Aufgabenblatt zur Heimarbeit 4

Mirco Bazzani, Amir Shehadeh & Luca Keiser

2022-05-26

## Aufgabenblatt zur Heimarbeit 4

[Aufgabenblatt zur Heimarbeit 4 1](#_Toc104748965)

[Methoden der sozialen Netzwerkanalyse 1](#_Toc104748966)

[Literaturverzeichnis 4](#_Toc104748967)

[Anhang - R-Code 4](#_Toc104748968)

[Bipartites Netzwerk 7](#_Toc104748969)

[Knotenattribute 9](#_Toc104748970)

[Grafiken 12](#_Toc104748971)

[Bipartites Netzwerk 12](#_Toc104748972)

[Erstellen der Projektionen 13](#_Toc104748973)

[Organisationen 1 15](#_Toc104748974)

[Organisationen 2 16](#_Toc104748975)

[Parlamentarier:innen 17](#_Toc104748976)

[Parlamentarier:innen - Community Detection 18](#_Toc104748977)

[Parlamentarier:innen - Communities 19](#_Toc104748978)

[Organisationen - Community Detection 21](#_Toc104748979)

[Organisationen - Community 22](#_Toc104748980)

[Tabelle 3 24](#_Toc104748981)

[Modelling 25](#_Toc104748982)

[Random Network Models 25](#_Toc104748983)

[Tabelle 4 26](#_Toc104748984)

[Abbildung 4 26](#_Toc104748985)

[Statistical Network Models 27](#_Toc104748986)

[Abbildung 5 28](#_Toc104748987)

[Tabelle 5 30](#_Toc104748988)

[Simulation 37](#_Toc104748989)

[Tabelle 5 37](#_Toc104748990)

[Abbildung 6 43](#_Toc104748991)

### 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 Organisationnetzwerk 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 Comminity 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 lasssen 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 weitervwerendet. 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 Comminity 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 Edgeliste (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 affiliiert sind (Abbildung xy).

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 Standerät:innen einen klaren mitte-rechts-Bias besitzt. Zusätzlich können wir beobachten, dass grosse Instiutionen, 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 Erdös-Rény), ein Small World Modell und schlussenldich 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 Erdös-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 Oragnisationen 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 Exponetial 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, idem 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>
* 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\_conflicts() ──  
## 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 meherer 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, parlame...  
## lgl (1): parlamentarier\_im\_rat\_bis  
## date (4): parlamentarier\_im\_rat\_seit, parlamentarier\_geburtstag, interessen...  
##   
## 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 n  
## <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. Organisationen)  
#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) Parlamentaier: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 industry\_org  
## <chr> <lgl> <chr> <chr> <chr> <chr> <chr> <chr>   
## 1 Arbeitskreis… FALSE <NA> <NA> <NA> Zürich Rüstungsind… Sicherheit   
## 2 Schweizerisc… FALSE <NA> <NA> <NA> Zürich KMU/Gewerbe… Wirtschaft   
## 3 FDP der Schw… FALSE <NA> <NA> <NA> Bern Parteien Staatspolit…  
## 4 Université d… FALSE <NA> <NA> <NA> Neuchâtel Bildung/Wis… Bildung   
## 5 IG Freiheit FALSE <NA> <NA> <NA> Zollikon Staatspolit… Staatspolit…  
## 6 Schweizerisc… FALSE <NA> <NA> <NA> Zürich Baunebengew… Wirtschaft   
## 7 Schweizerisc… FALSE <NA> <NA> <NA> Bern Advokaturen… Wirtschaft   
## 8 Swiss Medica… FALSE <NA> <NA> <NA> Fribourg Spitäler Gesundheit   
## 9 AJS ingénieu… FALSE <NA> <NA> <NA> Neuchâtel Bauhauptgew… Wirtschaft   
## 10 Société Anon… FALSE <NA> <NA> <NA> Neuchâtel Immobilien/… Wirtschaft   
## 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>

