

Heimarbeit 2

Julien Lattmann, Tobias Hoesli

2022-04-13

Als Teil des Seminars über soziale Netzwerke haben wir beschlossen, ein Fußballtransfer-Netzwerk zu erstellen. Wir sammelten alle Transaktionsdaten von Vereinen aus den beiden höchsten Schweizer Ligen. Dazu gehören 10 Teams aus der Super League (1. Liga) und 10 Teams aus der Challenge League (2. Liga):

Table 1: Super League Teams (CL)

Team	Kuerzel
BSC Young Boys	YB
FC Basel	BAS
Servette FC	SFC
FC Lugano	LUG
FC Luzern	LUZ
Lausanne-Sport	LS
FC St. Gallen	SG
FC Zürich	FCZ
FC Sion	SIO
Grasshoppers	GC

Table 2: Challenge League Teams (CL)

Team	Kuerzel
FC Vaduz	VAD
FC Thun	THU
Stade-Lausanne	SLO
FC Schaffhausen	FCS
FC Aarau	AAR
FC Winterthur	WIN
FC Wil 1900	Wil
SC Kriens	SCK
Neuchâtel Xamax	XAM
Yverdon Sport	YS

Wir haben jede einzelne Spielertransaktion zwischen Vereinen in diesen beiden Ligen für die gesamte Saison 2021/2022 analysiert. Diese Daten wurden von Transfermarkt.ch gesammelt. Die Transaktionen umfassten Transfers mit Abloesesumme (Abloese), wenn ein Spieler ausgeliehen wurde (Leihe) und unbekannt. Wir wissen, dass ein Spieler zu dem besagten Verein geschickt wurde, aber Transfermarkt kennt nicht die Art der Transaktion und ob er für eine Ablösesumme gewechselt hat.

Netzwerk

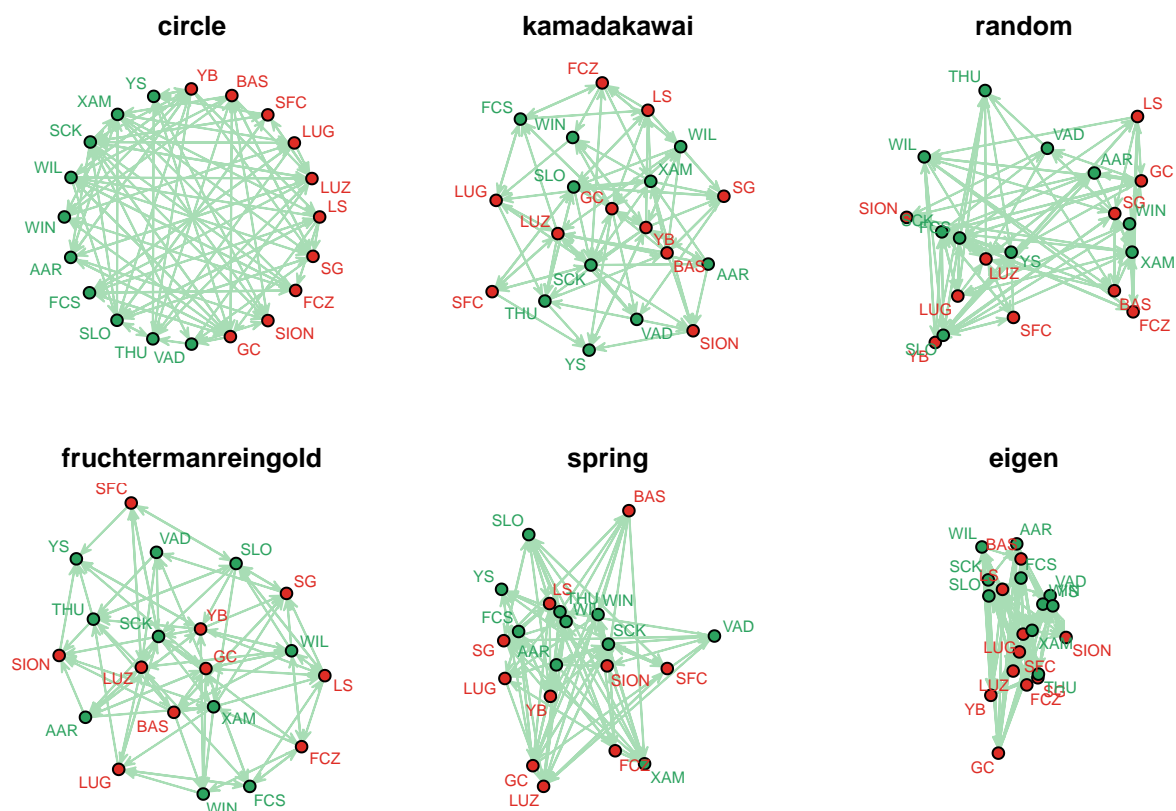
Nachdem wir alle Daten gesammelt haben, besteht unser Netzwerk aus 20 Knoten (Teams) und insgesamt 92 Kanten (Transaktionen). Die Kanten sind gerichtet, da bei jeder Transaktion ein Spieler von einer Mannschaft zu einer anderen Mannschaft wechselt.

