

# Einfluss der Wettbewerbsstruktur auf Gehaltsniveaus im Data Science-Bereich:

Eine Social Network Analyse

## Contents

|          |   |           |
|----------|---|-----------|
| <b>1</b> | <b>Einleitung</b>   | <b>2</b>  |
| 1.1      | Requirements . . . . .  | 2         |
| 1.2      | Motivation und Zielsetzung . . . . .  | 2         |
| 1.3      | Forschungsfrage . . . . .   | 3         |
| 1.4      | Datengrundlage . . . . .  | 3         |
| 1.4.1    | CSV einlesen . . . . .  | 3         |
| 1.4.2    | Erste Ansicht der Daten . . . . .   | 4         |
| <b>2</b> | <b>Analysestrategie</b>   | <b>8</b>  |
| <b>3</b> | <b>Analyse</b>  | <b>9</b>  |
| 3.1      | Datenbereinigung . . . . .  | 9         |
| 3.1.1    | Bereinigung für die geografische Analyse . . . . .                              | 9         |
| 3.1.2    | Überprüfung auf weitere fehlende Werte . . . . .                                | 10        |
| 3.1.3    | Entfernen irrelevanter Spalten . . . . .  | 12        |
| 3.1.4    | Bereinigung irrelevanter Jobtitel . . . . .                                     | 12        |
| 3.2      | Wettbewerbsnetzwerk . . . . .   | 13        |
| 3.2.1    | Erstellung eines Wettbewerbsnetzwerkes . . . . .                                | 13        |
| 3.2.2    | Betrachtung fokussiert auf direkte Wettbewerber . . . . .                       | 16        |
| 3.3      | Zentralitätsanalyse innerhalb des Netzwerkes . . . . .                          | 19        |
| 3.3.1    | Betweenness-Zentralität . . . . .   | 19        |
| 3.3.2    | Degree-Zentralität . . . . .  | 22        |
| 3.3.3    | Eigenvector-Zentralität . . . . .   | 24        |
| 3.4      | Cluster-Analyse . . . . .   | 25        |
| 3.4.1    | Clustering Identifikation . . . . .   | 25        |
| 3.4.2    | Gehaltsunterschiede zwischen den Clustern . . . . .                             | 30        |
| 3.4.3    | Branchenstruktur der Cluster . . . . .  | 31        |
| 3.4.4    | Errechnung der durchschnittlichen Gehälter pro Branche . . . . .                | 32        |
| <b>4</b> | <b>Abschließende Betrachtung mittels interaktiver Visualisierung</b>            | <b>33</b> |
| <b>5</b> | <b>Conclusion</b>   | <b>35</b> |
| <b>6</b> | <b>Anhang</b>   | <b>36</b> |
| 6.1      | Geografische Betrachtung . . . . .  | 36        |
| 6.1.1    | Erstellung eines Geografischen Netzwerkes . . . . .                             | 36        |
| 6.1.2    | Vergleich der Gehälter zwischen den Hotspot- und den anderen Regionen . . . . . | 38        |
| <b>7</b> | <b>Literaturverzeichnis</b>   | <b>41</b> |

# 1 Einleitung

## 1.1 Requirements

Zunächst müssen die benötigten Bibliotheken geladen werden:

```
library(tidyverse)
library(igraph)
library(visNetwork)
library(dplyr)
library(tidyr)
library(knitr)
library(kableExtra)
library(webshot)
library(ggplot2)
library(RColorBrewer)
library(skimr)
library(htmlwidgets)
```

## 1.2 Motivation und Zielsetzung

In ihrem Artikel “Data Scientist: The Sexiest Job of the 21st Century” betonen Davenport und Patil, dass Data Scientists durch ihre Fähigkeiten in Informatik, Statistik und ihr Fachwissen allgemein einen erheblichen Mehrwert für Unternehmen schaffen.<sup>1</sup> Die Fähigkeit, aus komplexen, unstrukturierten Daten wertvolle Erkenntnisse zu gewinnen, macht Data Scientists in vielen Branchen zu einer unverzichtbaren Ressource.<sup>2</sup> Die Nutzung ihrer Kompetenzen verschafft Unternehmen einen Wettbewerbsvorteil, da sie datengetriebene Entscheidungen, Produktinnovationen und Effizienzsteigerungen ermöglicht.<sup>3</sup>

Darüber ob Data Scientists immer noch the “Sexiest Job” des 21. Jahrhunderts sind, lässt sich streiten. Fakt ist jedoch, dass die Nachfrage nach Data Scientists in den letzten Jahren stark gestiegen ist und voraussichtlich immer weiter steigen wird. Dieser Trend ist auch in den Google-Suchanfragen zu den Begriffen erkenntlich:<sup>4</sup>

```
ggplot(data, aes(x = Monat)) +
  geom_line(aes(y = `data science`, color = "data science")) +
  geom_line(aes(y = `data scientist`, color = "data scientist")) +
  labs(title = "Google Suchtrend für 'data science' und 'data scientist'",
        x = "Datum",
        y = "Interesse",
        color = "Suchbegriff") +
  theme_minimal()
```

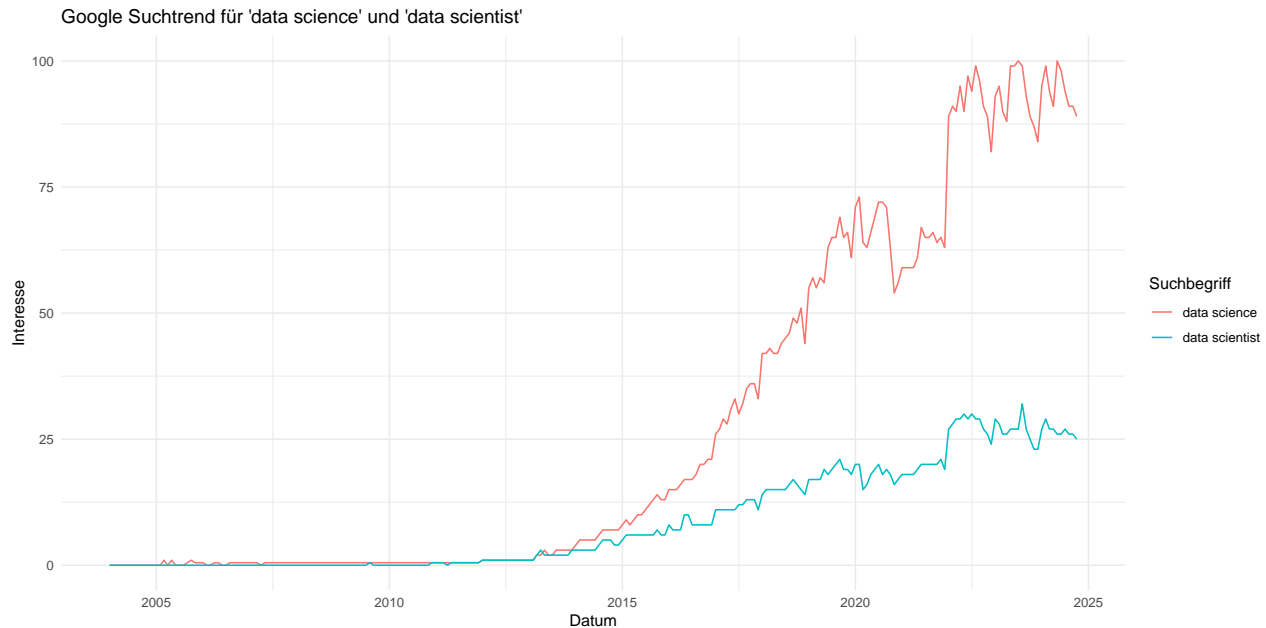
---

<sup>1</sup>Davenport, Patil 2012

<sup>2</sup>Davenport, Patil 2012

<sup>3</sup>Davenport, Patil 2012

<sup>4</sup>Google Trends, abgerufen am 30.10.2024



Das wachsende Interesse an Data Science stellt eine große Chance für Arbeitnehmer dar. Ziel dieser Arbeit ist es einen Überblick über den Data-Science-Jobmarkt zu geben, um Arbeitnehmern bei der Jobsuche zu helfen und andererseits einen Überblick über die Gehälter und die Rolle von Geographie und Wettbewerb bei Jobangeboten und Gehältern zu geben.

### 1.3 Forschungsfrage

Im Rahmen der vorliegenden Arbeit wird die folgende Forschungsfrage bearbeitet:

Es soll untersucht werden, inwiefern die Position und Zentralität von Unternehmen im Wettbewerbsnetzwerk die Gehaltsstrukturen von Data-Science-Positionen beeinflusst. Zudem soll erörtert werden, welche Rolle die Intensität des Wettbewerbs bei der Erklärung von Gehaltsunterschieden spielt.

Dabei sollen folgende Aspekte untersucht werden:

Bieten Unternehmen, die als zentrale Knoten in einem dichten Wettbewerbsnetzwerk agieren, tendenziell höhere Löhne an?

Inwieweit beeinflusst die Clusterbildung im Netzwerk die Gehaltsstrukturen und gibt es spezifische Cluster, in denen höhere Gehälter häufiger angeboten werden?

Welche Zentralitätsmaße korrelieren signifikant mit überdurchschnittlichen Gehältern und können Unternehmen mit bestimmten Netzwerkpositionen spezifische Wettbewerbsvorteile erzielen?

### 1.4 Datengrundlage

Nachdem die Daten in Python extern als Vorbereitung aufbereitet wurden, kann nun die Datengrundlage für diese Arbeit in R eingelesen werden. Dabei wurde sich an <https://www.kaggle.com/code/fahadrehman07/data-science-job-salary-prediction-glassdoor> orientiert.

#### 1.4.1 CSV einlesen

```
data <- read_csv("data/Glassdoor_DataScience_Salary_modified.csv", show_col_types = FALSE)
```

Die vorliegende Arbeit basiert auf einem Datensatz von Kaggle, der Informationen über Data Science Jobs in verschiedenen Unternehmen für den US-amerikanischen Markt enthält. Der Datensatz umfasst 742 Zeilen und 28 Spalten, was auf eine Anzahl von 742 verschiedenen Jobangeboten hindeutet. Diese Anzahl ist kann

für die Zwecke dieser Arbeit als ausreichend zu betrachten, auch wenn eine höhere Zahl an Beobachtungen möglicherweise zu präziseren Schlussfolgerungen geführt hätte.

Der Datensatz beruht auf Daten, die von Glassdoor extrahiert wurden, eine für Stellenanzeigen und Unternehmensbewertung bekannte Website, und bietet detaillierte Informationen über Data-Science-Jobs sowie deren Gehälter. Der Datensatz beinhaltet wesentliche Informationen, darunter Jobtitel, geschätzte Gehälter, Stellenbeschreibungen, Unternehmensbewertungen sowie relevante Unternehmensdaten wie Standort, Größe und Branche. Eine detaillierte Beschreibung dieser Daten erfolgt im späteren Verlauf. Der Datensatz eignet sich in besonderem Maße für den Zweck dieser Arbeit, aber auch für Analysen des Arbeitsmarktes, beispielsweise zur Untersuchung von Gehaltstrends oder zur Identifizierung der am besten bewerteten Unternehmen.

Der Datensatz umfasst konkret die folgenden Spalten:

#### 1.4.2 Erste Ansicht der Daten

```
head(data, 5)

## # A tibble: 5 x 28
##   `Job Title` `Salary Estimate` `Job Description` Rating `Company Name` Location
##   <chr>          <dbl> <chr>          <dbl> <chr>      <chr>
## 1 Data Scien~          72  "Data Scientist\~    3.8 Tecolote Rese~  Albuque~
## 2 Healthcare~        87.5  "What You Will D~    3.4 University of~ Linthic~
## 3 Data Scien~          85  "KnowBe4, Inc. i~    4.8 KnowBe4      Clearwa~
## 4 Data Scien~        76.5  "*Organization a~    3.8 PNNL          Richlan~
## 5 Data Scien~       114.  "Data Scientist\~    2.9 Affinity Solu~ New Yor~
## # i 22 more variables: Headquarters <chr>, Size <chr>, Founded <dbl>,
## #   `Type of ownership` <chr>, Industry <chr>, Sector <chr>, Revenue <chr>,
## #   Competitors <chr>, Min_Salary <dbl>, Max_Salary <dbl>, State <chr>,
## #   `Same State` <dbl>, Age <dbl>, Python_yn <dbl>, `R Studio` <dbl>,
## #   Spark <dbl>, AWS_yn <dbl>, Excel_yn <dbl>, Job_simp <chr>, job_state <chr>,
## #   desc_len <dbl>, Num_comp <dbl>
## # Erstellen einer schönen Zusammenfassung des Datensatzes
skim(data)
```

Table 1: Data summary

|                        |      |
|------------------------|------|
| Name                   | data |
| Number of rows         | 742  |
| Number of columns      | 28   |
| Column type frequency: |      |
| character              | 14   |
| numeric                | 14   |
| Group variables        | None |

