

Aufgabenblatt zur Heimarbeit 4

Mirco Bazzani, Amir Shehadeh & Luca Keiser

2022-05-26

Aufgabenblatt zur Heimarbeit 4

Inhaltsverzeichnis

<i>Aufgabenblatt zur Heimarbeit 4</i>	1
Methoden der sozialen Netzwerkanalyse	2
<i>Literaturverzeichnis</i>	6
<i>Anhang - R-Code</i>	6
<i>Bipartites Netzwerk</i>	9
<i>Knotenattribute</i>	11
<i>Grafiken</i>	14
Bipartites Netzwerk	14
Erstellen der Projektionen	15
<i>Organisationen 1</i>	16
<i>Organisationen 2</i>	17
<i>Parlamentarier:innen</i>	18
<i>Parlamentarier:innen - Community Detection</i>	20
Parlamentarier:innen - Communities	21
<i>Organisationen - Community Detection</i>	22
<i>Organisationen - Community</i>	23
<i>Tabelle 3</i>	25
Modelling	26
<i>Random Network Models</i>	26
<i>Tabelle 4</i>	27
<i>Abbildung 4</i>	27
<i>Statistical Network Models</i>	28
<i>Abbildung 5</i>	29
<i>Tabelle 5</i>	31
Simulation	38
<i>Tabelle 5</i>	38
<i>Abbildung 6</i>	44

Methoden der sozialen Netzwerkanalyse

Bis anhin haben wir unser Netzwerk als Ein-Knoten-Netzwerk dargestellt. Kurz zusammengefasst stellten die Knoten die einzelnen Ständerät:innen dar. Die Kanten bezogen sich auf die gemeinsamen Mitgliedschaften in den jeweiligen Lobby-Organisationen. Alle Kanten des Graphen galten dabei als ungerichtet, da wir davon ausgingen, dass sich die Personen jeweils gegenseitig kennen und wahrnehmen. Wir stellten Verbindungen zwischen Ständeräten dar, die nicht in der gleichen Partei tätig sind und verglichen diese mit dem dichteren Netzwerk mit allen innerparteilicher Edges.

In der Heimarbeit 4 fassen wir die wichtigsten Ergebnisse zusammen. Einerseits zeigen wir auf, wie die Parlamentarier:innen über die Organisationen miteinander verbunden sind und welche Communities entstanden sind. Wir betrachten zuerst die Ergebnisse von den Organisationen und Parlamentarier:innen getrennt und fügen die Netzwerke als Bipartites Netzwerk mit zwei verschiedenen Knotenattribute zusammen.

Das Ständerätliche Netzwerk (Abbildung 1.) zeigt auf wie die jeweiligen Ständeräte durch die gemeinsamen Einsitze in den Organisationen miteinander verbunden sind. Je breiter die Kanten, desto mehr Sitze teilen sich die Parlamentarier:innen in derselben Organisation. Die Parlamentarier:innen unterscheiden sich durch die Färbung der Knoten, welche die jeweilige Parteifarben signalisieren. Wie bereits in den vorherigen Heimarbeiten erwähnt, sind insbesondere Mitte-Partei-Mitglieder zentrale Akteure des Netzwerkes. Neu betrachten wir zudem das Organisationennetzwerk des Ständerates (Abbildung 2). Die Knoten stellen die Lobbyorganisationen dar und diese werden mit Parlamentarier:innen, welche in beiden oder mehrere Organisationen ein Verbindung aufweisen, vernetzt. Unterteilt wurden die Organisationen mit der Färbung der Knoten durch Branchenzugehörigkeit. Aus Platzgründen wurden die jeweiligen Kanten entfernt, die eine Gewichtung von ≤ 1 aufweisen. Des Weiteren wollen wir innerhalb der oben beschriebenen Netzwerke Teilgruppen und Communities bestimmen. In der Netzwerksoziologie wird eine Community als Subste der Nodes mit einer höheren Kantendichte als der Rest des Netzwerkes definiert (Vergleich Radichhi et. al 2004). Um diese Subsets zu identifizieren wird eine Auswahl an Algorithmen verwendet, wie sie auch von Douglas A. Luke in seiner User's Guide to Network Analysis vorgeschlagen werden. Die Community Detection wird sowohl auf die Parlamentarier:innen als auch das Organisationsnetzwerk angewendet. Basierend auf Douglas A. Lukes Methode (vgl. Luke 2015: 118) versuchten wir in einem ersten Schritt den Algorithmus zu

finden, welcher den höchsten Modularitätswert vorweist. Im Falle der Parlamentarier:innen wäre dies *Louvian*.

Wendet man den Algorithmus auf das Netzwerk an, so werden insgesamt sieben verschiedene Communities identifiziert. Zwei dieser sieben lassen sich dadurch erklären, dass sie atomisierte und innerhalb des Netzwerkes nicht verbundene Politiker:innen darstellen (Community 6 und 7, Abbildung 3). Communities eins und zwei weisen eine starke Überschneidung auf und hier sind keine klaren Parteimuster zu erkennen. Gerade in Community 2 sind die Parteien mit vier Grünen und jeweils drei Politiker:innen der SP und SVP relativ ausgeglichen. Die Community 3 besteht ausschliesslich aus FDP und Mitte-Politiker:innen.

Die Interpretation der identifizierten Communities bleibt dabei in den Händen der Forschenden und die Algorithmen machen jeweils keine qualitative Einordnung der identifizierten Subsets.

Wesentlich spannender und eindeutiger gestaltet sich die Community Detection bei den Organisationen. Alle verwendeten Algorithmen kommen hier auf ähnliche Modularitätswert. Der Einfachheit und Vergleichbarkeit halber wurde auch hier der *Louvian*-Algorithmus weiterverwendet. Insgesamt konnten 12 distinkte Communities (Abbildung 4) identifiziert werden. Darunter das primär wirtschaftlich geprägte Netzwerk um die Universität Zürich (Community 1), das ostschweizerische Netzwerk um die Universität St.Gallen (Community 2), Sport und Freizeit in Community 3, Politische Inklusion mit einem Flavour von Ökologie in Community 5, die Glarner Community in Community 6, das Zentralschweizer Netz in Community 7, Umwelt und Verkehr in Communities 8 und 9. Communities 4, 10, 11 und 12 konnten nicht klar identifiziert werden, eine Übersicht findet sich in der Tabelle 3.

Zusätzlich betrachten wir unser Netzwerk als Bipartites Netzwerk und erstellen diese anhand der Edgelist (Parlamentarier:in und Organisation (Event)). Dabei differenzierten wir die Knoten zwischen Parlamentarier:innen (45 Parlamentarier: innen) und die jeweiligen Lobbyorganisationen (636 Organisationen), und zeigen auf über welche Organisationen die jeweiligen Parlamentarier:innen miteinander affiliert sind.

Hervorheben möchten wir in unserer Netzwerkanalyse, dass es klare Erkennungen der Verbindungen durch die Communities gibt. Wie bereits oben erwähnt besteht Community 3 ausschliesslich aus FDP und Mitte-Politiker:innen. Dies zeigt auf, dass das Netzwerk der Ständerät:innen einen klaren mitte-rechts-Bias besitzt. Zusätzlich können wir beobachten, dass

grosse Institutionen, wie die Universitäten Zürich und St. Gallen einen hohen Einflussgrad innerhalb des Organisationsnetzwerkes haben. Auffallend sind auch die Interessen – und standortsbedingten Verbindungen. Dies zeigt sich anhand der Glarner-Community oder auch durch die Verbindungen von VCS, Pro Natura und WWF innerhalb der Community 2.

Basierend auf den Parametern des Organisationsnetzwerkes unserer Arbeit wurden anschliessend randomisierte Modelle erstellt. Aufbauend auf Kapitel 10 von Douglas A. Lukes Werk wurden dafür drei Modelltypen verwendet, das Poisson Random Graph Model (oder Erdős-Rény), ein Small World Modell und schlussendlich noch ein Scale Free Model (vgl. Luke 2015: 147 ff.). Randomisierte Netzwerke werden in der Regel dazu verwendet, grundlegende Dynamiken in Netzwerken herauszuarbeiten und so Informationen über Strukturdynamiken in empirischen Netzwerken zu erhalten. In unserem Fall und ebenfalls basierend auf Douglas A. Lukes vorgehen haben wir aber mehrere Netzwerkmodelle anhand der gegebenen Parameter unseres empirischen Netzwerkes erstellt und diese im Anschluss verglichen (Tabelle 4).

Obwohl das Erdős-Rény-Game als eines der simpelsten randomisierten Modelle in der Netzwerkforschung gehandelt wird hat es in unserem Fall ein Netzwerk erstellt, welches erstaunlich dicht an den Original-Kennzahlen des Empirischen Organisationsnetzwerkes liegen (siehe Tabelle 4). Dies liegt daran, dass das Originalnetzwerk der Organisationen einen unheimlich dicht vernetzten Kern aufweist, um diesen Kern aber eine Vielzahl atomisierter Organisationen ohne edges verteilt sind. Diese blumenförmige Anordnung der Nodes konnte im simulierten Modell nicht nachgeahmt werden.

Im Anschluss an die Random Graph Models wurde noch ein statistisches Modell anhand des Organisationsnetzwerkes erarbeitet. Anders als bei dem vorherigen Beispiel wurden hier die Filterkriterien wieder verwendet, um das stark atomisierte Netzwerk etwas zu verdichten und somit auch die Anzahl erklärender Variablen zu vermindern.

Wie bei den Visualisierungen des Bipartiten Netzwerkes wurde eine Degree-Schwelle von 35 gewählt.

In Anlehnung an das elfte Kapitel von Douglas A. Luke wurde ein Exponential Random Graph Model (kurz ERGM) erstellt. ERGMs verwenden Charakteristiken einzelner Netzwerkelemente (in unserem Falle Organisationen) und versucht anhand dieser die Charakteristiken des gesamten Modells vorherzusagen. Verwendet wird dabei eine Monte Carlo Markov Chain Maximum

Likelihood-Estimation, welche die Wahrscheinlichkeit einer Verbindung zwischen zwei Nodes mit gegebenen erklärenden Variablen vorherzusagen versucht (vgl. Luke 2015: 164).

Als Vergleichsvignette haben wir ein Nullmodell erstellt, welches ohne erklärende Variablen auskommt. Dieses Nullmodell soll nun mit weiteren Node-Attributen angereichert werden. Wir sind dabei von der Hypothese ausgegangen, dass die Vernetzungswahrscheinlichkeit basierend auf der Branche und dem Standort der Organisation steigt (Tabelle 5 & Abbildung 5).

Da der AIC mit der Hinzunahme der beiden Variablen gesunken ist gehen wir von einer verbesserten Erklärungsleistung aus. Es ist jedoch anzunehmen, dass die Industrie durch den grösseren Anteil signifikanter Kategorien einen grösseren Anteil zur Erklärungsleistung beiträgt (Tabelle 6).

Die Simulation geht von einer viel stärkeren Vernetzung innerhalb unseres Graphen aus, als sie empirisch festgestellt werden kann. Es wäre also zu empfehlen, die Simulation in einer folgestudie noch durch lokale strukturelle vorhersagen zu ergänzen (vgl. Luke 2015: 177). Dies wurde aktuell aufgrund der hohen Rechenintensivität und gesteigerten Komplexität unterlassen.

