

Your Title

```
# Installieren von Abhängigkeiten
# install.packages("igraph")
# install.packages("igraphdata")

library(igraph)

##
## Attaching package: 'igraph'

## The following objects are masked from 'package:stats':
##
##      decompose, spectrum

## The following object is masked from 'package:base':
##
##      union

library(igraphdata)
```

Gliederung

1. Einleitung und Forschungsfrage
2. Analysestrategie
3. Analyse 3.1 Degree-Centrality 3.2 Betweenness-Centrality 3.3 Closeness-Centrality
4. Conclusion
5. Anhang

1. Einleitung und Forschungsfrage

Die Folgende Arbeit führte eine Analyse des sozialen Netzwerkes der Fakultät einer Britischen Universität durch. Dazu wird der igraphdata-Datensatz “UKfaculty” verwendet, welcher die Beziehungen einer Gruppe von Studierenden gewichtet und gerichtet enthält.

```
# Daten laden
data("UKfaculty")
```

Daten untersuchen

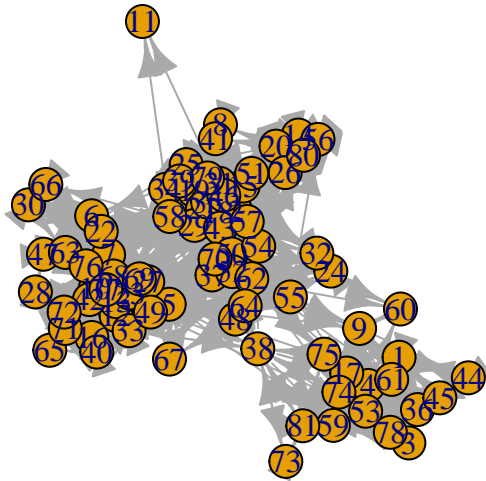
```
# Gesamtdaten
class(UKfaculty)

## [1] "igraph"

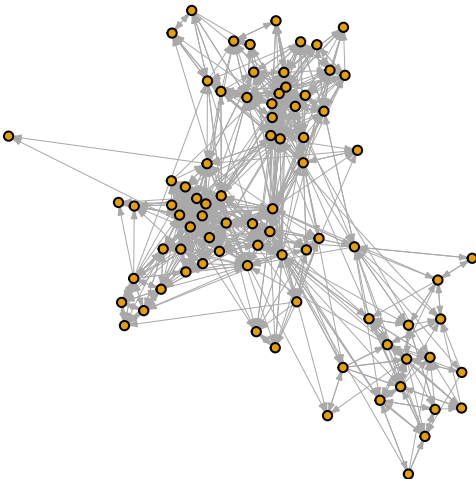
# View(UKfaculty) # Dieser Befehl führt zu Problemen beim knitten und ist daher auskommentiert

# Netzwerk roh abbilden
plot(UKfaculty)
```

```
## This graph was created by an old(er) igraph version.
## Call upgrade_graph() on it to use with the current igraph version
## For now we convert it on the fly...
```



```
# Netzwerk vereinfacht und entzerrt abbilden
plot(UKfaculty, vertex.size = 4, vertex.label = NA, edge.arrow.size = 0.2, edge.width = 0.5)
```



```
summary(UKfaculty)
```

```
## IGRAPH 6f42903 D-W- 81 817 --
## + attr: Type (g/c), Date (g/c), Citation (g/c), Author (g/c), Group
## | (v/n), weight (e/n)
```

```
# Ergebnis: 81 Vertices (Personen) und 817 Edges (Verbindungen)
```

```
V(UKfaculty)$Group
```

```
## [1] 3 1 3 3 2 2 2 1 3 2 1 2 2 1 1 2 3 1 1 1 1 2 2 1 1 1 2 2 1 2 1 1 2 1 1 3 1 3
## [39] 1 2 1 2 1 3 3 1 2 1 2 4 1 1 3 1 1 1 1 1 3 3 3 3 2 1 2 2 2 2 2 4 2 2 3 3 3 2
## [77] 2 3 1 1 3
```

```
E(UKfaculty)$weight
```

```
## [1] 4 14 4 4 10 2 6 2 4 4 1 3 1 2 2 2 8 1 2 1 10 6 3 8 2
## [26] 4 1 4 2 1 4 2 1 3 1 4 14 2 10 10 1 2 2 3 8 1 2 4 1 3
## [51] 2 4 1 14 2 8 4 1 1 1 1 2 10 12 6 2 14 4 2 2 8 4 6 1 3
```

```

## [76] 2 1 10 12 4 2 6 2 16 14 8 5 8 1 1 1 1 6 10 6 1 1 6 12 4
## [101] 2 12 2 8 1 1 2 12 8 2 10 1 6 1 2 4 12 2 4 14 16 1 2 2 14
## [126] 1 1 4 10 1 1 14 3 6 4 10 8 3 8 1 2 2 2 6 3 4 14 8 4 2
## [151] 1 12 1 10 14 6 12 6 2 16 6 4 12 2 2 12 1 14 12 10 6 1 6 16 3
## [176] 6 2 1 1 4 14 1 6 2 2 7 10 1 2 1 3 1 4 1 8 2 6 6 2 8
## [201] 1 2 1 1 4 6 4 6 2 10 2 2 4 14 4 2 2 2 16 2 2 1 2 2 10
## [226] 2 1 1 14 1 16 1 4 1 4 2 1 6 1 2 8 2 1 1 8 2 1 5 2 4
## [251] 4 2 10 12 14 2 2 2 2 2 1 6 1 1 6 2 10 4 1 12 2 2 14 2 1
## [276] 3 1 14 2 4 1 1 16 2 1 2 16 2 6 2 1 1 2 1 8 4 1 2 10 6
## [301] 4 2 4 1 2 3 1 8 2 14 2 8 2 1 6 1 1 8 1 1 1 2 8 4 2
## [326] 2 2 1 3 1 1 1 16 12 4 14 2 1 4 3 12 16 6 8 10 14 6 1 1 4
## [351] 16 2 6 8 2 2 1 1 2 6 2 1 2 2 2 14 2 1 3 1 2 4 2 4 4
## [376] 2 4 4 1 6 6 2 1 12 10 4 6 8 1 1 2 14 1 6 6 1 1 6 12 6
## [401] 2 6 12 1 16 4 6 2 8 14 2 4 4 2 1 12 6 4 1 2 2 1 1 16 1
## [426] 2 1 2 4 8 1 1 12 2 1 2 12 6 2 6 2 4 4 1 3 8 2 4 4 4
## [451] 6 2 2 2 2 4 2 4 6 10 10 6 2 6 1 2 2 4 8 2 2 2 2 2 3
## [476] 6 1 2 3 2 2 1 1 1 1 1 10 14 2 2 4 4 1 1 6 1 1 1 2 14
## [501] 4 1 6 2 4 3 6 3 12 6 8 4 1 10 8 8 4 2 2 1 1 6 10 3 8
## [526] 2 3 2 10 6 10 12 14 4 6 8 8 4 4 1 1 4 1 8 1 16 1 1 4 1
## [551] 8 10 2 6 6 8 1 3 3 2 4 6 1 12 4 4 1 2 2 8 12 6 1 16 16
## [576] 1 2 6 1 6 2 2 8 1 2 4 1 2 1 6 2 2 4 10 14 2 4 4 1 2
## [601] 2 2 16 16 1 2 4 2 1 2 1 2 1 2 2 4 16 5 2 3 10 8 2 4 2
## [626] 2 2 2 2 8 1 1 1 2 12 2 1 1 8 2 1 14 1 1 2 6 1 5 12 12
## [651] 4 1 12 6 4 10 1 6 1 2 12 12 1 4 4 8 16 1 1 2 1 6 8 3 12
## [676] 2 2 1 2 6 6 1 6 10 2 4 2 7 2 1 12 1 4 1 1 1 1 1 1 2
## [701] 2 2 1 4 12 4 10 8 1 3 4 1 14 12 10 6 6 12 1 2 1 2 2 2 3
## [726] 1 2 10 8 4 4 8 1 12 1 1 4 4 2 1 8 4 12 6 1 1 14 2 1 4
## [751] 6 1 4 12 10 6 8 4 1 14 1 1 2 1 2 4 2 1 1 3 4 16 4 5 4
## [776] 1 6 4 3 1 10 2 2 2 14 14 2 2 6 2 2 2 2 2 8 2 6 1 1 2
## [801] 1 4 1 14 6 4 4 2 6 1 8 2 2 16 1 1 2

```

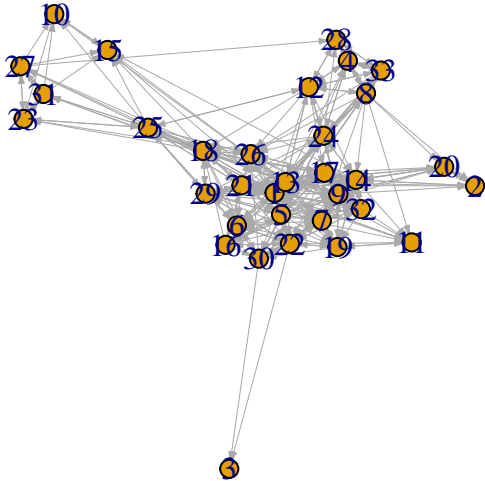
Im aufgezeichneten Netzwerk der Universität befinden sich 81 Personen (Vertices), welche durch 817 Verbindungen (Edges) miteinander verbunden sind. Durch die hohe Anzahl an Verbindungen lässt sich aus der ersten Darstellung des Netzwerkes nur wenig erkennen. Die 81 Personen sind in 4 Gruppen (1-4) eingeteilt, deren wissenschaftlicher Hintergrund sich jedoch nicht identifizieren lässt. Es ist zu vermuten, dass es sich dabei um einzelne Fakultäten handelt. Im Folgenden werden die einzelnen Gruppen genauer untersucht.

