type = "edgelist") #from network package
```

```
a <- as.matrix.network(g, matrix.type = "adjacency" ) #again, from the network package
```

[illegible]

Done!

Assignment

Now try the following:

1. Add the “personal support” network (for this school class!) as a second network to the project.
2. Add “sex” and ” age” as attribute data.
3. Combine the friendship and support networks, assuming for the resulting network that there is a (directed) tie if pupil A nominates pupil B as a friend, and/or reports receiving support from B. (Also think: what is really the *direction* of the tie measured by the “personal support” name generator question?)
4. Try plotting the network.