

Heimarbeit4

Tobias Hoesli, Julien Lattmann

2022-05-22

Wie bereits bei der letzten Arbeit beschrieben stellt unser Netzwerk die Transfers zwischen allen Teams aus den beiden höchsten Fussballligen der Schweiz in der Saison 2021/22 dar. Die Knoten stellen die 20 Mannschaften dar (je 10 pro Liga), während die Kanten jeweils einen Transfer, also einen Spieler, welcher von einem Team zu einem anderen gewechselt ist, verkörpern. Weil die Richtung des Wechsels bestimmt ist, handelt es sich um ein gerichtetes Netzwerk.

Unsere Hauptvisualisierung (Abbildung 1) gibt im Grunde alle Informationen über unser Netzwerk auf einen Blick wieder. Die Kanten unterscheiden sich farblich durch den Transfertyp und ihre Grösse repräsentiert die Transaktionssumme. Das Netzwerk hat erwartungsgemäss eine Grösse von 20, was auch der Anzahl der Knotenpunkte entspricht. Die Knoten unterscheiden sich farblich zwischen den beiden Schweizer Ligen und die Grösse repräsentiert zusätzlich den Stellenwert des einzelnen Knotens im Netzwerk, gemäss Betweeness. Bei der Betrachtung der verschiedenen Zentralitäts-Masse offenbaren sich einige Unterschiede bezüglich der Varianz der Zentralität zwischen den Knoten. Bei der degree-basierten Zentralisierung variieren die Werte für alle Knoten in einem überschaubaren Bereich von 3-7. Dies drückt aus, wie viele direkte Verbindungen zu anderen Knoten bei jedem Knoten vorhanden sind (Jansen, 2006). Gemäss Jansen weist das Mass auf die “mögliche Kommunikationsaktivität” hin, womit die Aktivität der Teams bezüglich dem Abgeben von Spielern nicht allzu fest variiert und somit auch kein Team einen extrem “zentralen” Platz im Graphen einnimmt. Die betweenness-basierte Zentralisierung offenbart grössere Unterschiede zwischen den einzelnen Knoten, da die Unterschiede bezüglich kürzesten Pfaden, welche durch den betrachteten Knoten führen, grösser sind als jene bezüglich direkten Verbindungen. Der Wertebereich reicht somit von 1.7 bis 28.8 und drückt gemäss Jansen (2006) die “mögliche Kommunikationskontrolle” aus. Das “zentralste” Team im Netzwerk wäre somit YB, welches über ein hohes Mass an “Kontrolle” darüber verfügt, wie Spieler innerhalb der Liga transferiert werden. Zuletzt kann noch die closeness-basierte Zentralisierung betrachtet werden, welche aufgrund der Werte, welche alle nahe bei 0.5 liegen und nur wenig variieren, nahe legt, dass alle Knoten ein ähnlich hohes Mass an Zentralität aufweisen. Die tiefen Werte hier implizieren, dass zwischen sämtlichen Knoten durchaus geringe durchschnittliche Distanzen bestehen. Gemäss Jansen (2006) dient das Mass auch als Indikator für die Unabhängigkeit bzw. Effizienz der Knoten, welche somit bei allen Teams hoch sein dürfte.

Für das Clustering (Abbildung 2) haben wir uns für drei verschiedene Methoden entschieden, um bestimmte “Communities” in unserem Netzwerk zu bestimmen. Die von uns gewählten Methoden waren “Walktrap, Edge-Betweeness und Infomap”. Die Entscheidung für die beiden letztgenannten Methoden war darauf zurückzuführen, dass sie die einzigen waren, die gerichtete Netzwerke unterstützen. Edge-Betweeness und Infomap Clustering zeigten leider einen niedrigen Modularitätswert (zwischen 1,11 - 1,13), was uns zeigt, dass die Algorithmen nicht zuversichtlich waren, Untergruppenstrukturen in unserem Netzwerk zu finden. Die Modularität zeigt also, dass unser Netzwerk keine einzelnen besonders dichten Verbindungen zwischen den Knoten aufweist. Interessant ist, dass das Clustering, das auf den ersten Blick die “deutlichste” Unterscheidbarkeit von Clustern darstellt (Walktrap), den niedrigsten Modularitätswert (0,06) aufweist.

