Amazon Network Analysis

Assignment im Rahmen der Vorlesung 'Social Network Analyis'

Ferdinand Bubeck

2021-11-17

Inhaltsverzeichnis

1 Einleitung

Im Rahmen der Vorlesung “Social Network Analyses” von Philipp Mendoza an der DHBW Stuttgart soll eine Netzwerkanalyse auf Basis eines gewählten Datensatzes abgegeben werden. Der Autor dieser Arbeit hat sich für einen Amazon Produktdatensatz entschieden, welcher im Laufe der Arbeit vorgestellt wird.

1.1 Zielsetzung

Zielsetzung ist es, auf Basis der Daten eine Forschungsfrage zu überlegen und diese netzwerkanalytisch zu beantworten. Dabei sollen erlernte Konzepte aus der Vorlesung einfließen und mindestens eine Netzwerk Visualisierung enthalten sein.

1.2 Vorgehensweise

Als Vorgehensweise wird in diesem Projekt das für das Feld Data Science etablierte Standard-Vorgehen CRISP-DM gewählt (Cross Industry Standard Process for Data Mining). In mehreren Phasen werden so von dem richtigen Verständnis der Daten, dem Data Wrangling und Data Preprocessing bis hin zum Modelfitting und der Evaluation alle entscheidenen Schritte strukturiert durchlaufen, um ein optimales Ergebnis aus den Daten zu generieren. In der Abbildung 1 ist das Vorgehensmodell abgebildet. Da es sich in diesem Projekt um ein PoC handelt, wird die letzte Phase ‘Deployment’ ausgelassen.

Figure 1: CRISP-DM (Source: https://statistik-dresden.de/archives/1128))

2 Hauptteil

2.1 Business Understanding

Netzwerkanalyse beschäftigt sich mit der Analyse von verschiedenen Arten von Netzwerken. Dabei liegt der Fokus auf den Beziehungen und vorallem den Beziehungsstrukturen zwischen mehreren Knoten. Die Merkmale der Knoten spielen ebenfalls eine Rolle, das Hauptaugenmerk liegt allerdings auf den Strukturen und Dynamiken.

2.1.1 Datensatz

Der dieser Arbeit zugrunde liegende Datensatz stammt aus der Datensatz-Bibliothek der Stanford University und bildet ein Netzwerk von einer Vielzahl an Amazon Produkten. Es handelt sich bei dem Datensatz um *ready-made Daten*, da der Datensatz als Nebenprodukt einer API entsteht.
"If a product i is frequently co-purchased with product j, the graph contains a directed edge from i to j"
Die Beschreibung des Datensatzes von der Website lässt die Vermutung zu, dass es Produkte geben muss, welche im Netzwerk zentral sind und häufig in Verbindung mit anderen Produkten gekauft werden.

2.1.2 Fragestellung

Aus diesem Grund sollen in dieser Arbeit die folgenden Frage beantwortet werden:

- Welche Produkte werden in Verbindung mit den meisten anderen Produkten gekauft?
- Welche Produkte werden hauptsächlich eigenständig gekauft?

2.2 Data Understanding

2.2.1 Laden der Libraries

Um mit der Datenanalyse und -aufbereitung zu beginnen, müssen zuerst Libraries geladen, welche relevant sind. *Tidyverse* ist eine Library, welche eine Vielzahl an Tools in einer eigenen Designphilosophy mit eigener Grammatik bereitstellt. So gehört zum Beispiel das Paket *ggplot2* für die Datenvisualisierung zur Libary dazu. *Tidygraph*, *ggraph* und *igraph* sind für die Visualisierung notwendig. Das Paket *tinytex* beinhaltet die Sprache LaTeX für die Kompilierung der Skript-Befehle. Die folgende Funktion überprüft, ob alle Pakete in der Liste bereits installiert sind und installiert gegenfalls alle nicht installierten Libraries. Danach werden alle Libraries geladen.

```
packages = c("tidyverse", "tidygraph",
             "igraph", "ggraph", "tinytex")

package.check <- lapply(
  packages,
```

```
  FUN = function(x) {
    if (!require(x, character.only = TRUE)) {
      install.packages(x, dependencies = TRUE)
      library(x, character.only = TRUE)
    }
  }
)
```