Visualisierung

Layoutwahl

Um das richtige Layout zu wählen, haben wir 6 mögliche Modi visualisiert: Circle, Kamadakawai, Random, Fruchtermanreingold, Spring und Eigen.

(Luke 2015)



Wir haben schnell festgestellt, dass wir in unserem Netz eine relativ hohe Dichte von 25 % und eine hohe Anzahl von Kanten haben. Dadurch wirkt das Netz zuerst ziemlich “chaotisch”.

Wie in (Luke 2015) erwähnt, haben wir bei unseren Layout-Optionen nach bestimmten Faktoren gesucht. Da die Anzahl der Kanten recht hoch ist, ist eine Minimierung der Kantenkreuzungen kaum realisierbar. Andererseits war ein Layout, das die Symmetrie des Netzes maximiert und die Variabilität der Kantenlänge minimiert, in Reichweite und umsetzbar. Wir entschieden uns schließlich zwischen dem Kreis-Layout und dem Fruchtermanreingold-Layout, wobei die endgültige Entscheidung auf Fruchtermanreingold fiel. Obwohl der Aspekt der Maximierung der Symmetrie und der Minimierung der Kantenlänge bei beiden in ähnlicher Weise gegeben ist. Fruchtermanreingold gibt einen viel tieferen Einblick in die Form des Netzes und die Zentralität. Außerdem deckt es einen weiteren wichtigen Punkt besser ab, nämlich die Maximierung des Winkels zwischen den Kanten, wenn sie Knoten kreuzen oder verbinden (Luke 2015).

```
bet <-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 <-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
```

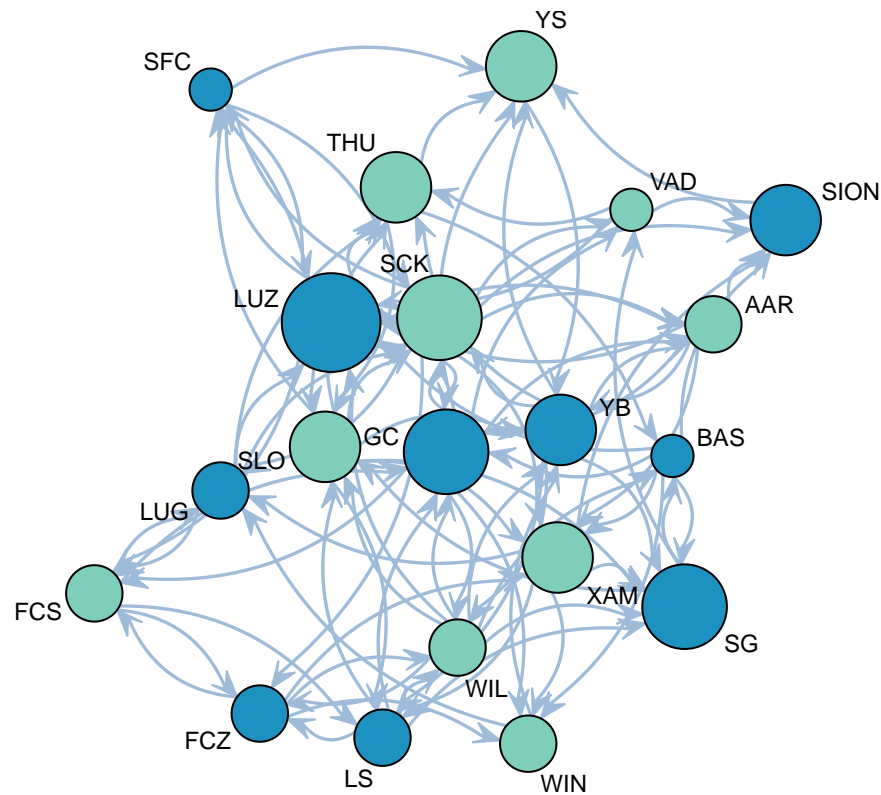
```
cls <-closeness(transfer_net, gmode = "graph")
cls
```

```
## [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
```