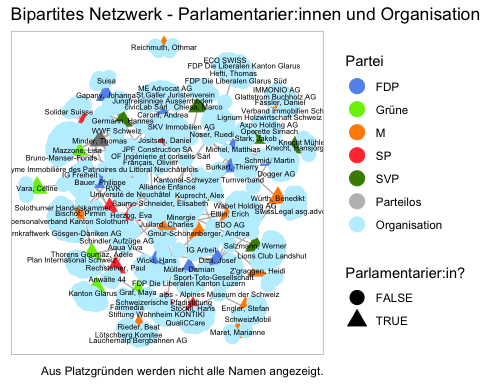
tibble(  
 name = V(SR\_bipartite\_igraph)$name,  
 type = V(SR\_bipartite\_igraph)$type,  
 sex = V(SR\_bipartite\_igraph)$sex,  
 party = V(SR\_bipartite\_igraph)$party,  
 canton = V(SR\_bipartite\_igraph)$canton,  
 place = V(SR\_bipartite\_igraph)$location\_org,  
 interest = V(SR\_bipartite\_igraph)$interest\_org,  
 branche = V(SR\_bipartite\_igraph)$industry\_org  
) %>%   
 group\_by(type) %>%   
 slice\_head(n = 10)

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



### Erstellen der Projektionen

# Projektionen  
  
SR\_projection <- bipartite.projection(SR\_bipartite\_tidy)  
head(SR\_projection)

## $proj1  
## IGRAPH 21684e7 UNW- 636 7680 --   
## + attr: name (v/c), sex (v/c), party (v/c), canton (v/c), location\_org  
## | (v/c), interest\_org (v/c), industry\_org (v/c), weight (e/n)  
## + edges from 21684e7 (vertex names):  
## [1] Arbeitskreis Sicherheit und Wehrtechnik--Schweizerischer Arbeitgeberverband  
## [2] Arbeitskreis Sicherheit und Wehrtechnik--FDP der Schweiz   
## [3] Arbeitskreis Sicherheit und Wehrtechnik--Université de Neuchâtel   
## [4] Arbeitskreis Sicherheit und Wehrtechnik--IG Freiheit   
## + ... omitted several edges  
##   
## $proj2  
## IGRAPH c1e7365 UNW- 45 118 --   
## + attr: name (v/c), sex (v/c), party (v/c), canton (v/c), location\_org  
## | (v/c), interest\_org (v/c), industry\_org (v/c), weight (e/n)  
## + edges from c1e7365 (vertex names):  
## [1] Bauer, Philippe --Häberli-Koller, Brigitte   
## [2] Bauer, Philippe --Kuprecht, Alex   
## [3] Bauer, Philippe --Salzmann, Werner   
## [4] Bauer, Philippe --Caroni, Andrea   
## [5] Bauer, Philippe --Baume-Schneider, Elisabeth  
## [6] Bauer, Philippe --Bischof, Pirmin   
## [7] Bauer, Philippe --Jositsch, Daniel   
## + ... omitted several edges

# 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

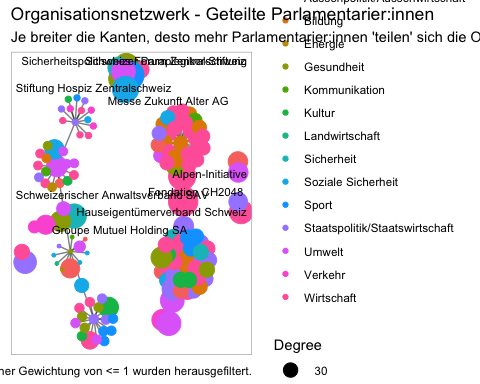
SR\_parlamentarier\_tidy <- as\_tbl\_graph(SR\_parlamentarier\_igraph) %>%   
 mutate(degree = centrality\_degree(),  
 closeness = centrality\_closeness())