#### Variable type: character

| skim_variable   | n_missing | complete_rate | min | max   | empty | n_unique | whitespace |
|-----------------|-----------|---------------|-----|-------|-------|----------|------------|
| Job Title       | 0         | 1             | 9   | 98    | 0     | 264      | 0          |
| Job Description | 0         | 1             | 407 | 10051 | 0     | 463      | 0          |
| Company Name    | 0         | 1             | 2   | 51    | 0     | 343      | 0          |
| Location        | 0         | 1             | 8   | 33    | 0     | 200      | 0          |

| skim_variable     | n_missing | complete_rate | min | max | empty | n_unique | whitespace |
|-------------------|-----------|---------------|-----|-----|-------|----------|------------|
| Headquarters      | 0         | 1             | 2   | 26  | 0     | 198      | 0          |
| Size              | 0         | 1             | 2   | 23  | 0     | 9        | 0          |
| Type of ownership | 0         | 1             | 2   | 30  | 0     | 11       | 0          |
| Industry          | 0         | 1             | 2   | 40  | 0     | 60       | 0          |
| Sector            | 0         | 1             | 2   | 34  | 0     | 25       | 0          |
| Revenue           | 0         | 1             | 2   | 32  | 0     | 14       | 0          |
| Competitors       | 0         | 1             | 2   | 92  | 0     | 128      | 0          |
| State             | 0         | 1             | 2   | 11  | 0     | 38       | 0          |
| Job_simp          | 0         | 1             | 2   | 14  | 0     | 7        | 0          |
| job_state         | 0         | 1             | 2   | 2   | 0     | 37       | 0          |

### Variable type: numeric

| skim_variable | n_missing | complete_rate | mean    | sd      | p0    | p25    | p50    | p75    | p100  | hist |
|---------------|-----------|---------------|---------|---------|-------|--------|--------|--------|-------|------|
| Salary        | 0         | 1             | 100.63  | 38.86   | 13.5  | 73.5   | 97.5   | 122.5  | 254   |      |
| Estimate      |           |               |         |         |       |        |        |        |       |      |
| Rating        | 0         | 1             | 3.62    | 0.80    | -1.0  | 3.3    | 3.7    | 4.0    | 5     |      |
| Founded       | 0         | 1             | 1837.15 | 497.18  | -1.0  | 1939.0 | 1988.0 | 2007.0 | 2019  |      |
| Min_Salary    | 0         | 1             | 74.72   | 30.98   | 15.0  | 52.0   | 69.5   | 91.0   | 202   |      |
| Max_Salary    | 0         | 1             | 127.18  | 46.91   | 16.0  | 96.0   | 124.0  | 155.0  | 306   |      |
| Same State    | 0         | 1             | 0.56    | 0.50    | 0.0   | 0.0    | 1.0    | 1.0    | 1     |      |
| Age           | 0         | 1             | 49.39   | 53.96   | -1.0  | 14.0   | 27.0   | 62.0   | 279   |      |
| Python_yn     | 0         | 1             | 0.53    | 0.50    | 0.0   | 0.0    | 1.0    | 1.0    | 1     |      |
| R Studio      | 0         | 1             | 0.00    | 0.05    | 0.0   | 0.0    | 0.0    | 0.0    | 1     |      |
| Spark         | 0         | 1             | 0.23    | 0.42    | 0.0   | 0.0    | 0.0    | 0.0    | 1     |      |
| AWS_yn        | 0         | 1             | 0.24    | 0.43    | 0.0   | 0.0    | 0.0    | 0.0    | 1     |      |
| Excel_yn      | 0         | 1             | 0.52    | 0.50    | 0.0   | 0.0    | 1.0    | 1.0    | 1     |      |
| desc_len      | 0         | 1             | 3869.55 | 1521.50 | 407.0 | 2801.0 | 3731.0 | 4740.0 | 10051 |      |
| Num_comp      | 0         | 1             | 1.05    | 1.38    | 0.0   | 0.0    | 0.0    | 3.0    | 4     |      |

Im Folgenden wird eine Übersicht der wesentlichen Spalten präsentiert:

- **Job Title:** Die Berufsbezeichnung, sie gibt Aufschluss über die Tätigkeit.
- **Salary Estimate:** Die geschätzte Gehalt, in tausend Dollar pro Jahr. Es basiert auf dem Durchschnitt von dem minimalen und maximalen Gehalt.
- **Job Description, Job\_simp:** Die Beschreibung der Stelle, die Aufgaben und Anforderungen enthält. Auch die vereinfachte Version der Berufsbezeichnung.
- **Rating:** Die Bewertung des Unternehmens, sie weist eine Spannbreite von 1 bis 5 auf, wobei die Bewertung "1" bei jeder Spalte für fehlende Bewertungen steht.
- **Company Name, Location, Headquarters, Size, Founded:** Unternehmensbezogene Daten wie Name, Standort, Sitz, Größe und Gründungsjahr des Unternehmens.
- **Type of ownership, Industry, Sector, Revenue:** Weitere Unternehmensmerkmale, diese umfassen die Eigentumsart, die Branche, den Sektor sowie die Einnahmen.
- **Competitors:** Die Wettbewerber des Unternehmens, die im Zusammenhang dieser Arbeit von besonderer Bedeutung sind.
- **Skills (Python\_yn, R Studio, Spark, AWS\_yn, Excel\_yn):** Spalten, aus denen hervorgeht, ob die betreffende Kompetenz in der Stellenbeschreibung verlangt wird (0 = nein, 1 = ja).
- **Min\_salary, Max\_salary:** Minimale und maximale Gehaltsschätzungen.
- **State, Same State, job\_state, Age, desc\_len, Num\_comp:** Zusätzliche Informationen wie Standort der Stelle, Alter des Unternehmens, Länge der Stellenbeschreibung und Anzahl der Mitbewerber.

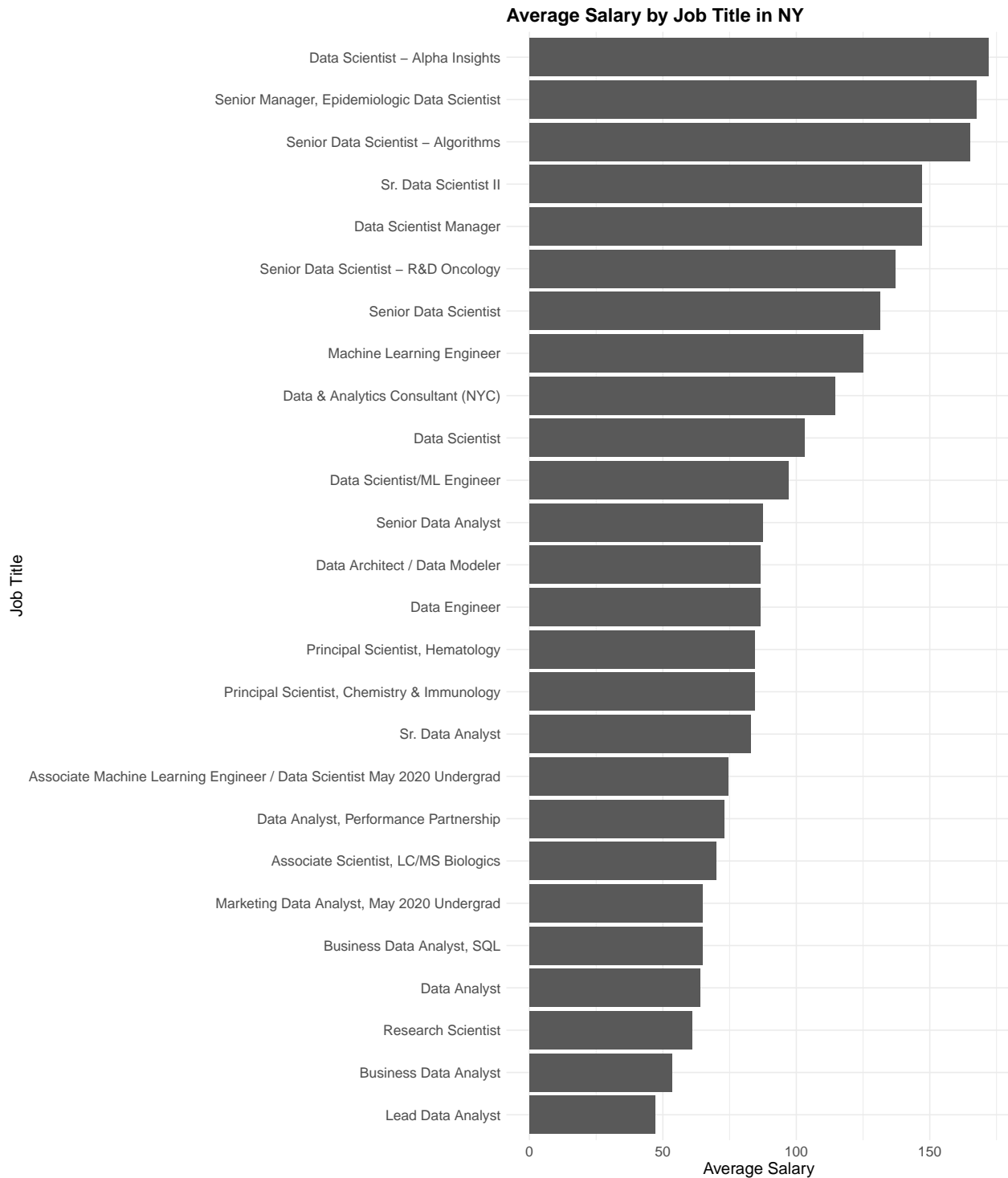
Es zeigt sich, dass eine Vielzahl von Spalten für die vorliegende Untersuchung irrelevant ist. Infolgedessen werden in einem späteren Teil der Arbeit irrelevante Spalten, wie beispielsweise die Kenntnisse in Python, R Studio, Spark und ähnlichen Programmen, welche ursprünglich aus der Jobbeschreibung extrahiert wurden, entfernt.

Im Folgenden wird eine erste Betrachtung der Daten vorgenommen. Zu diesem Zweck werden die Jobs in New York nach ihren jeweiligen Vergütungen geordnet und in Form eines Balkendiagramms dargestellt.

```
# Filterung der Daten für New York
data_ny <- data %>%
  filter(State == "NY")

# Durchschnittsgehalt nach Berufsbezeichnung
avg_salary_by_job_ny <- data_ny %>%
  group_by(`Job Title`) %>%
  summarise(Average_Salary = mean(`Salary Estimate`, na.rm = TRUE)) %>%
  arrange(desc(Average_Salary))

# Bar Plot
ggplot(avg_salary_by_job_ny,
  aes(x = reorder(`Job Title`, Average_Salary), y = Average_Salary)) +
  geom_bar(stat = "identity") +
  coord_flip() +
  labs(title = "Average Salary by Job Title in NY",
    x = "Job Title",
    y = "Average Salary") +
  theme_minimal() +
  theme(
    axis.title = element_text(size = 14),
    axis.text = element_text(size = 12),
    plot.title = element_text(size = 16, face = "bold")
  )
```



Es lässt sich eine signifikante Variation der Gehälter in New York beobachten. Zudem zeigt sich eine Korrelation zwischen Höhe des Gehalts und der Position: Mit steigendem Einkommen nehmen die Anzahl und der Rang der Senior- und Managerpositionen zu.

Da die Datengrundlage nicht in einem `igraph`-Objekt vorliegt und ungerichtet ist, ist es notwendig Knoten, Kanten sowie relevante Attribute wie beispielsweise Gewichtungen zu definieren, um überhaupt Netzwerkvisualisierungen in R durchführen zu können. Doch dazu mehr im nächsten Kapitel.

## 2 Analysestrategie

Im Rahmen meiner Analyse werde ich die verschiedenen Zentralitätsmaße zur Untersuchung der Position und Bedeutung von Unternehmen im Wettbewerbsnetzwerk heranziehen. Die Anwendung dieser Maße ermöglicht unterschiedlicher Einblicke in die strategische Verknüpfung von Unternehmen im Netzwerk sowie in die potenzielle Auswirkung dieser Verknüpfung auf die Gehaltspolitik und Attraktivität von Unternehmen im Data-Science-Bereich.

### 1. Datenaufbereitung und Netzwerkerstellung

In einem ersten Schritt wird ein Netzwerk aufgebaut, in dem jedes Unternehmen als Knoten dargestellt wird. Die Verbindungen zwischen den Knoten basieren auf der Spalte "Competitors". Dabei werden Unternehmen, die als Wettbewerber aufgeführt sind, durch eine Kante verbunden. Im nächsten Schritt erfolgt eine Aufbereitung der Daten. Dazu werden doppelte oder fehlerhafte Einträge entfernt sowie Unternehmensnamen standardisiert, um eine korrekte Kantenbildung zu gewährleisten.

### 2. Berechnung von Netzwerkmetriken und Zentralitätsmaßen

In der nächsten Phase erfolgt die Berechnung wichtiger Zentralitätsmetriken, welche die Bedeutung der einzelnen Unternehmen im Wettbewerbsnetzwerk bestimmen. Die Betweenness-Zentralität ermöglicht die Identifikation von Unternehmen, die eine Brückenfunktion zwischen anderen Unternehmen einnehmen. Diese Unternehmen sind von entscheidender Bedeutung für die Kommunikation und den Austausch von Informationen innerhalb des Netzwerkes. Die Degree-Zentralität gibt Aufschluss über die Anzahl direkter Wettbewerber, mit denen ein Unternehmen in Beziehung steht. Dies reflektiert die Wettbewerbsintensität und gibt Aufschluss über die Integration eines Unternehmens in das Wettbewerbsumfeld. Die Eigenvektor-Zentralität demonstriert die besondere Einflussnahme von Unternehmen, welche durch eine exzellente Vernetzung mit anderen zentralen Wettbewerbern gekennzeichnet ist.