Edges untersuchen - Gruppen

```
# Untersuchen der einzelnen Gruppen:
```

```
# Gruppe 1:
```

```
UKfaculty_group1 <- delete_vertices(UKfaculty, V(UKfaculty)[Group != 1])
plot(UKfaculty_group1, vertex.size = 8, edge.arrow.size = 0.2, edge.width = 0.5)
```

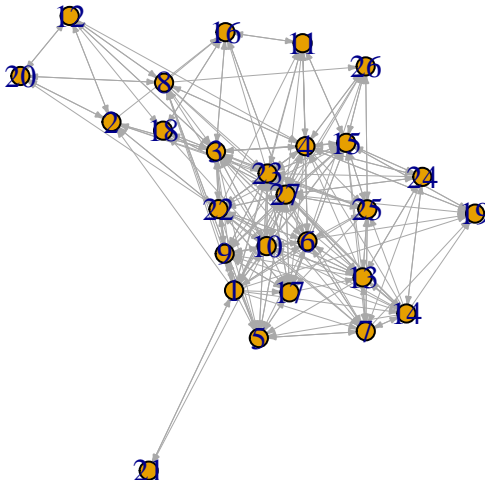


```
summary(UKfaculty_group1)
```

```
## IGRAPH 3e3c0b8 D-W- 33 317 --
## + attr: Type (g/c), Date (g/c), Citation (g/c), Author (g/c), Group
## | (v/n), weight (e/n)
```

```
# Gruppe 2:
```

```
UKfaculty_group2 <- delete_vertices(UKfaculty, V(UKfaculty)[Group != 2])
plot(UKfaculty_group2, vertex.size = 8, edge.arrow.size = 0.2, edge.width = 0.5)
```

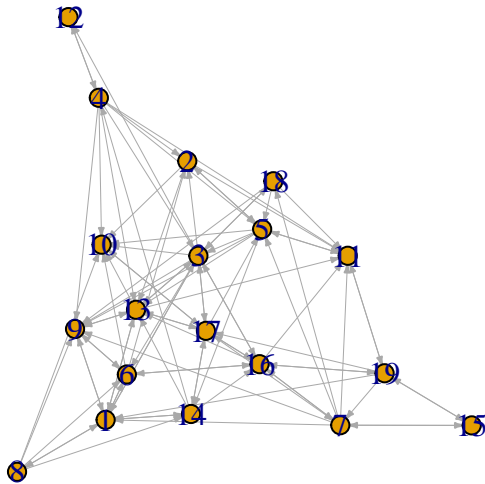


```
summary(UKfaculty_group2)
```

```
## IGRAPH 820a597 D-W- 27 250 --
## + attr: Type (g/c), Date (g/c), Citation (g/c), Author (g/c), Group
## | (v/n), weight (e/n)
```

```
# Gruppe 3:
```

```
UKfaculty_group3 <- delete_vertices(UKfaculty, V(UKfaculty)[Group != 3])
plot(UKfaculty_group3, vertex.size = 8, edge.arrow.size = 0.2, edge.width = 0.5)
```

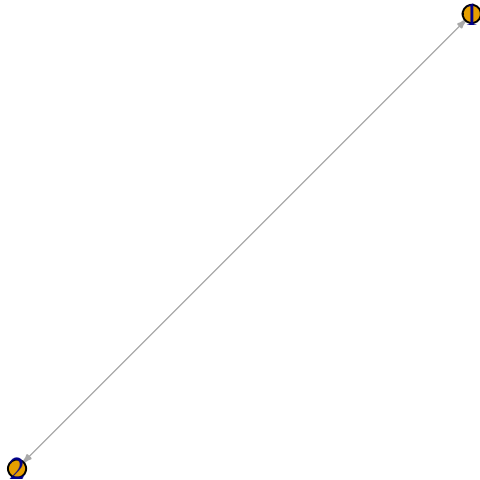


```
summary(UKfaculty_group3)
```

```
## IGRAPH 59e0691 D-W- 19 96 --
## + attr: Type (g/c), Date (g/c), Citation (g/c), Author (g/c), Group
## | (v/n), weight (e/n)
```

Gruppe 4:

```
UKfaculty_group4 <- delete_vertices(UKfaculty, V(UKfaculty)[Group != 4])
plot(UKfaculty_group4, vertex.size = 8, edge.arrow.size = 0.2, edge.width = 0.5)
```



```
summary(UKfaculty_group4)
```

```
## IGRAPH 09c3789 D-W- 2 2 --
## + attr: Type (g/c), Date (g/c), Citation (g/c), Author (g/c), Group
## | (v/n), weight (e/n)
```

Auffällig bei der Untersuchung der Gruppen sind die unterschiedliche Gruppengrößen. Während Gruppe 1 mit 33 Mitglieder die größte Gruppe darstellt, enthält Gruppe 4 nur 2 Mitglieder. Auch die Verbindungs-dichte, die Anzahl der Verbindungen innerhalb der Gruppen, unterscheidet sich stark.

Übersichtliche Zusammenfassung der Ergebnisse: - Gruppe 1: 33 Edges / 317 Vertices - Gruppe 2: 25 Edges / 250 Vertices - Gruppe 3: 19 Edges / 96 Vertices - Gruppe 4: 2 Edges / 2 Vertices

Edges untersuchen - Gewichtung

```
edges_weight <- (E(UKfaculty)$weight)
class(edges_weight)
```

```
## [1] "numeric"
```

```
edges_weight
```

```
## [1] 4 14 4 4 10 2 6 2 4 4 1 3 1 2 2 2 8 1 2 1 10 6 3 8 2
## [26] 4 1 4 2 1 4 2 1 3 1 4 14 2 10 10 1 2 2 3 8 1 2 4 1 3
## [51] 2 4 1 14 2 8 4 1 1 1 1 2 10 12 6 2 14 4 2 2 8 4 6 1 3
## [76] 2 1 10 12 4 2 6 2 16 14 8 5 8 1 1 1 1 6 10 6 1 1 6 12 4
## [101] 2 12 2 8 1 1 2 12 8 2 10 1 6 1 2 4 12 2 4 14 16 1 2 2 14
## [126] 1 1 4 10 1 1 14 3 6 4 10 8 3 8 1 2 2 2 6 3 4 14 8 4 2
## [151] 1 12 1 10 14 6 12 6 2 16 6 4 12 2 2 12 1 14 12 10 6 1 6 16 3
## [176] 6 2 1 1 4 14 1 6 2 2 7 10 1 2 1 3 1 4 1 8 2 6 6 2 8
## [201] 1 2 1 1 4 6 4 6 2 10 2 2 4 14 4 2 2 2 16 2 2 1 2 2 10
## [226] 2 1 1 14 1 16 1 4 1 4 2 1 6 1 2 8 2 1 1 8 2 1 5 2 4
## [251] 4 2 10 12 14 2 2 2 2 2 1 6 1 1 6 2 10 4 1 12 2 2 14 2 1
## [276] 3 1 14 2 4 1 1 16 2 1 2 16 2 6 2 1 1 2 1 8 4 1 2 10 6
## [301] 4 2 4 1 2 3 1 8 2 14 2 8 2 1 6 1 1 8 1 1 1 2 8 4 2
## [326] 2 2 1 3 1 1 1 16 12 4 14 2 1 4 3 12 16 6 8 10 14 6 1 1 4
## [351] 16 2 6 8 2 2 1 1 2 6 2 1 2 2 2 14 2 1 3 1 2 4 2 4 4
## [376] 2 4 4 1 6 6 2 1 12 10 4 6 8 1 1 2 14 1 6 6 1 1 6 12 6
## [401] 2 6 12 1 16 4 6 2 8 14 2 4 4 2 1 12 6 4 1 2 2 1 1 16 1
## [426] 2 1 2 4 8 1 1 12 2 1 2 12 6 2 6 2 4 4 1 3 8 2 4 4 4
## [451] 6 2 2 2 2 4 2 4 6 10 10 6 2 6 1 2 2 4 8 2 2 2 2 2 3
## [476] 6 1 2 3 2 2 1 1 1 1 1 10 14 2 2 4 4 1 1 6 1 1 1 2 14
## [501] 4 1 6 2 4 3 6 3 12 6 8 4 1 10 8 8 4 2 2 1 1 6 10 3 8
## [526] 2 3 2 10 6 10 12 14 4 6 8 8 4 4 1 1 4 1 8 1 16 1 1 4 1
## [551] 8 10 2 6 6 8 1 3 3 2 4 6 1 12 4 4 1 2 2 8 12 6 1 16 16
## [576] 1 2 6 1 6 2 2 8 1 2 4 1 2 1 6 2 2 4 10 14 2 4 4 1 2
## [601] 2 2 16 16 1 2 4 2 1 2 1 2 1 2 2 4 16 5 2 3 10 8 2 4 2
## [626] 2 2 2 2 8 1 1 1 2 12 2 1 1 8 2 1 14 1 1 2 6 1 5 12 12
## [651] 4 1 12 6 4 10 1 6 1 2 12 12 1 4 4 8 16 1 1 2 1 6 8 3 12
## [676] 2 2 1 2 6 6 1 6 10 2 4 2 7 2 1 12 1 4 1 1 1 1 1 1 2
## [701] 2 2 1 4 12 4 10 8 1 3 4 1 14 12 10 6 6 12 1 2 1 2 2 2 3
## [726] 1 2 10 8 4 4 8 1 12 1 1 4 4 2 1 8 4 12 6 1 1 14 2 1 4
## [751] 6 1 4 12 10 6 8 4 1 14 1 1 2 1 2 4 2 1 1 3 4 16 4 5 4
## [776] 1 6 4 3 1 10 2 2 2 14 14 2 2 6 2 2 2 2 2 8 2 6 1 1 2
## [801] 1 4 1 14 6 4 4 2 6 1 8 2 2 16 1 1 2
```

```
# Vorhandene Gewichtungen mit Häufigkeit
as.data.frame(table(edges_weight))
```

```
## edges_weight Freq
## 1 1 202
## 2 2 207
## 3 3 31
## 4 4 107
## 5 5 5
## 6 6 80
## 7 7 2
## 8 8 51
## 9 10 37
```

```
## 10      12    39
## 11      14    34
## 12      16    22
```

```
# Gesamtgewichtung
sum(edges_weight)
```

```
## [1] 3730
```

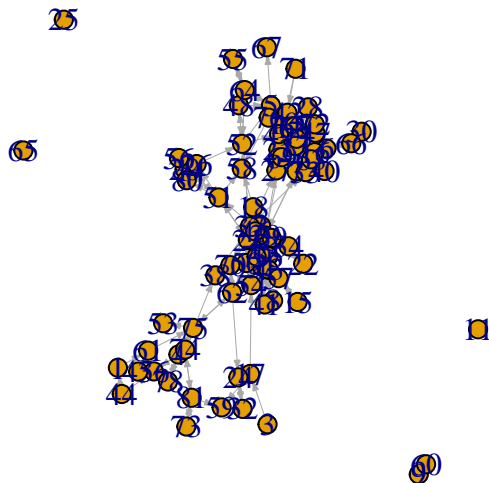
```
# durchschnittliche Gewichtung (Durchschnitt)
mean(edges_weight)
```

```
## [1] 4.565483
```

```
# durchschnittliche Gewichtung (Median)
median(edges_weight)
```

```
## [1] 2
```

```
# Plot für bestimmte Gewichtung
UKfaculty_min5 <- delete_edges(UKfaculty, E(UKfaculty)[weight <= 5])
plot(UKfaculty_min5, vertex.size = 8, edge.arrow.size = 0.2, edge.width = 0.5)
```



Die Beziehungen der einzelnen Studierenden sind unterschiedlich gewichtet, wodurch die Häufigkeit oder Intensität der einzelnen Beziehungen dargestellt wird. Bei einer Untersuchung der Gewichtungen lässt sich feststellen, dass die am häufigsten vorkommende Gewichtung 1 und 2 ist. Insgesamt lassen sich viele niedrige Gewichtungen erkennen, wobei die durchschnittliche Gewichtung bei 4,57 liegt.