2.2.2 Importieren der Daten

Die Daten stammen aus der Datensatz-Bibliothek der Stanford University und können als .txt unter folgendem Link heruntergeladen werden. (Link: https://snap.stanford.edu/data/amazon0302.html)

Zum Einlesen der Daten kommt im Folgenden die Funktion *read.table* zum Einsatz.

```
amazon <- read.table("Data/Amazon0302.txt")
```

2.2.3 Datenexploration

Die Daten basieren auf dem Grundsatz "Kunden, die Artikel A gekauft haben, haben auch Artikel B gekauft". Wenn ein Produkt i häufig zusammen mit Produkt j gekauft wird , enthält der Graph eine gerichtete Kante von i nach j .

Um einen ersten Einblick in die Daten zu erhalten, wird mit der Funktion *head* die ersten Zeilen des Datensatzes ausgegeben. Zusätzlich dazu ist es von entscheidender Rolle, die Qualität der Daten zu bewerten. Aus diesem Grund werden alle fehlenden Werte, sogenannte NAs gezählt und ausgegeben.

```
head(amazon)
```

```
##   V1 V2
## 1  0  1
## 2  0  2
## 3  0  3
## 4  0  4
## 5  0  5
## 6  1  0
```

```
# Count NAs
which(is.na(amazon))
```

```
## integer(0)
```

Der Dataframe besteht aus 3 Spalten: einer ID Spalte, und zwei Kantenspalten. Des Weiteren weisen die Daten keine Lücken und fehlenden Werte auf, sodass der komplette Datensatz für

das weitere Vorgehen genutzt werden kann.

2.3 Data Preparation

Auf Basis der vorangegangen Schritte müssen nun weitere Anpassungen der Daten erfolgen, um damit arbeiten zu können. Zum Einen werden die Kantenspalten von ihren ursprünglichen Namen in sprechendere Bezeichnungen umbenannt. Im gleichen Schritt werden alle Werte um 1 erhöht, sodass keine Nullen mehr existieren.

```
dat <- amazon %>%
  rename(
    from = V1,
    to = V2
  ) %>%
  mutate(
    from = from+1,
    to = to+1
  )
```

2.4 Modeling

Nach der Datenbearbeitung kann nun das Netz gefittet werden. Hierzu wird die Funktion *as tbl graph* angewendet, um ein Netz zu erstellen.

```
net <- as_tbl_graph(dat)
net
```

```
## # A tbl_graph: 262111 nodes and 1234877 edges
## #
## # A directed simple graph with 1 component
## #
## # Node Data: 262,111 x 1 (active)
##   name
##   <chr>
## 1 1
## 2 2
## 3 3
## 4 4
## 5 5
## 6 6
## # ... with 262,105 more rows
## #
## # Edge Data: 1,234,877 x 2
##    from    to
```

```
##   <int> <int>
## 1     1     2
## 2     1     3
## 3     1     4
## # ... with 1,234,874 more rows
```

Die beiden Spalten aus dem Ursprungsdatensatz wurden in ein Netz, bestehend aus 262111 Knoten und 1234877 Kanten, konvertiert. Es handelt sich, wie aus der Zusammenfassung des Netzes zu entnehmen ist, um einen gericheteten Graphen. Die Knotennamen sind in diesem Fall die Ziffern der Kantendaten. Leider liegt dem Autor dieser Arbeit keine Zuordnungsliste von Knotenziffern zu realen Amazonprodukten vor. Aus diesem Grund wird im Folgenden mit den Ziffern der Knoten weitergearbeitet.

```
# Calculate Degree of Vertices
degree <- degree(net)

# Adjacency Matrix
adjacencyMatrix <- net[]
```