## 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

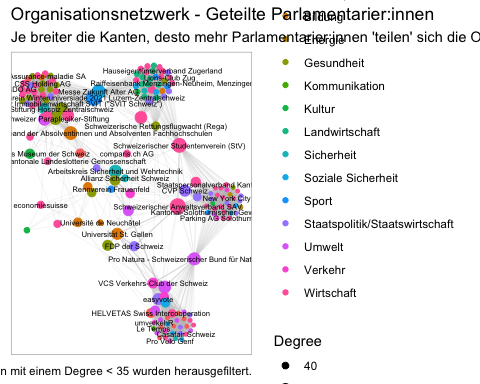
# in einem ersten Schritt muss dazu ein Filterungsindex erstellt werden  
index\_org <- SR\_organisationen\_tidy %>%  
 activate(edges) %>%   
 filter(weight > 1) %>%   
 as\_tibble() %>%   
 select(from, to)  
  
index\_org <- c(index\_org$from, index\_org$to) %>%   
 unique()  
  
set.seed(12345)  
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 herausgefiltert.")

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



## Organisationen 2

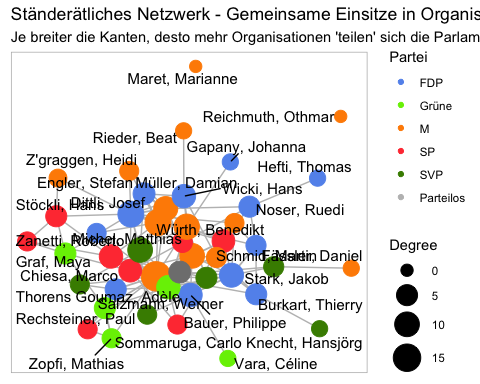
set.seed(12345)  
SR\_organisationen\_tidy %>%  
 activate(nodes) %>%   
 filter(degree >= 35) %>%   
 ggraph(layout = "fr") +  
 geom\_edge\_link(aes(width = weight),  
 color = "grey",  
 alpha = 0.5,  
 show.legend = FALSE) +  
 geom\_node\_point(aes(color = industry\_org,  
 size = degree)) +  
 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.")



## 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 Organisationen",  
 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



## 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 community  
## detection might not make sense -- modularity treats edge weights as similarities  
## while edge betwenness 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-Netzwerks basierend auf verschiedenen Community-Algorithmen")

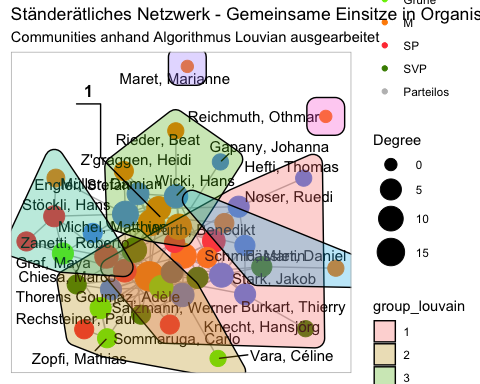
Tabelle 1: Modularity Scores des Parlamentarier:innen-Netzwerks basierend auf verschiedenen Community-Algorithmen

|  |  |
| --- | --- |
| 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 Organisationen | Communities",  
 subtitle = "Communities anhand Algorithmus Louvian ausgearbeitet",  
 size = "Degree",  
 color = "Partei")

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



## 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 community  
## detection might not make sense -- modularity treats edge weights as similarities  
## while edge betwenness 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 basierend auf verschiedenen Community-Algorithmen")

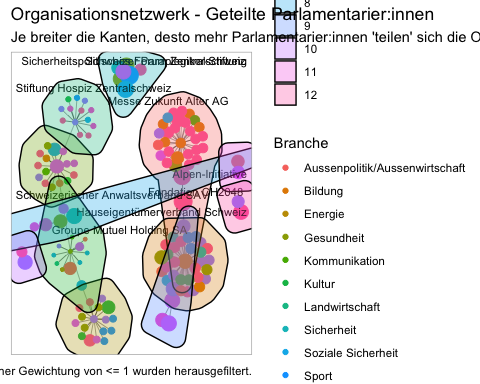
Tabelle 2:Modularity Scores des Organisationsnetzwerks basierend auf verschiedenen Community-Algorithmen

|  |  |
| --- | --- |
| 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("Parlimenatry 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\_mode\_param <- parameters\_network("Barabasi Game", bg\_model)