Forschungsfrage

Im Folgenden wird untersucht, welche Person die größte Wichtigkeit für das soziale Netzwerk der Universität darstellt. Die Forschungsfrage lautet: "Wer ist am wichtigsten für den Zusammenhalt in der Gruppe".

2. Analysestrategie

Um die Forschungsfrage ausführlich zu beantworten, werden verschiedene Analysestrategien angewendet, welche am Ende zusammengeführt werden. Die Teilnehmenden des Netzwerks werden auf drei Verschiedene Arten auf ihre Wichtigkeit respektive Zentralität und Stellung im Netzwerk untersucht. Zuerst wird dazu die Degree-Centrality - die Anzahl der Kontakte - jeder Person untersucht. Danach wird die Closeness und letztlich die Betweenness errechnet. Für jeden dieser untersuchten Parameter wird ein Rang von 1-81 innerhalb des Netzwerks vergeben, wodurch am Ende durch das aufsummieren der einzelnen Ränge ein

Gesamt-Rang ermittelt werden kann. Die Person mit dem niedrigsten Gesamt-Rang wird als am wichtigsten für das Netzwerk angenommen.

3. Analyse

In diesem Kapitel wird das Netzwerk auf die einzelnen Parameter untersucht. Dabei wird jeweils ein Scatterplot erstellt, in welchem die gesamte Verteilung der Werte erkennbar ist. Zusätzlich wird eine sortierte Tabelle der Ergebnisse abgebildet, in welche die exakten Werte der Parameter nachzulesen sind.

3.1 Wer hat am meisten Kontakte (Degree-Centrality)?

```
# Identifizierungs-Name der Vertices/Personen
personId <- c(as_ids(V(UKfaculty)))
class(personId)

## [1] "integer"

# Zugehörigkeit zu Gruppe
personGroup <- c(V(UKfaculty)$Group)
class(personGroup)

## [1] "numeric"

# Anzahl der Kontakte
personDegree <- c(degree(UKfaculty))
class(personDegree)

## [1] "numeric"

# Zusammenfassung in Dataframe
personDegree_df <- data.frame(name = personId, group = personGroup, degree = personDegree)
personDegree_df
```

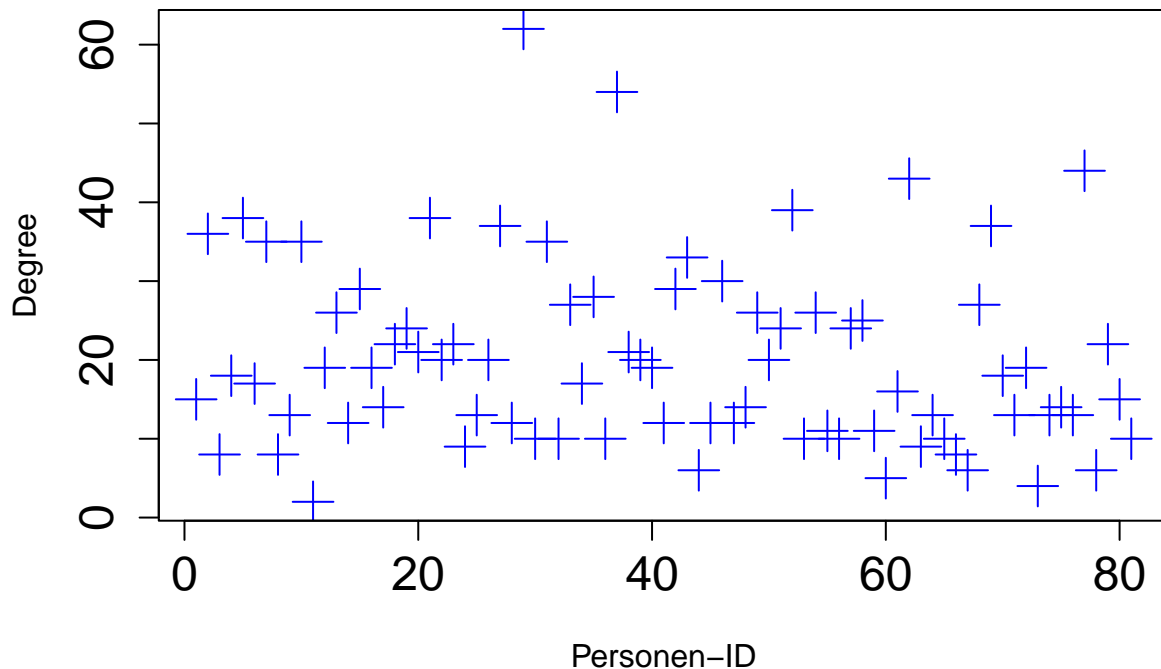
	name	group	degree
## 1	1	3	15
## 2	2	1	36
## 3	3	3	8
## 4	4	3	18
## 5	5	2	38
## 6	6	2	17
## 7	7	2	35
## 8	8	1	8
## 9	9	3	13
## 10	10	2	35
## 11	11	1	2
## 12	12	2	19
## 13	13	2	26
## 14	14	1	12
## 15	15	1	29
## 16	16	2	19
## 17	17	3	14
## 18	18	1	22
## 19	19	1	24
## 20	20	1	21
## 21	21	1	38
## 22	22	2	20

##	23	23	2	22
##	24	24	1	9
##	25	25	1	13
##	26	26	1	20
##	27	27	2	37
##	28	28	2	12
##	29	29	1	62
##	30	30	2	10
##	31	31	1	35
##	32	32	1	10
##	33	33	2	27
##	34	34	1	17
##	35	35	1	28
##	36	36	3	10
##	37	37	1	54
##	38	38	3	21
##	39	39	1	20
##	40	40	2	19
##	41	41	1	12
##	42	42	2	29
##	43	43	1	33
##	44	44	3	6
##	45	45	3	12
##	46	46	1	30
##	47	47	2	12
##	48	48	1	14
##	49	49	2	26
##	50	50	4	20
##	51	51	1	24
##	52	52	1	39
##	53	53	3	10
##	54	54	1	26
##	55	55	1	11
##	56	56	1	10
##	57	57	1	24
##	58	58	1	25
##	59	59	3	11
##	60	60	3	5
##	61	61	3	16
##	62	62	3	43
##	63	63	2	9
##	64	64	1	13
##	65	65	2	10
##	66	66	2	8
##	67	67	2	6
##	68	68	2	27
##	69	69	2	37
##	70	70	4	18
##	71	71	2	13
##	72	72	2	19
##	73	73	3	4
##	74	74	3	13
##	75	75	3	14
##	76	76	2	13

```
## 77 77 2 44
## 78 78 3 6
## 79 79 1 22
## 80 80 1 15
## 81 81 3 10
```

```
# Darstellen der Daten in Scatterplot für Anzahl der Kontakte je Person
plot(personDegree_df$name, personDegree_df$degree,
     xlab = "Personen-ID",
     ylab = "Degree",
     main = "Darstellung der Anzahl der Kontakte (Degree Centrality)",
     pch = 3,
     col = "blue",
     cex.axis = 1.5,
     cex.main = 1.5,
     cex = 2)
```

Darstellung der Anzahl der Kontakte (Degree Centralit



```
# Sortieren nach Anzahl der Kontakte
personDegree_df_sorted <- personDegree_df[order(personDegree_df$degree, decreasing = TRUE),]

# Ausgabe der 10 Personen mit den meisten Kontakten
head(personDegree_df_sorted, 10)
```

```
##   name group degree
## 29  29     1     62
## 37  37     1     54
## 77  77     2     44
## 62  62     3     43
## 52  52     1     39
## 5   5      2     38
## 21  21     1     38
```

```
## 27 27 2 37
## 69 69 2 37
## 2 2 1 36
```

Aus dem Scatterplot lässt sich erkennen, dass sich Person 29 und 37 besonders von den übrigen abheben. Eine tiefergehende Untersuchung dieser Tatsache muss in einer weiteren Untersuchung des Netzwerkes erfolgen.

3.2 Wer hat die größte Betweenness?

```
# Identifizierungs-Name der Vertices/Personen
personId <- c(as_ids(V(UKfaculty)))
class(personId)
```

```
## [1] "integer"
```

```
# Zugehörigkeit zu Gruppe
personGroup <- c(V(UKfaculty)$Group)
class(personGroup)
```

```
## [1] "numeric"
```

```
# Betweenness
personBetweenness <- round(c(betweenness(UKfaculty)), 2)
```

```
# Zusammenfassung in Dataframe
```

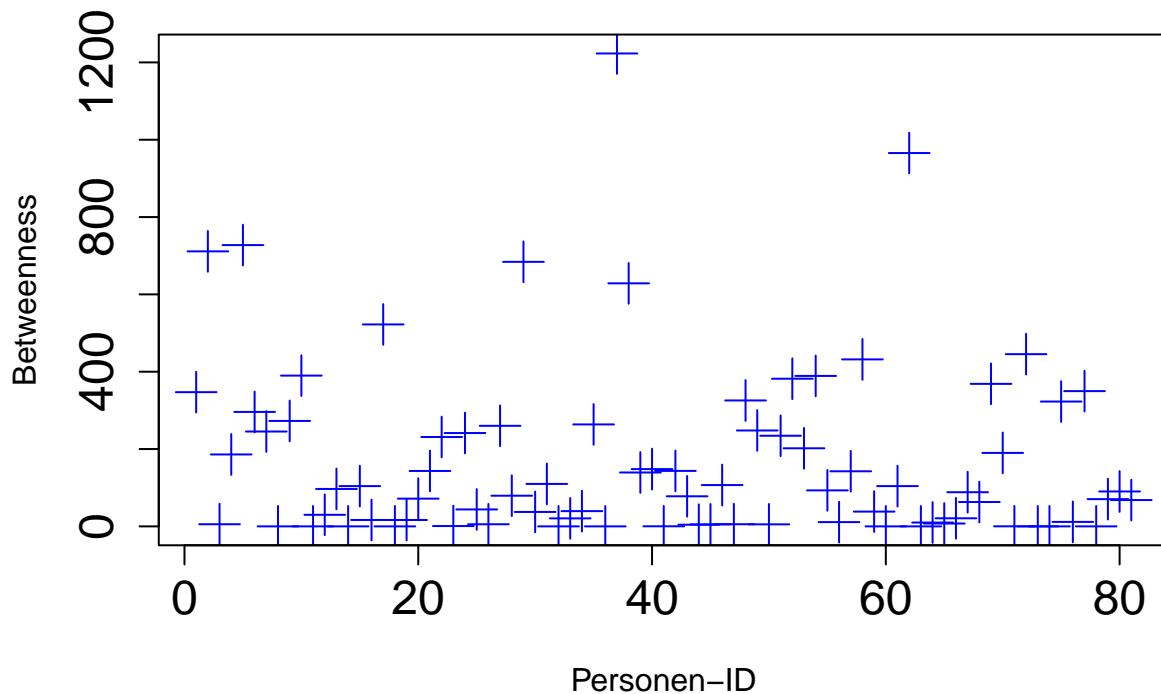
```
personBetweenness_df <- data.frame(name = personId, group = personGroup, betweenness = personBetweenness)
personBetweenness_df
```

```
##   name group betweenness
## 1    1     3    347.12
## 2    2     1    711.28
## 3    3     3     5.50
## 4    4     3    185.98
## 5    5     2    727.46
## 6    6     2    296.01
## 7    7     2    245.29
## 8    8     1     0.00
## 9    9     3    272.71
## 10  10     2    389.92
## 11  11     1     0.00
## 12  12     2     29.90
## 13  13     2     96.77
## 14  14     1     0.00
## 15  15     1    104.09
## 16  16     2     16.50
## 17  17     3    522.26
## 18  18     1     0.00
## 19  19     1     16.46
## 20  20     1     71.69
## 21  21     1    143.32
## 22  22     2    231.11
## 23  23     2     0.83
## 24  24     1    241.39
## 25  25     1     43.78
## 26  26     1     5.42
## 27  27     2    260.07
```