Aus dem Netz kann nun der Degree abgeleitet und abgespeichert werden. Der Degree oder Grad eines Knoten ist die Anzahl von Kanten, die an ihn angrenzen. Für die Analyse ist die Verteilung der Grade der Knoten interessant. Gibt es eine überwiegende Mehrheit an Knoten, welche die gleiche Anzahl an Kanten besitzen? Gibt es Ausreißer mit vielen Kanten? Ähnelt die Verteilung einer Normalverteilung, ist die links oder rechts verschoben?
Um diese Fragen zu beantworten, wird im nächsten Schritt ein Histogramm erzeugt, welches die Knotengrade des Netzwerkes visualisiert.

2.5 Data Visualization

Um die Degrees für die Visualisierung nutzen zu können, müssen diese zuvor in ein Dataframe umgewandelt werden. Dies geschieht mit der Funktion *as.data.frame*. Anschließend wird die Library *ggplot2* für das Histogramm angewendet.

```
degree_df <- as.data.frame(degree)


hist_of_degrees <- ggplot(data = degree_df, aes(x=degree))+
  geom_bar(fill = "#e2001a", colour = "#e2001a", alpha=.5)+
  scale_y_continuous(trans='log10')+
  xlim(0,120)+
  labs(title = "Histogram of Node-Degrees",
       subtitle = "Amazon Network Analysis",
       y = "Frequency (log10 scale)",
       x = "Degree of Vertices (xlim = 120)")+
```

```
  theme_classic()

hist_of_degrees
```