Abschliessend möchten wir noch auf eine aktuelle Studie von Rossier et al. 2022 verweisen. Die Studie befasst sich mit dem Netzwerk Schweizerischer Eliten und dessen zeitliche Entwicklung respektive Veränderung. Konkret beschreiben die Autoren, wie sich die Beziehungen innerhalb des Elitenetzwerkes zwischen den Jahren 1910 und 2015 verändert haben. Es handelt sich nicht um eine Lückenlose Darstellung. Der Untersuchungszeitraum wurde auf sieben Schlüsseljahre beschränkt (1910, 1937, 1957, 1980, 200, 2010 und 2015). Das Netzwerk seinerseits setzt sich aus insgesamt 22'262 Elitepersonen aus Wirtschaft, Politik und Wissenschaft und 2587 Organisationen aus acht Schlüsselindustrien zusammen. Es zeigte sich, dass sich der Kern des Netzwerks über den Untersuchungszeitraum hinweg stets von Wirtschaftsakteuren dominiert wurde. Während Krisenzeiten (nach dem 2. Weltkrieg und der Finanzkrise) wurden der Kern jedoch diversifiziert, indem andere Akteure aus Politik und Gewerkschaften inkludiert wurden (vgl. Rossier et al. 2022).

Anzahl Wörter: 1'292

Literaturverzeichnis

- Radicchi, Filippo; Claudio Castellano; Federico Cecconi; Vittorio Loreto; Domenico Parisi (2004): Defining and identifying communities in networks, Proceedings of the National Academy of Sciences, Volume 101, [doi:10.1073/pnas.0400054101](https://doi.org/10.1073/pnas.0400054101)
- Rossier, T., Ellersgaard, C. H., Larsen, A. G. & Lunding, J. A. (2022). From integrated to fragmented elites. The core of Swiss elite networks 1910–2015. The British Journal of Sociology, 73, 315– 335. <https://doi.org/10.1111/1468-4446.12929>

Anhang - R-Code

```
# Pakete Laden
library(tidyverse)

## — Attaching packages ————— tidyverse 1.
3.1 —

## ✓ ggplot2 3.3.5      ✓ purrr 0.3.4
## ✓ tibble 3.1.6      ✓ dplyr 1.0.8
## ✓ tidyr 1.2.0       ✓ stringr 1.4.0
## ✓ readr 2.1.2       ✓ forcats 0.5.1

## — Conflicts ————— tidyverse_conflict
s() —
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()

library(tidygraph)

##
## Attache Paket: 'tidygraph'

## Das folgende Objekt ist maskiert 'package:stats':
##
##   filter

library(ggraph)
library(igraph)

##
## Attache Paket: 'igraph'

## Das folgende Objekt ist maskiert 'package:tidygraph':
##
##   groups

## Die folgenden Objekte sind maskiert von 'package:dplyr':
##
##   as_data_frame, groups, union
```

```

## Die folgenden Objekte sind maskiert von 'package:purrr':
##
##   compose, simplify

## Das folgende Objekt ist maskiert 'package:tidyr':
##
##   crossing

## Das folgende Objekt ist maskiert 'package:tibble':
##
##   as_data_frame

## Die folgenden Objekte sind maskiert von 'package:stats':
##
##   decompose, spectrum

## Das folgende Objekt ist maskiert 'package:base':
##
##   union

library(knitr)
library(ggforce)
library(sna)

## Lade nötiges Paket: statnet.common

##
## Attache Paket: 'statnet.common'

## Die folgenden Objekte sind maskiert von 'package:base':
##
##   attr, order

## Lade nötiges Paket: network

##
## 'network' 1.17.1 (2021-06-12), part of the Statnet Project
## * 'news(package="network")' for changes since last version
## * 'citation("network")' for citation information
## * 'https://statnet.org' for help, support, and other information

##
## Attache Paket: 'network'

## Die folgenden Objekte sind maskiert von 'package:igraph':
##
##   %c%, %s%, add.edges, add.vertices, delete.edges, delete.vertices,
##   get.edge.attribute, get.edges, get.vertex.attribute, is.bipartite,
##   is.directed, list.edge.attributes, list.vertex.attributes,
##   set.edge.attribute, set.vertex.attribute

## sna: Tools for Social Network Analysis
## Version 2.6 created on 2020-10-5.
## copyright (c) 2005, Carter T. Butts, University of California-Irvine

```

```

## For citation information, type citation("sna").
## Type help(package="sna") to get started.

##
## Attache Paket: 'sna'

## Die folgenden Objekte sind maskiert von 'package:igraph':
##
##     betweenness, bonpow, closeness, components, degree, dyad.census,
##     evcent, hierarchy, is.connected, neighborhood, triad.census

library(janitor)

##
## Attache Paket: 'janitor'

## Die folgenden Objekte sind maskiert von 'package:stats':
##
##     chisq.test, fisher.test

library(gridExtra)

##
## Attache Paket: 'gridExtra'

## Das folgende Objekt ist maskiert 'package:dplyr':
##
##     combine

library(ergm)

##
## 'ergm' 4.2.1 (2022-05-10), part of the Statnet Project
## * 'news(package="ergm")' for changes since last version
## * 'citation("ergm")' for citation information
## * 'https://statnet.org' for help, support, and other information

## 'ergm' 4 is a major update that introduces some backwards-incompatible
## changes. Please type 'news(package="ergm")' for a list of major
## changes.

##
## Attache Paket: 'ergm'

## Das folgende Objekt ist maskiert 'package:statnet.common':
##
##     snctrl

theme_set(theme_light())

# Einsatz mehrerer Prozessoren
doParallel::registerDoParallel(cores = 4)

```



```

# Datensatz Laden
doc <- read_delim(here::here("Data", "Lobbywatch", "cartesian_minimal_parlamentarier_interessenbindung.csv"),
                  delim = "\t")

## Rows: 6773 Columns: 37

## — Column specification —————
## Delimiter: "\t"
## chr   (25): parlamentarier_name, parlamentarier_rat, parlamentarier_kanton, p...
## dbl   (7): parlamentarier_id, parlamentarier_parlament_biografie_id, parla me...
## lgl   (1): parlamentarier_im_rat_bis
## date  (4): parlamentarier_im_rat_seit, parlamentarier_geburtstag, interess en...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.

doc_SR <- doc %>%
  filter(parlamentarier_rat == "SR") %>%
  filter(organisation_rechtsform != "Parlamentarische Gruppe")

```

Bipartites Netzwerk

Erstellen eines bipartiten Netzwerks anhand der Edgelist

Edgelist (Parlamentarier:in und Organisation (Event))

```

edgelist <- doc_SR %>%
  select(parlamentarier_name, organisation_name)

```

ansehen

#head(edgelist)

```

edgelist %>%
  count(parlamentarier_name,
        sort = TRUE)

```

```

## # A tibble: 45 × 2
##   parlamentarier_name      n
##   <chr>                <int>
## 1 Ettlin, Erich         38
## 2 Hegglin, Peter        37
## 3 Bischof, Pirmin       36
## 4 Mazzone, Lisa         36
## 5 Caroni, Andrea        35

```

```

## 6 Jositsch, Daniel          31
## 7 Dittli, Josef            28
## 8 Häberli-Koller, Brigitte  28
## 9 Salzmann, Werner         24
## 10 Schmid, Martin          23
## # ... with 35 more rows

edgelist %>% count(organisation_name,
                  sort = TRUE)

## # A tibble: 636 × 2
##   organisation_name
##   <chr>
##   <int>
## 1 Aqua Viva
##   5
## 2 Verein Winteruniversiade 2021 Luzern-Zentralschweiz
##   5
## 3 Arbeitskreis Sicherheit und Wehrtechnik
##   4
## 4 FH SCHWEIZ, Dachverband der Absolventinnen und Absolventen Fachhochsch...
##   4
## 5 Groupe Mutuel Holding SA
##   4
## 6 Nachhaltigkeitsnetzwerk Zentralschweiz
##   4
## 7 Schweizer Paraplegiker-Stiftung
##   4
## 8 Schweizerischer Anwaltsverband SAV
##   4
## 9 Universität St. Gallen
##   4
## 10 alps - Alpines Museum der Schweiz
##   3
## # ... with 626 more rows

# Bipartites Netzwerk
SR_bipartite_igraph <- graph.data.frame(edgelist,
                                         directed = TRUE)

# Noch nicht ganz...
#SR_bipartite_igraph
#list.vertex.attributes(SR_bipartite_igraph)

# Die Konten müssen noch differenziert werden (Parlamentarier:innen vs. Organ
isationen)
#V(SR_bipartite_igraph)$name
#edgelist[,1]
#edgelist[,2]

```

```
V(SR_bipartite_igraph)$type <- V(SR_bipartite_igraph)$name %in% edgelist[,1]$
parlamentarier_name
```

```
# Jetzt sieht es gut aus!
#SR_bipartite_igraph
#list.vertex.attributes(SR_bipartite_igraph)

# 45 Parlamentarier:innen und 636 Organisationen
#table(get.vertex.attribute(SR_bipartite_igraph)$type)
```

Knotenattribute

```
# Hinzufügen weiterer Knotenattribute mit tidygraph
SR_bipartite_tidy <- as_tbl_graph(SR_bipartite_igraph)

# Definieren der Knotenattribute

# 1) Parlamentarier:innen
node_attributes_parl <- doc_SR %>%
  select(parlamentarier_name, parlamentarier_geschlecht, parlamentarier_partei,
    parlamentarier_kanton) %>%
  distinct()

# 2) Organisationen
node_attributes_org <- doc_SR %>%
  select(organisation_name, organisation_ort, organisation_interessengruppe1,
    organisation_interessengruppe1_branche) %>%
  distinct()

# Hinzufügen
SR_bipartite_tidy <- SR_bipartite_tidy %>%

  # Knotenattribute (Parlamentarier)
  left_join(node_attributes_parl,
    by = c("name" = "parlamentarier_name")) %>%
  # Korrektur für Thomas Minder (Parteilos)
  mutate(parlamentarier_partei = ifelse(name == "Minder, Thomas", "Parleilos",
    parlamentarier_partei)) %>%

  # Knotenattribute (Organisationen)
  left_join(node_attributes_org,
    by = c("name" = "organisation_name")) %>%
  rename(sex = parlamentarier_geschlecht,
    party = parlamentarier_partei,
    canton = parlamentarier_kanton,
    location_org = organisation_ort,
    interest_org = organisation_interessengruppe1,
    industry_org = organisation_interessengruppe1_branche)
```

```

# aktualisieren des igraph-objekts
SR_bipartite_igraph <- as.igraph(SR_bipartite_tidy)

#list.vertex.attributes(SR_bipartite_igraph)

# ansehen
SR_bipartite_tidy %>%
  activate(nodes) %>%
  as_tibble() %>%
  group_by(type) %>%
  slice_head(n = 10)

## # A tibble: 20 × 8
## # Groups:   type [2]
##   name                type sex party canton location_org interest_org indust
ry_org
##   <chr>              <lgl> <chr> <chr> <chr> <chr> <chr> <chr>
## 1 Arbeitskreis... FALSE <NA> <NA> <NA> Zürich Rüstungsind... Sicher
heit
## 2 Schweizerisc... FALSE <NA> <NA> <NA> Zürich KMU/Gewerbe... Wirtsc
haft
## 3 FDP der Schw... FALSE <NA> <NA> <NA> Bern Parteien Staats
polit...
## 4 Université d... FALSE <NA> <NA> <NA> Neuchâtel Bildung/Wis... Bildun
g
## 5 IG Freiheit FALSE <NA> <NA> <NA> Zollikon Staatspolit... Staats
polit...
## 6 Schweizerisc... FALSE <NA> <NA> <NA> Zürich Baunebengew... Wirtsc
haft
## 7 Schweizerisc... FALSE <NA> <NA> <NA> Bern Advokaturen... Wirtsc
haft
## 8 Swiss Medica... FALSE <NA> <NA> <NA> Fribourg Spitäler Gesund
heit
## 9 AJS ingénieu... FALSE <NA> <NA> <NA> Neuchâtel Bauhauptgew... Wirtsc
haft
## 10 Société Anon... FALSE <NA> <NA> <NA> Neuchâtel Immobilien/... Wirtsc
haft
## 11 Bauer, Phili... TRUE M FDP NE <NA> <NA> <NA>
## 12 Baume-Schnei... TRUE F SP JU <NA> <NA> <NA>
## 13 Bischof, Pir... TRUE M M SO <NA> <NA> <NA>
## 14 Burkart, Thi... TRUE M FDP AG <NA> <NA> <NA>
## 15 Carobbio Gus... TRUE F SP TI <NA> <NA> <NA>
## 16 Caroni, Andr... TRUE M FDP AR <NA> <NA> <NA>
## 17 Chiesa, Marco TRUE M SVP TI <NA> <NA> <NA>
## 18 Dittli, Josef TRUE M FDP UR <NA> <NA> <NA>
## 19 Engler, Stef... TRUE M M GR <NA> <NA> <NA>
## 20 Ettlin, Erich TRUE M M OW <NA> <NA> <NA>