### 3. Visualisierung und Strukturanalyse des Wettbewerbsnetzwerks

Im Folgenden erfolgt eine Anwendung von Algorithmen zur Erkennung von Communities, wie beispielsweise Louvain, mit dem Ziel, Cluster innerhalb des Netzwerks zu identifizieren. Die Analyse zielt darauf ab, natürliche Gruppen von Unternehmen zu identifizieren, die durch intensive Wettbewerbsverhältnisse gekennzeichnet sind. Des Weiteren erfolgt innerhalb der identifizierten Cluster eine Analyse der Gehaltsstrukturen sowie der Branchenzugehörigkeit, mit dem Ziel, etwaige Muster und Zusammenhänge zu identifizieren.

### 4. Explorative Gehaltsanalyse im Netzwerk

Abschließen wird im Rahmen einer explorativen Gehaltsanalyse dem Leser die Möglichkeit gegeben die Gehaltsstrukturen innerhalb des Netzwerkes gemeinsam mit den Zentralitätsmaßen selbstständig zu erforschen.



## 3 Analyse

### 3.1 Datenbereinigung

#### 3.1.1 Bereinigung für die geografische Analyse

Bei der Durchsicht des Datensatzes viel auf, dass die Spalten “Same State” und “job\_state” von der Logik her ähnlich sind. Dies soll nun näher untersucht werden, um spätere Fehler, vor allem bei geografischen Netzwerken(siehe Anhang), vorzubeugen.

```
# Auswahl der "State" und "job_state" Spalten
selected_data <- data %>%
  select(State, job_state)

# Heading der ausgewählten Spalten
head(selected_data, 15)
```

```
## # A tibble: 15 x 2
##   State job_state
##   <chr> <chr>
##  1 NM    NM
##  2 MD    MD
##  3 FL    FL
##  4 WA    WA
##  5 NY    NY
##  6 TX    TX
##  7 MD    MD
##  8 CA    CA
##  9 NY    NY
## 10 NY    NY
## 11 CA    CA
## 12 VA    VA
## 13 TX    TX
## 14 WA    WA
## 15 MA    MA
```

Es scheint, dass beide Spalten identisch sind. Diesbezüglich ist jedoch eine Prüfung erforderlich.

```
if (all(selected_data$State == selected_data$job_state, na.rm = TRUE)) {
  print("Alle Werte in 'State' und 'job_state' sind identisch.")
} else {
  print("Es gibt Unterschiede zwischen 'State' und 'job_state'.")
}
```

```
## [1] "Es gibt Unterschiede zwischen 'State' und 'job_state'."
```

Diese Annahme erweist sich jedoch als unzutreffend, da sich in der Tat Unterschiede feststellen lassen.

```
# Auswahl der Zeilen, in denen "State" und "job_state" unterschiedlich sind
different_states <- selected_data %>%
  filter(State != job_state)

print(different_states, n = Inf)
```

```
## # A tibble: 1 x 2
##   State      job_state
##   <chr>      <chr>
## 1 Los Angeles CA
```

Es fällt auf, dass LA und Los Angeles nicht einheitlich verwendet werden. Außerdem ist Los Angeles kein eigener Bundesstaat, sondern ein Teil von Kalifornien (CA). Dies sollte korrigiert werden.

Außerdem sollte bei der weiteren Vorgehensweise darauf geachtet werden, dass Werte wie "Na" oder "-1" vor den Auswertungen entfernt werden.

```
# Ersetzen von "Los Angeles" durch "LA" und "LA" durch "CA"
data <- data %>%
  mutate(State = ifelse(State == "Los Angeles", "LA", State),
         job_state = ifelse(job_state == "Los Angeles", "LA", job_state))

data <- data %>%
  mutate(State = ifelse(State == "LA", "CA", State),
         job_state = ifelse(job_state == "LA", "CA", job_state))

# Erneute Überprüfung
selected_data <- data %>%
  select(State, job_state)

if (all(selected_data$State == selected_data$job_state, na.rm = TRUE)) {
  print("Alle Werte in 'State' und 'job_state' sind identisch.")
} else {
  print("Es gibt Unterschiede zwischen 'State' und 'job_state'.")
}
```

```
## [1] "Alle Werte in 'State' und 'job_state' sind identisch."
```

### 3.1.2 Überprüfung auf weitere fehlende Werte

Im nächsten Schritt wird der Datensatz auf weitere fehlende Werte überprüft, um sicherzustellen, dass keine fehlerhaften Daten in die Analyse einfließen.

```
# Überprüfen auf NA-Werte
na_counts <- colSums(is.na(data))

# Anzahl der NA-Werte pro Spalte:
print(na_counts)
```

```
##      Job Title  Salary Estimate  Job Description      Rating
##           0           0           0           0
## Company Name      Location  Headquarters      Size
##           0           0           0           0
##      Founded Type of ownership      Industry      Sector
##           0           0           0           0
##      Revenue      Competitors  Min_Salary  Max_Salary
##           0           0           0           0
##           State      Same State      Age      Python_yn
##           0           0           0           0
##      R Studio      Spark      AWS_yn      Excel_yn
##           0           0           0           0
##      Job_simp      job_state  desc_len      Num_comp
##           0           0           0           0
```

```
# Überprüfen auf "na"-Werte (kleingeschrieben)
na_string_counts <- sapply(data, function(x) sum(tolower(x) == "na", na.rm = TRUE))

# Anzahl der "na"-Werte pro Spalte:
```

```
print(na_string_counts)
```

```
##      Job Title  Salary Estimate  Job Description      Rating
##           0              0              0              0
##      Company Name      Location      Headquarters      Size
##           0              0              0              0
##           Founded Type of ownership      Industry      Sector
##           0              0              0              0
##           Revenue      Competitors      Min_Salary      Max_Salary
##           0              0              0              0
##           State      Same State      Age      Python_yn
##           0              0              0              0
##           R Studio      Spark      AWS_yn      Excel_yn
##           0              0              0              0
##           Job_simp      job_state      desc_len      Num_comp
##           184              0              0              0
```

```
# Überprüfen auf -1-Werte
```

```
neg_one_counts <- sapply(data, function(x) sum(x == -1, na.rm = TRUE))
```

```
# Anzahl der -1 Werte in der Spalte Competitors
```

```
print(neg_one_counts)
```

```
##      Job Title  Salary Estimate  Job Description      Rating
##           0              0              0              11
##      Company Name      Location      Headquarters      Size
##           0              0              1              1
##           Founded Type of ownership      Industry      Sector
##           50              1              10              10
##           Revenue      Competitors      Min_Salary      Max_Salary
##           1              460              0              0
##           State      Same State      Age      Python_yn
##           0              0              50              0
##           R Studio      Spark      AWS_yn      Excel_yn
##           0              0              0              0
##           Job_simp      job_state      desc_len      Num_comp
##           0              0              0              0
```

Es zeigt sich, dass lediglich in der Spalte "Competitors" relevante negative Werte zu finden sind, welche nun entfernt werden müssen. Die "na"-Werte in der Spalte "Job\_simp" hingegen haben keinen Einfluss auf die Analyse.

```
# Entfernen von Zeilen mit -1 Werten in der Spalte "Competitors"
```

```
data <- data %>%
```

```
  filter(Competitors != -1)
```

```
# Überprüfen auf -1-Werte nach Entfernung
```

```
neg_one_counts <- sapply(data, function(x) sum(x == -1, na.rm = TRUE))
```

```
# Anzahl der -1 Werte pro Spalte:
```

```
print(neg_one_counts)
```

```
##      Job Title  Salary Estimate  Job Description      Rating
##           0              0              0              0
##      Company Name      Location      Headquarters      Size
##           0              0              0              0
##           Founded Type of ownership      Industry      Sector
```

```
##           1           0           0           0
##      Revenue    Competitors    Min_Salary    Max_Salary
##           0           0           0           0
##           State    Same State           Age    Python_yn
##           0           0           1           0
##      R Studio      Spark      AWS_yn      Excel_yn
##           0           0           0           0
##      Job_simp      job_state    desc_len      Num_comp
##           0           0           0           0
```

### 3.1.3 Entfernen irrelevanter Spalten

In einem nächsten Schritt werden auf Basis der zuvor festgelegten Analysestrategie sowie der geplanten Analysen diejenigen Spalten identifiziert, die für die Analysen nicht erforderlich sind und anschließend entfernt.

```
# Entfernen irrelevanter Spalten
# Job Description, Rating, Headquarters, Size, Founded, Type of ownership, Sector, Revenue und Skills

data <- data %>%
  select(-c(`Job Description`, Rating, Headquarters, Size, Founded,
            `Type of ownership`, Sector, Revenue, Python_yn,
            `R Studio`, Spark, AWS_yn, Excel_yn, desc_len, Num_comp))

# Ausgeben der noch enthaltenen Spalten
print(data %>% names())
```

```
## [1] "Job Title"      "Salary Estimate" "Company Name"    "Location"
## [5] "Industry"       "Competitors"     "Min_Salary"      "Max_Salary"
## [9] "State"          "Same State"      "Age"             "Job_simp"
## [13] "job_state"
```

### 3.1.4 Bereinigung irrelevanter Jobtitel

Im Rahmen dieser Arbeit erfolgt eine Analyse von Gehältern und Wettbewerbsbeziehungen für den Data-Science-Jobmarkt. Dabei ist zu überprüfen, ob alle Spalten tatsächlich konkrete Data-Science-Jobs repräsentieren.

Bei einer ersten Betrachtung des in der Einleitung präsentierten Balkendiagramms wird ersichtlich, dass eine Reihe von Jobtiteln nicht unmittelbar mit dem Bereich der “Data Science” assoziiert werden können.

```
# ausgeben alles uniqueen Job Title und Job_simp
print(data %>% select(`Job Title`, `Job_simp`) %>% unique())

## # A tibble: 111 x 2
##   `Job Title`                               Job_simp
##   <chr>                                       <chr>
## 1 Data Scientist                           data scienti~
## 2 Staff Data Scientist - Technology        data scienti~
## 3 Scientist I/II, Biology                  na
## 4 Data Analyst                             analyst
## 5 Scientist                                na
## 6 Senior Data Scientist                    data scienti~
## 7 Lead Data Scientist                      data scienti~
## 8 Spectral Scientist/Engineer              na
## 9 College Hire - Data Scientist - Open to December 2019 Graduates data scienti~
## 10 Data Scientist, Office of Data Science  data scienti~
## # i 101 more rows
```

Es lässt sich feststellen, dass der Datensatz auch eine Reihe von Jobs von Wissenschaftlern umfasst, die nicht unmittelbar mit Data Science assoziiert werden. Diese Jobs sind zudem nicht mit einem vereinfachten Jobtitel (“Job\_simp”) versehen, der auf eine Tätigkeit im Bereich Data Science hinweist. Daher ist es erforderlich, alle Datensätze mit na-Werten in der Spalte “Job\_simp” zu eliminieren. Diese besitzen auch eine andere Schreibweise, deswegen wurden sie nicht direkt entfernt.

```
# Entfernen von Zeilen mit NA-Werten in der Spalte "Job_simp"
data <- data %>%
  filter(Job_simp != "na")

# Überprüfen auf "na"-Werte in der Spalte "Job_simp"
na_job_simp <- sum(data$Job_simp == "na")

print(paste("Anzahl der 'na'-Werte in der Spalte 'Job_simp':", na_job_simp))

## [1] "Anzahl der 'na'-Werte in der Spalte 'Job_simp': 0"

print("Alle verbleibenden Jobtitel:")

## [1] "Alle verbleibenden Jobtitel:"

print(data %>% select(`Job Title`, Job_simp) %>% unique())

## # A tibble: 73 x 2
##   `Job Title`                               Job_simp
##   <chr>                                       <chr>
## 1 Data Scientist                           data scienti~
## 2 Staff Data Scientist - Technology        data scienti~
## 3 Data Analyst                             analyst
## 4 Senior Data Scientist                    data scienti~
## 5 Lead Data Scientist                      data scienti~
## 6 College Hire - Data Scientist - Open to December 2019 Graduates data scienti~
## 7 Data Scientist, Office of Data Science   data scienti~
## 8 Data Scientist in Artificial Intelligence Early Career data scienti~
## 9 Data Scientist - Research                data scienti~
## 10 Data Scientist SR                       data scienti~
## # i 63 more rows
```

Nachdem die Bereinigung des Datensatzes nun abgeschlossen ist, kann mit der Analyse begonnen werden.

## 3.2 Wettbewerbsnetzwerk

In diesem Abschnitt wird mit der eigentlichen Analyse, dem Ziel dieser Arbeit, der Erstellung einer Wettbewerbsanalyse begonnen.

### 3.2.1 Erstellung eines Wettbewerbsnetzwerkes

Zu diesem Zweck wird ein Netzwerk erstellt, das auf den Wettbewerbsbeziehungen zwischen Unternehmen basiert.

Die Wettbewerbsbeziehungen werden anhand der in der Spalte “Competitors” aufgeführten Unternehmen definiert. Die Punkte im Netzwerk repräsentieren die Unternehmen, während die Kanten die Wettbewerbsbeziehungen zwischen ihnen darstellen.