## Tabelle 4

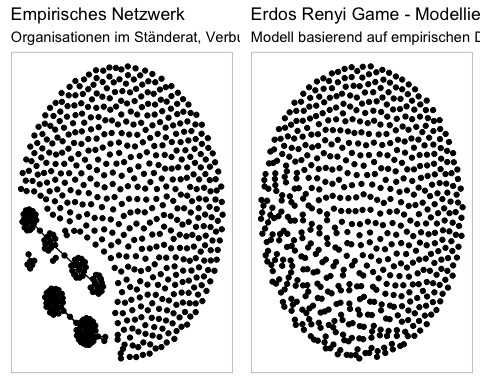
Dataframe\_Names <- c("Dataset Name", "Density", "Size (edges)", "Size (nodes)", "Mean Degree", "Transitivity")  
  
df\_model <- as.data.frame(do.call(rbind,list(Dataframe\_Names,  
 parliment\_orgs,  
 erg\_model\_param,  
 wsg\_model\_param,  
 bg\_mode\_param)))  
  
df\_model <- df\_model %>%   
 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 | Parlimenatry Network | 0.000723022829693458 | 146 | 636 | 0.459119496855346 | 0.0338743824982357 |
| 3 | Erdos Renyi Game | 0.000703213985044322 | 142 | 636 | 0.446540880503145 | 0 |
| 4 | Watts Strogatz Game | 0.0031496062992126 | 636 | 636 | 2 | 0 |
| 5 | Barabasi Game | 0.00310503639875204 | 627 | 636 | 1.97169811320755 | 0.0105820105820106 |

## 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)



## 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.

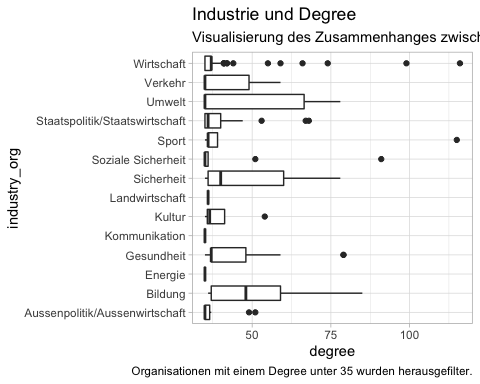
## Evaluating log-likelihood at the estimate.

summary(null\_model)

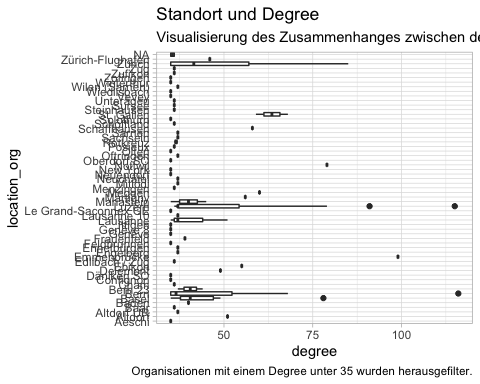
## Call:  
## ergm(formula = network\_base ~ edges, control = control.ergm(seed = 10))  
##   
## Maximum Likelihood Results:  
##   
## Estimate Std. Error MCMC % z value Pr(>|z|)   
## edges -4.19875 0.08338 0 -50.36 <1e-04 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 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 und dem Degree des Unternehmens",  
 caption = "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.")



