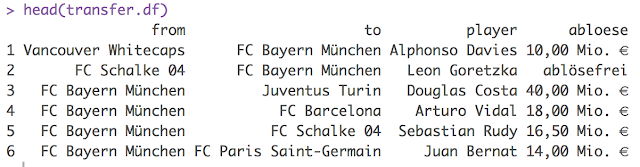
If you’re interested in the visualisation of networks or graphs, you might’ve heard of the great package “[visNetwork](https://datastorm-open.github.io/visNetwork/" \t "_blank)“. I think it’s a really great package and I love playing around with it. The scenarios of graph-based analyses are many and diverse: whenever you can describe your data in terms of “outgoing” and “receiving” entities, a graph-based analysis and/or visualisation is possible. During my work as a linguist, I already used graphs for different purposes like linking-structures within dictionaries, visualising co-occurence patterns of words and so on.

Today, I want to show you something completely different: transfers of male football players in the German “[1. Bundesliga](https://www.bundesliga.com/de/bundesliga/)“, the first division of male football in Germany. We can also describe this data in terms of outgoing and receiving entities (the nodes in the network): the clubs who are selling the players and the clubs who are buying the players. The edges (connections) within the network are the players themselves. And there are further attributes associated with the edges, e.g. the price of the player.

I’ll spare you the boring details of getting the data (please write a comment if you would like more details on that). I start with the raw data structure created by the scraping process. It’s a dataframe called transfer.df that looks like this:

[](https://i2.wp.com/3.bp.blogspot.com/-EaEw_SYV3NA/W5lVA-bl5_I/AAAAAAAADZM/bo7l0ax3GGgmlTEnkAB77xUi4GuNEQgVgCLcBGAs/s1600/transfer.df.png?ssl=1)

“abloese” (or “Ablöse”) means “transfer fee” and currently holds a string with certain codes and the currency: “ablösefrei” means that no transfer fee had to be payed (remember the famous [Bosman ruling](https://en.wikipedia.org/wiki/Bosman_ruling)?). “-” means that the information on a transfer fee doesn’t make sense (e.g., when a player finishes his career). “?” means that no information about the transfer fee is available. “Mio.” and “Tsd.” just encodes “million” or “thousand”, we have to deal with that later.

But we have to take care of something else first. In the dataframe, a player always appears twice if he changed teams **within** the 1. Bundesliga: the first time for his old club (as outgoing) and the second time for his new club (as incoming). I dealt with it this way (also, I loaded the required packages):

library(visNetwork)  
library(igraph)  
library(stringr)

transfer.df2 <- data.frame()  
all.players <- unique(transfer.df$player)  
for (pi in all.players) {  
  vork <- grep(pi, transfer.df$player, fixed = T)  
  if (length(vork) == 1) {  
    transfer.df2 <- rbind(transfer.df2, transfer.df[vork,])  
  } else {  
    transfer.df2 <- rbind(transfer.df2, transfer.df[vork[1],])  
  }  
}  
  
This basically means that, whenever a player appears more than once in transfer.df, the first appearance is kept and the second appearance is deleted. The resulting network wouldn’t be any different if we would keep the second appearance. So, now we are using transfer.df2 as our data structure.  
  
Now, we have to deal with the “abloese” (transfer fee) column:  
  
transfer.df2$abloese.num <- sapply(transfer.df2$abloese, USE.NAMES = F, FUN = function (x) {  
  if (x %in% c(“-“, “?”)) NA else {  
    if (x == “ablösefrei”) 0 else {  
      mio <- grepl(“Mio.”, x, fixed = T)  
      tsd <- grepl(“Tsd.”, x, fixed = T)  
      x2 <- gsub(“,”, “.”, x, fixed = T)  
      x3 <- gsub(“Mio. €”, “”, x2, fixed = T)  
      x4 <- as.numeric(str\_trim(gsub(“Tsd. €”, “”, x3, fixed = T)))  
      if (mio) x4\*1000000 else {  
        if (tsd) x4\*1000 else { “FEHLER” }  
      }  
    }  
  }