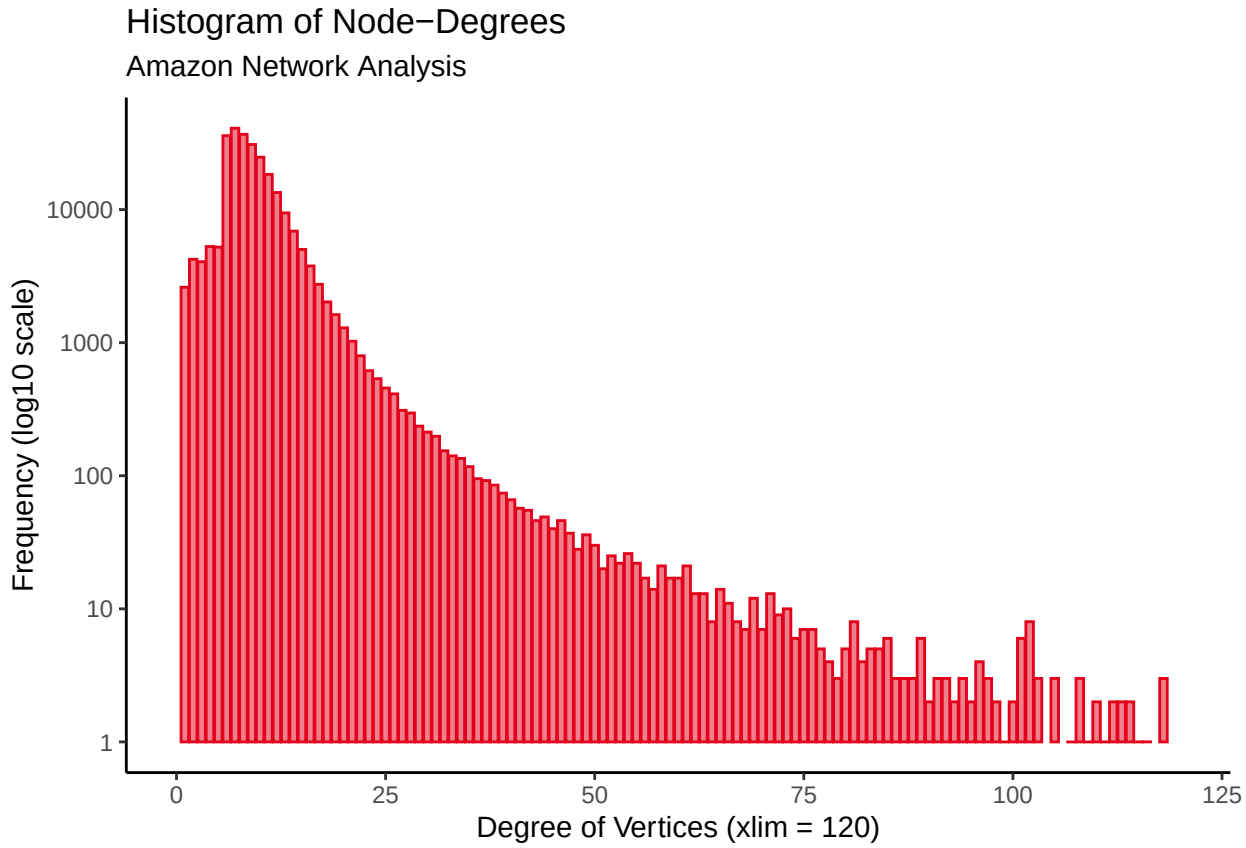


Figure 2: Knotengrad Histogramm

Das Histogramm der Knotengrade zeigt eine Mehrheit der Grade im Bereich 5-20. Dies bedeutet, dass eine Mehrheit der Knoten im Datensatz eine durchschnittliche Anzahl an Kanten von 5-20 aufweist. Weiterhin ist zu erkennen, dass einige Knoten 100 und mehr Kanten besitzen. Die großen Ausreißer wurden in diesem Plot weggelassen, doch selbst in dieser Darstellungsweise zeigt sich ein abflachender Bereich Richtung x –> unendlich. Um eine übersichtlichere Darstellung der Observationen um den Nullbereich der y-Achse zu gewährleisten, wurde die y-Achse nach dem dekadischen Logarithmus skaliert.

2.6 Experimental Data

Um die Laufzeit und die Übersichtlichkeit der Visualisierungen zu verbessern hat sich der Autor dieser Arbeit dazu entschieden, die Fragestellung anhand einer Teilmenge des gesamten Datensatzes durchzuführen. Es wird demnach ein Subset von *250 Zeilen* aus dem Datensatz verwendet.

```
# Subsetting Data
dat_exp <- dat[1:250,]

net_exp <- as_tbl_graph(dat_exp)

net_exp <- net_exp %>%
  activate(nodes) %>%
  mutate(
    degree = centrality_degree()
  )
```

2.6.1 Zentralitätsmaße

Um die Fragestellung dieser Arbeit zu beantworten, müssen weitere Erkenntnisse über die Netzwerk-Struktur analysiert werden. Hierfür sind Zentralitätsmaße eine gute Anlaufstelle, um "wichtigere Knoten" im Sinne des Maßes zu identifizieren. Der Autor hat sich für die Beantwortung der Fragestellung:
Welche Produkte werden in Verbindung mit den meisten anderen Produkten gekauft?
für die Degree-Centrality entschieden. Der Grad eines Knoten ist die Menge aller inzidenten Kanten. Somit wird ein Produkt mit einem hohen Grad in Verbindung mit vielen weiteren Produkten gekauft.

```
# Degree Centrality
centr_degree <- degree(net_exp)

df_centr_degree <- as.data.frame(centr_degree)

top_5_degree <- df_centr_degree %>%
  top_n(5) %>%  # highest values
  arrange(-centr_degree)

top_5_degree
```

```
##    centr_degree
## 9            16
## 8            12
## 12           11
## 14           11
## 19           11
## 24           11
## 25           11
## 31           11
```