## 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. Error  
## edges -2.747e+00 2.160e+00  
## nodefactor.location\_org.Altdorf -1.514e-01 1.545e+00  
## nodefactor.location\_org.Altdorf 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 -1.806e-02 1.631e+00  
## nodefactor.location\_org.Menzingen -1.512e-01 1.166e+00  
## nodefactor.location\_org.Mitlödi -9.108e-02 1.596e+00  
## nodefactor.location\_org.NA -1.776e-01 1.222e+00  
## nodefactor.location\_org.Neuchâtel 5.630e-01 1.205e+00  
## nodefactor.location\_org.Neuendorf -1.534e-01 1.583e+00  
## nodefactor.location\_org.New York -2.742e-01 1.448e+00  
## nodefactor.location\_org.Nottwil 3.238e-03 1.488e+00  
## nodefactor.location\_org.Oberdorf SO 2.690e+00 1.230e+00  
## nodefactor.location\_org.Oftringen -2.742e-01 1.448e+00  
## nodefactor.location\_org.Olten -1.866e-01 1.294e+00  
## nodefactor.location\_org.Posieux -5.626e-02 1.552e+00  
## nodefactor.location\_org.Rotkreuz -2.742e-01 1.259e+00  
## nodefactor.location\_org.Sachseln -2.742e-01 1.259e+00  
## nodefactor.location\_org.Sarnen 5.708e-01 1.082e+00  
## nodefactor.location\_org.Schaffhausen -1.514e-01 1.545e+00  
## nodefactor.location\_org.Schöftland -5.626e-02 1.378e+00  
## nodefactor.location\_org.Solothurn 1.733e+00 1.032e+00  
## nodefactor.location\_org.St. Gallen 1.860e+00 1.105e+00  
## nodefactor.location\_org.Steinhausen -1.909e-02 1.186e+00  
## nodefactor.location\_org.Sursee -5.626e-02 1.552e+00  
## nodefactor.location\_org.Unterägeri -5.626e-02 1.552e+00  
## nodefactor.location\_org.Wiedlisbach 5.631e-01 1.417e+00  
## nodefactor.location\_org.Wilen (Sarnen) -2.742e-01 1.448e+00  
## nodefactor.location\_org.Winterthur -2.742e-01 1.448e+00  
## 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 0.203567  
## nodefactor.location\_org.Altdorf 0 -0.098 0.921958  
## nodefactor.location\_org.Altdorf UR 0 -0.189 0.849771  
## nodefactor.location\_org.Baar 0 -0.189 0.849771  
## nodefactor.location\_org.Baden 0 -0.011 0.991167  
## nodefactor.location\_org.Basel 0 -0.805 0.421034  
## nodefactor.location\_org.Bern 0 0.067 0.946688  
## nodefactor.location\_org.Bern 23 0 0.228 0.820025  
## nodefactor.location\_org.Cham 0 -0.036 0.971089  
## nodefactor.location\_org.Däniken SO 0 0.983 0.325836  
## nodefactor.location\_org.Delémont 0 0.511 0.609413  
## nodefactor.location\_org.Ebikon 0 1.255 0.209645  
## nodefactor.location\_org.Edlibach / Zug 0 -0.602 0.547256  
## nodefactor.location\_org.Emmenbrücke 0 -0.189 0.849771  
## nodefactor.location\_org.Engelberg 0 0.000 1.000000  
## nodefactor.location\_org.Ennetbürgen 0 -0.189 0.849771  
## nodefactor.location\_org.Feldbrunnen 0 -0.189 0.849771  
## nodefactor.location\_org.Frauenfeld 0 -0.102 0.918420  
## nodefactor.location\_org.Lausanne 0 0.002 0.998264  
## nodefactor.location\_org.Lausanne 10 0 0.002 0.998264  
## nodefactor.location\_org.Luzern 0 -0.151 0.880096  
## nodefactor.location\_org.Mariastein 0 2.904 0.003688  
## nodefactor.location\_org.Martigny 0 0.002 0.998264  
## nodefactor.location\_org.Meggen 0 -0.011 0.991167  
## nodefactor.location\_org.Menzingen 0 -0.130 0.896819  
## nodefactor.location\_org.Mitlödi 0 -0.057 0.954491  
## nodefactor.location\_org.NA 0 -0.145 0.884449  
## nodefactor.location\_org.Neuchâtel 