Die Gewichtung der Kanten erfolgt wie folgt:

- Direkte Wettbewerber erhalten eine Gewichtung von 1.
- Unternehmen in derselben Branche, jedoch nicht als direkte Wettbewerber aufgeführt, erhalten eine Gewichtung von 0.5.

Branchenbezogene Wettbewerbsbeziehungen sind in blau dargestellt, während direkte Wettbewerber in rot hervorgehoben sind.

```
# Extrahiere Unternehmen und ihre Wettbewerber
edges <- data %>%
  separate_rows(Competitors, sep = ", ") %>%
  select(`Company Name`, Competitors) %>%
  rename(from = `Company Name`, to = Competitors) %>%
  mutate(weight = 1) # Gewichtung für direkte Wettbewerber

# Füge Unternehmen in derselben Branche mit Gewichtung 0.5 hinzu
industry_edges <- data %>%
  select(`Company Name`, Industry) %>%
  inner_join(
    data %>% select(`Company Name`, Industry),
    by = "Industry",
    relationship = "many-to-many"
  ) %>%
  filter(`Company Name.x` != `Company Name.y`) %>%
  select(from = `Company Name.x`, to = `Company Name.y`) %>%
  mutate(weight = 0.5) # Gewichtung für gleiche Branche

# Kombiniere beide Datensätze
all_edges <- bind_rows(edges, industry_edges)

# Erstelle den Graphen
g_competitors <- graph_from_data_frame(all_edges, directed = FALSE)

# Entferne mehrere Kanten zwischen denselben Punkten
g_competitors <- simplify(g_competitors, remove_multiple = TRUE,
  edge.attr.comb = "first")

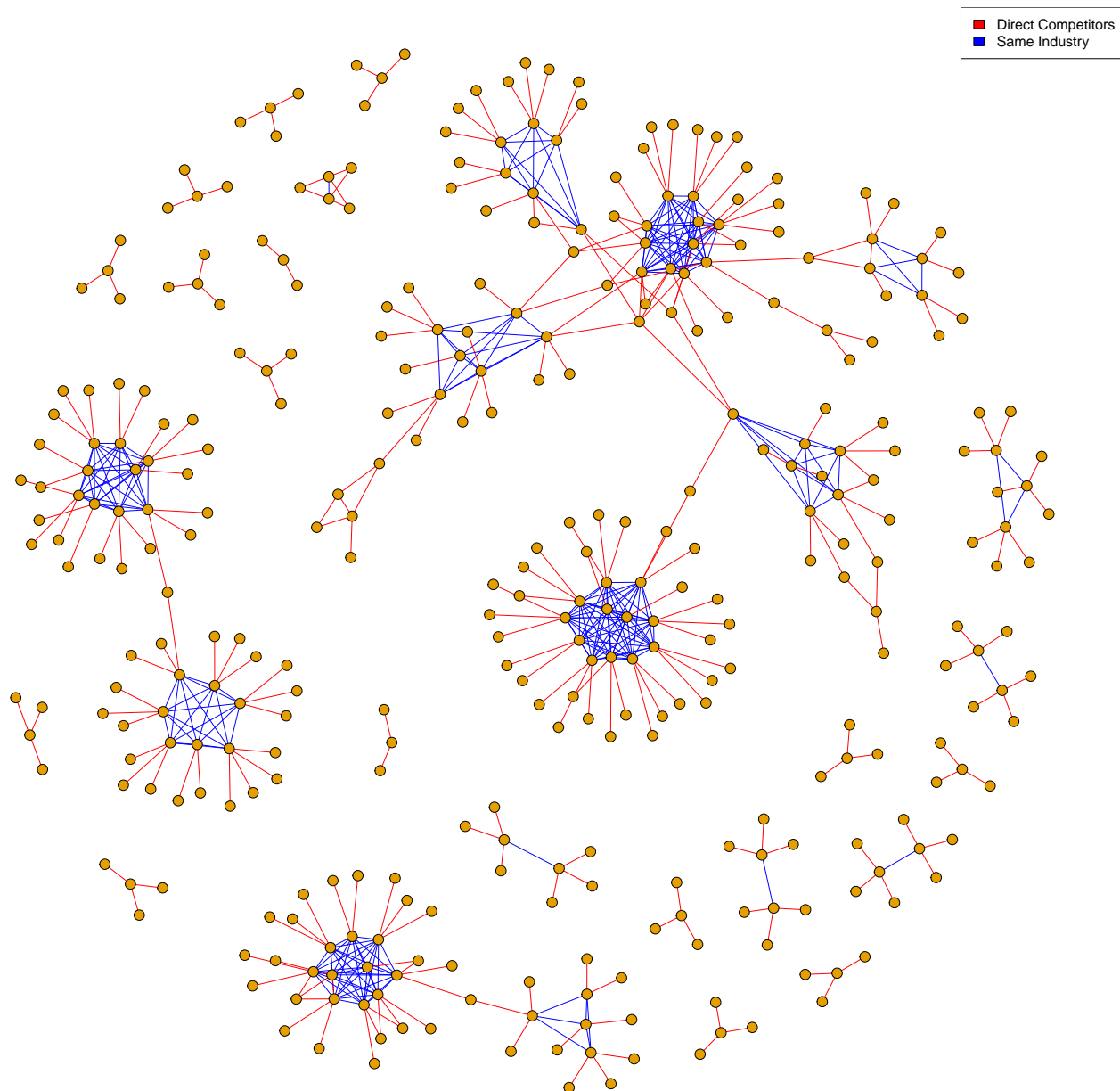
# Setze die Farben der Kanten basierend auf der Gewichtung
E(g_competitors)$color <- ifelse(E(g_competitors)$weight == 1, "red", "blue")

# Visualisiere das Netzwerk mit kleineren Knoten
plot(g_competitors, vertex.label = NA,
  vertex.size = 2, # Kleinere Knoten
  edge.width = E(g_competitors)$weight, # Gewichtung der Kanten
  edge.arrow.size = 0.5, # Kleinere Pfeile
  layout = layout_with_fr)

title(main = "Unternehmensnetzwerk basierend auf Wettbewerbern und Branchen", cex.main = 2)

# Legende für Kantenfarben
legend("topright", legend = c("Direct Competitors", "Same Industry"),
  fill = c("red", "blue"))
```

## Unternehmensnetzwerk basierend auf Wettbewerbern und Branchen



Es lassen sich einige interessante Beobachtungen aus dem Netzwerk ziehen. Einerseits sind ganz eindeutig Branchencluster zu erkennen, die auf die Branchenzugehörigkeit der Unternehmen hinweisen.

```
# Ausgeben der 10 häufigsten Branchen im Netzwerk
top_industries <- data %>%
  count(Industry, sort = TRUE) %>%
  head(10)

# Welche Unternehmen sind in mehreren Branchen vertreten?
multi_industry_companies <- data %>%
  group_by(`Company Name`) %>%
  summarise(Num_Industries = n_distinct(Industry)) %>%
  filter(Num_Industries > 1) %>%
  arrange(desc(Num_Industries))
```

```
# Sind Unternehmen enthalten, die in mehreren Branchen vertreten sind?
if (nrow(multi_industry_companies) == 0) {
  print("Keine Unternehmen in mehreren Branchen vertreten.")
} else {
  print("Unternehmen in mehreren Branchen vertreten.")
}
```

```
## [1] "Keine Unternehmen in mehreren Branchen vertreten."
```

Jedoch sind keine Unternehmen in mehreren Branchen vertreten, was darauf hindeutet, dass die Branchenzugehörigkeit eindeutig ist. Aber es gibt einige Unternehmen, die mit direkter Konkurrenz die verschiedenen Branchen verbinden. Dies könnte auf eine Diversifikation der Geschäftsfelder hindeuten, die eine breitere Wettbewerbsbasis schafft.

Da aber wie oberhalb dargestellt, die Branchenzugehörigkeit eindeutig ist, und somit die Branchenzugehörigkeit keinen Mehrwert für die Analyse bietet, wird diese nicht weiter verfolgt.

### 3.2.2 Betrachtung fokussiert auf direkte Wettbewerber

Aus diesem Grund wird das Netzwerk auf direkte Wettbewerber beschränkt, um die Analyse zu vereinfachen und die Relevanz der Wettbewerbsbeziehungen zu erhöhen.

```
# Extrahiere Unternehmen und ihre Wettbewerber
edges <- data %>%
  separate_rows(Competitors, sep = ", ") %>%
  select(`Company Name`, Competitors) %>%
  rename(from = `Company Name`, to = Competitors) %>%
  mutate(weight = 1) # Gewichtung für direkte Wettbewerber

# Summiere die Gewichtungen für mehrere Kanten zwischen denselben Punkten
edge_weights <- edges %>%
  group_by(from, to) %>%
  summarise(weight = sum(weight), .groups = 'drop')

# Erstelle den Graphen nur mit direkten Wettbewerbern
g_direct_competitors <- graph_from_data_frame(edge_weights, directed = FALSE)

# Setze die Gewichtungen der Kanten im Graphen
E(g_direct_competitors)$weight <- edge_weights$weight

# Füge die Gehaltsdaten hinzu und berechne das durchschnittliche Gehalt pro Unternehmen
salary_data <- data %>%
  group_by(`Company Name`) %>%
  summarise(AverageSalary = mean(`Salary Estimate`, na.rm = TRUE))

# Füge die Gehaltsdaten zu den Knoten des Graphen hinzu
V(g_direct_competitors)$salary <- salary_data$AverageSalary[match(V(g_direct_competitors)$name, salary_data$name)]

# Setze die Farben der Knoten basierend auf den Gehältern
salary_quantiles <- quantile(V(g_direct_competitors)$salary, probs = seq(0, 1, length.out = 6), na.rm = TRUE)
color_palette <- brewer.pal(5, "YlOrRd")
V(g_direct_competitors)$color <- cut(V(g_direct_competitors)$salary,
  breaks = salary_quantiles,
  labels = FALSE,
  include.lowest = TRUE)
```



```

V(g_direct_competitors)$color <- color_palette[V(g_direct_competitors)$color]

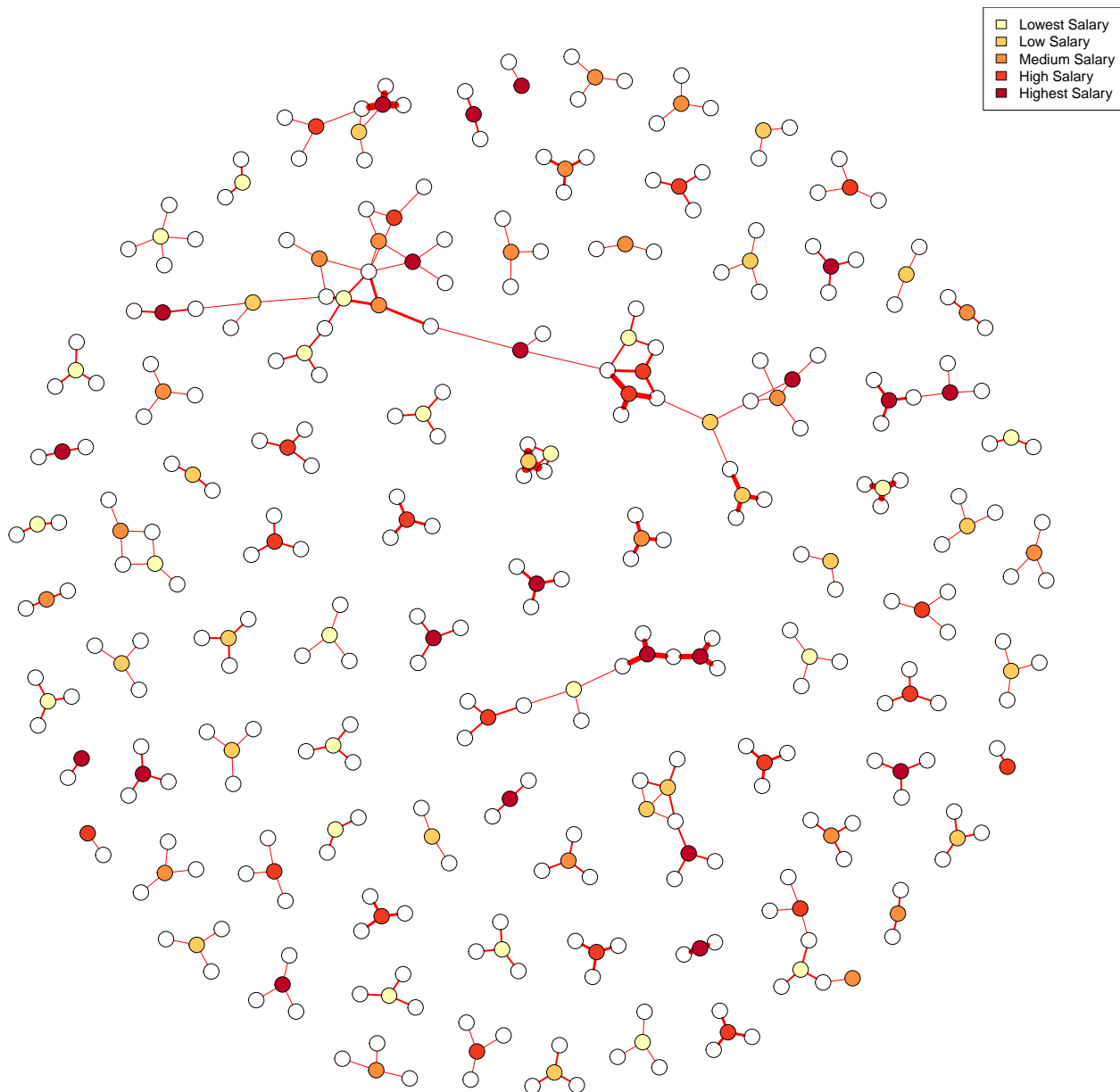
# Setze die Farben der Kanten basierend auf der Gewichtung
E(g_direct_competitors)$color <- "red"

# Visualisiere das Netzwerk mit kleineren Knoten
plot(g_direct_competitors, vertex.label = NA,
     vertex.size = 3, # Kleinere Knoten
     edge.width = 1 * E(g_direct_competitors)$weight, # Reduzierte Gewichtung der Kanten
     edge.arrow.size = 1,
     layout = layout_with_fr
)
title(main = "Unternehmensnetzwerk basierend auf direkten Wettbewerbern", cex.main = 2)

# Legende für Knotenfarben
legend("topright", legend = c("Lowest Salary", "Low Salary", "Medium Salary", "High Salary", "Highest Salary"),
      fill = color_palette)