```

```

tibble(
  name = V(SR_bipartite_igraph)$name,
  type = V(SR_bipartite_igraph)$type,
  sex = V(SR_bipartite_igraph)$sex,
  party = V(SR_bipartite_igraph)$party,
  canton = V(SR_bipartite_igraph)$canton,
  place = V(SR_bipartite_igraph)$location_org,
  interest = V(SR_bipartite_igraph)$interest_org,
  branche = V(SR_bipartite_igraph)$industry_org
) %>%
  group_by(type) %>%
  slice_head(n = 10)

## # A tibble: 20 × 8
## # Groups:   type [2]
##   name                                type  sex  party canton place interest b
##   <chr>                                <lgl> <chr> <chr> <chr> <chr> <chr> <
## 1 Arbeitskreis Sicherheit und ... FALSE <NA> <NA> <NA> Züri... Rüstung... S
## 2 Schweizerischer Arbeitgeberv... FALSE <NA> <NA> <NA> Züri... KMU/Gew... W
## 3 FDP der Schweiz                 FALSE <NA> <NA> <NA> Bern  Parteien S
## 4 Université de Neuchâtel         FALSE <NA> <NA> <NA> Neuc... Bildung... B
## 5 IG Freiheit                     FALSE <NA> <NA> <NA> Zoll... Staatsp... S
## 6 Schweizerisch-Liechtensteini... FALSE <NA> <NA> <NA> Züri... Baunebe... W
## 7 Schweizerischer Anwaltsverba... FALSE <NA> <NA> <NA> Bern  Advokat... W
## 8 Swiss Medical Network Hospit... FALSE <NA> <NA> <NA> Frib... Spitäler G
## 9 AJS ingénieurs civils SA        FALSE <NA> <NA> <NA> Neuc... Bauhaup... W
## 10 Société Anonyme Immobilière ... FALSE <NA> <NA> <NA> Neuc... Immobil... W
## 11 Bauer, Philippe                 TRUE  M    FDP  NE    <NA> <NA> <
## 12 Baume-Schneider, Elisabeth     TRUE  F    SP   JU    <NA> <NA> <
## 13 Bischof, Pirmin                 TRUE  M    M    SO    <NA> <NA> <
## 14 Burkart, Thierry               TRUE  M    FDP  AG    <NA> <NA> <
## 15 Carobbio Guscetti, Marina      TRUE  F    SP   TI    <NA> <NA> <
## 16 Caroni, Andrea                 TRUE  M    FDP  AR    <NA> <NA> <

```

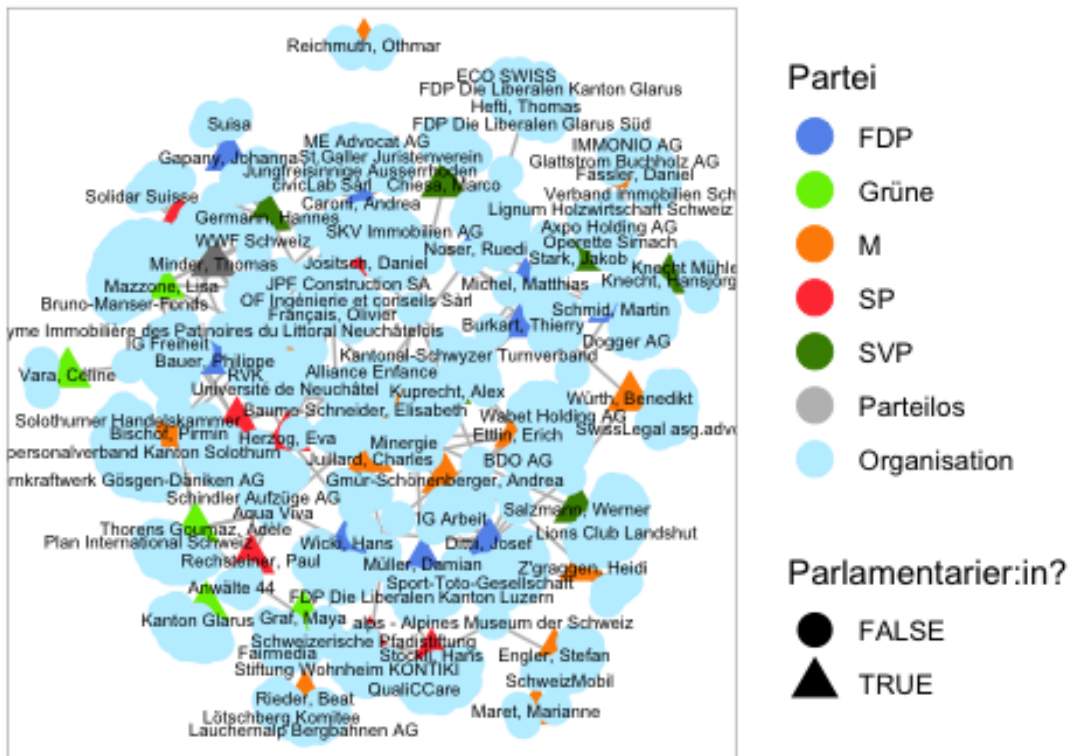
## 17 Chiesa, Marco	TRUE	M	SVP	TI	<NA>	<NA>	<
NA>							
## 18 Dittli, Josef	TRUE	M	FDP	UR	<NA>	<NA>	<
NA>							
## 19 Engler, Stefan	TRUE	M	M	GR	<NA>	<NA>	<
NA>							
## 20 Ettlin, Erich	TRUE	M	M	OW	<NA>	<NA>	<
NA>							

Grafiken

Bipartites Netzwerk

```
set.seed(12345)
SR_bipartite_tidy %>%
  activate(nodes) %>%
  mutate(party = ifelse(is.na(party), "Organisation", party)) %>%
  ggraph(layout = "fr") +
  geom_edge_link(color = "grey") +
  geom_node_point(aes(shape = type,
                      color = party),
                 size = 5) +
  geom_node_text(aes(label = name,
                    size = 2,
                    check_overlap = TRUE) +
  scale_color_manual(values = c("FDP" = "cornflowerblue",
                                "Grüne" = "chartreuse2",
                                "M" = "darkorange",
                                "SP" = "brown1",
                                "SVP" = "chartreuse4",
                                "Parteilos" = "grey",
                                "Organisation" = "lightblue1")) +
  labs(title = "Bipartites Netzwerk - Parlamentarier:innen und Organisationen",
       caption = "Aus Platzgründen werden nicht alle Namen angezeigt.",
       color = "Partei",
       shape = "Parlamentarier:in?")
```

Bipartites Netzwerk - Parlamentarier:innen und Organisation



Aus Platzgründen werden nicht alle Namen angezeigt.

Erstellen der Projektionen

Projektionen

```
SR_projection <- bipartite.projection(SR_bipartite_tidy)
head(SR_projection)
```

1) Organisationen

```
SR_organisationen_igraph <- SR_projection$proj1
```

2) Parlamentarier:innen

```
SR_parlamentarier_igraph <- SR_projection$proj2
```

Gewichtung

```
get.adjacency(SR_organisationen_igraph,
  sparse = FALSE,
  attr = "weight")
```

```
get.adjacency(SR_parlamentarier_igraph,
  sparse = FALSE,
  attr = "weight")
```

```

# ansehen
# sehr spärlich...
#table(E(SR_organisationen_igraph)$weight)
#table(E(SR_parlamentarier_igraph)$weight)

# Grafiken werden mit tidygraph erstellt
SR_organisationen_tidy <- as_tbl_graph(SR_organisationen_igraph) %>%
  activate(nodes) %>%
  mutate(degree = centrality_degree(),
         closeness = centrality_closeness())

## Warning in closeness(graph = graph, vids = V(graph), mode = mode, weights
=
## weights, : At centrality.c:2874 :closeness centrality is not well-defined
for
## disconnected graphs

SR_parlamentarier_tidy <- as_tbl_graph(SR_parlamentarier_igraph) %>%
  mutate(degree = centrality_degree(),
         closeness = centrality_closeness())

## Warning in closeness(graph = graph, vids = V(graph), mode = mode, weights
=
## weights, : At centrality.c:2874 :closeness centrality is not well-defined
for
## disconnected graphs

```

Organisationen 1

```

# in einem ersten Schritt muss dazu ein Filterungsindex erstellt werden
index_org <- SR_organisationen_tidy %>%
  activate(edges) %>%
  filter(weight > 1) %>%
  as_tibble() %>%
  select(from, to)

index_org <- c(index_org$from, index_org$to) %>%
  unique()

set.seed(12345)
SR_organisationen_tidy %>%
  activate(edges) %>%
  filter(weight > 1) %>%
  activate(nodes) %>%
  slice(index_org) %>%
  ggraph(layout = "fr") +
  geom_edge_link(aes(width = weight),
                alpha = 0.5,
                show.legend = FALSE) +
  geom_node_point(aes(color = industry_org,

```



```

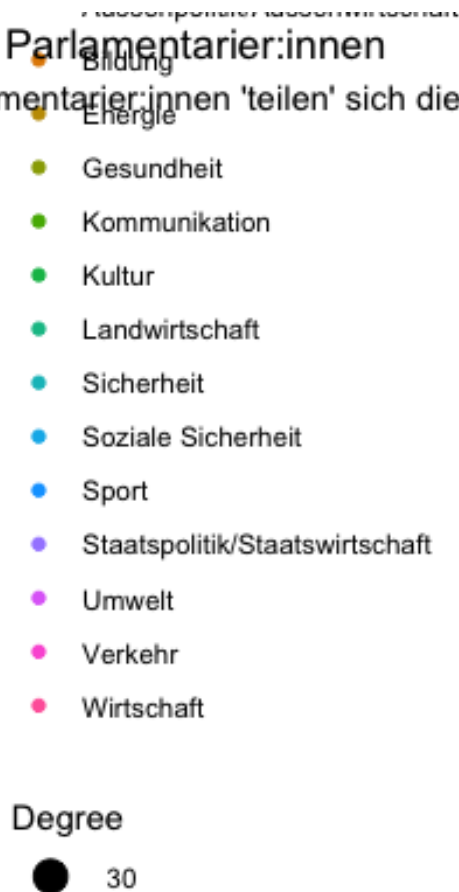
    size = degree)) +
geom_node_text(aes(label = name),
               repel = TRUE,
               size = 3) +
scale_edge_width(range = c(0.5, 3)) +
scale_size(range = c(1, 10)) +
labs(title = "Organisationsnetzwerk - Geteilte Parlamentarier:innen",
     subtitle = "Je breiter die Kanten, desto mehr Parlamentarier:innen 'teilen' sich die Organisationen",
     size = "Degree",
     color = "Branche",
     caption = "Organisationen mit einer Gewichtung von <= 1 wurden herausg
efiltert.")

## Warning: ggrepel: 132 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps

```

Organisationsnetzwerk - Geteilte Parlamentarier:innen

Je breiter die Kanten, desto mehr Parlamentarier:innen 'teilen' sich die O



er Gewichtung von <= 1 wurden herausgefiltert.