0 0.467 0.640283  
## nodefactor.location\_org.Neuendorf 0 -0.097 0.922825  
## nodefactor.location\_org.New York 0 -0.189 0.849771  
## nodefactor.location\_org.Nottwil 0 0.002 0.998264  
## nodefactor.location\_org.Oberdorf SO 0 2.187 0.028779  
## nodefactor.location\_org.Oftringen 0 -0.189 0.849771  
## nodefactor.location\_org.Olten 0 -0.144 0.885313  
## nodefactor.location\_org.Posieux 0 -0.036 0.971089  
## nodefactor.location\_org.Rotkreuz 0 -0.218 0.827545  
## nodefactor.location\_org.Sachseln 0 -0.218 0.827545  
## nodefactor.location\_org.Sarnen 0 0.528 0.597655  
## nodefactor.location\_org.Schaffhausen 0 -0.098 0.921958  
## nodefactor.location\_org.Schöftland 0 -0.041 0.967428  
## nodefactor.location\_org.Solothurn 0 1.679 0.093062  
## nodefactor.location\_org.St. Gallen 0 1.683 0.092351  
## nodefactor.location\_org.Steinhausen 0 -0.016 0.987161  
## nodefactor.location\_org.Sursee 0 -0.036 0.971089  
## nodefactor.location\_org.Unterägeri 0 -0.036 0.971089  
## nodefactor.location\_org.Wiedlisbach 0 0.397 0.691022  
## nodefactor.location\_org.Wilen (Sarnen) 0 -0.189 0.849771  
## nodefactor.location\_org.Winterthur 0 -0.189 0.849771  
## nodefactor.location\_org.Zofingen 0 -0.098 0.921958  
## nodefactor.location\_org.Zufikon 0 -0.189 0.849771  
## nodefactor.location\_org.Zug 0 -0.130 0.896763  
## nodefactor.location\_org.Zürich 0 0.091 0.927716  
## nodefactor.location\_org.Zürich-Flughafen 0 0.002 0.998264  
## nodefactor.industry\_org.Bildung 0 -1.465 0.142931  
## nodefactor.industry\_org.Energie 0 -3.016 0.002559  
## nodefactor.industry\_org.Gesundheit 0 -3.359 0.000783  
## nodefactor.industry\_org.Kultur 0 -1.903 0.056996  
## nodefactor.industry\_org.Landwirtschaft 0 -2.202 0.027682  
## nodefactor.industry\_org.Sicherheit 0 -1.864 0.062316  
## nodefactor.industry\_org.Soziale Sicherheit 0 -1.867 0.061844  
## nodefactor.industry\_org.Sport 0 -2.035 0.041875  
## nodefactor.industry\_org.Staatspolitik/Staatswirtschaft 0 -3.921 < 1e-04  
## nodefactor.industry\_org.Umwelt 0 -2.338 0.019397  
## nodefactor.industry\_org.Verkehr 0 -3.816 0.000136  
## nodefactor.industry\_org.Wirtschaft 0 -3.694 0.000221  
##   
## 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 SO   
## 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 SO \*   
## nodefactor.location\_org.Oftringen   
## nodefactor.location\_org.Olten   
## nodefactor.location\_org.Posieux   
## nodefactor.location\_org.Rotkreuz   
## nodefactor.location\_org.Sachseln   
## 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/Treuhand   
## 10 9   
## Immobilien/Hauseigentümer Parteien   
## 8 7   
## Bildung/Wissenschaft Zünfte, Verbindungen und Serviceclubs   
## 6 6   
## KMU/Gewerbe/Arbeitgeber Religion   
## 5 5   
## Umwelt & Natur Öffentlicher Verkehr   
## 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 agriss   
## 1 1   
## Allianz Sicherheit Schweiz Alpen-Initiative   
## 1 1   
## alps - Alpines Museum der Schweiz amitola, gemeinnützige GmbH für Kinder   
## 1 1   
## Aqua Viva Arbeitskreis Sicherheit und Wehrtechnik   
## 1 1   
## Arcosana AG Association Ouestrail   
## 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.")



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.")