Knotenattribute

Um ein Knotenattribute zu visualisieren, haben wir uns für die zwei verschiedenen Ligen entschieden, aus denen die Teams stammen, indem wir sie entsprechend ihrer Ligazugehörigkeit einfärben. Zusätzlich berechneten wir den Degree jedes Knotens, was sich dann in der Größe jedes Knotens widerspiegelte. Zudem haben wir mit gebogenen Pfeilen anstelle von geraden Pfeilen herumgespielt.

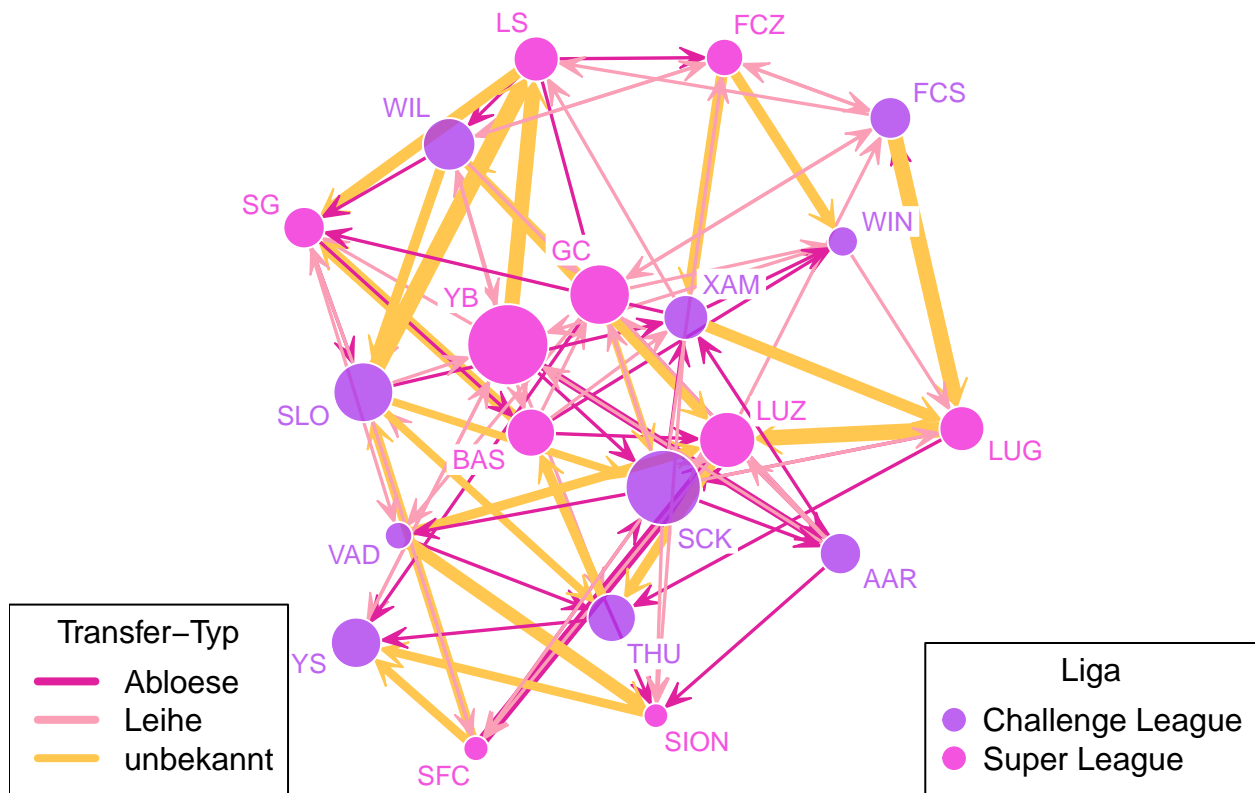
Interessant dabei war, dass der Degree als Knotenattribut nicht so signifikant zu sein schien, denn jeder Knoten hatte mindestens 3 Verbindungen, wobei der höchste Wert 7 war und der Mittelwert bei 4,6 lag.