Zu sehen sind die top 5 Gradwerte der Teilmenge des Datensatzes. Dies bedeutet, dass Knoten Nummer 9 der Knoten ist, welcher die meisten Kanten zu anderen Knoten hat. In dieser Teilmenge des Datensatzes ist Knoten Nummer 9 das Produkt, welches am meisten in

Verbindung mit anderen Produkten gekauft wird.

```
# Betweenness Centrality
centr_betweenness <- betweenness(
  net_exp,
  directed = TRUE,
  weights = NULL,
  nobigint = TRUE,
  normalized = FALSE
)

df_centr_betweenness <- as.data.frame(centr_betweenness)

top_5_betweenness <- df_centr_betweenness %>%
  top_n(5) %>%  # highest values
  arrange(-centr_betweenness)

top_5_betweenness
```

```
##    centr_betweenness
## 9          1154.5000
## 21          585.7333
## 23          470.0000
## 31          428.7500
## 12          343.0667
```

Auch die Betweenness-Zentralität gibt als zentralsten Knoten den Knoten Nummer 9 zurück. Betweenness misst das Ausmaß, in dem ein Knoten auf kürzesten Pfaden zwischen anderen Knoten im Graphen positioniert ist. Somit wird in dieser Teilmenge des Datensatzes der Knoten 9 und damit das Produkt Nummer 9 als am meisten mit anderen Produkten gekaufte Produkt verstanden.

Um die Fragestellung *Welche Produkte werden hauptsächlich eigenständig gekauft?* zu beantworten, werden im Folgenden die Top 5 der Knoten ausgegeben, welche die wenigsten Kanten zu Nachbarknoten haben.

```
lowest_5_degree <- df_centr_degree %>%
  top_n(-5)  # lowest values

lowest_5_degree
```

```
##     centr_degree
## 64             1
## 65             1
## 66             1
## 67             1
```

```
## 68              1
## 291             1
## 138             1
## 139             1
## 140             1
## 261             1
## 262             1
## 263             1
## 264             1
## 265             1
## 141             1
## 142             1
## 69              1
## 70              1
## 71              1
## 72              1
## 73              1
## 75              1
## 165             1
## 112             1
## 143             1
## 266             1
## 267             1
## 268             1
## 269             1
## 77              1
## 79              1
## 80              1
## 81              1
## 82              1
## 83              1
## 144             1
## 145             1
## 146             1
## 147             1
## 148             1
## 84              1
## 85              1
## 86              1
## 149             1
## 150             1
## 151             1
## 94              1
## 95              1
## 96              1
```

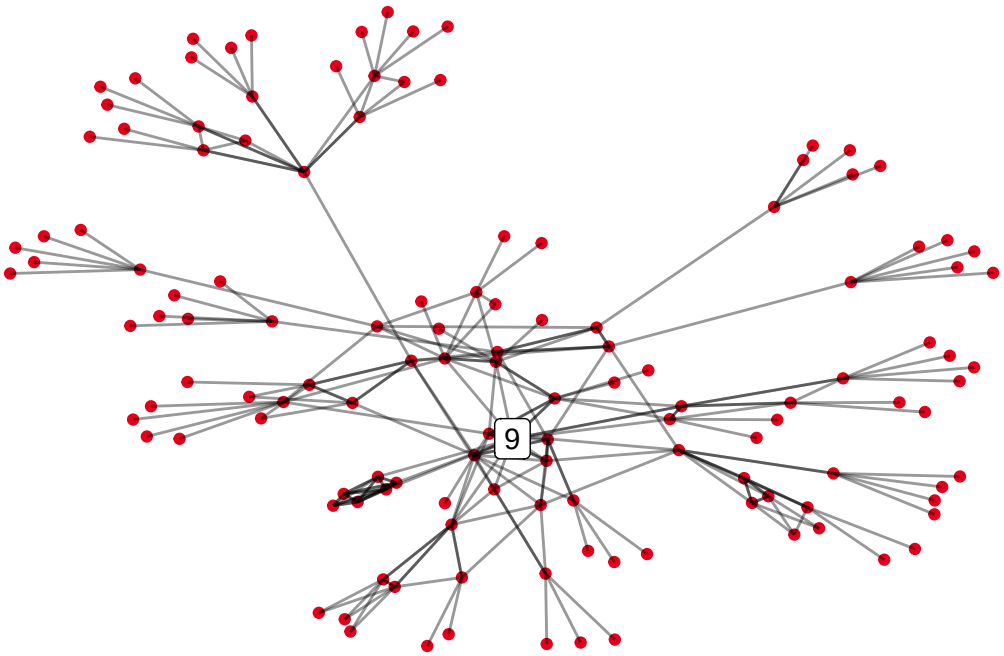
```
## 152             1
## 53              1
## 54              1
## 55              1
## 56              1
## 153             1
## 154             1
## 155             1
## 156             1
## 98              1
## 270             1
## 271             1
## 272             1
## 273             1
## 274             1
## 87              1
## 88              1
## 58              1
## 59              1
## 90              1
## 91              1
```