##	28	28	2	79.31
##	29	29	1	684.20
##	30	30	2	37.28
##	31	31	1	109.79
##	32	32	1	0.00
##	33	33	2	20.65
##	34	34	1	39.42
##	35	35	1	263.63
##	36	36	3	0.00
##	37	37	1	1223.08
##	38	38	3	628.55
##	39	39	1	139.35
##	40	40	2	148.15
##	41	41	1	0.20
##	42	42	2	143.46
##	43	43	1	77.80
##	44	44	3	4.27
##	45	45	3	5.17
##	46	46	1	106.81
##	47	47	2	5.57
##	48	48	1	325.63
##	49	49	2	248.02
##	50	50	4	5.23
##	51	51	1	234.38
##	52	52	1	381.84
##	53	53	3	201.90
##	54	54	1	389.11
##	55	55	1	93.11
##	56	56	1	11.00
##	57	57	1	142.47
##	58	58	1	431.86
##	59	59	3	38.20
##	60	60	3	0.00
##	61	61	3	104.10
##	62	62	3	965.60
##	63	63	2	0.00
##	64	64	1	9.94
##	65	65	2	7.33
##	66	66	2	20.87
##	67	67	2	88.33
##	68	68	2	62.92
##	69	69	2	368.58
##	70	70	4	190.02
##	71	71	2	0.12
##	72	72	2	445.42
##	73	73	3	0.00
##	74	74	3	0.00
##	75	75	3	322.80
##	76	76	2	11.53
##	77	77	2	349.77
##	78	78	3	0.00
##	79	79	1	70.39
##	80	80	1	90.25
##	81	81	3	68.02

```
# Darstellen der Daten in Scatterplot für Betweenness je Person
plot(personBetweenness_df$name, personBetweenness_df$betweenness,
     xlab = "Personen-ID",
     ylab = "Betweenness",
     main = "Darstellung der Betweenness",
     pch = 3,
     col = "blue",
     cex.axis = 1.5,
     cex.main = 1.5,
     cex = 2)
```

Darstellung der Betweenness



```
# Sortieren nach Betweenness
personBetweenness_df_sorted <- personBetweenness_df[order(personBetweenness_df$betweenness, decreasing = TRUE), ]
personBetweenness_df_sorted
```

##	name	group	betweenness
## 37	37	1	1223.08
## 62	62	3	965.60
## 5	5	2	727.46
## 2	2	1	711.28
## 29	29	1	684.20
## 38	38	3	628.55
## 17	17	3	522.26
## 72	72	2	445.42
## 58	58	1	431.86
## 10	10	2	389.92
## 54	54	1	389.11
## 52	52	1	381.84
## 69	69	2	368.58
## 77	77	2	349.77

## 1	1	3	347.12
## 48	48	1	325.63
## 75	75	3	322.80
## 6	6	2	296.01
## 9	9	3	272.71
## 35	35	1	263.63
## 27	27	2	260.07
## 49	49	2	248.02
## 7	7	2	245.29
## 24	24	1	241.39
## 51	51	1	234.38
## 22	22	2	231.11
## 53	53	3	201.90
## 70	70	4	190.02
## 4	4	3	185.98
## 40	40	2	148.15
## 42	42	2	143.46
## 21	21	1	143.32
## 57	57	1	142.47
## 39	39	1	139.35
## 31	31	1	109.79
## 46	46	1	106.81
## 61	61	3	104.10
## 15	15	1	104.09
## 13	13	2	96.77
## 55	55	1	93.11
## 80	80	1	90.25
## 67	67	2	88.33
## 28	28	2	79.31
## 43	43	1	77.80
## 20	20	1	71.69
## 79	79	1	70.39
## 81	81	3	68.02
## 68	68	2	62.92
## 25	25	1	43.78
## 34	34	1	39.42
## 59	59	3	38.20
## 30	30	2	37.28
## 12	12	2	29.90
## 66	66	2	20.87
## 33	33	2	20.65
## 16	16	2	16.50
## 19	19	1	16.46
## 76	76	2	11.53
## 56	56	1	11.00
## 64	64	1	9.94
## 65	65	2	7.33
## 47	47	2	5.57
## 3	3	3	5.50
## 26	26	1	5.42
## 50	50	4	5.23
## 45	45	3	5.17
## 44	44	3	4.27
## 23	23	2	0.83

```
## 41 41 1 0.20
## 71 71 2 0.12
## 8 8 1 0.00
## 11 11 1 0.00
## 14 14 1 0.00
## 18 18 1 0.00
## 32 32 1 0.00
## 36 36 3 0.00
## 60 60 3 0.00
## 63 63 2 0.00
## 73 73 3 0.00
## 74 74 3 0.00
## 78 78 3 0.00
```

```
# Ausgabe der 10 Personen mit den meisten Kontakten
head(personBetweenness_df_sorted, 10)
```

```
## name group betweenness
## 37 37 1 1223.08
## 62 62 3 965.60
## 5 5 2 727.46
## 2 2 1 711.28
## 29 29 1 684.20
## 38 38 3 628.55
## 17 17 3 522.26
## 72 72 2 445.42
## 58 58 1 431.86
## 10 10 2 389.92
```

```
# Ausgabe der 10 Personen mit den wenigsten Kontakten
tail(personBetweenness_df_sorted, 10)
```

```
## name group betweenness
## 11 11 1 0
## 14 14 1 0
## 18 18 1 0
## 32 32 1 0
## 36 36 3 0
## 60 60 3 0
## 63 63 2 0
## 73 73 3 0
## 74 74 3 0
## 78 78 3 0
```

Eine Betrachtung des Scatterplots zeigt ein hohes Vorkommen niedriger Werte und nur wenige setzen sich davon ab. Es besteht eine hohe Varianz der Werte. Hier ist eine detailliertere Untersuchung der Gegebenheiten in einer weiterführenden Untersuchung des Netzwerkes zu empfehlen.

3.3 Wer hat die größte Closeness?

```
# Identifizierungs-Name der Vertices/Personen
personId <- c(as_ids(V(UKfaculty)))

# Zugehörigkeit zu Gruppe
personGroup <- c(V(UKfaculty)$Group)
```

```

# Closeness
personCloseness <- c(closeness(UKfaculty))

# Zusammenfassung in Dataframe
personCloseness_df <- data.frame(name = personId, group = personGroup, closeness = personCloseness)
personCloseness_df

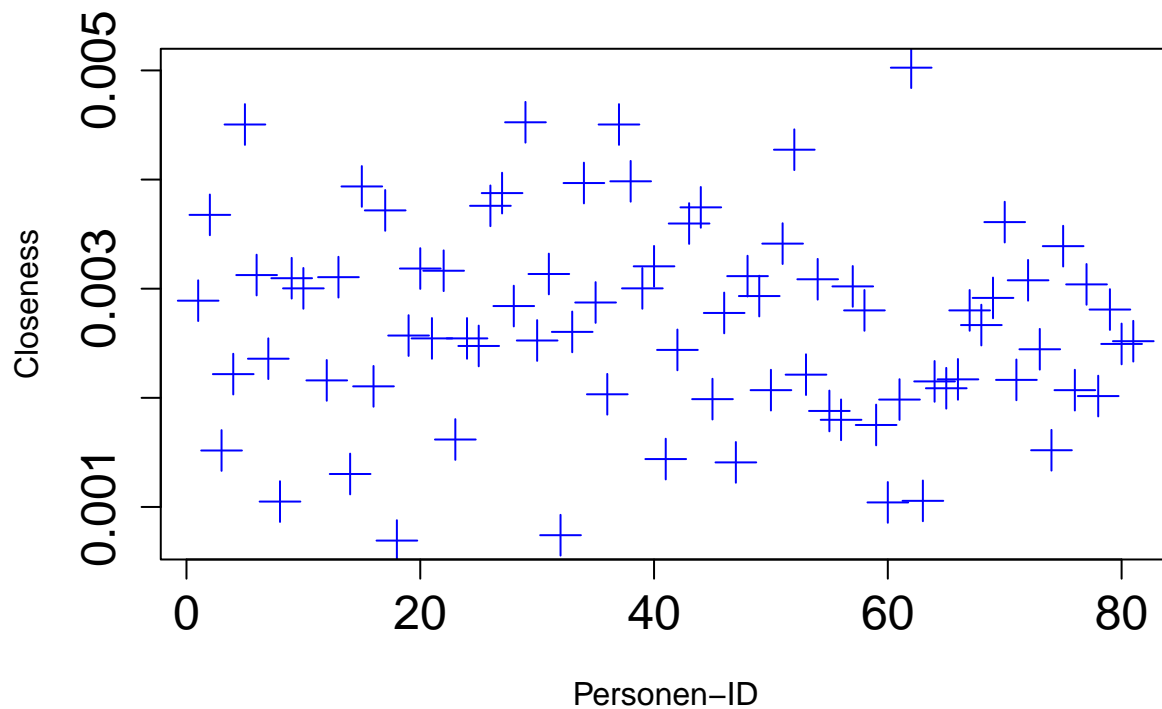
```