Organisationen 2

```

set.seed(12345)
SR_organisationen_tidy %>%
  activate(nodes) %>%
  filter(degree >= 35) %>%
  ggraph(layout = "fr") +
  geom_edge_link(aes(width = weight),

```

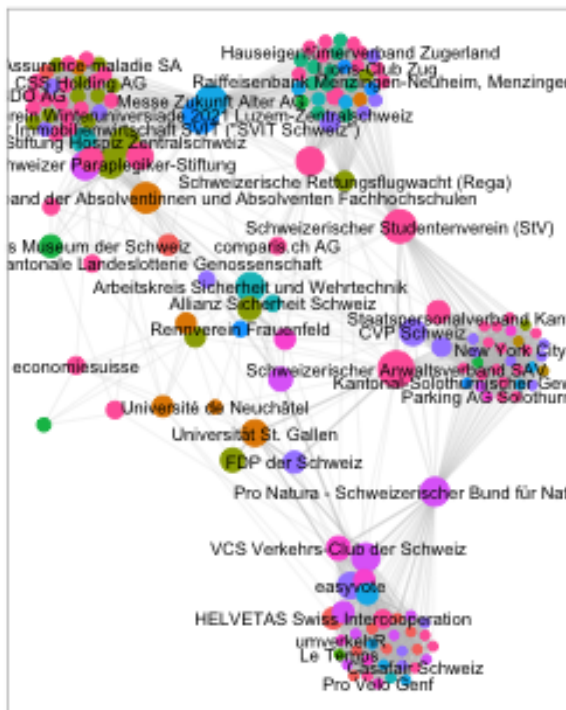
```

    color = "grey",
    alpha = 0.5,
    show.legend = FALSE) +
geom_node_point(aes(color = industry_org,
    size = degree)) +
geom_node_text(aes(label = name),
    check_overlap = TRUE,
    size = 2) +
scale_edge_width(range = c(0.1, 0.75)) +
scale_size(range = c(1, 5)) +
labs(title = "Organisationsnetzwerk - Geteilte Parlamentarier:innen",
    subtitle = "Je breiter die Kanten, desto mehr Parlamentarier:innen 'teilen' sich die Organisationen",
    size = "Degree",
    color = "Branche",
    caption = "Organisationen mit einem Degree < 35 wurden herausgefiltert
.")

```

Organisationsnetzwerk - Geteilte Parlamentarier:innen

Je breiter die Kanten, desto mehr Parlamentarier:innen 'teilen' sich die O



- Gesundheit
- Kommunikation
- Kultur
- Landwirtschaft
- Sicherheit
- Soziale Sicherheit
- Sport
- Staatspolitik/Staatswirtschaft
- Umwelt
- Verkehr
- Wirtschaft

Degree

● 40

-

n mit einem Degree < 35 wurden herausgefiltert.

Parlamentarier:innen

```

set.seed(12345)
SR_parlamentarier_tidy %>%
  ggraph(layout = "fr") +
  geom_edge_link(aes(width = weight),
    color = "grey",

```

```

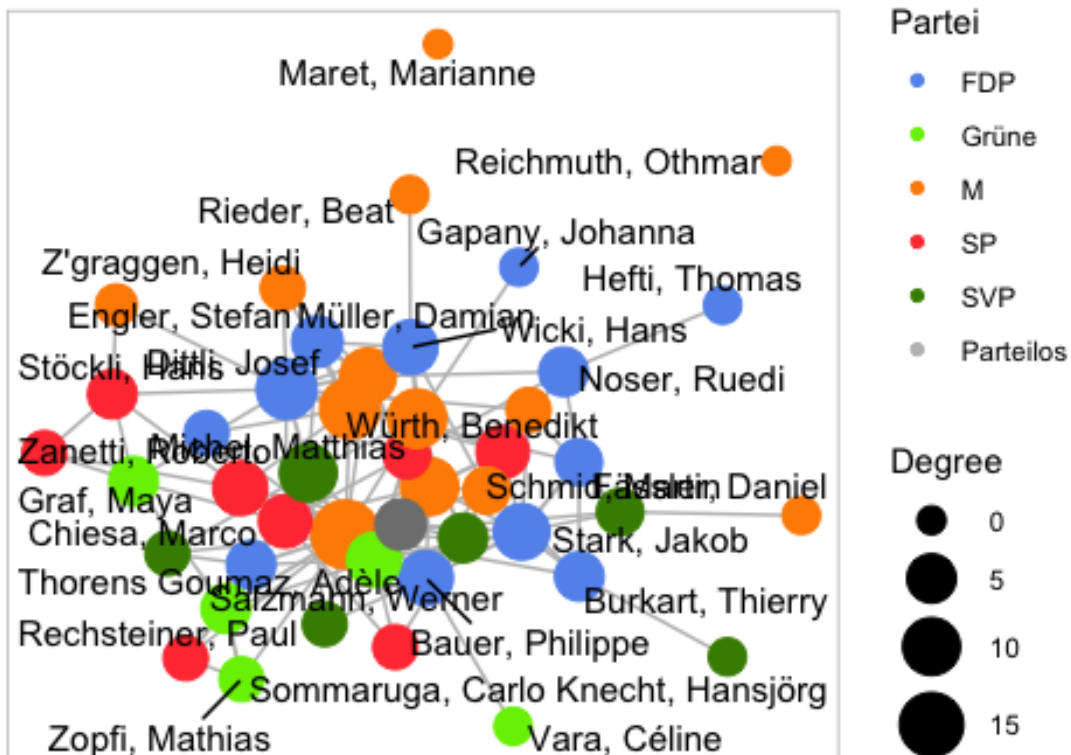
    show.legend = FALSE) +
  geom_node_point(aes(color = party,
                      size = degree)) +
  geom_node_text(aes(label = name),
                size = 4,
                repel = TRUE) +
  scale_color_manual(values = c("FDP" = "cornflowerblue",
                                "Grüne" = "chartreuse2",
                                "M" = "darkorange",
                                "SP" = "brown1",
                                "SVP" = "chartreuse4",
                                "Parteilos" = "grey")) +
  scale_edge_width(range = c(0.5, 1)) +
  scale_size(range = c(4, 10)) +
  labs(title = "Ständerätliches Netzwerk - Gemeinsame Einsitze in Organisatio-
nen",
       subtitle = "Je breiter die Kanten, desto mehr Organisationen 'teilen'
sich die Parlamentarier:innen",
       size = "Degree",
       color = "Partei")

## Warning: ggrepel: 16 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps

```

Ständerätliches Netzwerk - Gemeinsame Einsitze in Organis

Je breiter die Kanten, desto mehr Organisationen 'teilen' sich die Parlam



Parlamentarier:innen - Community Detection

Auswahl des Algorithmus mit der besten Modularität für das Parlamentarier:innen-Netzwerk

Fast & Greedy

```
fg_p <- modularity(  
  cluster_fast_greedy(SR_parlamentarier_tidy)  
)
```

Infomap

```
in_p <- modularity(  
  cluster_infomap(SR_parlamentarier_tidy)  
)
```

Edge Betweenness

```
eb_p <- modularity(  
  cluster_edge_betweenness(SR_parlamentarier_tidy)  
)
```

```
## Warning in cluster_edge_betweenness(SR_parlamentarier_tidy): At  
## community.c:461 :Membership vector will be selected based on the lowest  
## modularity score.
```

```
## Warning in cluster_edge_betweenness(SR_parlamentarier_tidy): At  
## community.c:468 :Modularity calculation with weighted edge betweenness com  
munity  
## detection might not make sense -- modularity treats edge weights as simila  
rities  
## while edge betweenness treats them as distances
```

Louvian

```
lv_p <- modularity(  
  cluster_louvain(SR_parlamentarier_tidy)  
)
```

Dataframe

```
Algorithm <- c("Fast & Greedy",  
              "Infomap",  
              "Edge Betweenness",  
              "Louvian")
```

```
Modularity <- c(fg_p,  
               in_p,  
               eb_p,  
               lv_p )
```

```
kable(data.frame(Algorithm,Modularity),  
      caption = "Tabelle 1: Modularity Scores des Parlamentarier:innen-Netzwe  
rks basierend auf verschiedenen Community-Algorithmen")
```

Tabelle 1: Modularity Scores des Parlamentarier:innen-Netzwerks basierend auf verschiedenen Community-Algorithmen

Algorithm	Modularity
Fast & Greedy	0.3743198
Infomap	0.1333036
Edge Betweenness	0.1953697
Louvian	0.3918418

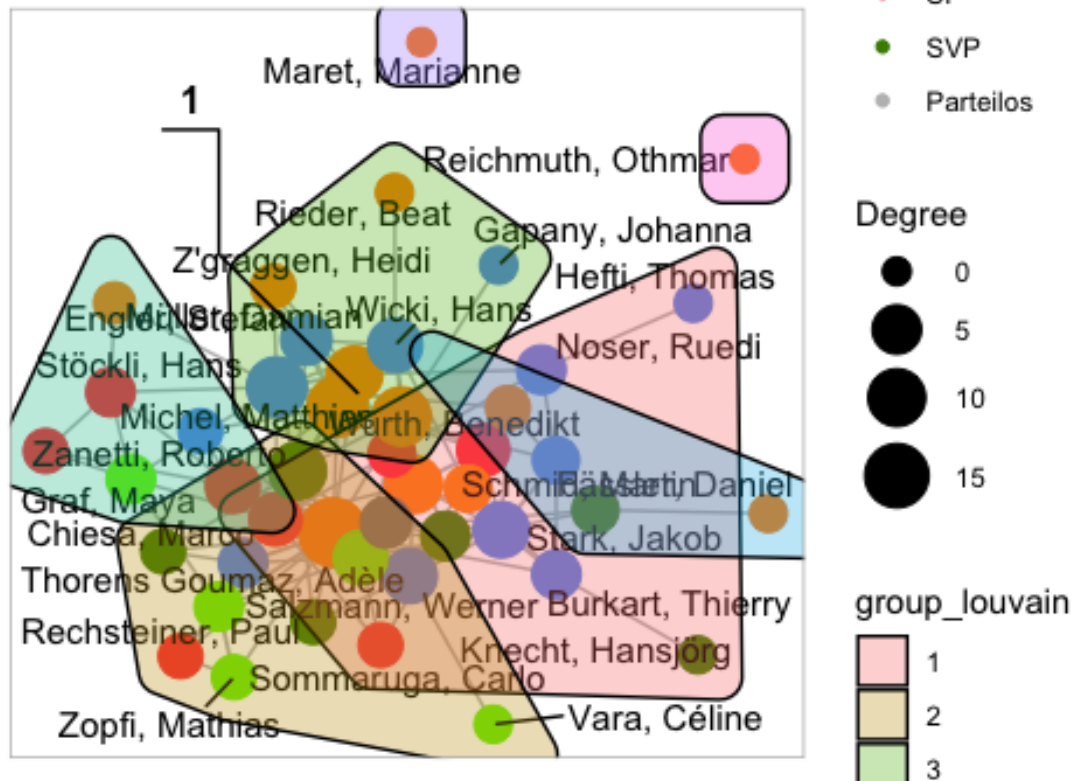
Parlamentarier:innen - Communities

```
set.seed(12345)
SR_parlamentarier_tidy %>%
  mutate(group_louvain= as.factor(group_louvain())) %>%
  ggraph(layout = "fr") +
  geom_edge_link(aes(width = weight),
                 color = "grey",
                 show.legend = FALSE) +
  geom_node_point(aes(color = party,
                     size = degree)) +
  geom_node_text(aes(label = name),
                size = 4,
                repel = TRUE) +
  geom_mark_hull(aes(x = x,
                    y = y,
                    fill = group_louvain,
                    label = group_louvain),
                concavity = 6) +
  scale_color_manual(values = c("FDP" = "cornflowerblue",
                                "Grüne" = "chartreuse2",
                                "M" = "darkorange",
                                "SP" = "brown1",
                                "SVP" = "chartreuse4",
                                "Parteilos" = "grey")) +
  scale_edge_width(range = c(0.5, 1)) +
  scale_size(range = c(4, 10)) +
  labs(title = "Ständerätliches Netzwerk - Gemeinsame Einsitze in Organisatio
nen | Communities",
       subtitle = "Communities anhand Algorithmus Louvian ausgearbeitet",
       size = "Degree",
       color = "Partei")

## Warning: ggrepel: 18 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps
```