Kantenattribute

Als quantitatives Kantenattribut haben wir eine Liste von “Ausgaben” für jede Transaktion ausgewertet. Darüber hinaus konnten wir die Art der Transaktion als qualitatives Attribut verwenden (Abloese, Leihe, unbekannt). Leihgaben wurden als 0-Ausgaben betrachtet, Ablöse wurde mit der tatsächlichen Transfer-summe betrachtet und für die Unbekannten haben wir, anstatt auf eine Null-Ausgabe zurückzufallen, den von Transfermarkt ermittelten Markwert des Spielers zu Transaktionszeitpunkt als Ausgabensumme verwendet.

Die Transaktionssumme spiegelt sich in der Größe des Pfeils und die Transaktionsart in der Farbe des Pfeils wider. Außerdem basiert die Grösse jedes Knotens nun auf der Betweeness des Knotens im Netzwerk.



Zentrale Botschaften

Interessant war die Entdeckung, dass bei der Visualisierung die Betweenness eine viel grössere Bedeutung hatten als der Degree Wert. Alle Knoten hatten eine ähnliche Anzahl von Verbindungen im Graphen, aber der "kürzeste" Weg zwischen zwei Punkten führte meist durch eine Handvoll von Knoten. Dies zeigt sich dadurch, dass der Median von Betweenness bei 8,1 liegt, die Teams YB und SC Kriens aber Werte von 28,8 und 24,5 aufweisen. Sie übertreffen damit den Rest der Knotenpunkte bei weitem.

Die hohe Zahl der unbekannten Transaktionsarten und -summen lässt auch Raum für Spekulationen. Die höchste bestätigte Summe, die in dieser Saison zwischen Schweizer Teams bezahlt wurde, betrug 380'000 CHF, was nach unserer Logik nur die zehnthöchste Transaktion wäre. Wir können also davon ausgehen, dass die unbekannten Transaktionssummen für diese Spieler deutlich unter ihrem damaligen Marktwert lagen.

Appendix: Gesamten R-Code für den Bericht

```
knitr::opts_chunk$set(echo = TRUE)
# install.packages('statnet')
library("RColorBrewer")
library("statnet")
library("scales")
library("formatR")
library("knitr")
library("readxl")
library("tidyr")
```

```

library("kableExtra")
library("dplyr")
team_table <- read_excel("data/Edge_List_Transfers.xlsx", sheet = "Teams")
team_table_sl <- team_table %>%
  filter(Liga == "Super League") %>%
  select(Team, Kuerzel)
team_table_cl <- team_table %>%
  filter(Liga == "Challenge League") %>%
  select(Team, Kuerzel)
kable(team_table_sl, caption = "Super League Teams (CL)", booktabs = T) %>%
  kable_styling(latex_options = "HOLD_position")

```

Table 3: Super League Teams (CL)

Team	Kuerzel
BSC Young Boys	YB
FC Basel	BAS
Servette FC	SFC
FC Lugano	LUG
FC Luzern	LUZ
Lausanne-Sport	LS
FC St. Gallen	SG
FC Zürich	FCZ
FC Sion	SIO
Grasshoppers	GC

```

kable(team_table_cl, caption = "Challenge League Teams (CL)",
  booktabs = T) %>%
  kable_styling(latex_options = "HOLD_position")

```

Table 4: Challenge League Teams (CL)

Team	Kuerzel
FC Vaduz	VAD
FC Thun	THU
Stade-Lausanne	SLO
FC Schaffhausen	FCS
FC Aarau	AAR
FC Winterthur	WIN
FC Wil 1900	Wil
SC Kriens	SCK
Neuchâtel Xamax	XAM
Yverdon Sport	YS

```

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 benennen network.vertex.names(transfer_net) <- c(
# 'BSC Young Boys', 'FC Basel', 'Servette FC', 'FC Lugano',
# 'FC Luzern', 'Lausanne-Sport', 'FC St. Gallen', 'FC
# Zürich', 'FC Sion', 'Grasshoppers', 'FC Vaduz', 'FC
# Thun', 'Stade-Lausanne', 'FC Schaffhausen', 'FC Aarau',
# 'FC Winterthur', 'FC Wil 1900', 'SC Kriens', 'Neuchâtel
# Xamax', 'Yverdon Sport' )
# 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
set.vertex.attribute(transfer_net, "league", c("SL", "SL", "SL",
    "SL", "SL", "SL", "SL", "SL", "ChL", "ChL", "ChL",
    "ChL", "ChL", "ChL", "ChL", "ChL", "ChL", "ChL"))
# Kanten-Attribut 'spendings' definieren: Transfersumme in
# Tsd.
spendings <- read.csv("data/Spendings.csv", sep = ";")
set.edge.attribute(transfer_net, "spendings", spendings)
# Kanten-Attribut 'type' definieren:
# Spieler-Transaktionstyp (Ablöse, Leihe, unbekannt)
type <- read.csv("data/Type_ID.csv", sep = ";")
set.edge.attribute(transfer_net, "type", type)
# Knoten-Attribut 'alldeg' definieren (abgekürzte Variante)
transfer_net %v% "alldeg" <- degree(transfer_net)
summary(transfer_net, print.adj = FALSE)