##	name	group	closeness
## 1	1	3	0.0028901734
## 2	2	1	0.0036764706
## 3	3	3	0.0015174507
## 4	4	3	0.0022172949
## 5	5	2	0.0045045045
## 6	6	2	0.0031250000
## 7	7	2	0.0023584906
## 8	8	1	0.0010493179
## 9	9	3	0.0030959752
## 10	10	2	0.0030030030
## 11	11	1	NaN
## 12	12	2	0.0021598272
## 13	13	2	0.0031055901
## 14	14	1	0.0013020833
## 15	15	1	0.0039370079
## 16	16	2	0.0021052632
## 17	17	3	0.0037174721
## 18	18	1	0.0006920415
## 19	19	1	0.0025706941
## 20	20	1	0.0031847134
## 21	21	1	0.0025445293
## 22	22	2	0.0031645570
## 23	23	2	0.0016181230
## 24	24	1	0.0025445293
## 25	25	1	0.0024752475
## 26	26	1	0.0037593985
## 27	27	2	0.0038759690
## 28	28	2	0.0028409091
## 29	29	1	0.0045248869
## 30	30	2	0.0025252525
## 31	31	1	0.0031347962
## 32	32	1	0.0007412898
## 33	33	2	0.0026041667
## 34	34	1	0.0039682540
## 35	35	1	0.0028735632
## 36	36	3	0.0020325203
## 37	37	1	0.0045045045
## 38	38	3	0.0039840637
## 39	39	1	0.0030030030
## 40	40	2	0.0032051282
## 41	41	1	0.0014388489
## 42	42	2	0.0024390244
## 43	43	1	0.0035971223
## 44	44	3	0.0037453184
## 45	45	3	0.0019880716
## 46	46	1	0.0027777778


```
## 47 47 2 0.0014084507
## 48 48 1 0.0031152648
## 49 49 2 0.0029325513
## 50 50 4 0.0020703934
## 51 51 1 0.0034129693
## 52 52 1 0.0042735043
## 53 53 3 0.0022123894
## 54 54 1 0.0030864198
## 55 55 1 0.0018796992
## 56 56 1 0.0017985612
## 57 57 1 0.0030211480
## 58 58 1 0.0028011204
## 59 59 3 0.0017513135
## 60 60 3 0.0010416667
## 61 61 3 0.0019841270
## 62 62 3 0.0050251256
## 63 63 2 0.0010559662
## 64 64 1 0.0021505376
## 65 65 2 0.0020876827
## 66 66 2 0.0021691974
## 67 67 2 0.0028011204
## 68 68 2 0.0026666667
## 69 69 2 0.0029154519
## 70 70 4 0.0036101083
## 71 71 2 0.0021645022
## 72 72 2 0.0030769231
## 73 73 3 0.0024449878
## 74 74 3 0.0015197568
## 75 75 3 0.0033898305
## 76 76 2 0.0020703934
## 77 77 2 0.0030395137
## 78 78 3 0.0020161290
## 79 79 1 0.0028089888
## 80 80 1 0.0024937656
## 81 81 3 0.0025188917
```

```
# Darstellen der Daten in Scatterplot für Closeness je Person
plot(personCloseness_df$name, personCloseness_df$closeness,
      xlab = "Personen-ID",
      ylab = "Closeness",
      main = "Darstellung der Closeness",
      pch = 3,
      col = "blue",
      cex.axis = 1.5,
      cex.main = 1.5,
      cex = 2)
```

Darstellung der Closeness



```
# Sortieren nach Closeness
```

```
personCloseness_df_sorted <- personCloseness_df[order(personCloseness_df$closeness, decreasing = TRUE),]
personCloseness_df_sorted
```

##	name	group	closeness
## 62	62	3	0.0050251256
## 29	29	1	0.0045248869
## 5	5	2	0.0045045045
## 37	37	1	0.0045045045
## 52	52	1	0.0042735043
## 38	38	3	0.0039840637
## 34	34	1	0.0039682540
## 15	15	1	0.0039370079
## 27	27	2	0.0038759690
## 26	26	1	0.0037593985
## 44	44	3	0.0037453184
## 17	17	3	0.0037174721
## 2	2	1	0.0036764706
## 70	70	4	0.0036101083
## 43	43	1	0.0035971223
## 51	51	1	0.0034129693
## 75	75	3	0.0033898305
## 40	40	2	0.0032051282
## 20	20	1	0.0031847134
## 22	22	2	0.0031645570
## 31	31	1	0.0031347962
## 6	6	2	0.0031250000
## 48	48	1	0.0031152648
## 13	13	2	0.0031055901
## 9	9	3	0.0030959752

## 54	54	1 0.0030864198
## 72	72	2 0.0030769231
## 77	77	2 0.0030395137
## 57	57	1 0.0030211480
## 10	10	2 0.0030030030
## 39	39	1 0.0030030030
## 49	49	2 0.0029325513
## 69	69	2 0.0029154519
## 1	1	3 0.0028901734
## 35	35	1 0.0028735632
## 28	28	2 0.0028409091
## 79	79	1 0.0028089888
## 58	58	1 0.0028011204
## 67	67	2 0.0028011204
## 46	46	1 0.0027777778
## 68	68	2 0.0026666667
## 33	33	2 0.0026041667
## 19	19	1 0.0025706941
## 21	21	1 0.0025445293
## 24	24	1 0.0025445293
## 30	30	2 0.0025252525
## 81	81	3 0.0025188917
## 80	80	1 0.0024937656
## 25	25	1 0.0024752475
## 73	73	3 0.0024449878
## 42	42	2 0.0024390244
## 7	7	2 0.0023584906
## 4	4	3 0.0022172949
## 53	53	3 0.0022123894
## 66	66	2 0.0021691974
## 71	71	2 0.0021645022
## 12	12	2 0.0021598272
## 64	64	1 0.0021505376
## 16	16	2 0.0021052632
## 65	65	2 0.0020876827
## 50	50	4 0.0020703934
## 76	76	2 0.0020703934
## 36	36	3 0.0020325203
## 78	78	3 0.0020161290
## 45	45	3 0.0019880716
## 61	61	3 0.0019841270
## 55	55	1 0.0018796992
## 56	56	1 0.0017985612
## 59	59	3 0.0017513135
## 23	23	2 0.0016181230
## 74	74	3 0.0015197568
## 3	3	3 0.0015174507
## 41	41	1 0.0014388489
## 47	47	2 0.0014084507
## 14	14	1 0.0013020833
## 63	63	2 0.0010559662
## 8	8	1 0.0010493179
## 60	60	3 0.0010416667
## 32	32	1 0.0007412898

```
## 18 18 1 0.0006920415
## 11 11 1 NaN
# Ausgabe der 10 Personen mit der höchsten Closeness
head(personCloseness_df_sorted, 10)
```

```
##   name group closeness
## 62  62     3 0.005025126
## 29  29     1 0.004524887
## 5   5     2 0.004504505
## 37  37     1 0.004504505
## 52  52     1 0.004273504
## 38  38     3 0.003984064
## 34  34     1 0.003968254
## 15  15     1 0.003937008
## 27  27     2 0.003875969
## 26  26     1 0.003759398
```

```
# Ausgabe der 10 Personen mit der niedrigsten Closeness
tail(personCloseness_df_sorted, 10)
```

```
##   name group closeness
## 3    3     3 0.0015174507
## 41   41     1 0.0014388489
## 47   47     2 0.0014084507
## 14   14     1 0.0013020833
## 63   63     2 0.0010559662
## 8    8     1 0.0010493179
## 60   60     3 0.0010416667
## 32   32     1 0.0007412898
## 18   18     1 0.0006920415
## 11   11     1 NaN
```

Neben der vermältnismäßig mittleren Varianz der Werte und einer häufung der Werte zwischen 0.002 und 0.004, die sich im Scatterplot gut erkennen lässt, ist auch eine Person mit einem nicht vorhandenen Closeness-Wert zu erkennen. Auch dies muss in einer weitergehenden Untersuchung genauer betrachtet werden.

4. Conclusion

```
# Einführen eines Rangs für die 3 Kriterien als übergreifende Bewertung
personID <- c(as_ids(V(UKfaculty)))

# Degree:
personDegree_df_sorted$degreeRank <- 1:nrow(personDegree_df_sorted)
personDegree_df_sorted
```

```
##   name group degree degreeRank
## 29  29     1     62           1
## 37  37     1     54           2
## 77  77     2     44           3
## 62  62     3     43           4
## 52  52     1     39           5
## 5   5     2     38           6
## 21  21     1     38           7
## 27  27     2     37           8
```

##	69	69	2	37	9
##	2	2	1	36	10
##	7	7	2	35	11
##	10	10	2	35	12
##	31	31	1	35	13
##	43	43	1	33	14
##	46	46	1	30	15
##	15	15	1	29	16
##	42	42	2	29	17
##	35	35	1	28	18
##	33	33	2	27	19
##	68	68	2	27	20
##	13	13	2	26	21
##	49	49	2	26	22
##	54	54	1	26	23
##	58	58	1	25	24
##	19	19	1	24	25
##	51	51	1	24	26
##	57	57	1	24	27
##	18	18	1	22	28
##	23	23	2	22	29
##	79	79	1	22	30
##	20	20	1	21	31
##	38	38	3	21	32
##	22	22	2	20	33
##	26	26	1	20	34
##	39	39	1	20	35
##	50	50	4	20	36
##	12	12	2	19	37
##	16	16	2	19	38
##	40	40	2	19	39
##	72	72	2	19	40
##	4	4	3	18	41
##	70	70	4	18	42
##	6	6	2	17	43
##	34	34	1	17	44
##	61	61	3	16	45
##	1	1	3	15	46
##	80	80	1	15	47
##	17	17	3	14	48
##	48	48	1	14	49
##	75	75	3	14	50
##	9	9	3	13	51
##	25	25	1	13	52
##	64	64	1	13	53
##	71	71	2	13	54
##	74	74	3	13	55
##	76	76	2	13	56
##	14	14	1	12	57
##	28	28	2	12	58
##	41	41	1	12	59
##	45	45	3	12	60
##	47	47	2	12	61
##	55	55	1	11	62

```
## 59 59 3 11 63
## 30 30 2 10 64
## 32 32 1 10 65
## 36 36 3 10 66
## 53 53 3 10 67
## 56 56 1 10 68
## 65 65 2 10 69
## 81 81 3 10 70
## 24 24 1 9 71
## 63 63 2 9 72
## 3 3 3 8 73
## 8 8 1 8 74
## 66 66 2 8 75
## 44 44 3 6 76
## 67 67 2 6 77
## 78 78 3 6 78
## 60 60 3 5 79
## 73 73 3 4 80
## 11 11 1 2 81
```

```
# Betweenness:
```

```
personBetweenness_df_sorted$betweennessRank <- 1:nrow(personBetweenness_df_sorted)
personBetweenness_df_sorted
```

```
## name group betweenness betweennessRank
## 37 37 1 1223.08 1
## 62 62 3 965.60 2
## 5 5 2 727.46 3
## 2 2 1 711.28 4
## 29 29 1 684.20 5
## 38 38 3 628.55 6
## 17 17 3 522.26 7
## 72 72 2 445.42 8
## 58 58 1 431.86 9
## 10 10 2 389.92 10
## 54 54 1 389.11 11
## 52 52 1 381.84 12
## 69 69 2 368.58 13
## 77 77 2 349.77 14
## 1 1 3 347.12 15
## 48 48 1 325.63 16
## 75 75 3 322.80 17
## 6 6 2 296.01 18
## 9 9 3 272.71 19
## 35 35 1 263.63 20
## 27 27 2 260.07 21
## 49 49 2 248.02 22
## 7 7 2 245.29 23
## 24 24 1 241.39 24
## 51 51 1 234.38 25
## 22 22 2 231.11 26
## 53 53 3 201.90 27
## 70 70 4 190.02 28
## 4 4 3 185.98 29
## 40 40 2 148.15 30
```