Zusätzlich haben wir Modellierung (Abbildung 3) auf dem Netzwerk betrieben. Die Modellierung ist nützlich bei der Analyse von Netzwerkmodellen und dient als Grundlage für den Vergleich empirischer sozialer Netzwerke und als Bausteine für komplexe Netzwerksimulationen (Luke 2015). Wir führten die Simulation von 3 verschiedenen Modellen mit der gleichen Netzwerkgrösse durch. Interessant dabei war, dass die Modelle

in beiden Spektren gebaut wurden, das Erdos-Renyi-Modell erzeugte ein etwas dichteres Modell als unser eigenes. Noch beeindruckender war jedoch, dass das Barabasi Game Model im Vergleich zu unserem ursprünglichen Modell eine sehr geringe Dichte (0.1 im Vergleich zu 0.45) aufwies. Das Modell erzeugte sogar Isolates und Komponenten.

Als literarische Referenz haben wir uns auf Ali Sevilmiş Papier bezogen, das die Transfers in der türkischen Liga untersucht (Sevilmiş 2020). In seiner Untersuchung kam er zu dem Schluss, dass es eine beträchtliche Anzahl von Transfers zwischen anatolischen Mannschaften gab, einige sogar mit ähnlicher Vereinskultur. Solche kulturgeschichtlichen Entdeckungen und weitere Forschungen wären für die Umsetzung in unserem Netzwerk sehr interessant gewesen. Aufgrund der Dichte des Transfernetzes 2021/22 war es schwierig, irgendwelche "Communities" zu identifizieren. Weitere Forschungen zur Kumulierung von Transfertransaktionen der vergangenen Jahre könnten wiederum zur Entdeckung kultureller oder freundschaftlicher Bindungen zwischen Vereinen führen, die wir noch nicht entdecken konnten.

Anzahl Wörter: 685

Appendix: Gesamten R-Code für den Bericht

```
## ----set up of packages,
## include=FALSE-----
## install.packages('statnet')
library("RColorBrewer")
library("statnet")
library("scales")
library("formatR")
library("knitr")
library("readxl")
library("tidyr")
library("kableExtra")
library("dplyr")
## ----set up of edge list ,
## tidy=TRUE-----
edge_list <- read.csv("data/Edge_List_Values.csv", sep = ";")
# Erstellen der Kanten-Liste aufgrund der gesammelten Daten
netmat2 <- edge_list
# Netzwerk-Objekt erstellen
transfer_net <- network(netmat2, matrix.type = "edgelist", directed = TRUE)
# Knoten Acronyms
network.vertex.names(transfer_net) <- c("YB", "BAS", "SFC", "LUG",
    "LUZ", "LS", "SG", "FCZ", "SION", "GC", "VAD", "THU", "SLO",
    "FCS", "AAR", "WIN", "WIL", "SCK", "XAM", "YS")
# Knoten-Attribut 'league' definieren
network::set.vertex.attribute(transfer_net, "league", c("SL",
    "SL", "SL", "SL", "SL", "SL", "SL", "SL", "SL", "ChL",
    "ChL", "ChL", "ChL", "ChL", "ChL", "ChL", "ChL", "ChL"))
# Kanten-Attribut 'spendings' definieren: Transfersumme in
# Tsd.
spendings <- read.csv("data/Spendings.csv", sep = ";")
network::set.edge.attribute(transfer_net, "spendings", spendings)
# Kanten-Attribut 'type' definieren:
# Spieler-Transaktionstyp (Ablöse, Leihe, unbekannt)
type <- read.csv("data/Type_ID.csv", sep = ";")
network::set.edge.attribute(transfer_net, "type", type)
```

```
# Knoten-Attribut 'alldeg' definieren (abgekürzte Variante)
transfer_net %v% "alldeg" <- sna::degree(transfer_net)
class(transfer_net)
```

```
## [1] "network"
```

```
bet <- sna::betweenness(transfer_net, gmode = "graph")
summary(bet)
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  1.743   6.383   8.169   9.875  11.917  28.787
```

```
bet
```

```
## [1] 28.787271  9.448153  1.844246  8.142316 13.235220  8.196140  6.755556
## [8]  5.327055  1.742857 15.538220  2.412302  9.550992 15.167995  6.878083
## [15]  6.734830  3.152686 11.477543 24.545531  8.092170 10.470833
```