```

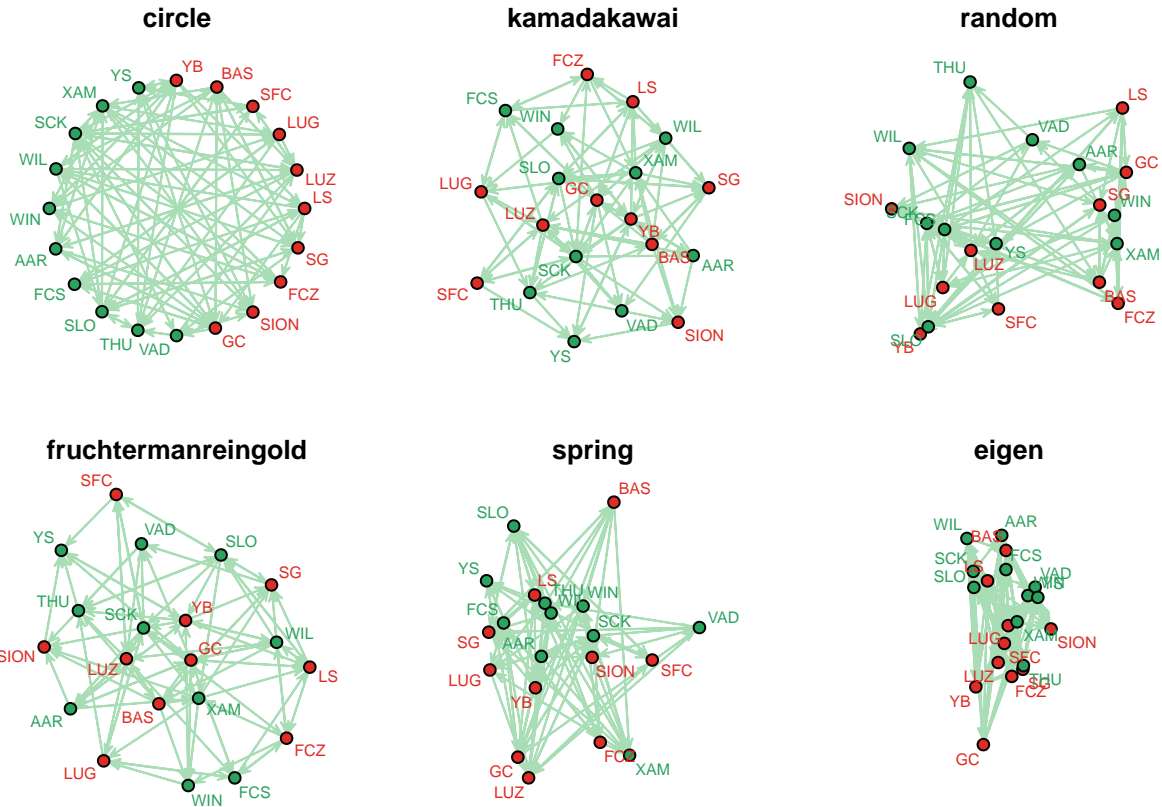
```

## Network attributes:
##   vertices = 20
##   directed = TRUE
##   hyper = FALSE
##   loops = FALSE
##   multiple = FALSE
##   bipartite = FALSE
##   total edges = 92
##   missing edges = 0
##   non-missing edges = 92
##   density = 0.2421053
##
## Vertex attributes:
##
##   alldeg:
##     numeric valued attribute
##     attribute summary:
##       Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##       6.00   7.75   9.00   9.20  11.25  15.00
##
##   league:
##     character valued attribute
##     attribute summary:
##   ChL  SL