```

## Unternehmensnetzwerk basierend auf direkten Wettbewerbern



```
# Ausgabe der Gehälter der Unternehmen
salary_data <- data %>%
  group_by(`Company Name`) %>%
  summarise(AverageSalary = mean(`Salary Estimate`, na.rm = TRUE)) %>%
  arrange(desc(AverageSalary))
```

```
# Ausgabe der 10 Unternehmen mit den höchsten Gehältern
print(salary_data, n = 10)
```

```
## # A tibble: 107 x 2
##   `Company Name`      AverageSalary
##   <chr>              <dbl>
## 1 Gallup              238.
## 2 Credit Sesame       205
```

```
## 3 The Climate Corporation          194
## 4 Liberty Mutual Insurance        179.
## 5 Samsung Research America        177
## 6 Western Digital                 172.
## 7 Glassdoor                       162
## 8 Netskope                        154.
## 9 Factual                         153
## 10 Visa Inc.                      153.
## # i 97 more rows
```

TODO

### 3.3 Zentralitätsanalyse innerhalb des Netzwerkes

Die hier präsentierten Zentralitätsmaße lassen sich unmittelbar auf die Gehaltsstrukturen im Kontext des Wettbewerbsnetzwerks anwenden. Auf diese Weise könnten sie dazu beitragen, die Position eines Unternehmens in Bezug auf seine Wettbewerber hinsichtlich der Gehälter zu bestimmen.

Folgend wird die Berechnung der Zentralitätsmaße für die Unternehmen im Wettbewerbsnetzwerk dargelegt.

```
# Datenauswahl für die Zentralitätsanalyse
selected_data <- data %>%
  select(`Company Name`, Competitors, `Salary Estimate`)

# Extrahiere Unternehmen und ihre Wettbewerber
edges <- selected_data %>%
  separate_rows(Competitors, sep = ", ") %>%
  select(`Company Name`, Competitors) %>%
  rename(from = `Company Name`, to = Competitors)

# Erstelle den Graphen
g_competitors_zentralität <- graph_from_data_frame(edges, directed = FALSE)

# Berechne die Zentralitätsmaße für die Knoten
degree_centrality <- degree(g_competitors_zentralität, mode = "all")
betweenness_centrality <- betweenness(g_competitors_zentralität, directed = FALSE)
eigenvector_centrality <- evcent(g_competitors_zentralität, directed = FALSE)

## Warning: `evcent()` was deprecated in igraph 2.0.0.
## i Please use `eigen_centrality()` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

#### 3.3.1 Betweenness-Zentralität

Jetzt soll das igraph-Paket in R verwendet werden, um die Betweenness-Zentralität für jeden Knoten zu berechnen. Unternehmen mit hoher Betweenness-Centralität könnten strategische Wettbewerbsvorteile aufweisen, da sie als Vermittler zwischen mehreren Konkurrenten fungieren und dadurch den Informationsfluss beeinflussen können.

Die Unternehmen könnten höhere Gehälter anbieten, um hochqualifizierte Arbeitskräfte zu gewinnen, die dazu beitragen, ihre zentrale Position und die damit verbundenen Wettbewerbsvorteile zu erhalten. Alternativ könnte ein hohes Gehalt auch als Indikator für eine hohe Nachfrage nach qualifizierten Mitarbeitenden in solchen Schlüsselpositionen dienen, da das Unternehmen sich in einem Bereich positioniert, der viele strategische Informationen benötigt.

```

# Berechne die Betweenness-Centrality und sortiere sie absteigend
top_betweenness <- head(sort(betweenness centrality, decreasing = TRUE), 5)

# Erstelle ein DataFrame mit den Namen der Unternehmen und ihrer Betweenness-Centrality
top_betweenness_df <- data.frame(
  Company = names(top_betweenness),
  Betweenness = as.numeric(top_betweenness),
  stringsAsFactors = FALSE
)

# Erstellen einer Tabelle
kable(top_betweenness_df, format = "latex", booktabs = TRUE, align = "l") %>%
kable_styling(latex_options = c("striped", "hold_position"), position = "left")

```

| Company                                 | Betweenness |
|---|-------------|
| PA Consulting                           | 492.8571    |
| Gallup                                  | 479.0000    |
| Booz Allen Hamilton                     | 477.1818    |
| McKinsey & Company                      | 462.0000    |
| General Dynamics Information Technology | 354.0000    |

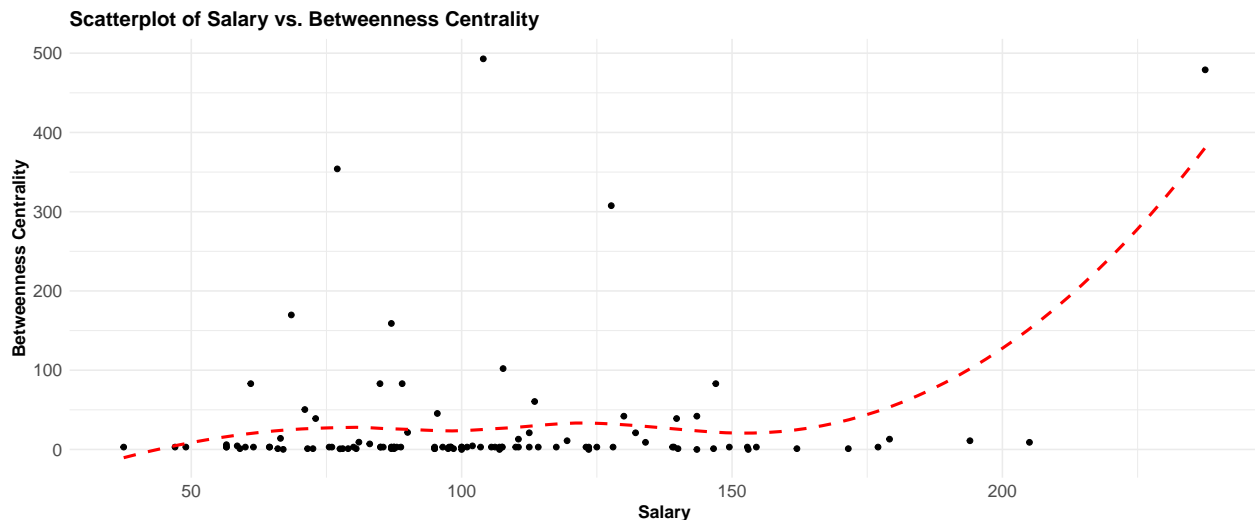
```

# Zusammenhang zwischen durchschnittliches Gehalt und Betweenness-Zentralität
salary_betweenness <- data.frame(
  Salary = salary_data$AverageSalary,
  Betweenness = betweenness centrality[match(salary_data$`Company Name`, names(betweenness centrality))]
)

# Scatterplot mit roter, gestrichelter Trendlinie
ggplot(salary_betweenness, aes(x = Salary, y = Betweenness)) +
  geom_point() +
  geom_smooth(method = "loess", se = FALSE, color = "red", linetype = "dashed") +
  labs(title = "Scatterplot of Salary vs. Betweenness Centrality",
       x = "Salary", y = "Betweenness Centrality") +
  theme_minimal() +
  theme(
    plot.title = element_text(size = 14, face = "bold"),
    axis.title = element_text(size = 12, face = "bold"),
    axis.text = element_text(size = 12)
  )

## `geom_smooth()` using formula = 'y ~ x'

```



Der Scatterplot veranschaulicht die Korrelation zwischen dem Gehalt auf der x-Achse und der Betweenness Centrality auf der y-Achse. Die rote Linie zeigt einen nichtlinearen Trend zwischen Gehalt und Betweenness Centrality, was auf eine mögliche Korrelation hinweisen könnte, insbesondere bei höheren Gehältern.

Die Mehrzahl der Unternehmen weist eine geringe Betweenness Centrality auf, wobei die Höhe der Vergütung eine eher untergeordnete Rolle spielt. Allerdings lässt sich eine Ausnahme beobachten. Ab ca. 150k Dollar auf der Gehaltsachse zeigt der Trend einen deutlichen Anstieg der Betweenness Centrality. Dies könnte bedeuten, dass Unternehmen mit sehr hohen Gehältern tendenziell auch höhere Betweenness Centrality Werte aufweisen und somit eine wichtige Vermittlerrolle im Netzwerk einnehmen. Jedoch wird dieser Anstieg bloß von einem Unternehmen repräsentiert, was auf eine Ausnahme hinweisen könnte. Dieses Unternehmen könnten jedoch ein besonders attraktiver Arbeitgeber für strategische Schlüsselpositionen sein und durch hohe Gehälter zusätzlich Talente anziehen.

Allgemeins lässt dies den Schluss zu, dass bestimmte Unternehmen eine zentrale Vermittlerrolle im Netzwerk einnehmen, ohne zwangsläufig die höchsten Gehälter zu offerieren.

Die wenigen Unternehmen mit einer sehr hohen Betweenness Centrality könnten als strategische Vermittler im Markt auftreten und dabei möglicherweise andere Vorteile nutzen, anstatt ihre Gehälter zu erhöhen. Die dargelegte Erkenntnis lässt die Hypothese zu, dass Unternehmen, die über zahlreiche Verbindungen im Wettbewerbsnetzwerk verfügen, ihre Attraktivität nicht allein durch monetäre Zuwendungen, sondern auch durch andere Vorteile oder Reputation aufrechterhalten könnten. Für hochspezialisierte Unternehmen, die in einer Nische tätig sind und weniger Konkurrenz haben, könnte es sich lohnen, gezielt in Gehälter zu investieren, um Talente anzuziehen.

Im folgenden Schritt soll sich der Außreißer mit dem höchsten Gehalt und der zweithöchsten Betweenness Centrality genauer betrachtet werden.

```
# Unternehmen mit dem höchsten Gehalt und der zweithöchsten Betweenness Centrality
outlier <- salary_betweenness %>%
  filter(Salary > 210 & Betweenness > 0.0001)

# Ausgabe des Unternehmens
print(outlier)
```

```
##           Salary Betweenness
## Gallup    237.5         479
```

Das Unternehmen mit dem höchsten Gehalt und der zweithöchsten Betweenness Centrality ist "Gallup".

```
# Berechne die Nachbarn von "Gallup"
gallup_neighbors <- neighbors(g_direct_competitors, "Gallup", mode = "all")
```

```
# Erstelle ein DataFrame mit den Namen der direkten Wettbewerber von "Gallup"
gallup_neighbors_df <- data.frame(
  Competitors = names(gallup_neighbors),
  stringsAsFactors = FALSE
)

# Erstelle eine schön formatierte Tabelle im LaTeX-Format
kable(gallup_neighbors_df, format = "latex", booktabs = TRUE, align = "l") %>%
  kable_styling(latex_options = c("striped", "hold_position"), position = "left") %>%
  column_spec(1, width = "5cm") # Passe die Spaltenbreite an
```

| Competitors         |
|---------------------|
| Booz Allen Hamilton |
| Advisory Board      |
| McKinsey & Company  |

“Gallup” hat 3 direkte Wettbewerber: “Booz Allen Hamilton”, “Advisory Board” und “McKinsey & Company”. Daraus lässt sich schließen, dass “Gallup” eine zentrale Vermittlerrolle zwischen diesen drei Unternehmen einnimmt, was durch die hohe Betweenness Centrality und das hohe Gehalt reflektiert wird.

### 3.3.2 Degree-Zentralität

Hier wird die Anzahl der Kanten gezählt, die an jedem Knoten hängen. Hohe Werte können auf starke Verbindungen zu anderen Unternehmen hinweisen. Unternehmen mit hoher Degree Centrality sind also in einem besonderem Maße der Konkurrenz durch eine Vielzahl anderer Firmen ausgesetzt, was zu einem hohen Druck innerhalb der Branche führen kann.

Ein hoher Degree Centrality-Wert lässt die Vermutung zu, dass es sich um ein Unternehmen handelt, welches sich durch höhere Gehälter von der Konkurrenz abheben und somit Talente anwerben möchte. Andererseits besteht für Unternehmen in stark besetzten Märkten die Möglichkeit, durch Maßnahmen wie eine attraktive Arbeitskultur oder Weiterbildungsmöglichkeiten für Mitarbeiter, trotz eines geringeren Gehalts, für Bewerber attraktiver zu sein. In diesem Fall können Unternehmen mit mittleren oder niedrigeren Degree-Centrality-Werten im Vergleich attraktivere Gehälter bieten, da sie weniger Wettbewerbsdruck haben und gezielt in Gehälter investieren können.