Ständerätliches Netzwerk - Gemeinsame Einsitze in Organisations

Communities anhand Algorithmus Louvian ausgearbeitet



Organisationen - Community Detection

Auswahl des Algorithmus mit der besten Modularität für das Parlamentarier: innen-Netzwerk

```
SR_organisationen_tidy_com <- SR_organisationen_tidy %>%
  activate(edges) %>%
  filter(weight > 1)
```

Fast & Greedy

```
fg_o <- modularity(
  cluster_fast_greedy(SR_organisationen_tidy_com)
)
```

Infomap

```
in_o <- modularity(
  cluster_infomap(SR_organisationen_tidy_com)
)
```

Edge Betweenness

```
eb_o <- modularity(
  cluster_edge_betweenness(SR_organisationen_tidy_com)
)
```

```
## Warning in cluster_edge_betweenness(SR_organisationen_tidy_com): At
## community.c:461 :Membership vector will be selected based on the lowest
## modularity score.

## Warning in cluster_edge_betweenness(SR_organisationen_tidy_com): At
## community.c:468 :Modularity calculation with weighted edge betweenness com
munity
## detection might not make sense -- modularity treats edge weights as simila
rities
## while edge betweenness treats them as distances

# Louvian
lv_o <- modularity(
  cluster_louvain(SR_organisationen_tidy_com)
)

## Dataframe
Algorithm <- c("Fast & Greedy",
               "Infomap",
               "Edge Betweenness",
               "Louvian")

Modularity <- c(fg_o,
               in_o,
               eb_o,
               lv_o)

kable(data.frame(Algorithm,Modularity),
      caption = "Tabelle 2:Modularity Scores des Organisationsnetzwerks basie
rend auf verschiedenen Community-Algorithmen")
```

Tabelle 2:Modularity Scores des Organisationsnetzwerks basierend auf verschiedenen Community-Algorithmen

Algorithm	Modularity
Fast & Greedy	0.8073257
Infomap	0.8043639
Edge Betweenness	0.8068629
Louvian	0.8072331

Organisationen - Community

```
set.seed(12345)

SR_organisationen_tidy_com %>%
  activate(nodes) %>%
  slice(index_org) %>%
  mutate(group_louvain= as.factor(group_louvain())) %>%
  ggraph(layout = "fr") +
  geom_edge_link(aes(width = weight),
                alpha = 0.5,
```



```

    show.legend = FALSE) +
  geom_node_point(aes(color = industry_org,
    size = degree)) +
  geom_node_text(aes(label = name),
    repel = TRUE,
    size = 3) +
  geom_mark_hull(aes(x = x,
    y = y,
    fill = group_louvain,
    label = group_louvain),
    concavity = 6) +
  scale_edge_width(range = c(0.2, 0.8)) +
  scale_size(range = c(1, 6)) +
  labs(title = "Organisationsnetzwerk - Geteilte Parlamentarier:innen",
    subtitle = "Je breiter die Kanten, desto mehr Parlamentarier:innen 'teilen' sich die Organisationen",
    size = "Degree",
    color = "Branche",
    caption = "Organisationen mit einer Gewichtung von <= 1 wurden herausgefiltert.")

```

```

## Warning: ggrepel: 132 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps

```

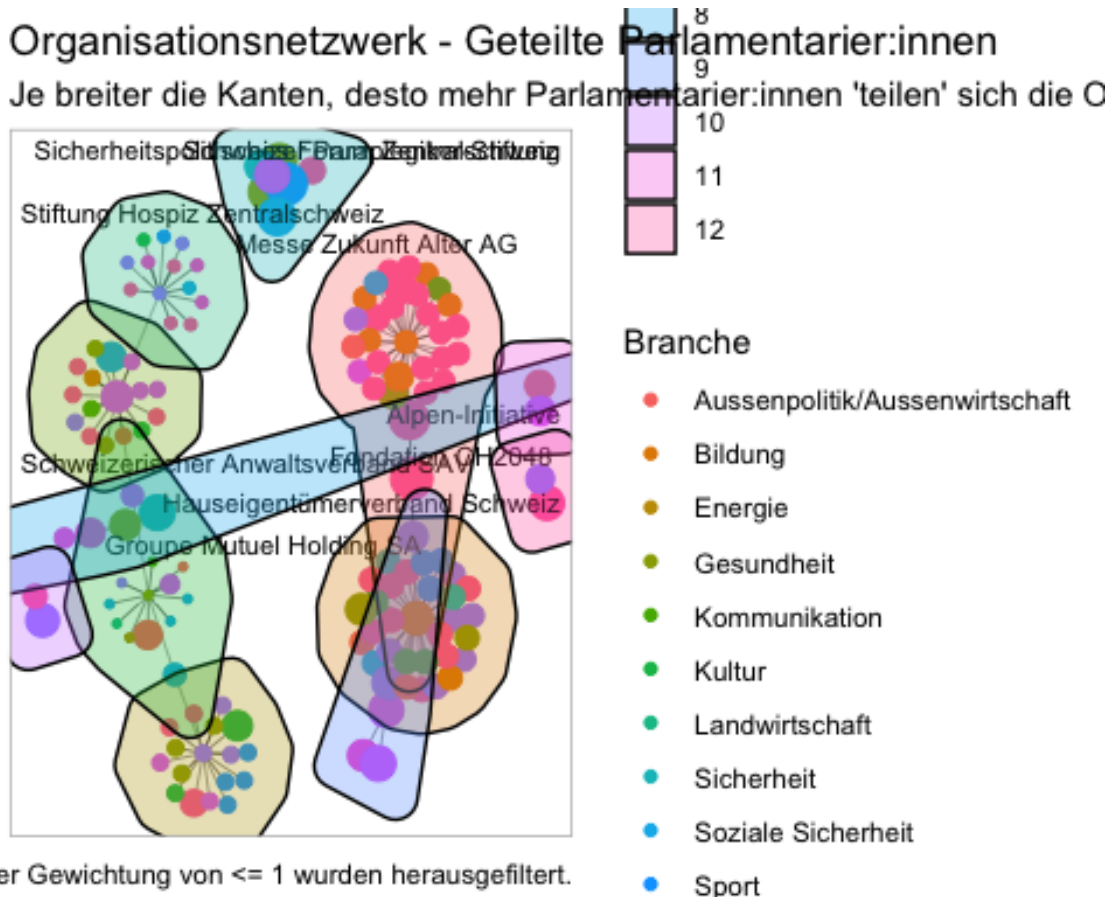


Tabelle 3

```
Communities <- c()

for(i in 1:12){
  text <- paste("Community ", as.character(i))
  Communities <- append(Communities, text)
}

Kategorie <- c("Wirtschaft Zürich",
               "Ostschweizer Netzwerk",
               "Sport und Freizeit",
               "-",
               "Politische Inklusion und Ökologie",
               "Glarner Community",
               "Zentralschweizer Netz",
               "Umwelt und Verkehr",
               "Umwelt und Verkehr",
               "-",
               "-",
               "-")

kable(as.data.frame(Communities, Kategorie),
      caption = "Tabelle 3: Interpretation der Communities im organisationalen Netzwerk")

## Warning in data.row.names(row.names, rowsi, i): einige row.names doppelt:
## 9,10,11,12 -> row.names NICHT benutzt
```

Tabelle 3: Interpretation der Communities im organisationalen Netzwerk

	Communities
Wirtschaft Zürich	Community 1
Ostschweizer Netzwerk	Community 2
Sport und Freizeit	Community 3
-	Community 4
Politische Inklusion und Ökologie	Community 5
Glarner Community	Community 6
Zentralschweizer Netz	Community 7
Umwelt und Verkehr	Community 8
Umwelt und Verkehr	Community 9
-	Community 10
-	Community 11
-	Community 12

Modelling

Random Network Models

Standardisierte Funktion wird geschrieben, damit wir in einem Schritt die Netzwerkparameter aller Modelle herausarbeiten können.

```
parameters_network <- function(name_net, network){

  # Density
  net_dens <- as.numeric(graph.density(network))

  # Size (edges)
  net_size <- as.numeric(gsize(network))

  # Size (nodes)
  net_nodes <- as.numeric(gorder(network))

  # Mean Degree
  net_mdeg <- as.numeric(mean(igraph::degree(network)))

  # Transitivity
  net_trans <- as.numeric(transitivity(network))

  output <- c(name_net, net_dens, net_size, net_nodes, net_mdeg, net_trans)

  return(output)
}

parliment_orgs <- parameters_network("Parlimentary Network", SR_organisationen_tidy_com)

net_nodes <- gorder(SR_organisationen_tidy_com)
net_dens <- graph.density(SR_organisationen_tidy_com)

erg_model <- erdos.renyi.game(n = net_nodes,
                             p.or.m = net_dens,
                             type='gnp')
wsg_model <- watts.strogatz.game(dim=1,
                                 size=net_nodes,
                                 nei = 1,
                                 p=.1)

bg_model <- barabasi.game(net_nodes,
                          out.dist=c(.33,.33,.33),
                          directed=FALSE,
                          zero.appeal=1)
```

```
erg_model_param <- parameters_network("Erdos Renyi Game", erg_model)
wsg_model_param <- parameters_network("Watts Strogatz Game", wsg_model)
bg_model_param <- parameters_network("Barabasi Game", bg_model)
```

Tabelle 4

```
Dataframe_Names <- c("Dataset Name", "Density", "Size (edges)", "Size (nodes)",
  "Mean Degree", "Transitivity")

df_model <- as.data.frame(do.call(rbind,list(Dataframe_Names,
  parliment_orgs,
  erg_model_param,
  wsg_model_param,
  bg_model_param)))

df_model <- df_model %>%
  row_to_names(row_number = 1)

kable(df_model, caption = "Tabelle 4: Model comparison empirical network vs.
modelled networks")
```

Tabelle 4: Model comparison empirical network vs. modelled networks

	Dataset Name	Density	Size (edges)	Size (nodes)	Mean Degree	Transitivity
2	Parliment try Network	0.0007230228296 93458	146	636	0.45911949685 5346	0.033874382498 2357
3	Erdos Renyi Game	0.0007032139850 44322	142	636	0.44654088050 3145	0
4	Watts Strogatz Game	0.0031496062992 126	636	636	2	0
5	Barabasi Game	0.0031050363987 5204	627	636	1.97169811320 755	0.010582010582 0106