```

```
## 10 10
## vertex.names:
## character valued attribute
## 20 valid vertex names
##
## Edge attributes:
##
## spendings:
## integer valued attribute
## 92values
##
## type:
## integer valued attribute
## 92values
```

```
my_pal = c("#2ca25f", "#de2d26")
league_cat = as.factor(get.vertex.attribute(transfer_net, "league"))
edge_pal = c("#a8ddb5")
set.seed(5)
par(mar = c(2, 0, 2, 0), mfrow = c(2, 3))
gplot(transfer_net, vertex.col = my_pal[league_cat], displaylabels = TRUE,
      label.cex = 0.75, edge.col = edge_pal, edge.lwd = 0.5, mode = "circle",
      main = "circle", label.col = my_pal[league_cat])
gplot(transfer_net, vertex.col = my_pal[league_cat], displaylabels = TRUE,
      label.cex = 0.75, edge.col = edge_pal, edge.lwd = 0.5, mode = "kamadakawai",
      main = "kamadakawai", label.col = my_pal[league_cat])
gplot(transfer_net, vertex.col = my_pal[league_cat], displaylabels = TRUE,
      label.cex = 0.75, edge.col = edge_pal, edge.lwd = 0.5, mode = "random",
      main = "random", label.col = my_pal[league_cat])
gplot(transfer_net, vertex.col = my_pal[league_cat], displaylabels = TRUE,
      label.cex = 0.75, edge.col = edge_pal, edge.lwd = 0.5, mode = "fruchtermanreingold",
      main = "fruchtermanreingold", label.col = my_pal[league_cat])
gplot(transfer_net, vertex.col = my_pal[league_cat], displaylabels = TRUE,
      label.cex = 0.75, edge.col = edge_pal, edge.lwd = 0.5, mode = "spring",
      main = "spring", label.col = my_pal[league_cat])
gplot(transfer_net, vertex.col = my_pal[league_cat], displaylabels = TRUE,
      label.cex = 0.75, edge.col = edge_pal, edge.lwd = 0.5, mode = "eigen",
      main = "eigen", label.col = my_pal[league_cat])
```

```
bet <- 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 <- 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
```