##	42	42	2	143.46	31
##	21	21	1	143.32	32
##	57	57	1	142.47	33
##	39	39	1	139.35	34
##	31	31	1	109.79	35
##	46	46	1	106.81	36
##	61	61	3	104.10	37
##	15	15	1	104.09	38
##	13	13	2	96.77	39
##	55	55	1	93.11	40
##	80	80	1	90.25	41
##	67	67	2	88.33	42
##	28	28	2	79.31	43
##	43	43	1	77.80	44
##	20	20	1	71.69	45
##	79	79	1	70.39	46
##	81	81	3	68.02	47
##	68	68	2	62.92	48
##	25	25	1	43.78	49
##	34	34	1	39.42	50
##	59	59	3	38.20	51
##	30	30	2	37.28	52
##	12	12	2	29.90	53
##	66	66	2	20.87	54
##	33	33	2	20.65	55
##	16	16	2	16.50	56
##	19	19	1	16.46	57
##	76	76	2	11.53	58
##	56	56	1	11.00	59
##	64	64	1	9.94	60
##	65	65	2	7.33	61
##	47	47	2	5.57	62
##	3	3	3	5.50	63
##	26	26	1	5.42	64
##	50	50	4	5.23	65
##	45	45	3	5.17	66
##	44	44	3	4.27	67
##	23	23	2	0.83	68
##	41	41	1	0.20	69
##	71	71	2	0.12	70
##	8	8	1	0.00	71
##	11	11	1	0.00	72
##	14	14	1	0.00	73
##	18	18	1	0.00	74
##	32	32	1	0.00	75
##	36	36	3	0.00	76
##	60	60	3	0.00	77
##	63	63	2	0.00	78
##	73	73	3	0.00	79
##	74	74	3	0.00	80
##	78	78	3	0.00	81

```
# Closeness:
```

```
personCloseness_df_sorted$closenessRank <- 1:nrow(personCloseness_df_sorted)
```

personCloseness_df_sorted

##	name	group	closeness	closenessRank
## 62	62	3	0.0050251256	1
## 29	29	1	0.0045248869	2
## 5	5	2	0.0045045045	3
## 37	37	1	0.0045045045	4
## 52	52	1	0.0042735043	5
## 38	38	3	0.0039840637	6
## 34	34	1	0.0039682540	7
## 15	15	1	0.0039370079	8
## 27	27	2	0.0038759690	9
## 26	26	1	0.0037593985	10
## 44	44	3	0.0037453184	11
## 17	17	3	0.0037174721	12
## 2	2	1	0.0036764706	13
## 70	70	4	0.0036101083	14
## 43	43	1	0.0035971223	15
## 51	51	1	0.0034129693	16
## 75	75	3	0.0033898305	17
## 40	40	2	0.0032051282	18
## 20	20	1	0.0031847134	19
## 22	22	2	0.0031645570	20
## 31	31	1	0.0031347962	21
## 6	6	2	0.0031250000	22
## 48	48	1	0.0031152648	23
## 13	13	2	0.0031055901	24
## 9	9	3	0.0030959752	25
## 54	54	1	0.0030864198	26
## 72	72	2	0.0030769231	27
## 77	77	2	0.0030395137	28
## 57	57	1	0.0030211480	29
## 10	10	2	0.0030030030	30
## 39	39	1	0.0030030030	31
## 49	49	2	0.0029325513	32
## 69	69	2	0.0029154519	33
## 1	1	3	0.0028901734	34
## 35	35	1	0.0028735632	35
## 28	28	2	0.0028409091	36
## 79	79	1	0.0028089888	37
## 58	58	1	0.0028011204	38
## 67	67	2	0.0028011204	39
## 46	46	1	0.0027777778	40
## 68	68	2	0.0026666667	41
## 33	33	2	0.0026041667	42
## 19	19	1	0.0025706941	43
## 21	21	1	0.0025445293	44
## 24	24	1	0.0025445293	45
## 30	30	2	0.0025252525	46
## 81	81	3	0.0025188917	47
## 80	80	1	0.0024937656	48
## 25	25	1	0.0024752475	49
## 73	73	3	0.0024449878	50
## 42	42	2	0.0024390244	51


```
## 7      7      2 0.0023584906      52
## 4      4      3 0.0022172949      53
## 53     53      3 0.0022123894      54
## 66     66      2 0.0021691974      55
## 71     71      2 0.0021645022      56
## 12     12      2 0.0021598272      57
## 64     64      1 0.0021505376      58
## 16     16      2 0.0021052632      59
## 65     65      2 0.0020876827      60
## 50     50      4 0.0020703934      61
## 76     76      2 0.0020703934      62
## 36     36      3 0.0020325203      63
## 78     78      3 0.0020161290      64
## 45     45      3 0.0019880716      65
## 61     61      3 0.0019841270      66
## 55     55      1 0.0018796992      67
## 56     56      1 0.0017985612      68
## 59     59      3 0.0017513135      69
## 23     23      2 0.0016181230      70
## 74     74      3 0.0015197568      71
## 3       3      3 0.0015174507      72
## 41     41      1 0.0014388489      73
## 47     47      2 0.0014084507      74
## 14     14      1 0.0013020833      75
## 63     63      2 0.0010559662      76
## 8       8      1 0.0010493179      77
## 60     60      3 0.0010416667      78
## 32     32      1 0.0007412898      79
## 18     18      1 0.0006920415      80
## 11     11      1      NaN      81
```

```
# Dataframe mit allen Rängen
```

```
combinedRank_df <- data.frame(name=personID)
```

```
combinedRank_df$degreeRank <- personDegree_df_sorted$degreeRank[match(combinedRank_df$name, personDegree_df$name)]
```

```
combinedRank_df$betweennessRank <- personBetweenness_df_sorted$betweennessRank[match(combinedRank_df$name, personBetweenness_df$name)]
```

```
combinedRank_df$closenessRank <- personCloseness_df_sorted$closenessRank[match(combinedRank_df$name, personCloseness_df$name)]
```

```
# Aufsummieren der Ränge
```

```
combinedRank_df$sumRank <- combinedRank_df$degreeRank + combinedRank_df$betweennessRank + combinedRank_df$closenessRank
combinedRank_df
```

```
##      name degreeRank betweennessRank closenessRank sumRank
## 1      1         46           15           34         95
## 2      2         10            4           13         27
## 3      3         73           63           72        208
## 4      4         41           29           53        123
## 5      5          6            3            3         12
## 6      6         43           18           22         83
## 7      7         11           23           52         86
## 8      8         74           71           77        222
## 9      9         51           19           25         95
## 10     10         12            10           30         52
## 11     11         81           72           81        234
## 12     12         37           53           57        147
```

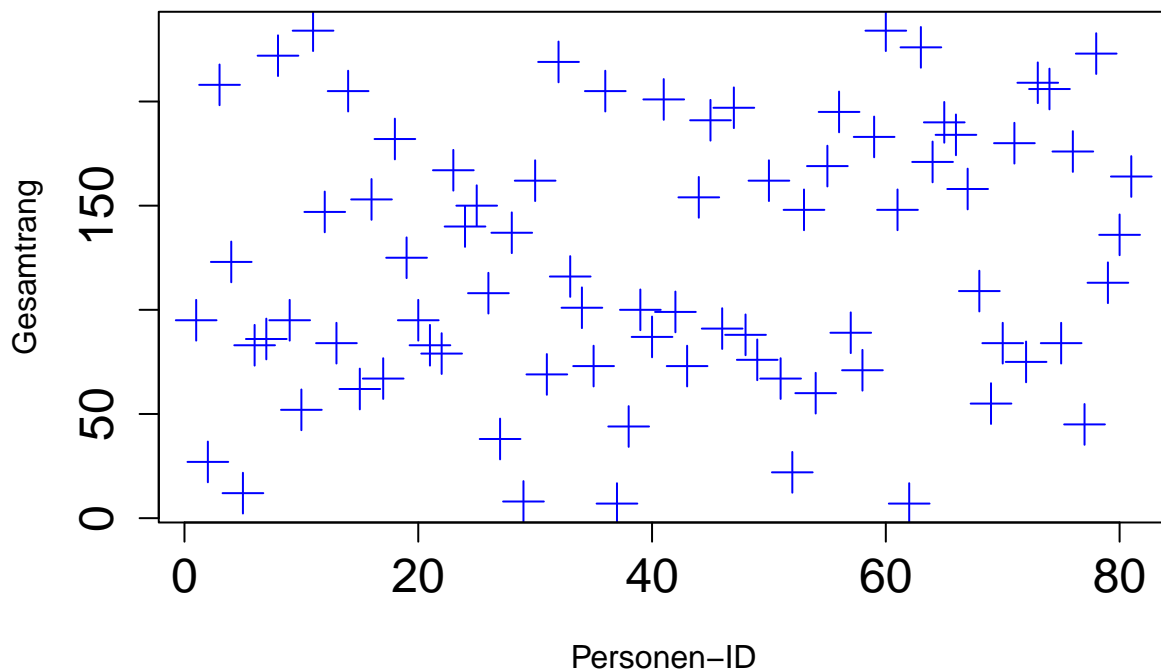
## 13	13	21	39	24	84
## 14	14	57	73	75	205
## 15	15	16	38	8	62
## 16	16	38	56	59	153
## 17	17	48	7	12	67
## 18	18	28	74	80	182
## 19	19	25	57	43	125
## 20	20	31	45	19	95
## 21	21	7	32	44	83
## 22	22	33	26	20	79
## 23	23	29	68	70	167
## 24	24	71	24	45	140
## 25	25	52	49	49	150
## 26	26	34	64	10	108
## 27	27	8	21	9	38
## 28	28	58	43	36	137
## 29	29	1	5	2	8
## 30	30	64	52	46	162
## 31	31	13	35	21	69
## 32	32	65	75	79	219
## 33	33	19	55	42	116
## 34	34	44	50	7	101
## 35	35	18	20	35	73
## 36	36	66	76	63	205
## 37	37	2	1	4	7
## 38	38	32	6	6	44
## 39	39	35	34	31	100
## 40	40	39	30	18	87
## 41	41	59	69	73	201
## 42	42	17	31	51	99
## 43	43	14	44	15	73
## 44	44	76	67	11	154
## 45	45	60	66	65	191
## 46	46	15	36	40	91
## 47	47	61	62	74	197
## 48	48	49	16	23	88
## 49	49	22	22	32	76
## 50	50	36	65	61	162
## 51	51	26	25	16	67
## 52	52	5	12	5	22
## 53	53	67	27	54	148
## 54	54	23	11	26	60
## 55	55	62	40	67	169
## 56	56	68	59	68	195
## 57	57	27	33	29	89
## 58	58	24	9	38	71
## 59	59	63	51	69	183
## 60	60	79	77	78	234
## 61	61	45	37	66	148
## 62	62	4	2	1	7
## 63	63	72	78	76	226
## 64	64	53	60	58	171
## 65	65	69	61	60	190
## 66	66	75	54	55	184

```
## 67 67 77 42 39 158
## 68 68 20 48 41 109
## 69 69 9 13 33 55
## 70 70 42 28 14 84
## 71 71 54 70 56 180
## 72 72 40 8 27 75
## 73 73 80 79 50 209
## 74 74 55 80 71 206
## 75 75 50 17 17 84
## 76 76 56 58 62 176
## 77 77 3 14 28 45
## 78 78 78 81 64 223
## 79 79 30 46 37 113
## 80 80 47 41 48 136
## 81 81 70 47 47 164
```

```
# Darstellen der Daten in Scatterplot für Gesamt-Rang je Person
```

```
plot(combinedRank_df$name, combinedRank_df$sumRank,
      xlab = "Personen-ID",
      ylab = "Gesamtrang",
      main = "Darstellung des Gesamtrangs",
      pch = 3,
      col = "blue",
      cex.axis = 1.5,
      cex.main = 1.5,
      cex = 2)
```