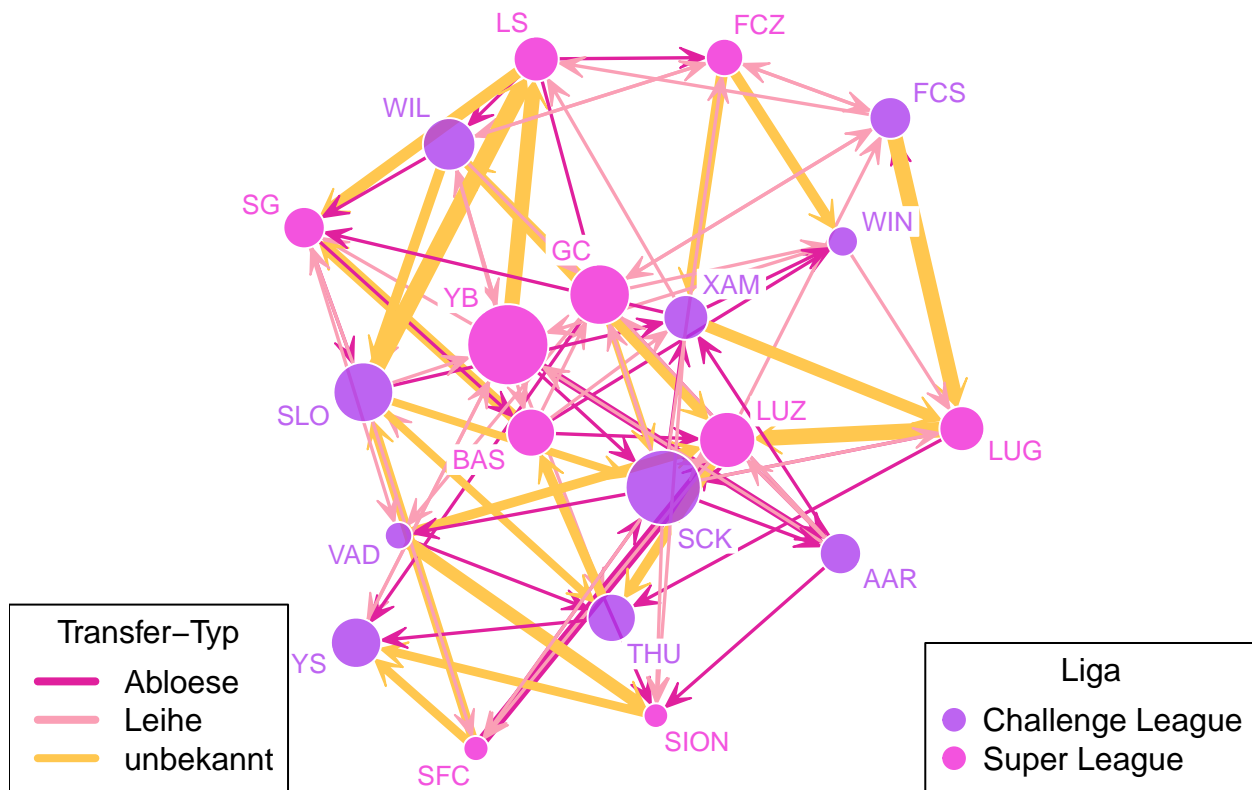
```
deg <- sna::degree(transfer_net, gmode = "graph")
deg
```

```
## [1] 5 3 3 4 7 4 6 4 5 6 3 5 5 4 4 4 4 6 5 5
```

```
cls <- sna::closeness(transfer_net, gmode = "graph")
cls
```

```
## [1] 0.6551724 0.6333333 0.5428571 0.5937500 0.6551724 0.6333333 0.6129032
## [8] 0.5588235 0.5757576 0.6551724 0.5937500 0.5937500 0.6333333 0.5588235
## [15] 0.5937500 0.5757576 0.6333333 0.7037037 0.6785714 0.5588235
```

```
par(mar = c(1, 0, 1, 0), mfrow = c(1, 1))
linecol_pal <- c("#e0209d", "#fa9fb5", "#ffc74f")
league_pal = c("#aa36e6", "#f353de")
type_cat <- as.factor(network::get.edge.attribute(transfer_net,
  "type"))
league_cat = as.factor(network::get.vertex.attribute(transfer_net,
  "league"))
set.seed(8)
gplot(transfer_net, vertex.col = league_pal[league_cat], displaylabels = TRUE,
  vertex.cex = sqrt(bet + 1)/1.85, vertex.border = "white",
  edge.col = linecol_pal[transfer_net %e% "type"], edge.lwd = sqrt(transfer_net %e%
    "spendings") * 0.35, mode = "fruchtermanreingold", boxed.labels = TRUE,
  label.border = "white", label.pos = 0, label.bg = "white",
  label.col = league_pal[league_cat], label.cex = 0.85, usearrows = TRUE)
legend("bottomleft", legend = c("Abloese", "Leihe", "unbekannt"),
  col = linecol_pal, lty = 1, lwd = 4, pt.cex = 1.5, bty = "o",
  title = "Transfer-Typ")
legend("bottomright", legend = c("Challenge League", "Super League"),
  col = league_pal, pch = 19, pt.cex = 1.5, bty = "o", title = "Liga")
```



```
library(igraph)
# detach(package:statnet, unload = TRUE)
# detach(package:network, unload = TRUE)
edge_list <- read.csv("data/Edge_List_Values.csv", sep = ";")
class(edge_list)
```