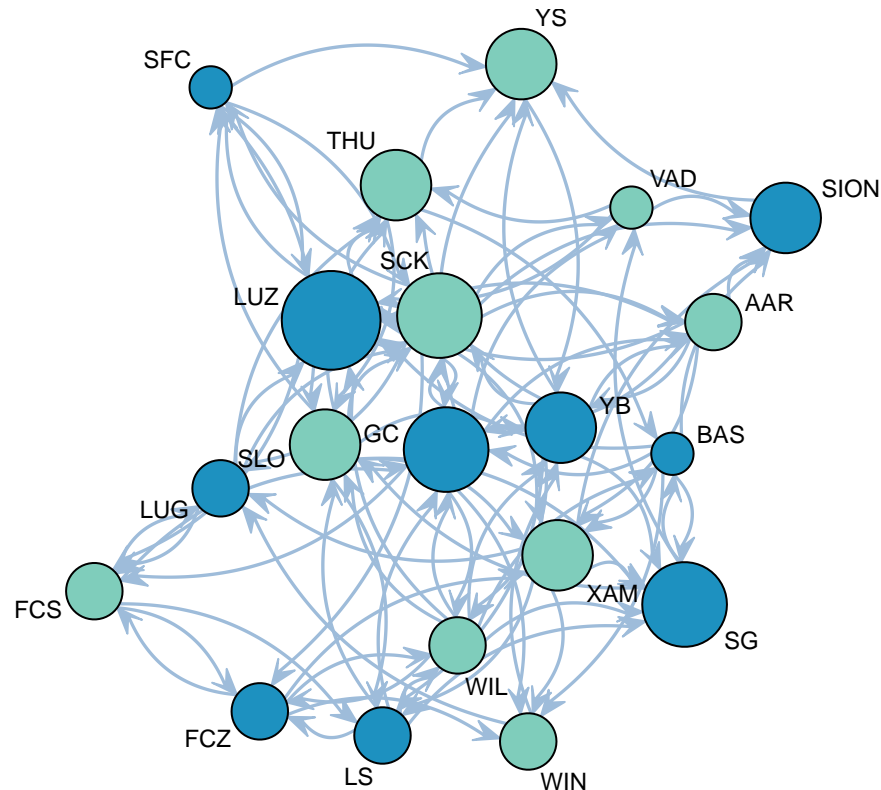
```
cls <- closeness(transfer_net, gmode = "graph")
cls
```

```
## [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(0, 0, 0, 0), mfrow = c(1, 1))
vertex_col <- c("#7fcdbb", "#1d91c0")
edge_col <- c("#9ebcda")
set.seed(5)
par(mar = c(1, 0, 1, 0), mfrow = c(1, 1))
gplot(transfer_net, vertex.col = vertex_col[league_cat], displaylabels = TRUE,
      vertex.cex = deg/1.95, edge.col = edge_col, mode = "fruchtermanreingold",
      label.cex = 0.75, label.pos = 0, usecurve = TRUE)

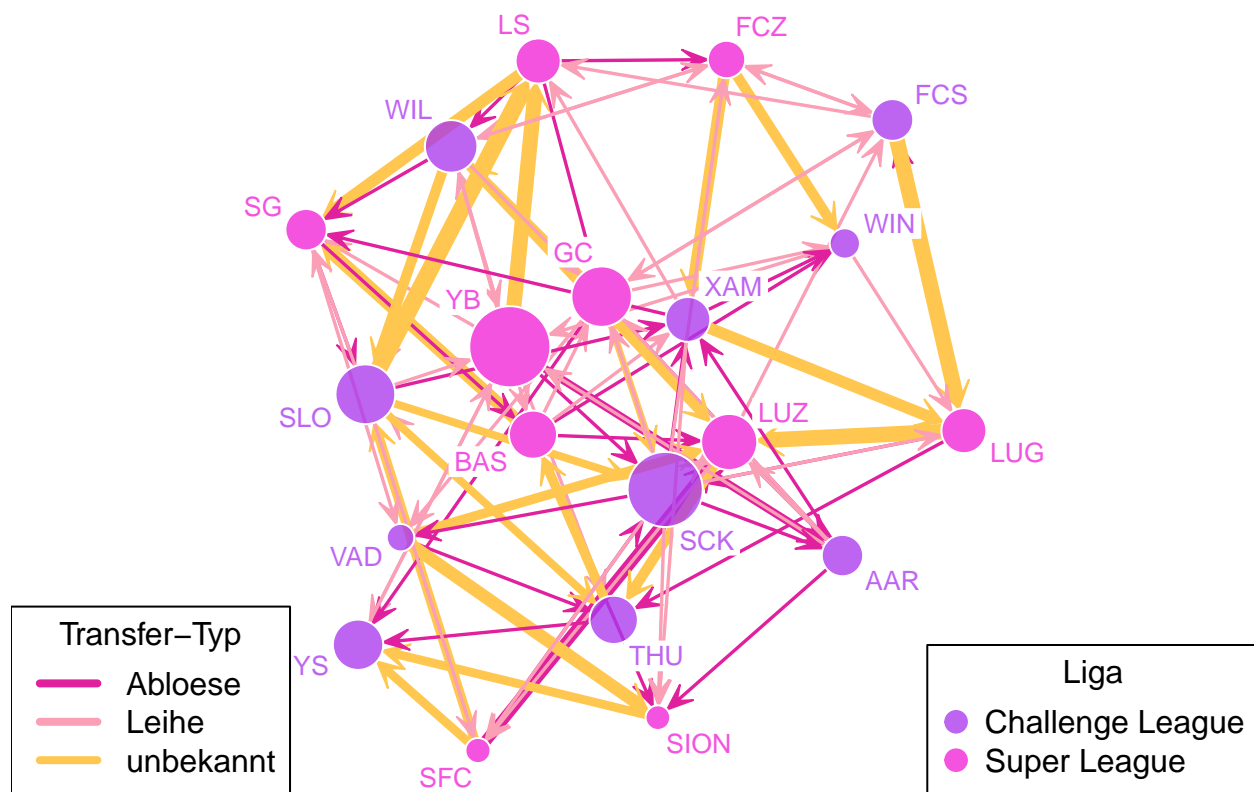
```



```

par(mar = c(1, 0, 1, 0), mfrow = c(1, 1))
linecol_pal <- c("#e0209d", "#fa9fb5", "#ffc74f")
league_pal = c("#aa36e4", "#f353de")
type_cat <- as.factor(get.edge.attribute(transfer_net, "type"))
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")

```



```
# knitr::purl('Heimarbeit2.rmd')
```

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