```
# Berechne die Degree-Centrality und sortiere sie absteigend
top_degree <- head(sort(degree centrality, decreasing = TRUE), 5)

# Erstelle ein DataFrame mit den Namen der Unternehmen und ihrer Degree-Centrality
top_degree_df <- data.frame(
  Company = names(top_degree),
  Degree = as.numeric(top_degree),
  stringsAsFactors = FALSE
)

# Erstelle die Tabelle und zentriere sie links
kable(gallup_neighbors_df, format = "latex", booktabs = TRUE, align = "l") %>%
  kable_styling(latex_options = c("striped", "hold_position"), position = "left")
```

| Competitors         |
|---------------------|
| Booz Allen Hamilton |
| Advisory Board      |
| McKinsey & Company  |

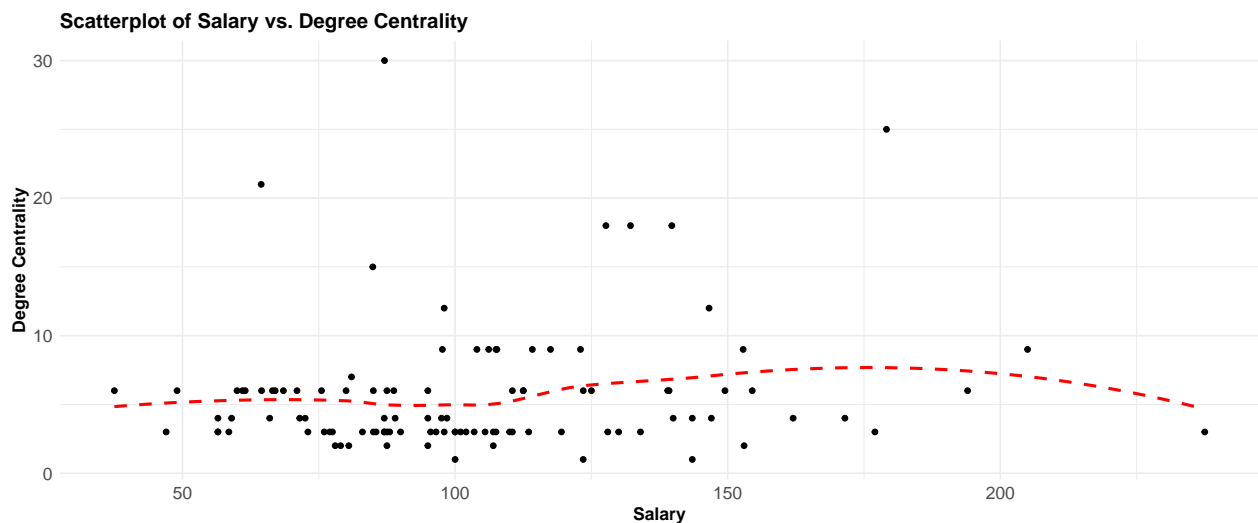
```

# Zusammenhang zwischen durchschnittliches Gehalt und Degree-Zentralität
salary_degree <- data.frame(
  Salary = salary_data$AverageSalary,
  Degree = degree_centrality[match(salary_data$`Company Name`, names(degree_centrality))]
)

# Scatterplot
ggplot(salary_degree, aes(x = Salary, y = Degree)) +
  geom_smooth(method = "loess", se = FALSE, color = "red", linetype = "dashed") +
  geom_point() +
  labs(title = "Scatterplot of Salary vs. Degree Centrality",
       x = "Salary", y = "Degree Centrality") +
  theme_minimal() +
  theme(
    plot.title = element_text(size = 14, face = "bold"),
    axis.title = element_text(size = 12, face = "bold"),
    axis.text = element_text(size = 12)
  )

```

```
## `geom_smooth()` using formula = 'y ~ x'
```



Die rote Linie zeigt einen flachen Trend ohne starke Korrelation in diesem Scatterplot zwischen Gehalt und Degree Centrality.

Der weitgehend flache Verlauf der Trendlinie lässt den Schluss zu, dass kein klarer Zusammenhang zwischen der Degree Centrality und den Gehältern besteht. Dies könnte implizieren, dass die Anzahl direkter Wettbewerber (Degree Centrality) für sich genommen keinen signifikanten Einfluss auf das Gehaltsniveau eines Unternehmens ausübt.

Ein kleiner Anstieg des Gehalts ist insbesondere im Bereich zwischen 100 und 150 zu verzeichnen, was darauf hindeutet, dass Unternehmen mit mittlerem Gehaltsniveau tendenziell mehr direkte Wettbewerber haben. Diese Entwicklung lässt sich dadurch erklären, dass Unternehmen mit mittleren Gehaltsniveaus in stärker besetzten Märkten tätig sind, in denen eine größere Anzahl an Unternehmen um Talente konkurriert.

Die meisten Werte der Degree Centrality sind relativ niedrig, was darauf hindeutet, dass eine Vielzahl von Unternehmen lediglich eine geringe Anzahl direkter Wettbewerber aufweist. Einige Unternehmen mit hoher Degree Centrality und niedrigen bis mittleren Gehältern heben sich jedoch von der Gesamtheit ab. Es ist möglich, dass diese Unternehmen in stark umkämpften Märkten tätig sind, jedoch nicht über die erforderlichen Ressourcen verfügen, um hohe Gehälter zu offerieren.

Ein Vergleich der Degree Centrality mit der Betweenness Centrality zeigt, dass die Degree Centrality einen geringeren Einfluss auf das Gehalt zu haben scheint. Dies lässt den Schluss zu, dass nicht die Anzahl der direkten Wettbewerber, sondern die Position eines Unternehmens als strategischer Vermittler im Netzwerk (Betweenness) einflussreicher für die Höhe des Gehalts ist.

### 3.3.3 Eigenvector-Zentralität

Unternehmen mit hoher Eigenvector Centrality stehen nicht nur in Konkurrenz zu einer Vielzahl von Unternehmen, sondern insbesondere zu besonders einflussreichen Wettbewerbern im Netzwerk.

Unternehmen mit hoher Eigenvector Centrality könnten in der Konsequenz wettbewerbsfähige Gehälter anbieten müssen, um sich im Wettbewerb mit zentralen und attraktiven Arbeitgebern zu behaupten. Folglich sind auch die umliegenden Firmen gezwungen, ihre Angebote anzupassen, um für die Talente am Arbeitsmarkt attraktiv zu bleiben. Es besteht die Möglichkeit, dass diese Unternehmen die Gehälter für spezifische, wettbewerbsrelevante Rollen erhöhen, um den Marktstandards und den Anforderungen eines zentralen Wettbewerbsnetzwerks gerecht zu werden.

```
# Berechne die Eigenvector-Centrality und sortiere sie absteigend
top_eigenvector <- head(sort(eigenvector_centrality$vector, decreasing = TRUE), 5)

# Erstelle ein DataFrame mit den Namen der Unternehmen und ihrer Eigenvector-Centrality
top_eigenvector_df <- data.frame(
  Company = names(top_eigenvector),
  Eigenvector = as.numeric(top_eigenvector),
  stringsAsFactors = FALSE
)

# Erstelle die Tabelle und zentriere sie links
kable(top_eigenvector_df, format = "latex", booktabs = TRUE, align = "l") %>%
  kable_styling(latex_options = c("striped", "hold_position"), position = "left")
```

| Company                               | Eigenvector |
|---------------------------------------|-------------|
| PNNL                                  | 1.0000000   |
| Oak Ridge National Laboratory         | 0.5887841   |
| National Renewable Energy Lab         | 0.5887841   |
| Los Alamos National Laboratory        | 0.5887841   |
| Pacific Northwest National Laboratory | 0.2000000   |

```
# Berechne die Eigenvector-Centrality und sortiere sie absteigend
eigenvector_centrality <- evcent(g_competitors_zentralität, directed = FALSE)$vector

# Zusammenhang zwischen durchschnittliches Gehalt und Eigenvector-Zentralität
salary_eigenvector <- data.frame(
  Salary = salary_data$AverageSalary,
  Eigenvector = eigenvector_centrality[match(salary_data$`Company Name`, names(eigenvector_centrality))
)

# Entferne NA-Werte, die durch fehlende Übereinstimmungen entstehen könnten
salary_eigenvector <- na.omit(salary_eigenvector)

# Scatterplot mit roter, gestrichelter Trendlinie
ggplot(salary_eigenvector, aes(x = Salary, y = Eigenvector)) +
  geom_point() +
  geom_smooth(method = "loess", se = FALSE, color = "red", linetype = "dashed") +
  labs(title = "Scatterplot of Salary vs. Eigenvector Centrality",
```

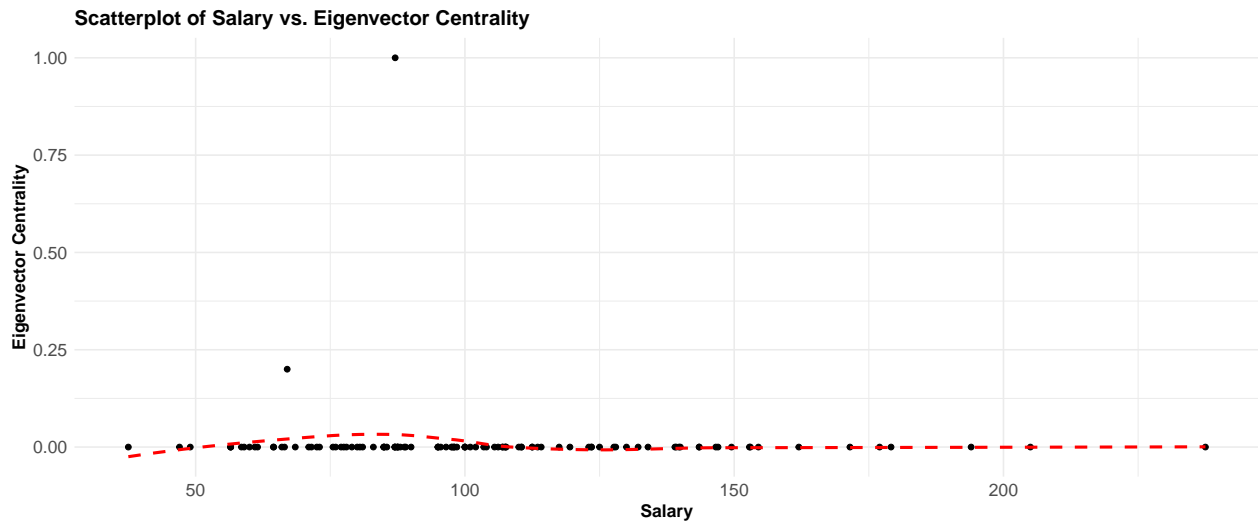


```

    x = "Salary", y = "Eigenvector Centrality") +
  theme_minimal() +
  theme(
    plot.title = element_text(size = 14, face = "bold"),
    axis.title = element_text(size = 12, face = "bold"),
    axis.text = element_text(size = 12)
  )

```

```
## `geom_smooth()` using formula = 'y ~ x'
```



### 3.4 Cluster-Analyse

Die Anwendung von Community-Detection-Methoden, wie beispielsweise Louvain oder Walktrap, ermöglicht die Identifikation von Clustern von Unternehmen, die sich durch einen besonders engen Wettbewerb auszeichnen.

Im Rahmen der Klassifizierung und Visualisierung erfolgt eine farbliche Markierung der identifizierten Cluster im Netzwerk, um eine bessere Übersicht über die Wettbewerbsgruppen zu gewinnen.

Im Rahmen der Untersuchung wird ermittelt, ob Cluster existieren, in denen besonders hohe oder niedrige Gehälter überwiegen, und ob diese Cluster eine gemeinsame Markt- oder Branchenstruktur aufweisen.

#### 3.4.1 Clustering Identifikation

```

# Louvain Community Detection
louvain_community <- cluster_louvain(g_direct_competitors, resolution = 0.1)

# Walktrap Community Detection
walktrap_community <- cluster_walktrap(g_direct_competitors, steps = 2)

# Wähle Louvain als Community-Detection-Methode
communities <- louvain_community

# Füge die Community-Informationen zu den Knoten des Graphen hinzu
V(g_direct_competitors)$community <- membership(communities)