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

```
graph <- graph_from_data_frame(edge_list, directed = FALSE, vertices = )
V(graph)$name <- c("YB", "BAS", "SFC", "LUG", "LUZ", "LS", "SG",
  "FCZ", "SION", "GC", "VAD", "THU", "SLO", "FCS", "AAR", "WIN",
  "WIL", "SCK", "XAM", "YS")
V(graph)$league <- c("SL", "SL", "SL", "SL", "SL", "SL", "SL",
  "SL", "SL", "SL", "ChL", "ChL", "ChL", "ChL", "ChL", "ChL",
  "ChL", "ChL", "ChL", "ChL")
# net <- network(edge_list, directed = TRUE, matrix.type =
# 'edgelist', vertex.attrnames = vertex_names)
type <- read.csv("data/Type_ID.csv", sep = ";")
E(graph)$type <- type
spendings <- read.csv("data/Spending.csv", sep = ";")
E(graph)$spending <- spendings
# Erstellen der Kanten-Liste aufgrund der gesammelten Daten
# netmat2 <- edge_list
V(graph)
```

```
## + 20/20 vertices, named, from cc107eb:
## [1] YB BAS SFC LUG LUZ LS SG FCZ SION GC VAD THU SLO FCS AAR
## [16] WIN WIL SCK XAM YS
```

```
graph
```

```
## IGRAPH cc107eb UN-- 20 92 --
## + attr: name (v/c), league (v/c), type (e/x), spending (e/x)
## + edges from cc107eb (vertex names):
## [1] YB --AAR YB --SCK YB --LS YB --WIL YB --SG YB --THU
## [7] YB --YS BAS --WIN BAS --LUZ BAS --SION BAS --SG BAS --XAM
## [13] BAS --GC SFC --LUZ SFC --SCK SFC --YS SFC --SLO LUG --FCS
## [19] LUG --SCK LUG --THU LUG --LUZ LUZ --AAR LUZ --THU LUZ --SCK
## [25] LUZ --FCS SFC --LUZ LS --WIL LS --GC LS --FCZ LS --SG
## [31] LS --SLO BAS --SG SG --SLO SG --VAD FCZ --WIN FCZ --XAM
## [37] FCZ --WIL FCZ --FCS SION--YS GC --AAR GC --YS GC --WIL
## [43] GC --SCK GC --FCS GC --WIN GC --VAD VAD --THU SION--VAD
## + ... omitted several edges
```

```
cw <- cluster_walktrap(graph)
membership(cw)
```

```
## YB BAS SFC LUG LUZ LS SG FCZ SION GC VAD THU SLO FCS AAR WIN
## 1 1 1 1 1 2 1 2 1 1 1 1 1 2 1 2
## WIL SCK XAM YS
## 2 1 1 1
```

```
modularity(cw)
```

```
## [1] 0.06970698
```

```
ceb <- cluster_edge_betweenness(graph)
membership(ceb)
```

```
## YB BAS SFC LUG LUZ LS SG FCZ SION GC VAD THU SLO FCS AAR WIN
## 1 2 3 3 3 1 2 1 4 3 5 1 1 3 3 6
## WIL SCK XAM YS
## 1 3 7 1
```

```
modularity(ceb)
```

```
## [1] 0.1372873
```

```
imap <- cluster_infomap(graph)
membership(imap)
```