Darstellung des Gesamtrangs



```
# Sortieren nach Gesamt-Rang
```

```
combinedRank_df_sorted <- combinedRank_df[order(combinedRank_df$sumRank, decreasing = FALSE),]
combinedRank_df_sorted
```

##	name	degreeRank	betweennessRank	closenessRank	sumRank
## 37	37	2	1	4	7
## 62	62	4	2	1	7
## 29	29	1	5	2	8
## 5	5	6	3	3	12
## 52	52	5	12	5	22
## 2	2	10	4	13	27
## 27	27	8	21	9	38
## 38	38	32	6	6	44
## 77	77	3	14	28	45
## 10	10	12	10	30	52
## 69	69	9	13	33	55
## 54	54	23	11	26	60
## 15	15	16	38	8	62
## 17	17	48	7	12	67
## 51	51	26	25	16	67
## 31	31	13	35	21	69
## 58	58	24	9	38	71
## 35	35	18	20	35	73
## 43	43	14	44	15	73
## 72	72	40	8	27	75
## 49	49	22	22	32	76
## 22	22	33	26	20	79
## 6	6	43	18	22	83
## 21	21	7	32	44	83
## 13	13	21	39	24	84
## 70	70	42	28	14	84
## 75	75	50	17	17	84
## 7	7	11	23	52	86
## 40	40	39	30	18	87
## 48	48	49	16	23	88
## 57	57	27	33	29	89
## 46	46	15	36	40	91
## 1	1	46	15	34	95
## 9	9	51	19	25	95
## 20	20	31	45	19	95
## 42	42	17	31	51	99
## 39	39	35	34	31	100
## 34	34	44	50	7	101
## 26	26	34	64	10	108
## 68	68	20	48	41	109
## 79	79	30	46	37	113
## 33	33	19	55	42	116
## 4	4	41	29	53	123
## 19	19	25	57	43	125
## 80	80	47	41	48	136
## 28	28	58	43	36	137
## 24	24	71	24	45	140
## 12	12	37	53	57	147
## 53	53	67	27	54	148
## 61	61	45	37	66	148
## 25	25	52	49	49	150
## 16	16	38	56	59	153
## 44	44	76	67	11	154

```
## 67 67 77 42 39 158
## 30 30 64 52 46 162
## 50 50 36 65 61 162
## 81 81 70 47 47 164
## 23 23 29 68 70 167
## 55 55 62 40 67 169
## 64 64 53 60 58 171
## 76 76 56 58 62 176
## 71 71 54 70 56 180
## 18 18 28 74 80 182
## 59 59 63 51 69 183
## 66 66 75 54 55 184
## 65 65 69 61 60 190
## 45 45 60 66 65 191
## 56 56 68 59 68 195
## 47 47 61 62 74 197
## 41 41 59 69 73 201
## 14 14 57 73 75 205
## 36 36 66 76 63 205
## 74 74 55 80 71 206
## 3 3 73 63 72 208
## 73 73 80 79 50 209
## 32 32 65 75 79 219
## 8 8 74 71 77 222
## 78 78 78 81 64 223
## 63 63 72 78 76 226
## 11 11 81 72 81 234
## 60 60 79 77 78 234
```

Spalten benennen

```
colnames(combinedRank_df_sorted)[1] <- "Personen-ID"
colnames(combinedRank_df_sorted)[2] <- "Degree-Rang"
colnames(combinedRank_df_sorted)[3] <- "Betweenness-Rang"
colnames(combinedRank_df_sorted)[4] <- "Closeness-Rang"
colnames(combinedRank_df_sorted)[5] <- "Gesamt-Rang"
```

Spalten sortieren

```
combinedRank_df_sorted <- combinedRank_df_sorted[, c(1, 5, 2, 3, 4)]
```

Ausgabe des ganzen Dataframes

```
combinedRank_df_sorted
```

```
## Personen-ID Gesamt-Rang Degree-Rang Betweenness-Rang Closeness-Rang
## 37 37 7 2 1 4
## 62 62 7 4 2 1
## 29 29 8 1 5 2
## 5 5 12 6 3 3
## 52 52 22 5 12 5
## 2 2 27 10 4 13
## 27 27 38 8 21 9
## 38 38 44 32 6 6
## 77 77 45 3 14 28
## 10 10 52 12 10 30
## 69 69 55 9 13 33
## 54 54 60 23 11 26
```

## 15	15	62	16	38	8
## 17	17	67	48	7	12
## 51	51	67	26	25	16
## 31	31	69	13	35	21
## 58	58	71	24	9	38
## 35	35	73	18	20	35
## 43	43	73	14	44	15
## 72	72	75	40	8	27
## 49	49	76	22	22	32
## 22	22	79	33	26	20
## 6	6	83	43	18	22
## 21	21	83	7	32	44
## 13	13	84	21	39	24
## 70	70	84	42	28	14
## 75	75	84	50	17	17
## 7	7	86	11	23	52
## 40	40	87	39	30	18
## 48	48	88	49	16	23
## 57	57	89	27	33	29
## 46	46	91	15	36	40
## 1	1	95	46	15	34
## 9	9	95	51	19	25
## 20	20	95	31	45	19
## 42	42	99	17	31	51
## 39	39	100	35	34	31
## 34	34	101	44	50	7
## 26	26	108	34	64	10
## 68	68	109	20	48	41
## 79	79	113	30	46	37
## 33	33	116	19	55	42
## 4	4	123	41	29	53
## 19	19	125	25	57	43
## 80	80	136	47	41	48
## 28	28	137	58	43	36
## 24	24	140	71	24	45
## 12	12	147	37	53	57
## 53	53	148	67	27	54
## 61	61	148	45	37	66
## 25	25	150	52	49	49
## 16	16	153	38	56	59
## 44	44	154	76	67	11
## 67	67	158	77	42	39
## 30	30	162	64	52	46
## 50	50	162	36	65	61
## 81	81	164	70	47	47
## 23	23	167	29	68	70
## 55	55	169	62	40	67
## 64	64	171	53	60	58
## 76	76	176	56	58	62
## 71	71	180	54	70	56
## 18	18	182	28	74	80
## 59	59	183	63	51	69
## 66	66	184	75	54	55
## 65	65	190	69	61	60

## 45	45	191	60	66	65
## 56	56	195	68	59	68
## 47	47	197	61	62	74
## 41	41	201	59	69	73
## 14	14	205	57	73	75
## 36	36	205	66	76	63
## 74	74	206	55	80	71
## 3	3	208	73	63	72
## 73	73	209	80	79	50
## 32	32	219	65	75	79
## 8	8	222	74	71	77
## 78	78	223	78	81	64
## 63	63	226	72	78	76
## 11	11	234	81	72	81
## 60	60	234	79	77	78

```
# Ausgabe der 10 Personen mit dem niedrigsten Gesamtrang
head(combinedRank_df_sorted, 10)
```

##	Personen-ID	Gesamt-Rang	Degree-Rang	Betweenness-Rang	Closeness-Rang
## 37	37	7	2	1	4
## 62	62	7	4	2	1
## 29	29	8	1	5	2
## 5	5	12	6	3	3
## 52	52	22	5	12	5
## 2	2	27	10	4	13
## 27	27	38	8	21	9
## 38	38	44	32	6	6
## 77	77	45	3	14	28
## 10	10	52	12	10	30

```
# Ausgabe der 10 Personen mit dem höchsten Gesamtrang
tail(combinedRank_df_sorted, 10)
```

##	Personen-ID	Gesamt-Rang	Degree-Rang	Betweenness-Rang	Closeness-Rang
## 36	36	205	66	76	63
## 74	74	206	55	80	71
## 3	3	208	73	63	72
## 73	73	209	80	79	50
## 32	32	219	65	75	79
## 8	8	222	74	71	77
## 78	78	223	78	81	64
## 63	63	226	72	78	76
## 11	11	234	81	72	81
## 60	60	234	79	77	78

Wie sich erkennen lässt, haben die Personen 37 und 62 mit 7 und Person 29 mit 8 den niedrigsten Gesamt-Rang. Da die Gesamt-Ränge sehr nah beieinanderliegen, ist es schwierig die zu anfangs aufgestellte Forschungsfrage eindeutig zu beantworten. Im Rahmen dieser Arbeit lässt sich sagen, dass alle drei Personen (37, 62, 29) eine sehr wichtige Rolle in der Netzwerkstruktur spielen.

In einer abschließenden Darstellung des Netzwerkes werden die erarbeiteten Ergebnisse noch einmal ganzheitlich dargestellt. Dazu werden die drei wichtigsten Personen und ihre Beziehungen zu anderen Kontakten des Netzwerkes farblich markiert. Außerdem wird durch die Größe der einzelnen Knoten, der Rang der jeweiligen Person dargestellt. Je größer der Knoten, desto besser der Rang. Um eine lineare Verteilung der des Gesamt-Rangs zu ermöglichen, der auch dargestellt werden kann, wurde zur Berechnung des Ranges die Funktion $\text{"(-1/39)*(V(UKfaculty_result)\$rank)+(268/29)"}$ verwendet. Diese hat für den

niedrigsten Rang (7) den Wert 9 und für den höchsten Rang (234) den Wert 3, sodass jeder Knoten noch gut sichtbar dargestellt werden kann.