Abbildung 4

Vergleich empirisches Netzwerk mit Modell

```
p1 <- SR_organisationen_tidy_com %>%
  activate(edges) %>%
  ggraph(layout = "fr") +
  geom_edge_link() +
  geom_node_point() +
  scale_edge_width(range = c(0.2, 0.8)) +
  scale_size(range = c(1, 6)) +
  labs(title = "Empirisches Netzwerk",
```

```

      subtitle = "Organisationen im Ständerat, Verbunden über Parlamentarier
:innen")

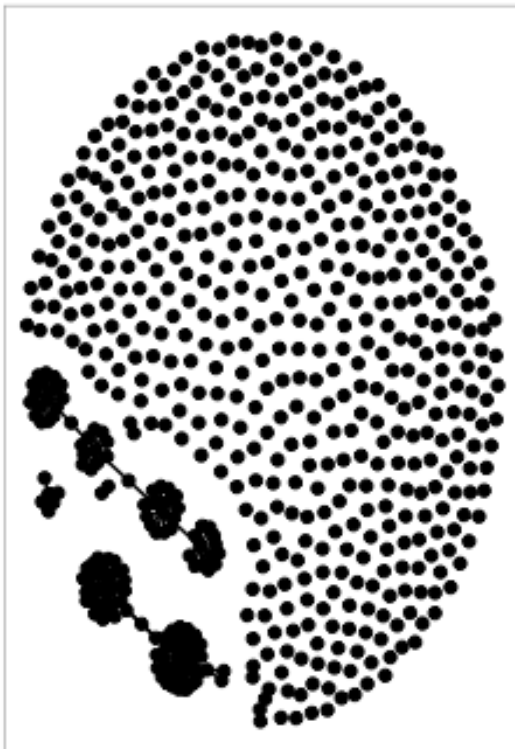
p2 <- as_tbl_graph(erg_model) %>%
  activate(edges) %>%
  ggraph(layout = "fr") +
  geom_edge_link() +
  geom_node_point() +
  scale_edge_width(range = c(0.2, 0.8)) +
  scale_size(range = c(1, 6)) +
  labs(title = "Erdos Renyi Game - Modellierung",
       subtitle = "Modell basierend auf empirischen Daten")

grid.arrange(p1, p2, ncol = 2)

```

Empirisches Netzwerk

Organisationen im Ständerat, Verbu



Erdos Renyi Game - Modellie

Modell basierend auf empirischen L



Statistical Network Models

```

edges <- SR_organisationen_tidy_com %>%
  activate(edges) %>%
  data.frame()

nodes <- SR_organisationen_tidy_com %>%
  activate(nodes) %>%
  filter(degree >= 35) %>%
  data.frame() %>%

```

```

  replace_na(list(location_org = "NA", industry_org = "NA"))

network_base <- network(edges, vertex.attr = nodes, matrix.type = "edgelist",
ignore.eval = FALSE, directed = FALSE)

data(network_base)

## Warning in data(network_base): data set 'network_base' not found

null_model <- ergm(network_base ~ edges, control = control.ergm(seed = 10))

## Starting maximum pseudolikelihood estimation (MPLE):

## Evaluating the predictor and response matrix.

## Maximizing the pseudolikelihood.

## Finished MPLE.

## Stopping at the initial estimate.

## Evaluating log-likelihood at the estimate.

summary(null_model)

## Call:
## ergm(formula = network_base ~ edges, control = control.ergm(seed = 10))
##
## Maximum Likelihood Results:
##
##      Estimate Std. Error MCMC % z value Pr(>|z|)
## edges -4.19875    0.08338      0 -50.36  <1e-04 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##      Null Deviance: 13683  on 9870  degrees of freedom
##      Residual Deviance: 1520  on 9869  degrees of freedom
##
## AIC: 1522  BIC: 1529  (Smaller is better. MC Std. Err. = 0)

```

Abbildung 5

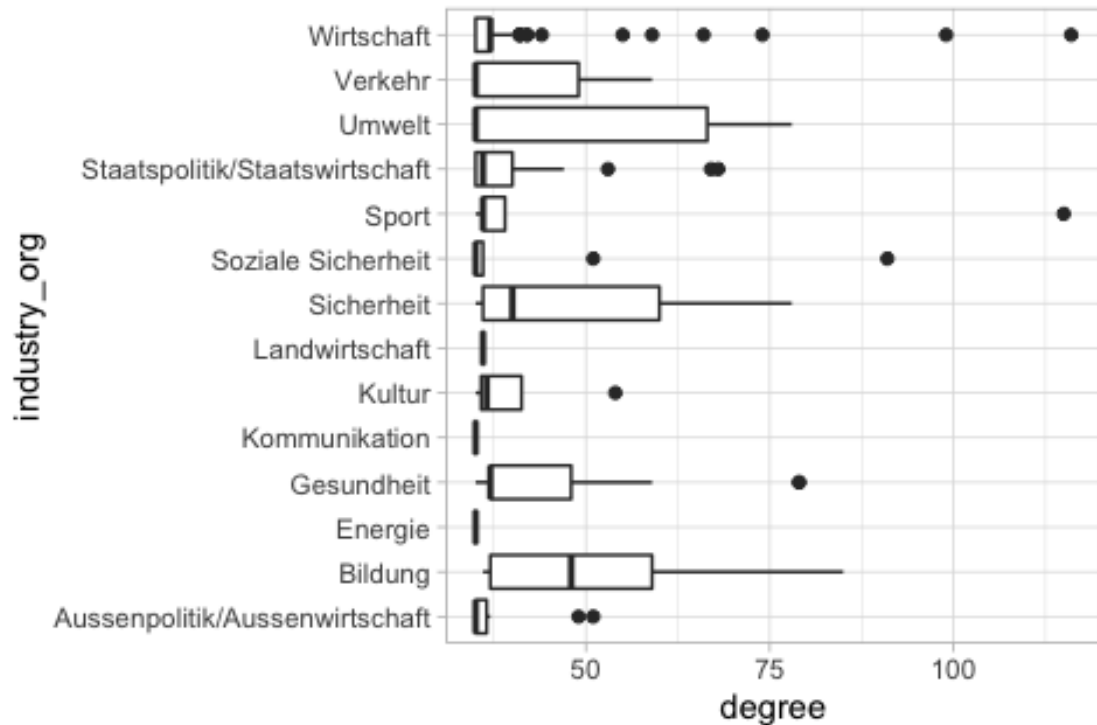
```

SR_organisationen_tidy_com %>%
  activate(nodes) %>%
  filter(degree >= 35) %>%
  data.frame() %>%
  ggplot(aes(x = industry_org, y = degree)) +
  geom_boxplot() +
  coord_flip() +
  labs(title = "Industrie und Degree",
        subtitle = "Visualisierung des Zusammenhanges zwischen der Industrie u
nd dem Degree des Unternehmens",
        caption = "Organisationen mit einem Degree unter 35 wurden herausgefil
ter.")

```

Industrie und Degree

Visualisierung des Zusammenhanges zwisch

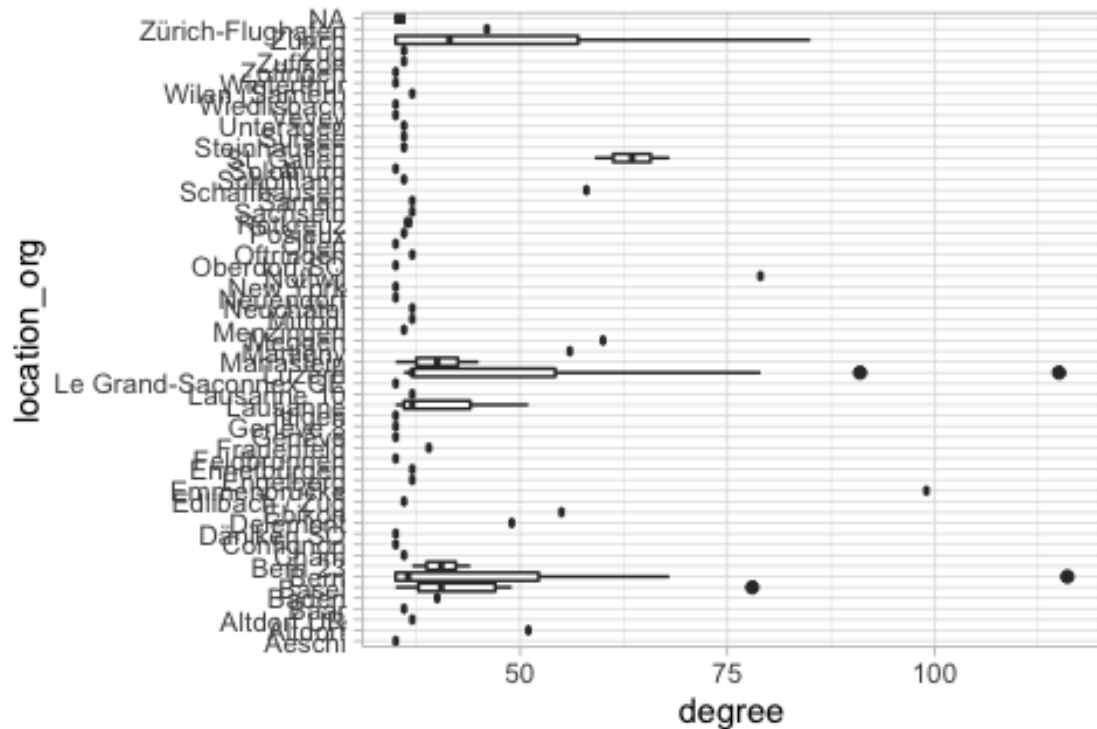


Organisationen mit einem Degree unter 35 wurden herausgefilter.

```
SR_organisationen_tidy_com %>%
  activate(nodes) %>%
  filter(degree >= 35) %>%
  data.frame() %>%
  ggplot(aes(x = location_org, y = degree)) +
  geom_boxplot() +
  coord_flip() +
  labs(title = "Standort und Degree",
        subtitle = "Visualisierung des Zusammenhanges zwischen dem Standort und dem Degree des Unternehmens",
        caption = "Organisationen mit einem Degree unter 35 wurden herausgefilter.")
```

Standort und Degree

Visualisierung des Zusammenhanges zwischen d



Organisationen mit einem Degree unter 35 wurden herausgefiltert.