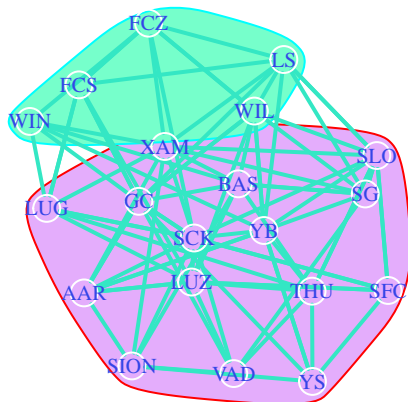
```
## YB BAS SFC LUG LUZ LS SG FCZ SION GC VAD THU SLO FCS AAR WIN
## 1 2 3 4 3 5 2 6 1 7 4 5 5 5 7 1
## WIL SCK XAM YS
## 6 4 7 1
```

```
modularity(imap)
```

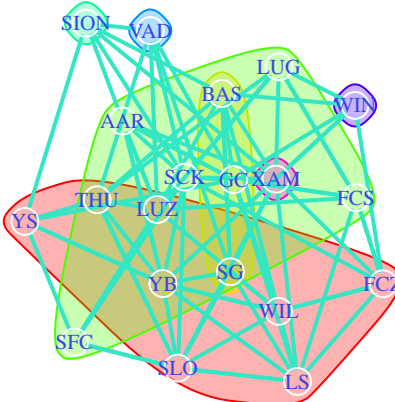
```
## [1] 0.07525992
```

```
graph_pal <- c("#FF5733", "#f353de")
V(graph)$color <- graph_pal[V(graph)$league]
# plot(graph, vertex.label=V(graph)$name)
set.seed(40)
par(mar = c(2, 0, 2, 0), mfrow = c(1, 3))
plot(graph, vertex.label = V(graph)$names, mark.groups = cw,
     mark.col = c("#E4ADFC", "#76FFCB"), vertex.label = NA, vertex.color = V(graph)$color,
     edge.color = "#34E7C4", edge.width = 2, vertex.label.color = "#3445E7",
     vertex.frame.color = "#ffffff", edge.curved = FALSE, main = "Walktrap")
plot(graph, vertex.label = V(graph)$names, mark.groups = ceb,
     vertex.label = NA, vertex.color = V(graph)$color, edge.color = "#34E7C4",
     edge.width = 2, vertex.label.color = "#3445E7", vertex.frame.color = "#ffffff",
     edge.curved = FALSE, main = "CEB")
plot(graph, vertex.label = V(graph)$names, mark.groups = imap,
     vertex.label = NA, vertex.color = V(graph)$color, edge.color = "#34E7C4",
     edge.width = 2, vertex.label.color = "#3445E7", vertex.frame.color = "#ffffff",
     edge.curved = FALSE, main = "Imap")
```