})

Basically, this is what we are doing:

* If abloese is “-” or “?”, we are using NA
* If abloese is “ablösefrei” we are putting in 0
* Then we see whether “Mio.” appears in the string.
* Then we see whether “Tsd.” appears in the string.
* Then we are deleting these substrings and the EUR sign and
* trim the string and convert it to a numeric value.
* If “Mio.” appeared in the string, we are multiplying the result with one million and if “Tsd.” appeared in the string, we are multiplying the result with one thousand (both will never appear in the string, it doesn’t make sense).

Alright, now we have the column abloese.num and can move on to group the different transfer fees because we want to assign different colours to the edges in the network dependent on the transfer sum. The thresholds are arbitrary.

transfer.df2$abl.group <- cut(transfer.df2$abloese.num, c(0, 200\*1000, 1000\*1000, 2000\*1000, 5000\*1000, 10000\*1000, 60000\*1000), include.lowest = T)  
  
transfer.df2$abl.col <- ifelse(transfer.df2$abloese.num == 0, “green”,  
                               ifelse(transfer.df2$abl.group == “[0,2e+05]”, “#ffffcc”,  
                                      ifelse(transfer.df2$abl.group == “(2e+05,1e+06]”, “#fed976”,  
                                             ifelse(transfer.df2$abl.group == “(1e+06,2e+06]”, “#feb24c”,  
                                                    ifelse(transfer.df2$abl.group == “(2e+06,5e+06]”, “#fc4e2a”,  
                                                           ifelse(transfer.df2$abl.group == “(5e+06,1e+07]”, “#e31a1c”,  
                                                                  ifelse(transfer.df2$abl.group == “(1e+07,6e+07]”, “#800026”, “grey”)))))))  
transfer.df2$abl.col <- ifelse([is.na](http://is.na)(transfer.df2$abl.group), “grey”, transfer.df2$abl.col)   
  
Now, I am converting the dataframe to an igraph object and this object to visNetwork object. I’m sure the igraph step could be skipped, but this works like a charm and doesn’t take much time.  
  
graph <- graph.data.frame(transfer.df2)

vn <- toVisNetworkData(graph)

I am assigning color codes to the nodes:

vn$nodes$color <- ifelse(vn$nodes$id %in% clubs, “tomato”,

                         ifelse(vn$nodes$id == “Vereinslos”, “green”,

                                ifelse(vn$nodes$id == “Karriereende”, “blue”, “grey”)))

All clubs in the 1. Bundesliga get “tomato” (clubs is an object I defined earlier) all clubs that are not in the 1. Bundesliga (e.g., Hamburger SV) get “grey”. There are two other special “clubs”: “Karriereende” for “end of career” and “Vereinslos” for “no club”, both get “green”.

Three things left to be done:

vn$edges$title <- paste(vn$edges$player, vn$edges$abloese, sep = ” – “)

vn$edges$color <- vn$edges$abl.col

vn$edges$width <- 4

* Assign a title to the edges that consists of the player name and the transfer fee. This appears upon hovering the edge.
* Assign the grouped transfer fee color we defined earlier.
* Increase the width of the edges to make the color more visible.

No, for creating the HTML file for the graph:

visNetwork(nodes = vn$nodes, edges = vn$edges, height = “1000px”, width = “100%”) %>%

  visOptions(highlightNearest = TRUE) %>%

  #visIgraphLayout(layout = “layout\_with\_dh”) %>%

  visEdges(arrows = “to”, arrowStrikethrough = F) %>% visSave(file = “~/Desktop/transfers.html”, selfcontained = T)

Please visit my [personal webspace](http://www.wolferonline.de/uploads/transfers.html) for the final result. The “redder” an edge in the network is, the more expensive the transfer was. You can also click on the nodes to only highlight all adjacent nodes (selling and buying clubs), drag nodes around (graph physics!) and hover over edges to see the specific player being transfered. Of course, zooming is enabled.