# Erstelle eine Farbpalette für die Communities

```

```

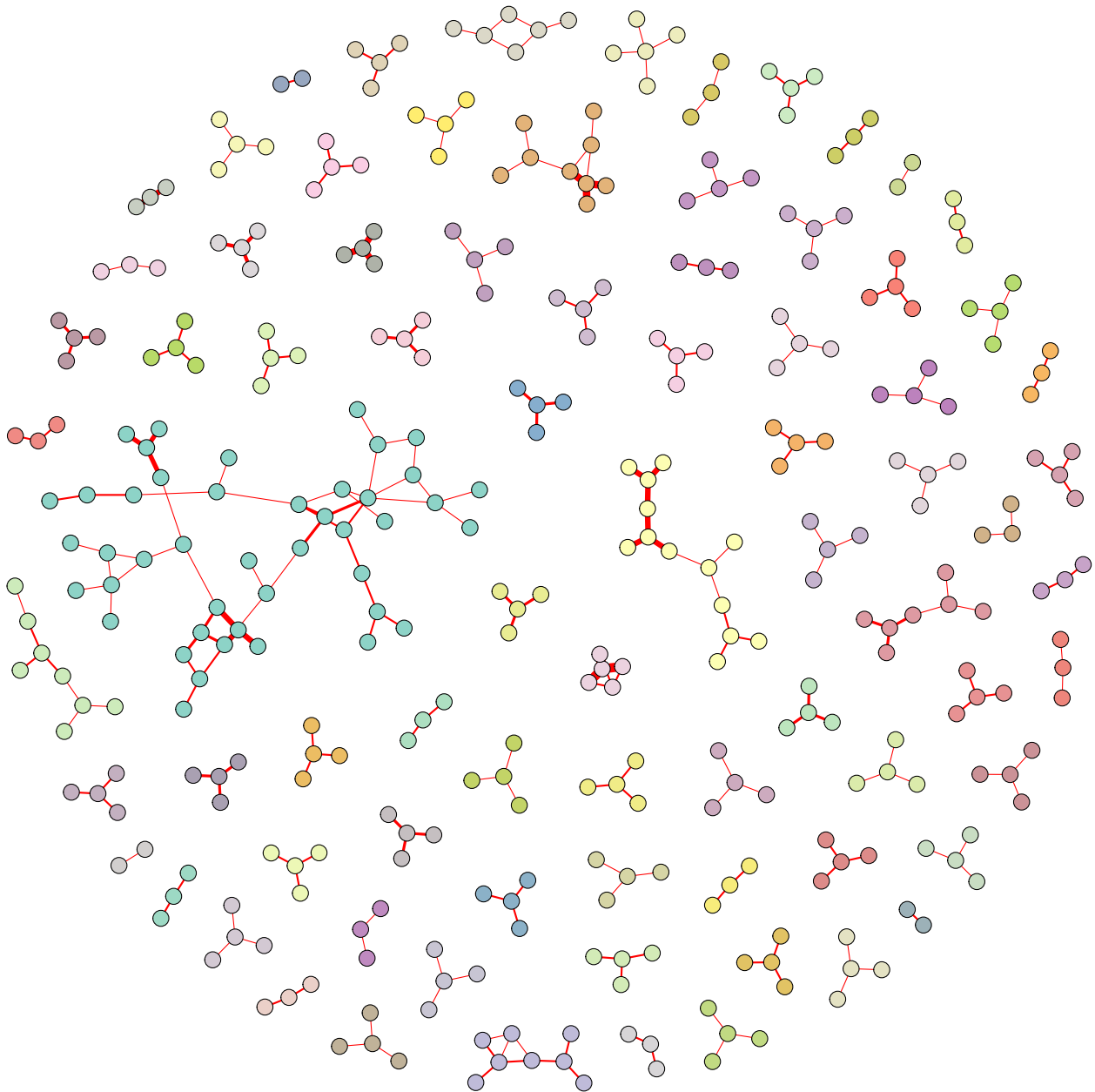
num_communities <- length(unique(V(g_direct_competitors)$community))
if (num_communities > 12) {
  community_colors <- colorRampPalette(brewer.pal(12, "Set3"))(num_communities)
} else {
  community_colors <- brewer.pal(num_communities, "Set3")
}

# Weise die Farben den Knoten basierend auf ihrer Community zu
V(g_direct_competitors)$color <- community_colors[V(g_direct_competitors)$community]

# Visualisiere das Netzwerk mit den Community-Farben
plot(g_direct_competitors, vertex.label = NA,
     vertex.size = 3, # Kleinere Knoten
     edge.width = 1 * E(g_direct_competitors)$weight, # Reduzierte Gewichtung der Kanten
     edge.arrow.size = 1,
     layout = layout_with_fr,
)
title(main = "Unternehmensnetzwerk mit Community-Detection (Louvain)", cex.main = 2)

```

## Unternehmensnetzwerk mit Community-Detection (Louvain)



```
# Berechne die Größe der Cluster
cluster_sizes <- table(membership(communities))

# Sortiere die Cluster nach Größe absteigend
sorted_clusters <- sort(cluster_sizes, decreasing = TRUE)

# Erstelle ein DataFrame mit den Cluster-Informationen
cluster_info <- data.frame(
  Cluster = names(sorted_clusters),
  Size = as.numeric(sorted_clusters),
  stringsAsFactors = FALSE
```

```

)

# Ausgeben wie viele Cluster es gibt
print(paste("Anzahl der Cluster:", length(cluster_info$Cluster)))

## [1] "Anzahl der Cluster: 79"

# Auswahl der 7 größten Cluster
top_clusters <- head(sorted_clusters, 7)

# Ausgabe der 7 größten Cluster
top_clusters_df <- data.frame(
  Cluster = names(top_clusters),
  Size = as.numeric(top_clusters),
  stringsAsFactors = FALSE
)

# Erstelle die Tabelle und zentriere sie links
kable(top_clusters_df, format = "latex", booktabs = TRUE, align = "l") %>%
  kable_styling(latex_options = c("striped", "hold_position"),
    position = "left")

```

| Cluster | Size |
|---------|------|
| 1       | 44   |
| 8       | 13   |
| 35      | 9    |
| 5       | 8    |
| 15      | 8    |
| 19      | 7    |
| 12      | 6    |

Da das Netzwerk sehr fragmentiert ist, werden im folgenden Schritt bloß die 7 größten Cluster betrachtet.

```

# Berechne die Größe der Cluster
cluster_sizes <- table(membership(communities))

# Sortiere die Cluster nach Größe absteigend
sorted_clusters <- sort(cluster_sizes, decreasing = TRUE)

# Auswahl der 7 größten Cluster
top_clusters <- head(sorted_clusters, 7)

# Erstelle eine Farbpalette für die Top 7 Cluster
top_cluster_ids <- names(top_clusters)
num_top_clusters <- length(top_cluster_ids)
top_cluster_colors <- brewer.pal(min(num_top_clusters, 12), "Set3")

# Weise die Farben den Knoten basierend auf ihrer Zugehörigkeit zu den Top 7 Clustern zu
V(g_direct_competitors)$color <- ifelse(V(g_direct_competitors)$community %in% top_cluster_ids,
  top_cluster_colors[match(V(g_direct_competitors)$community, top_cluster_ids)],
  "grey")

# Visualisiere das Netzwerk mit den Community-Farben
plot(g_direct_competitors, vertex.label = NA,

```

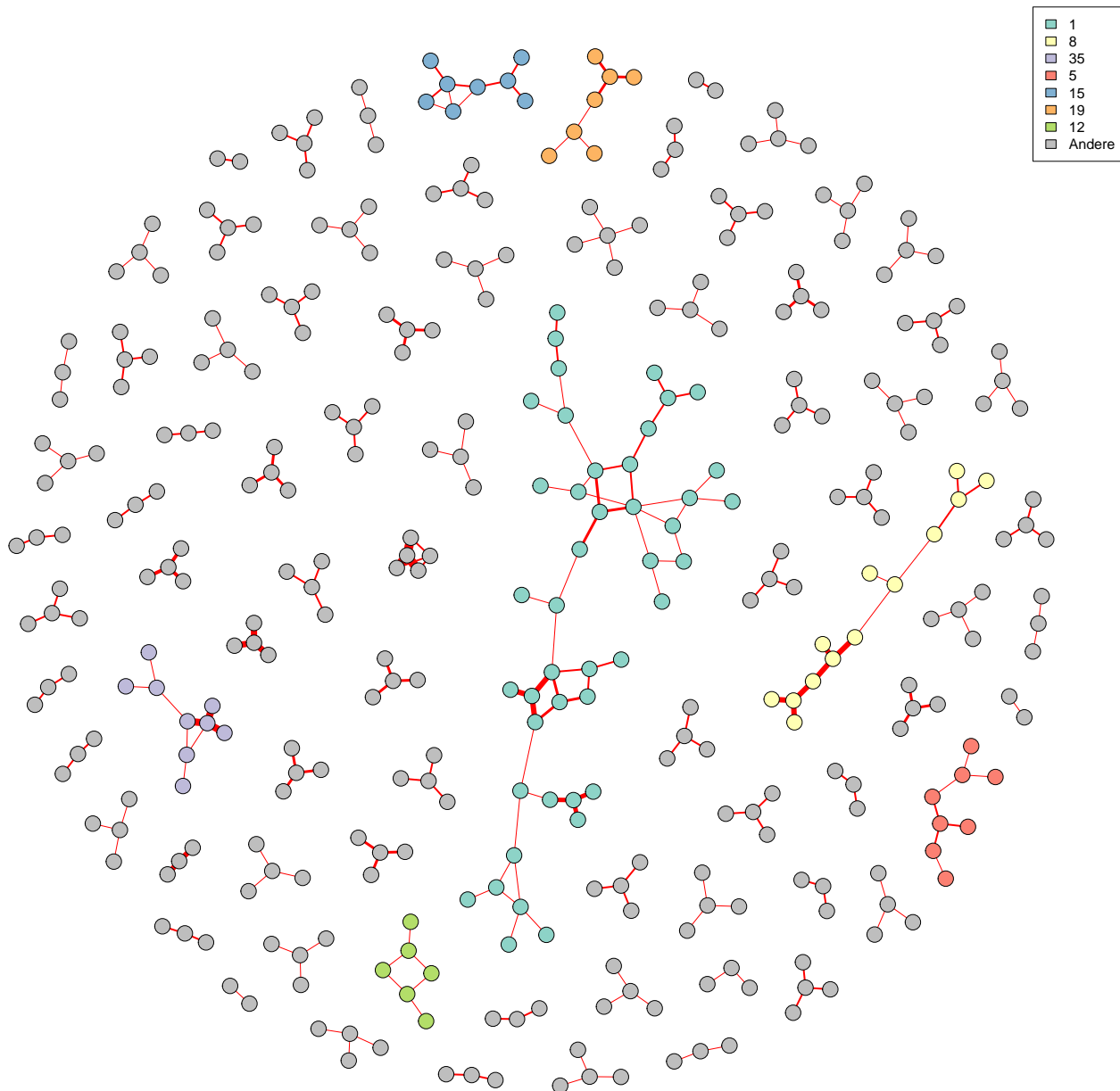
```

vertex.size = 3, # Kleinere Knoten
edge.width = 1 * E(g_direct_competitors)$weight, # Reduzierte Gewichtung der Kanten
edge.arrow.size = 1,
layout = layout_with_fr
)
title(main = "Die 7 größten Cluster im Unternehmensnetzwerk", cex.main = 2)

# Legende für Cluster-Farben
legend("topright", legend = c(top_cluster_ids, "Andere"),
      fill = c(top_cluster_colors, "grey"))

```

## Die 7 größten Cluster im Unternehmensnetzwerk



### 3.4.2 Gehaltsunterschiede zwischen den Clustern

Der Boxplot zeigt die Verteilung der Durchschnittsgehälter in den Top 7 Clustern.

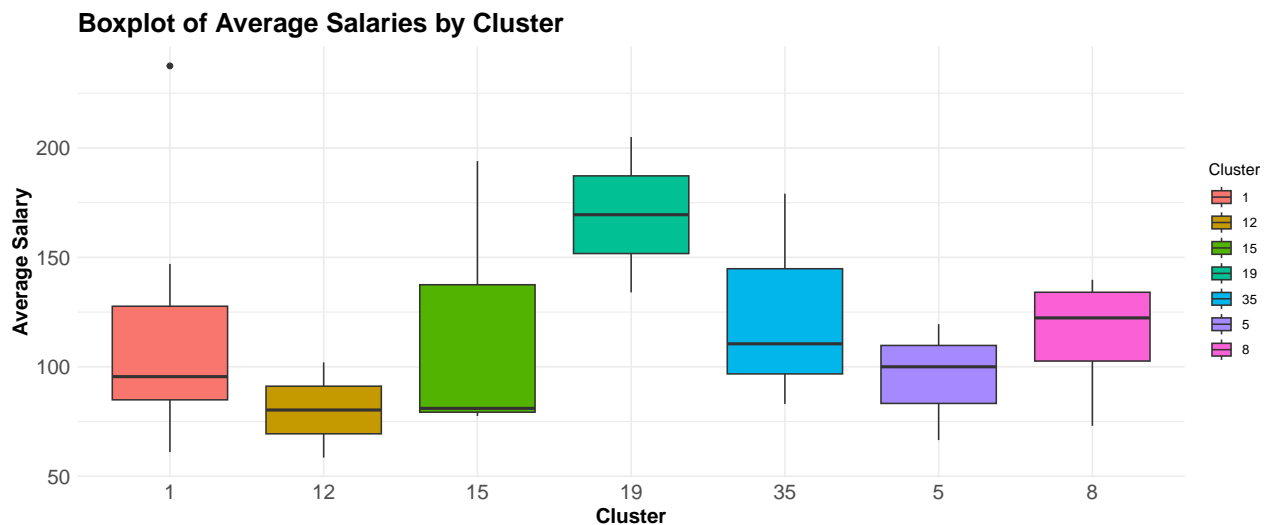
```
# Nimmt die top_cluster und fügt sie in ein Dataframe
top_clusters_df <- data.frame(
  Cluster = names(top_clusters),
  Size = as.numeric(top_clusters),
  stringsAsFactors = FALSE
)

# Erstelle eine leere Liste für die Durchschnittsgehälter der Cluster
cluster_salaries <- list()

# Berechne die Durchschnittsgehälter für jeden Cluster
for (cluster_id in names(top_clusters)) {
  cluster_members <- V(g_direct_competitors)$name[V(g_direct_competitors)$community == cluster_id]
  cluster_salaries[[cluster_id]] <- salary_data$AverageSalary[salary_data$`Company Name` %in% cluster_m
}

# Erstelle ein DataFrame mit den Durchschnittsgehältern der Cluster
cluster_salaries_df <- data.frame(
  Cluster = rep(names(top_clusters), sapply(cluster_salaries, length)),
  Salary = unlist(cluster_salaries),
  stringsAsFactors = FALSE
)

# Erstelle ein Boxplot der Gehälter nach Cluster
ggplot(cluster_salaries_df, aes(x = Cluster, y = Salary, fill = Cluster)) +
  geom_boxplot() +
  labs(title = "Boxplot of Average Salaries by Cluster",
       x = "Cluster", y = "Average Salary") +
  theme_minimal() +
  theme(
    plot.title = element_text(size = 18, face = "bold"),
    axis.title = element_text(size = 14, face = "bold"),
    axis.text = element_text(size = 14)
  )
)
```



### 3.4.3 Branchenstruktur der Cluster

Die Branchenzugehörigkeit der Unternehmen in den Top 7 Clustern wird im folgenden Schritt untersucht, um festzustellen, ob die Cluster eine gemeinsame Markt- oder Branchenstruktur aufweisen.