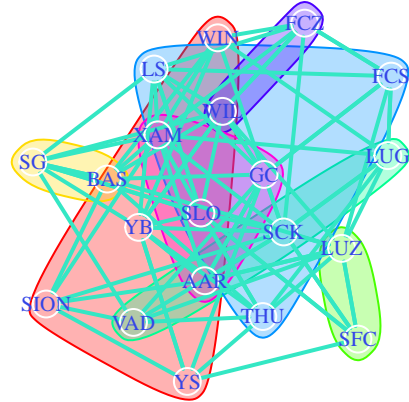
Walktrap



CEB



Imap



```
library(ggplot2)
graph
```

```
## IGRAPH cc107eb UN-- 20 92 --
## + attr: name (v/c), league (v/c), color (v/c), type (e/x), spending
## | (e/x)
## + edges from cc107eb (vertex names):
## [1] YB --AAR YB --SCK YB --LS YB --WIL YB --SG YB --THU
## [7] YB --YS BAS --WIN BAS --LUZ BAS --SION BAS --SG BAS --XAM
## [13] BAS --GC SFC --LUZ SFC --SCK SFC --YS SFC --SLO LUG --FCS
## [19] LUG --SCK LUG --THU LUG --LUZ LUZ --AAR LUZ --THU LUZ --SCK
## [25] LUZ --FCS SFC --LUZ LS --WIL LS --GC LS --FCZ LS --SG
## [31] LS --SLO BAS --SG SG --SLO SG --VAD FCZ --WIN FCZ --XAM
## [37] FCZ --WIL FCZ --FCS SION--YS GC --AAR GC --YS GC --WIL
## + ... omitted several edges
```

```
graph.density(graph)
```

```
## [1] 0.4842105
```

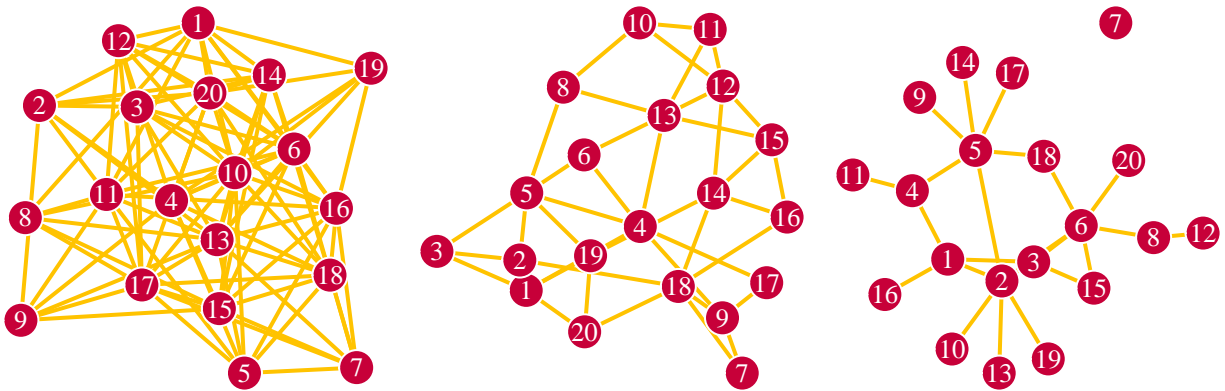
```
transfer_degree <- mean(igraph::degree(graph))
transfer_degree
```

```
## [1] 9.2
```