Tabelle 5

```
data(network_base)

## Warning in data(network_base): data set 'network_base' not found

full_model <- ergm(network_base ~ edges + nodefactor("location_org") + nodefactor("industry_org"),
                   control = control.ergm(seed = 10))

## Starting maximum pseudolikelihood estimation (MPLE):
## Evaluating the predictor and response matrix.
## Maximizing the pseudolikelihood.
## Finished MPLE.
## Stopping at the initial estimate.
## Evaluating log-likelihood at the estimate.

summary(full_model)

## Call:
## ergm(formula = network_base ~ edges + nodefactor("location_org") +
##       nodefactor("industry_org"), control = control.ergm(seed = 10))
```

```

##
## Maximum Likelihood Results:
##
##                                     Estimate Std. Err
or
## edges                             -2.747e+00  2.160e+
00
## nodefactor.location_org.Alt Dorf    -1.514e-01  1.545e+
00
## nodefactor.location_org.Alt Dorf UR -2.742e-01  1.448e+
00
## nodefactor.location_org.Baar        -2.742e-01  1.448e+
00
## nodefactor.location_org.Baden       -1.806e-02  1.631e+
00
## nodefactor.location_org.Basel       -9.276e-01  1.153e+
00
## nodefactor.location_org.Bern        7.026e-02  1.051e+
00
## nodefactor.location_org.Bern 23     2.727e-01  1.199e+
00
## nodefactor.location_org.Cham        -5.626e-02  1.552e+
00
## nodefactor.location_org.Däniken SO  1.733e+00  1.764e+
00
## nodefactor.location_org.Delémont    7.825e-01  1.532e+
00
## nodefactor.location_org.Ebikon       1.428e+00  1.138e+
00
## nodefactor.location_org.Edlibach / Zug -8.931e-01  1.484e+
00
## nodefactor.location_org.Emmenbrücke -2.742e-01  1.448e+
00
## nodefactor.location_org.Engelberg    1.281e-09  1.430e+
00
## nodefactor.location_org.Ennetbürgen -2.742e-01  1.448e+
00
## nodefactor.location_org.Feldbrunnen -2.742e-01  1.448e+
00
## nodefactor.location_org.Frauenfeld  -1.594e-01  1.556e+
00
## nodefactor.location_org.Lausanne     3.238e-03  1.488e+
00
## nodefactor.location_org.Lausanne 10  3.238e-03  1.488e+
00
## nodefactor.location_org.Luzern      -1.628e-01  1.079e+
00
## nodefactor.location_org.Mariastein   3.005e+00  1.035e+
00
## nodefactor.location_org.Martigny     3.238e-03  1.488e+
00

```


## nodefactor.location_org.Meggen 00	-1.806e-02	1.631e+
## nodefactor.location_org.Menzingen 00	-1.512e-01	1.166e+
## nodefactor.location_org.Mitlödi 00	-9.108e-02	1.596e+
## nodefactor.location_org.NA 00	-1.776e-01	1.222e+
## nodefactor.location_org.Neuchâtel 00	5.630e-01	1.205e+
## nodefactor.location_org.Neuendorf 00	-1.534e-01	1.583e+
## nodefactor.location_org.New York 00	-2.742e-01	1.448e+
## nodefactor.location_org.Nottwil 00	3.238e-03	1.488e+
## nodefactor.location_org.Oberdorf SO 00	2.690e+00	1.230e+
## nodefactor.location_org.Oftringen 00	-2.742e-01	1.448e+
## nodefactor.location_org.Olten 00	-1.866e-01	1.294e+
## nodefactor.location_org.Posieux 00	-5.626e-02	1.552e+
## nodefactor.location_org.Rotkreuz 00	-2.742e-01	1.259e+
## nodefactor.location_org.Sachseln 00	-2.742e-01	1.259e+
## nodefactor.location_org.Sarnen 00	5.708e-01	1.082e+
## nodefactor.location_org.Schaffhausen 00	-1.514e-01	1.545e+
## nodefactor.location_org.Schöftland 00	-5.626e-02	1.378e+
## nodefactor.location_org.Solothurn 00	1.733e+00	1.032e+
## nodefactor.location_org.St. Gallen 00	1.860e+00	1.105e+
## nodefactor.location_org.Steinhausen 00	-1.909e-02	1.186e+
## nodefactor.location_org.Sursee 00	-5.626e-02	1.552e+
## nodefactor.location_org.Unterägeri 00	-5.626e-02	1.552e+
## nodefactor.location_org.Wiedlisbach 00	5.631e-01	1.417e+
## nodefactor.location_org.Wilen (Sarnen) 00	-2.742e-01	1.448e+
## nodefactor.location_org.Winterthur 00	-2.742e-01	1.448e+
## nodefactor.location_org.Zofingen	-1.514e-01	1.545e+

```

00
## nodefactor.location_org.Zufikon -2.742e-01 1.448e+
00
## nodefactor.location_org.Zug -1.462e-01 1.127e+
00
## nodefactor.location_org.Zürich 9.608e-02 1.059e+
00
## nodefactor.location_org.Zürich-Flughafen 3.238e-03 1.488e+
00
## nodefactor.industry_org.Bildung -6.062e-01 4.138e-
01
## nodefactor.industry_org.Energie -3.232e+00 1.072e+
00
## nodefactor.industry_org.Gesundheit -1.503e+00 4.474e-
01
## nodefactor.industry_org.Kultur -1.408e+00 7.399e-
01
## nodefactor.industry_org.Landwirtschaft -1.443e+00 6.554e-
01
## nodefactor.industry_org.Sicherheit -1.481e+00 7.947e-
01
## nodefactor.industry_org.Soziale Sicherheit -1.346e+00 7.208e-
01
## nodefactor.industry_org.Sport -1.340e+00 6.585e-
01
## nodefactor.industry_org.Staatspolitik/Staatswirtschaft -1.499e+00 3.823e-
01
## nodefactor.industry_org.Umwelt -1.348e+00 5.766e-
01
## nodefactor.industry_org.Verkehr -2.282e+00 5.980e-
01
## nodefactor.industry_org.Wirtschaft -1.225e+00 3.317e-
01
## MCMC % z value Pr(>
|z|)
## edges 0 -1.271 0.20
3567
## nodefactor.location_org.Altendorf 0 -0.098 0.92
1958
## nodefactor.location_org.Altendorf UR 0 -0.189 0.84
9771
## nodefactor.location_org.Baar 0 -0.189 0.84
9771
## nodefactor.location_org.Baden 0 -0.011 0.99
1167
## nodefactor.location_org.Basel 0 -0.805 0.42
1034
## nodefactor.location_org.Bern 0 0.067 0.94
6688
## nodefactor.location_org.Bern 23 0 0.228 0.82
0025

```

## nodefactor.location_org.Cham 1089	0	-0.036	0.97
## nodefactor.location_org.Däniken SO 5836	0	0.983	0.32
## nodefactor.location_org.Delémont 9413	0	0.511	0.60
## nodefactor.location_org.Ebikon 9645	0	1.255	0.20
## nodefactor.location_org.Edlibach / Zug 7256	0	-0.602	0.54
## nodefactor.location_org.Emmenbrücke 9771	0	-0.189	0.84
## nodefactor.location_org.Engelberg 0000	0	0.000	1.00
## nodefactor.location_org.Ennetbürgen 9771	0	-0.189	0.84
## nodefactor.location_org.Feldbrunnen 9771	0	-0.189	0.84
## nodefactor.location_org.Frauenfeld 8420	0	-0.102	0.91
## nodefactor.location_org.Lausanne 8264	0	0.002	0.99
## nodefactor.location_org.Lausanne 10 8264	0	0.002	0.99
## nodefactor.location_org.Luzern 0096	0	-0.151	0.88
## nodefactor.location_org.Mariastein 3688	0	2.904	0.00
## nodefactor.location_org.Martigny 8264	0	0.002	0.99
## nodefactor.location_org.Meggen 1167	0	-0.011	0.99
## nodefactor.location_org.Menzingen 6819	0	-0.130	0.89
## nodefactor.location_org.Mitlödi 4491	0	-0.057	0.95
## nodefactor.location_org.NA 4449	0	-0.145	0.88
## nodefactor.location_org.Neuchâtel 0283	0	0.467	0.64
## nodefactor.location_org.Neuendorf 2825	0	-0.097	0.92
## nodefactor.location_org.New York 9771	0	-0.189	0.84
## nodefactor.location_org.Nottwil 8264	0	0.002	0.99
## nodefactor.location_org.Oberdorf SO 8779	0	2.187	0.02
## nodefactor.location_org.Oftringen 9771	0	-0.189	0.84
## nodefactor.location_org.Olten	0	-0.144	0.88

5313			
## nodefactor.location_org.Posieux	0	-0.036	0.97
1089			
## nodefactor.location_org.Rotkreuz	0	-0.218	0.82
7545			
## nodefactor.location_org.Sachseln	0	-0.218	0.82
7545			
## nodefactor.location_org.Sarnen	0	0.528	0.59
7655			
## nodefactor.location_org.Schaffhausen	0	-0.098	0.92
1958			
## nodefactor.location_org.Schöftland	0	-0.041	0.96
7428			
## nodefactor.location_org.Solothurn	0	1.679	0.09
3062			
## nodefactor.location_org.St. Gallen	0	1.683	0.09
2351			
## nodefactor.location_org.Steinhausen	0	-0.016	0.98
7161			
## nodefactor.location_org.Sursee	0	-0.036	0.97
1089			
## nodefactor.location_org.Unterägeri	0	-0.036	0.97
1089			
## nodefactor.location_org.Wiedlisbach	0	0.397	0.69
1022			
## nodefactor.location_org.Wilen (Sarnen)	0	-0.189	0.84
9771			
## nodefactor.location_org.Winterthur	0	-0.189	0.84
9771			
## nodefactor.location_org.Zofingen	0	-0.098	0.92
1958			
## nodefactor.location_org.Zufikon	0	-0.189	0.84
9771			
## nodefactor.location_org.Zug	0	-0.130	0.89
6763			
## nodefactor.location_org.Zürich	0	0.091	0.92
7716			
## nodefactor.location_org.Zürich-Flughafen	0	0.002	0.99
8264			
## nodefactor.industry_org.Bildung	0	-1.465	0.14
2931			
## nodefactor.industry_org.Energie	0	-3.016	0.00
2559			
## nodefactor.industry_org.Gesundheit	0	-3.359	0.00
0783			
## nodefactor.industry_org.Kultur	0	-1.903	0.05
6996			
## nodefactor.industry_org.Landwirtschaft	0	-2.202	0.02
7682			
## nodefactor.industry_org.Sicherheit	0	-1.864	0.06
2316			

```

## nodefactor.industry_org.Soziale Sicherheit      0 -1.867 0.06
1844
## nodefactor.industry_org.Sport                   0 -2.035 0.04
1875
## nodefactor.industry_org.Staatspolitik/Staatswirtschaft 0 -3.921 < 1
e-04
## nodefactor.industry_org.Umwelt                  0 -2.338 0.01
9397
## nodefactor.industry_org.Verkehr                  0 -3.816 0.00
0136
## nodefactor.industry_org.Wirtschaft               0 -3.694 0.00
0221
##
## edges
## nodefactor.location_org.Altdorf
## nodefactor.location_org.Altdorf UR
## nodefactor.location_org.Baar
## nodefactor.location_org.Baden
## nodefactor.location_org.Basel
## nodefactor.location_org.Bern
## nodefactor.location_org.Bern 23
## nodefactor.location_org.Cham
## nodefactor.location_org.Däniken S0
## nodefactor.location_org.Delémont
## nodefactor.location_org.Ebikon
## nodefactor.location_org.Edlibach / Zug
## nodefactor.location_org.Emmenbrücke
## nodefactor.location_org.Engelberg
## nodefactor.location_org.Ennetbürgen
## nodefactor.location_org.Feldbrunnen
## nodefactor.location_org.Frauenfeld
## nodefactor.location_org.Lausanne
## nodefactor.location_org.Lausanne 10
## nodefactor.location_org.Luzern
## nodefactor.location_org.Mariastein              **
## nodefactor.location_org.Martigny
## nodefactor.location_org.Meggen
## nodefactor.location_org.Menzingen
## nodefactor.location_org.Mitlödi
## nodefactor.location_org.NA
## nodefactor.location_org.Neuchâtel
## nodefactor.location_org.Neuendorf
## nodefactor.location_org.New York
## nodefactor.location_org.Nottwil
## nodefactor.location_org.Oberdorf S0             *
## nodefactor.location_org.Oftringen
## nodefactor.location_org.Olten
## nodefactor.location_org.Posieux
## nodefactor.location_org.Rotkreuz
## nodefactor.location_org.Sachsln
## nodefactor.location_org.Sarnen