Wie zu erkennen ist, gibt es eine Vielzahl an Produkten, welche lediglich eine Kante besitzen und damit häufig in Verbindung mit einem weiteren Produkt gekauft werden. Es gibt allerdings keine Knoten, welche unabhängig sind und keine einzige Kante besitzen. Somit kann für diese Fragestellung kein eindeutiges Produkt identifiziert werden, welches eigenständig gekauft wird.

2.6.2 Visualisierung

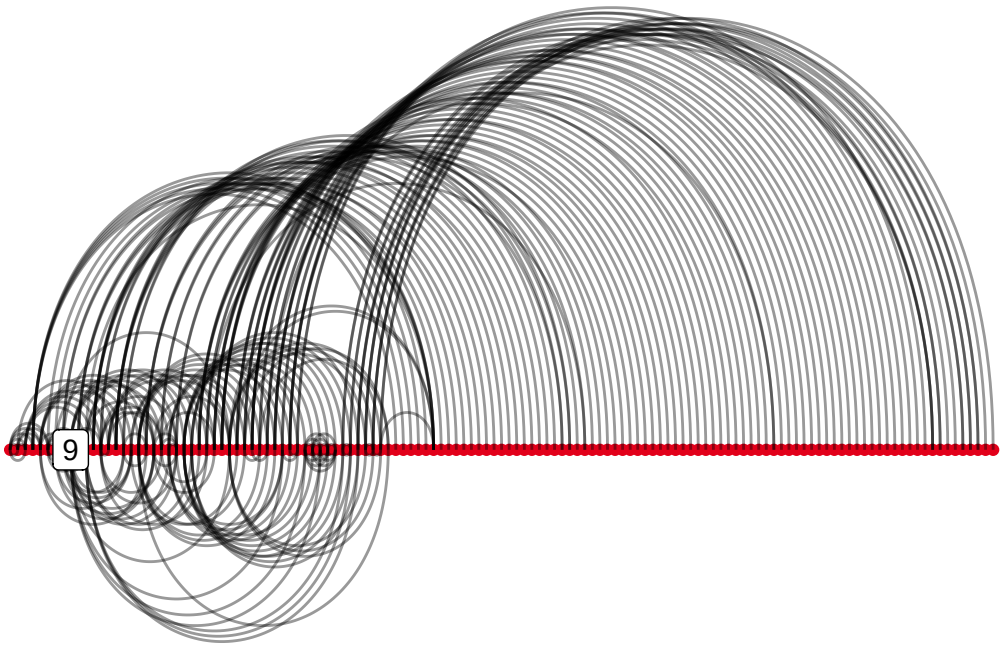
Um die Beziehungen des Knoten Nummer 9 besser verstehen zu können, werden im Folgenden drei Graph-Visualisierungen erstellt. Dabei ist Knoten 9 in jedem Graph mit seinem Namenslabel gekennzeichnet.

```
# Data Viz for Subset
# network diagramm
ggraph(net_exp, layout = 'fr', maxiter = 100) +
  geom_node_point(colour="#e2001a") +
  geom_edge_link(alpha = .4) +
  geom_node_label(aes(label=ifelse(name == "9", name, NA))) +
  theme_graph()
```