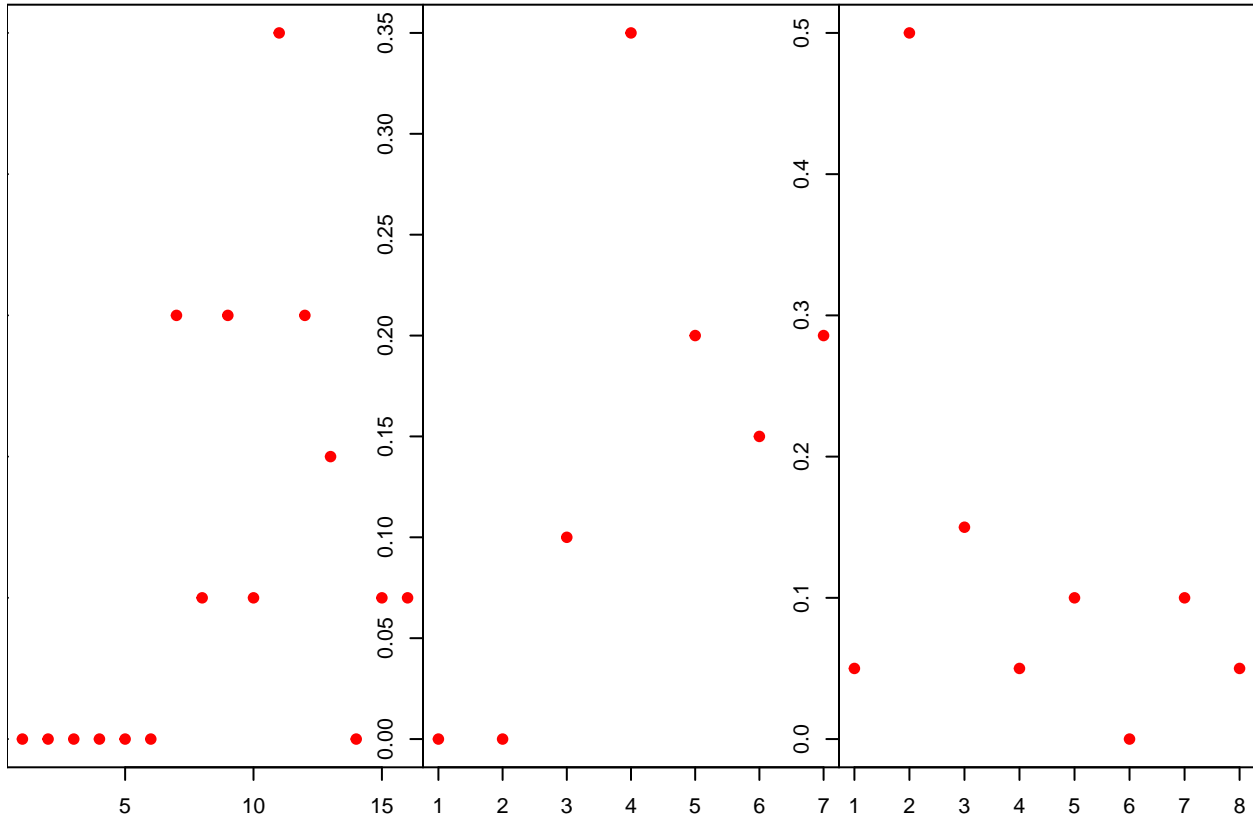
```
transfer_density <- graph.density(graph)
transfer_density
```

```
## [1] 0.4842105
```

```
original_graph <- c("transfer_network", transfer_density, transfer_degree)
g_rnd <- erdos.renyi.game(20, 0.484, type = "gnp")
g_smwrld <- watts.strogatz.game(dim = 1, size = 20, nei = 2,
  p = 0.25)
g_prfatt <- barabasi.game(20, out.dist = c(0.15, 0.6, 0.25),
  directed = FALSE, zero.appeal = 2)
set.seed(123)
par(mar = c(2, 0, 2, 0), mfrow = c(1, 3))
plot(g_rnd, edge.color = "#FFC300", vertex.color = "#C70039",
  vertex.label.color = "white", vertex.label.cex = 1.25, vertex.frame.color = "white",
  vertex.size = 20, edge.width = 2)
plot(g_smwrld, edge.color = "#FFC300", vertex.color = "#C70039",
  vertex.label.color = "white", vertex.label.cex = 1.25, vertex.frame.color = "white",
  vertex.size = 20, edge.width = 2)
plot(g_prfatt, edge.color = "#FFC300", vertex.color = "#C70039",
  vertex.label.color = "white", vertex.label.cex = 1.25, vertex.frame.color = "white",
  vertex.size = 20, edge.width = 2)
```



```
par(mar = c(2, 0, 2, 0), mfrow = c(1, 3))
plot(degree.distribution(g_rnd), xlab = "Degree", ylab = "Proportion",
     col = "red", pch = 19)
plot(degree.distribution(g_smrld), xlab = "Degree", ylab = "Proportion",
     col = "red", pch = 19)
plot(degree.distribution(g_prfatt), xlab = "Degree", ylab = "Proportion",
     col = "red", pch = 19)
```

```
# Modellierungs Tabelle
model_keys <- function(graph, model) {
  name <- deparse(substitute(model))
  density <- graph.density(model)
  degree <- mean(degree(model))
  df <- data.frame(modelname = name, density = density, degree = degree)
  return(df)
}
list_model_g_rnd <- model_keys(graph, g_rnd)
list_model_g_smwrld <- model_keys(graph, g_smwrld)
list_model_g_prfatt <- model_keys(graph, g_prfatt)
list_all_models <- rbind(original_graph, list_model_g_rnd, list_model_g_smwrld,
  list_model_g_prfatt)
kable(list_all_models, caption = "Modelling Networks", booktabs = T) %>%
  kable_styling(latex_options = "HOLD_position")
```