```

```

## nodefactor.location_org.Schaffhausen
## nodefactor.location_org.Schöftland
## nodefactor.location_org.Solothurn .
## nodefactor.location_org.St. Gallen .
## nodefactor.location_org.Steinhausen
## nodefactor.location_org.Sursee
## nodefactor.location_org.Unterägeri
## nodefactor.location_org.Wiedlisbach
## nodefactor.location_org.Wilen (Sarnen)
## nodefactor.location_org.Winterthur
## nodefactor.location_org.Zofingen
## nodefactor.location_org.Zufikon
## nodefactor.location_org.Zug
## nodefactor.location_org.Zürich
## nodefactor.location_org.Zürich-Flughafen
## nodefactor.industry_org.Bildung
## nodefactor.industry_org.Energie **
## nodefactor.industry_org.Gesundheit ***
## nodefactor.industry_org.Kultur .
## nodefactor.industry_org.Landwirtschaft *
## nodefactor.industry_org.Sicherheit .
## nodefactor.industry_org.Soziale Sicherheit .
## nodefactor.industry_org.Sport *
## nodefactor.industry_org.Staatspolitik/Staatswirtschaft ***
## nodefactor.industry_org.Umwelt *
## nodefactor.industry_org.Verkehr ***
## nodefactor.industry_org.Wirtschaft ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Null Deviance: 13683 on 9870 degrees of freedom
## Residual Deviance: 1256 on 9805 degrees of freedom
##
## AIC: 1386 BIC: 1854 (Smaller is better. MC Std. Err. = 0)

```

Simulation

Tabelle 5

```
simu <- simulate(full_model, nsim = 1, seed = 1234)
```

```
summary(simu)
```

```

## Network attributes:
## vertices = 141
## directed = FALSE
## hyper = FALSE
## loops = FALSE
## multiple = FALSE
## bipartite = FALSE
## total edges = 140

```

```

## missing edges = 0
## non-missing edges = 140
## density = 0.0141844
##
## Vertex attributes:
##
## canton:
## character valued attribute
## attribute summary:
## integer(0)
##
## closeness:
## numeric valued attribute
## attribute summary:
##      Min.   1st Qu.   Median     Mean   3rd Qu.     Max.
## 0.0001559 0.0001634 0.0001635 0.0001644 0.0001639 0.0001727
##
## degree:
## numeric valued attribute
## attribute summary:
##      Min. 1st Qu.  Median     Mean 3rd Qu.     Max.
##    35.00  36.00  37.00   43.83  45.00   116.00
##
## industry_org:
## character valued attribute
## attribute summary:
## the 10 most common values are:
##
##           Wirtschaft Staatspolitik/Staatswirtschaft
##                51                                18
##           Gesundheit Landwirtschaft
##                17                                10
##           Bildung                                Umwelt
##                7                                7
##           Verkehr Aussenpolitik/Aussenwirtschaft
##                6                                5
##           Soziale Sicherheit Sport
##                5                                5
##
## interest_org:
## character valued attribute
## attribute summary:
## the 10 most common values are:
##
##           Krankenkassen Advokaturen/Treuhan
##
##                10
##
##           Immobilien/Hauseigentümer Parteie
##
##                8
##
##           Bildung/Wissenschaft Zünfte, Verbindungen und Serviceclub

```

```

s
##                                     6
6
##                               KMU/Gewerbe/Arbeitgeber                               Religio
n
##                                     5
5
##                               Umwelt & Natur                               Öffentlicher Verkeh
r
##                                     5
4
##
## location_org:
##   character valued attribute
##   attribute summary:
##   the 10 most common values are:
##       Bern      Zürich      Luzern      Solothurn      Zug      Basel
##       17        17         16         15         7        5
##   Menzingen    Sarnen      NA Steinhausen
##       4         4         3         3
##
## name:
##   character valued attribute
##   attribute summary:
##   the 10 most common values are:
##                               AEK onyx AG                               a
griss
##                                     1
1
##                               Allianz Sicherheit Schweiz                               Alpen-Initi
ative
##                                     1
1
##       alps - Alpines Museum der Schweiz   amitola, gemeinnützige GmbH für K
inder
##                                     1
1
##                               Aqua Viva Arbeitskreis Sicherheit und Wehrte
chnik
##                                     1
1
##                               Arcosana AG                               Association Oues
trail
##                                     1
1
##
## party:
##   character valued attribute
##   attribute summary:
## integer(0)
##

```



```

## sex:
##   character valued attribute
##   attribute summary:
## integer(0)
##   vertex.names:
##   character valued attribute
##   141 valid vertex names
##
## No edge attributes
##
## Network edgelist matrix:
##      [,1] [,2]
## [1,]    2   17
## [2,]    2   27
## [3,]    2  118
## [4,]    3   23
## [5,]    3   41
## [6,]    4   20
## [7,]    4   32
## [8,]    5   29
## [9,]    5   41
## [10,]   6   24
## [11,]   6   41
## [12,]   6   43
## [13,]   6   52
## [14,]   6   79
## [15,]   6  132
## [16,]   6  139
## [17,]   8   24
## [18,]   9   18
## [19,]   9   20
## [20,]   9   21
## [21,]   9   32
## [22,]   9   45
## [23,]   9  103
## [24,]   9  137
## [25,]  13  100
## [26,]  14   28
## [27,]  14   80
## [28,]  14  126
## [29,]  15   16
## [30,]  15   49
## [31,]  17   83
## [32,]  17  110
## [33,]  17  137
## [34,]  18   27
## [35,]  18   30
## [36,]  18   32
## [37,]  18  117
## [38,]  19   21
## [39,]  19   42

```

##	[40,]	19	50
##	[41,]	20	24
##	[42,]	20	25
##	[43,]	20	32
##	[44,]	20	41
##	[45,]	20	97
##	[46,]	20	137
##	[47,]	21	29
##	[48,]	21	43
##	[49,]	21	63
##	[50,]	22	28
##	[51,]	22	30
##	[52,]	22	50
##	[53,]	22	56
##	[54,]	23	27
##	[55,]	23	47
##	[56,]	23	104
##	[57,]	24	28
##	[58,]	24	74
##	[59,]	24	138
##	[60,]	27	43
##	[61,]	27	103
##	[62,]	28	43
##	[63,]	28	135
##	[64,]	28	140
##	[65,]	29	97
##	[66,]	30	32
##	[67,]	30	38
##	[68,]	30	61
##	[69,]	30	70
##	[70,]	30	130
##	[71,]	30	131
##	[72,]	30	134
##	[73,]	30	137
##	[74,]	31	108
##	[75,]	32	38
##	[76,]	32	40
##	[77,]	32	41
##	[78,]	32	43
##	[79,]	32	50
##	[80,]	32	54
##	[81,]	32	56
##	[82,]	32	61
##	[83,]	32	66
##	[84,]	32	85
##	[85,]	32	97
##	[86,]	32	103
##	[87,]	32	122
##	[88,]	32	139
##	[89,]	32	140
##	[90,]	33	41

##	[91,]	33	141
##	[92,]	39	41
##	[93,]	39	45
##	[94,]	40	41
##	[95,]	40	74
##	[96,]	40	76
##	[97,]	40	88
##	[98,]	41	42
##	[99,]	41	43
##	[100,]	41	46
##	[101,]	41	50
##	[102,]	41	64
##	[103,]	41	78
##	[104,]	41	93
##	[105,]	41	100
##	[106,]	41	101
##	[107,]	41	112
##	[108,]	41	136
##	[109,]	41	138
##	[110,]	41	140
##	[111,]	41	141
##	[112,]	42	50
##	[113,]	42	65
##	[114,]	42	66
##	[115,]	43	50
##	[116,]	43	86
##	[117,]	43	93
##	[118,]	43	101
##	[119,]	43	112
##	[120,]	43	118
##	[121,]	43	119
##	[122,]	43	141
##	[123,]	46	89
##	[124,]	47	64
##	[125,]	47	97
##	[126,]	56	82
##	[127,]	56	127
##	[128,]	60	118
##	[129,]	61	137
##	[130,]	62	128
##	[131,]	74	78
##	[132,]	78	131
##	[133,]	79	130
##	[134,]	81	113
##	[135,]	86	98
##	[136,]	97	99
##	[137,]	112	122
##	[138,]	117	120
##	[139,]	125	141
##	[140,]	133	141

Abbildung 6

```
simu_tbl <- as_tbl_graph(simu)
```

```
set.seed(12345)
SR_organisationen_tidy_com %>%
  activate(nodes) %>%
  filter(degree >= 35) %>%
  ggraph(layout = "fr") +
  geom_edge_link(color = "grey",
                 alpha = 0.5,
                 show.legend = FALSE) +
  geom_node_point(aes(color = industry_org,
                      size = degree)) +
  geom_node_text(aes(label = name),
                 check_overlap = TRUE,
                 size = 2) +
  scale_edge_width(range = c(0.1, 0.75)) +
  scale_size(range = c(1, 6)) +
  labs(title = "Empirisches Organisationsnetzwerk",
       size = "Degree",
       color = "Branche",
       caption = "Organisationen mit einem Degree < 35 wurden herausgefiltert
.")
```

Empirisches Organisationsnetzwerk



- Bildung
- Energie
- Gesundheit
- Kommunikation
- Kultur
- Landwirtschaft
- Sicherheit
- Soziale Sicherheit
- Sport
- Staatspolitik/Staatswirtschaft
- Umwelt
- Verkehr
- Wirtschaft

Degree

- 40
- 60

n mit einem Degree < 35 wurden herausgefiltert.

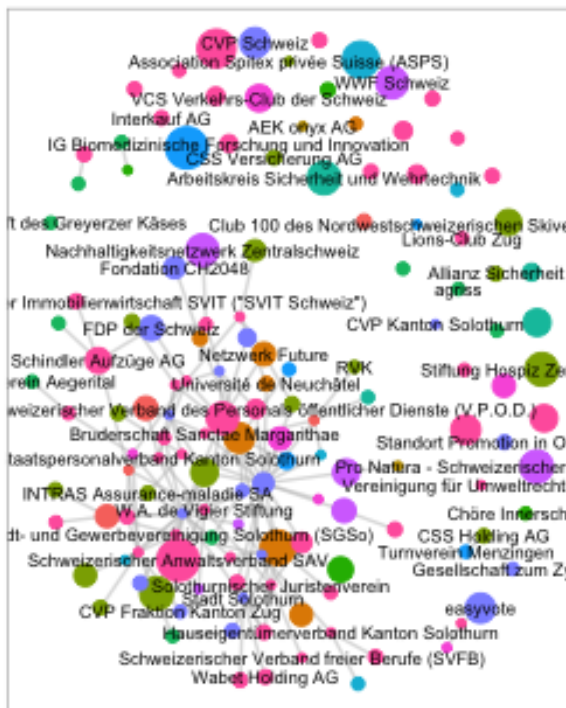
```

simu_tbl %>%
  activate(nodes) %>%
  filter(degree >= 35) %>%
  ggraph(layout = "fr") +
  geom_edge_link(color = "grey",
                alpha = 0.5,
                show.legend = FALSE) +
  geom_node_point(aes(color = industry_org,
                    size = degree)) +
  geom_node_text(aes(label = name),
                check_overlap = TRUE,
                size = 2) +
  scale_edge_width(range = c(0.1, 0.75)) +
  scale_size(range = c(1, 6)) +
  labs(title = "Simuliertes Organisationsnetzwerk",
       subtitle = "Methode: ERGM",
       size = "Degree",
       color = "Branche",
       caption = "Organisationen mit einem Degree < 35 wurden herausgefiltert
.")

```

Simuliertes Organisationsnetzwerk

Methode: ERGM



- Aussenpolitik/Aussenwirtschaft
- Bildung
- Energie
- Gesundheit
- Kultur
- Landwirtschaft
- Sicherheit
- Soziale Sicherheit
- Sport
- Staatspolitik/Staatswirtschaft
- Umwelt
- Verkehr
- Wirtschaft

Degree

- 40
- 60

n mit einem Degree < 35 wurden herausgefiltert.