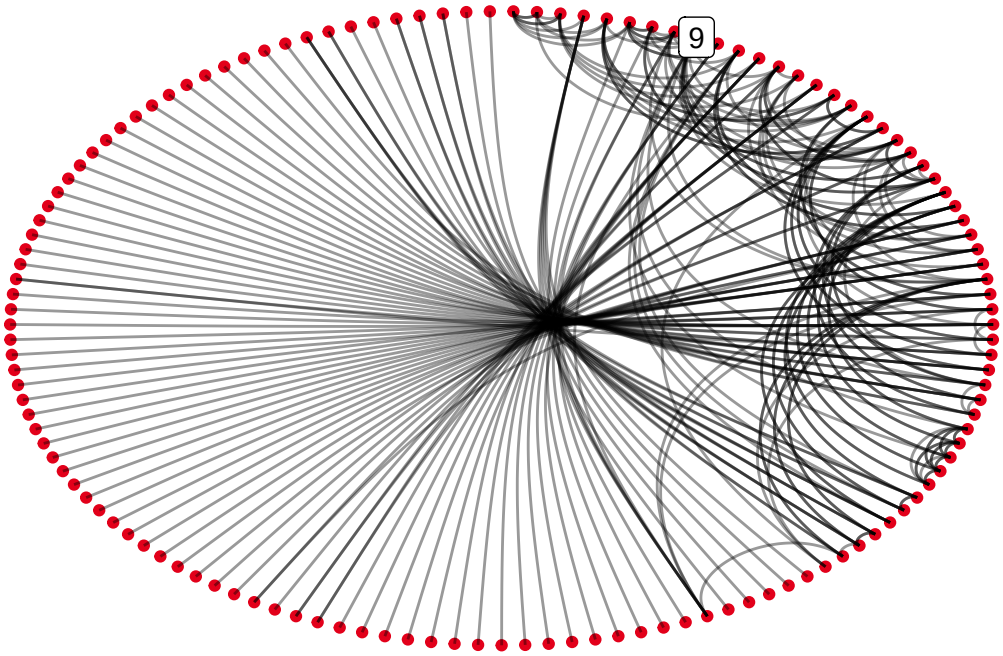
Anhand dieses Graphen kann man die Lage des Knoten 9 gut erkennen. Bei genauerem Betrachten kann man seine 16 Kanten zu weiteren Knoten erkennen. Die Visualisierung zeigt an den Randbereichen weiterhin alle Knoten mit nur einer einzigen Kante.

```
ggraph(net_exp, layout = 'linear') +
  geom_node_point(colour="#e2001a") +
  geom_edge_arc(alpha = .4) +
  geom_node_label(aes(label=ifelse(name == "9", name, NA))) +
  theme_graph()
```



Anhand dieser Visualisierung kann man sehr gut erkennen, dass Knoten 9 die meisten Verbindungen zu anderen Knoten besitzt. Alle Knoten sind bei diesem Layout sortiert auf einer Geraden angeordnet und die Kanten werden als Bögen dargestellt. So wird auch die Verbindung zu anderen Knotennummern sichtbar.

```
# coord diagramm
ggraph(net_exp, layout = 'linear', circular = TRUE) +
  geom_node_point(colour="#e2001a") +
  geom_edge_arc(alpha = .4) +
  geom_node_label(aes(label=ifelse(name == "9", name, NA))) +
  theme_graph()
```



3 Fazit

3.1 Evaluation der Ergebnisse

tbd

3.2 kritische Reflexion

tbd