Table 1: Modelling Networks

modelname	density	degree
transfer_network	0.484210526315789	9.2
g_rnd	0.510526315789474	9.7
g_smwrld	0.210526315789474	4
g_prfatt	0.121052631578947	2.3

```
graph.density(g_rnd)
```

```
## [1] 0.5105263
```

```
##### Start Hausarbeit 3 #####
```

```
# Grösse des Netzwerkes berechnen
```

```
NA
```

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

```
NA
```

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

```
# Dichte des Netzwerkes berechnen
```

```
Dichte <- gden(transfer_net)
```

```
Dichte
```

```
## [1] 0.2421053
```

```
# oder von Hand -->  $m/(n(n-1))$ 
```

```
Dichte_H <- 92/(20 * (20 - 1))
```

```
Dichte_H
```

```
## [1] 0.2421053
```

```
# Komponenten des Netzwerkes berechnen
```

```
Komp <- sna::components(transfer_net)
```

```
Komp
```

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

```
# Durchmesser des Netzwerkes berechnen
```

```
Dm <- geodist(transfer_net)
```

```
Dm_max <- max(Dm$gdist)
```

```
Dm_max
```

```
## [1] 5
```

```
# Clustering Coefficient des Netzwerkes berechnen
```

```
Clust <- gtrans(transfer_net)
```

```
Clust
```

```
## [1] 0.1760204
```

Luke, Douglas A. 2015. *A User's Guide to Network Analysis in R*. 1st ed. Use r! Cham, Switzerland: Springer International Publishing.

Sevilmiş, Ali. 2020. "Examining Transfer Directions in 2019-2020 Season in Turkey by Means of Social Network Analysis." *Turkish Journal of Sport and Exercise* 22 (December): 395–401. <https://doi.org/10.15314/tsed.724394>.