Darstellen der Ergebnisse

```
UKfaculty_result <- UKfaculty

# Gesamt-Rang hinzufügen
V(UKfaculty_result)$rank <- c(combinedRank_df$sumRank)
V(UKfaculty_result)$rank

## [1] 95 27 208 123 12 83 86 222 95 52 234 147 84 205 62 153 67 182 125
## [20] 95 83 79 167 140 150 108 38 137 8 162 69 219 116 101 73 205 7 44
## [39] 100 87 201 99 73 154 191 91 197 88 76 162 67 22 148 60 169 195 89
## [58] 71 183 234 148 7 226 171 190 184 158 109 55 84 180 75 209 206 84 176
## [77] 45 223 113 136 164

# Farben hinzufügen
V(UKfaculty_result)$color <- "#8a888882"
V(UKfaculty_result)$color

## [1] "#8a888882" "#8a888882" "#8a888882" "#8a888882" "#8a888882" "#8a888882"
## [7] "#8a888882" "#8a888882" "#8a888882" "#8a888882" "#8a888882" "#8a888882"
## [13] "#8a888882" "#8a888882" "#8a888882" "#8a888882" "#8a888882" "#8a888882"
## [19] "#8a888882" "#8a888882" "#8a888882" "#8a888882" "#8a888882" "#8a888882"
## [25] "#8a888882" "#8a888882" "#8a888882" "#8a888882" "#8a888882" "#8a888882"
## [31] "#8a888882" "#8a888882" "#8a888882" "#8a888882" "#8a888882" "#8a888882"
## [37] "#8a888882" "#8a888882" "#8a888882" "#8a888882" "#8a888882" "#8a888882"
## [43] "#8a888882" "#8a888882" "#8a888882" "#8a888882" "#8a888882" "#8a888882"
## [49] "#8a888882" "#8a888882" "#8a888882" "#8a888882" "#8a888882" "#8a888882"
## [55] "#8a888882" "#8a888882" "#8a888882" "#8a888882" "#8a888882" "#8a888882"
## [61] "#8a888882" "#8a888882" "#8a888882" "#8a888882" "#8a888882" "#8a888882"
## [67] "#8a888882" "#8a888882" "#8a888882" "#8a888882" "#8a888882" "#8a888882"
## [73] "#8a888882" "#8a888882" "#8a888882" "#8a888882" "#8a888882" "#8a888882"
## [79] "#8a888882" "#8a888882" "#8a888882"

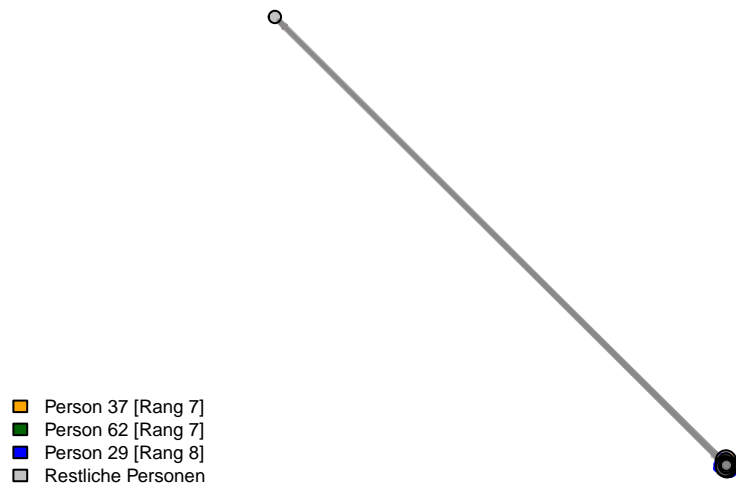
# Farben für die Top-3 Personen
V(UKfaculty_result)$color[match("37", personID)] <- "orange"
V(UKfaculty_result)$color[match("62", personID)] <- "darkgreen"
V(UKfaculty_result)$color[match("29", personID)] <- "blue"

# Einfärben der Edges mit den Farben der Vertices#0a0a#04790a
edge.start <- ends(UKfaculty_result, es=E(UKfaculty_result), names=F)[,1]
edge.col <- V(UKfaculty_result)$color[edge.start]

# Darstellung
plot(UKfaculty_result,
     vertex.size = (-1/39)*(V(UKfaculty_result)$rank)+(268/29),
     vertex.label = NA,
     layout = (layout_with_lgl(UKfaculty_result))*2.5,
     edge.color=edge.col,
     edge.arrow.size = 0.2,
     edge.width = E(UKfaculty_result)$weight/3.5,)
```



```
# Hinzufügen einer Legende
legend("bottomleft",
  legend = c("Person 37 [Rang 7]", "Person 62 [Rang 7]", "Person 29 [Rang 8]", "Restliche Personen"),
  fill = c("orange", "darkgreen", "blue", "#8a88882"),
  box.lty = 0,
  bty = "n",
  cex = 0.6)
```



Ausblick

Bereits im vorangegangenen Abschnitt wurden Auffälligkeiten festgestellt, für deren Untersuchung in dieser Arbeit keine Zeit mehr zur Verfügung stand. Es empfiehlt sich, diese in einer weiteren Betrachtung des Netzwerkes detaillierter zu untersuchen. Außerdem ist es möglich, weitere Parameter zur Wichtigkeit von Personen innerhalb eines Netzwerkes zu untersuchen.

Weitere Untersuchungen die während der Analyse des Netzwerkes aufgekommen sind, betreffe die detaillierte Untersuchung einzelner Personen. Hierzu ist im Anhang eine beispielhafte Untersuchung der drei Personen mit dem besten Gesamt-Rang zu finden. Auch eine Untersuchung der Personen mit dem schlechtesten Gesamt-Rang ist möglich. Eine weitere beispielhafte Untersuchung, welche sich im Anhang wiederfinden lässt, beschäftigt sich mit der Analyse der Einzelnen Untersuchungs-Parameter pro Gruppe. Es ist möglich, die Parameter für die einzelnen Gruppen zu vergleichen, um so Rückschlüsse auf Untergruppen und deren Vergleich zueinander treffen zu können. So scheint beispielsweise die Frage, ob Personen einer Untergruppen häufiger zentrale Positionen innerhalb des gesamten Netzwerkes einnehmen, sehr interessant.

Die Betrachtung des Netzwerkes mit Hilfe von Clustering-Methoden kann ebenfalls neue Erkenntnisse über das Netzwerk liefern. Besonders vielversprechend erscheint ein Vergleich der Ergebnisse der Clustering-Methoden mit den durch den Datensatz definierten Gruppen der Personen.

5. Anhang

Untersuchen der 3 Personen mit dem besten Rang

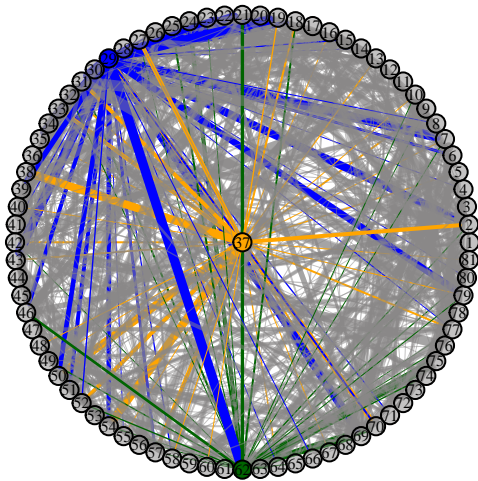
```
# Erneute Ausgabe der 3 Personen mit dem besten Rang
head(combinedRank_df_sorted, 3)
```

##	Personen-ID	Gesamt-Rang	Degree-Rang	Betweenness-Rang	Closeness-Rang
## 37	37	7	2	1	4
## 62	62	7	4	2	1
## 29	29	8	1	5	2

```

plot(UKfaculty_result,
     edge.width=(E(UKfaculty_result)$weight)/3,
     vertex.size=8,
     edge.arrow.size = 0.0,
     edge.color=edge.col,
     vertex.label.color="black",
     vertex.label.cex=0.5,
     layout=layout_as_star(UKfaculty_result, center=V(UKfaculty_result)[37]))

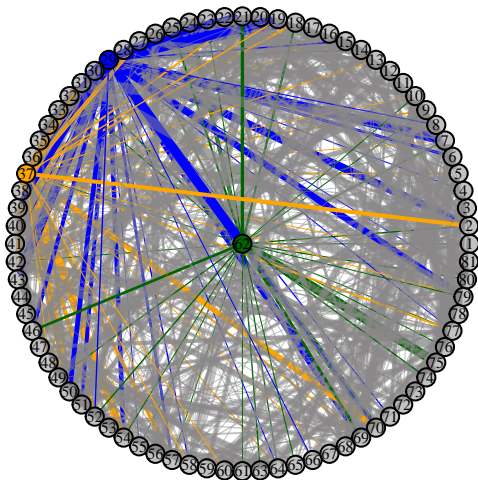
```



```

plot(UKfaculty_result,
     edge.width=(E(UKfaculty_result)$weight)/3,
     vertex.size=8,
     edge.arrow.size = 0.0,
     edge.color=edge.col,
     vertex.label.color="black",
     vertex.label.cex=0.5,
     layout=layout_as_star(UKfaculty_result, center=V(UKfaculty_result)[62]))

```

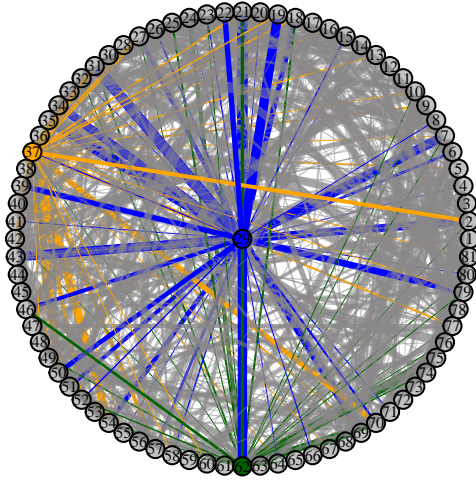


```

plot(UKfaculty_result,
     edge.width=(E(UKfaculty_result)$weight)/3,
     vertex.size=8,
     edge.arrow.size = 0.0,
     edge.color=edge.col,

```

```
vertex.label.color="black",
vertex.label.cex=0.5,
layout=layout_as_star(UKfaculty_result, center=V(UKfaculty_result)[29]))
```



Untersuchen der Gruppen auf Degree-Centrality

```
# Gesamte Anzahl der Kontakte je Gruppe
totalGroupDegree <- aggregate(personDegree_df$degree, by = list(personDegree_df$group), FUN = sum)
# totalGroupDegree

# Verhältnis der Kontakte je Gruppe in Prozent gerundet
totalGroupDegree$percent <- round(totalGroupDegree$x / sum(totalGroupDegree$x) * 100, 2)

# Anzahl der Personen je Gruppe in dataframe
groupCount <- c(table(personDegree_df$group))
groupCount_df <- data.frame(group = c(1,2,3,4), count = groupCount)
# groupCount_df
totalGroupDegree$members <- groupCount_df$count

# Durchschnittliche Anzahl der Kontakte je Gruppe
totalGroupDegree$avgDegree <- round(totalGroupDegree$x / totalGroupDegree$members, 2)

# Spalten benennen
colnames(totalGroupDegree)[1] <- "Gruppe"
colnames(totalGroupDegree)[2] <- "#Kontakte"
colnames(totalGroupDegree)[3] <- "Verteilung [%]"
colnames(totalGroupDegree)[4] <- "#Mitglieder"
colnames(totalGroupDegree)[5] <- "ØKontakte [%]"

# Spalten sortieren
totalGroupDegree <- totalGroupDegree[, c(1, 4, 2, 5, 3)]

# Ausgabe
totalGroupDegree
```

##	Gruppe	#Mitglieder	#Kontakte	ØKontakte [%]	Verteilung [%]
## 1	1	33	758	22.97	46.39
## 2	2	27	589	21.81	36.05

## 3	3	19	249	13.11	15.24
## 4	4	2	38	19.00	2.33