```
# Erstelle eine leere Liste für die Branchen der Cluster
cluster_industries <- list()

# Extrahiere die Branchen der Unternehmen in den Clustern
for (cluster_id in names(top_clusters)) {
  cluster_members <- V(g_direct_competitors)$name[V(g_direct_competitors)$community == cluster_id]
  cluster_industries[[cluster_id]] <- data$Industry[data$`Company Name` %in% cluster_members]
}

# Erstelle Tabellen für jedes Cluster
for (cluster_id in names(cluster_industries)) {
  # Berechne die Anzahl der einzigartigen Branchen in jedem Cluster
  industry_counts <- as.data.frame(table(cluster_industries[[cluster_id]]))
  colnames(industry_counts) <- c("Industry", "Count")

  # Erstelle eine Tabelle für das aktuelle Cluster
  cat(paste0("**Cluster ", cluster_id, "**\n\n"))
  kable_output <- kable(industry_counts, format = "latex", booktabs = TRUE, align = "l") %>%
    kable_styling(latex_options = c("striped", "hold_position"), position = "left")
  print(kable_output)
}
```

#### Cluster 1

| Industry                                | Count |
|---|-------|
| Aerospace & Defense                     | 2     |
| Biotech & Pharmaceuticals               | 2     |
| Computer Hardware & Software            | 2     |
| Consulting                              | 4     |
| Enterprise Software & Network Solutions | 7     |
| Federal Agencies                        | 5     |
| IT Services                             | 12    |

#### Cluster 8

| Industry                           | Count |
|------------------------------------|-------|
| Biotech & Pharmaceuticals          | 14    |
| Health Care Products Manufacturing | 1     |

#### Cluster 35

| Industry           | Count |
|--------------------|-------|
| Insurance Carriers | 10    |

#### Cluster 5

| Industry                         | Count |
|----------------------------------|-------|
| Health Care Services & Hospitals | 1     |
| Insurance Carriers               | 3     |

#### Cluster 15

| Industry                                | Count |
|---|-------|
| Enterprise Software & Network Solutions | 2     |
| Industrial Manufacturing                | 3     |

#### Cluster 19

| Industry                | Count |
|-------------------------|-------|
| Advertising & Marketing | 1     |
| Information Technology  | 3     |

#### Cluster 12

| Industry           | Count |
|--------------------|-------|
| Insurance Carriers | 2     |

### 3.4.4 Errechnung der durchschnittlichen Gehälter pro Branche

Im folgenden Schritt wird die durchschnittliche Gehaltsstruktur pro Branche in den Top 7 Clustern analysiert, um festzustellen, ob bestimmte Branchen tendenziell höhere oder niedrigere Gehälter anbieten. Dafür wird die Anzahl wie oft ein Unternehmen einer Branche in einem Cluster vorkommt berechnet und anschließend das Durchschnittsgehalt pro Branche ermittelt.

```
# Lade das Paket "viridis" für eine bessere Farbpalette
library(viridis)

## Loading required package: viridisLite

# Erstelle eine leere Liste für die Durchschnittsgehälter pro Branche
industry_salaries <- list()

# Extrahiere die Branchen und Gehälter der Unternehmen in den Clustern
for (cluster_id in names(top_clusters)) {
  cluster_members <- V(g_direct_competitors)$name[V(g_direct_competitors)$community == cluster_id]
  cluster_data <- data[data$`Company Name` %in% cluster_members, ]

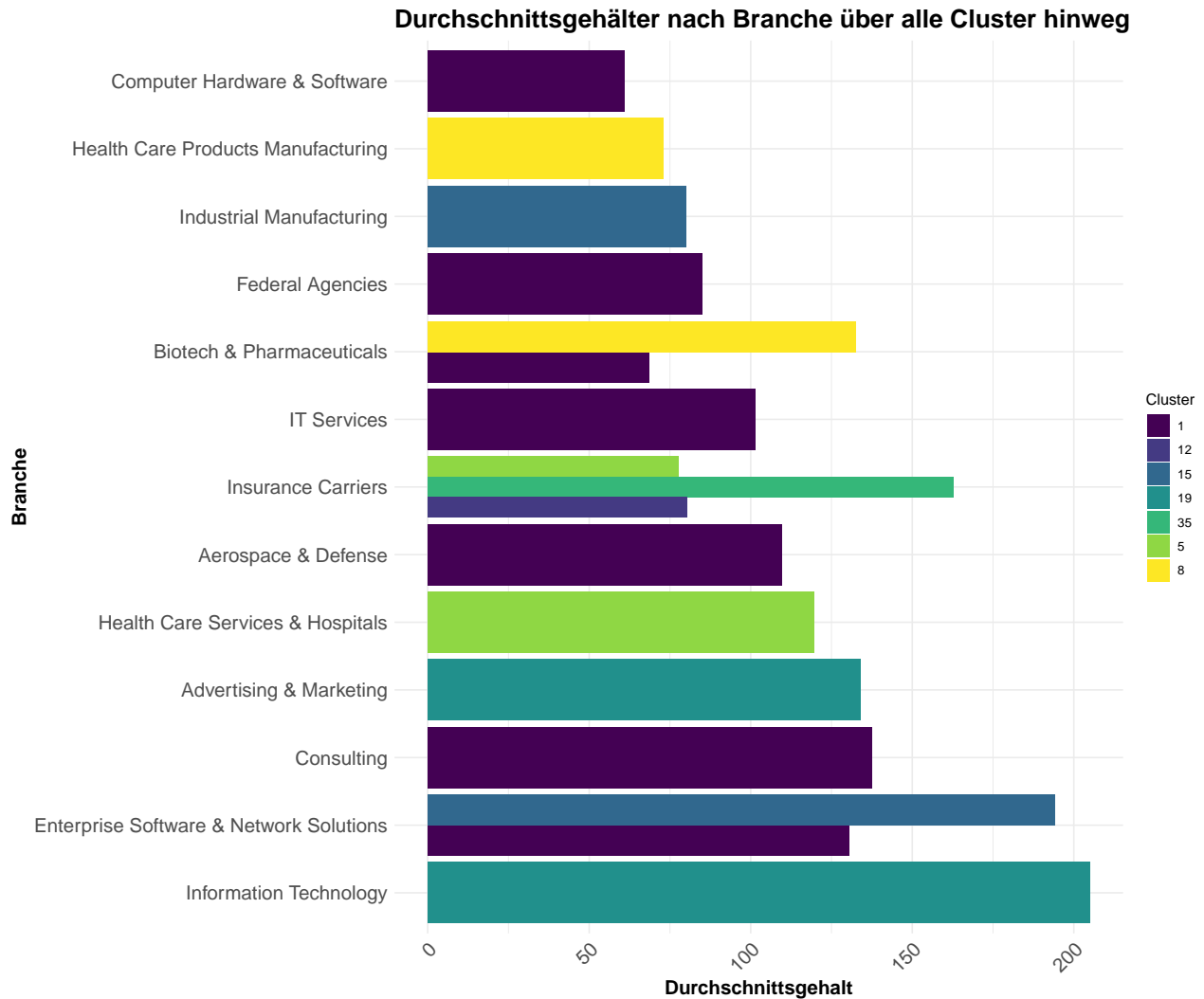
  # Berechne die Durchschnittsgehälter pro Branche
  industry_salaries[[cluster_id]] <- cluster_data %>%
    group_by(Industry) %>%
    summarise(AverageSalary = mean(`Salary Estimate`, na.rm = TRUE)) %>%
    mutate(Cluster = cluster_id) %>%
    arrange(desc(AverageSalary))
}

# Kombiniere die Daten aus allen Clustern
combined_industry_salaries <- bind_rows(industry_salaries)

# Visualisiere die Durchschnittsgehälter pro Branche über alle Cluster hinweg
ggplot(combined_industry_salaries, aes(x = reorder(Industry, -AverageSalary), y = AverageSalary, fill =
  geom_bar(stat = "identity", position = "dodge") +
  scale_fill_viridis_d() +
  labs(title = "Durchschnittsgehälter nach Branche über alle Cluster hinweg",
        x = "Branche", y = "Durchschnittsgehalt") +
  theme_minimal() +
  theme(
    plot.title = element_text(size = 18, face = "bold"),
```



```
axis.title = element_text(size = 14, face = "bold"),
axis.text = element_text(size = 14),
axis.text.x = element_text(angle = 45, hjust = 1)
) +
coord_flip() # Optional: Drehe das Diagramm für bessere Lesbarkeit
```



## 4 Abschließende Betrachtung mittels interaktiver Visualisierung

Im Folgenden wird dem Leser eine interaktive Visualisierung präsentiert, welche ihm die Möglichkeit bietet, die Ergebnisse der Wettbewerbsanalyse eigenständig zu explorieren, bevor die Schlussfolgerungen und Empfehlungen dargelegt werden.

```
closeness_centrality <- closeness(g_direct_competitors)
clustering_coeff <- transitivity(g_direct_competitors, type = "local")
```

```
# Bereite die Daten für visNetwork vor
nodes <- data.frame(id = V(g_direct_competitors)$name,
                    label = V(g_direct_competitors)$name,
                    group = membership(communities),
                    value = degree_centrality,
```

```

        title = paste("Degree:", degree centrality,
                      "<br>Betweenness:", betweenness centrality,
                      "<br>Closeness:", closeness centrality,
                      "<br>Eigenvector:", eigenvector centrality,
                      "<br>Average Salary:", V(g_direct_competitors)$salary),
        salary = V(g_direct_competitors)$salary)

edges <- data.frame(from = as.character(edges$from), to = as.character(edges$to))

# Erstelle eine Farbpalette basierend auf den Gehältern mit RColorBrewer
num_colors <- 9
color_palette <- brewer.pal(num_colors, "YlOrRd")
nodes$color <- color_palette[cut(nodes$salary, breaks = num_colors, labels = FALSE)]

# Erstelle die interaktive Netzwerkvisualisierung
network <- visNetwork(nodes, edges) %>%
  visOptions(highlightNearest = TRUE, nodesIdSelection = TRUE) %>%
  visGroups(groupname = "1", color = "red") %>%
  visGroups(groupname = "2", color = "blue") %>%
  visGroups(groupname = "3", color = "green") %>%
  visGroups(groupname = "4", color = "yellow") %>%
  visGroups(groupname = "5", color = "purple") %>%
  visGroups(groupname = "6", color = "orange") %>%
  visGroups(groupname = "7", color = "pink") %>%
  visLayout(randomSeed = 123) %>%
  visLegend() %>%
  visNodes(color = list(highlight = "orange"))

# Speichere die interaktive Visualisierung als HTML-Datei
saveWidget(network, "network.html", selfcontained = TRUE)
# Fügt ein Bild der interaktiven Netzwerkvisualisierung hinzu
# knitr::include_graphics("interaktive_Netzwerke_Bilder/Übersicht.png")

# knitr::include_graphics("interaktive_Netzwerke_Bilder/NVDIA.png")

```

Die Ausführung des Codes resultiert in der Speicherung der interaktiven Visualisierung unter dem Namen “network.html” im Arbeitsverzeichnis. Diese kann aufgerufen werden, um explorativ die Ergebnisse zu untersuchen.

## 5 Conclusion

Zusammenfassung der Ergebnisse: Fasse die zentralen Erkenntnisse zur Rolle des Wettbewerbs im Gehaltsgefüge von Data Science-Positionen zusammen, wie etwa: „Unternehmen in Clustern mit hoher Degree-Zentralität bieten im Durchschnitt 15 % höhere Gehälter als isolierte Unternehmen.“

Bedeutung der Ergebnisse: Diskutiere die Bedeutung der Wettbewerbsstruktur und welche Unternehmen durch ihre Position im Netzwerk von Wettbewerbsvorteilen und Talentrekrutierung profitieren.

Praktische Empfehlungen: Leite Handlungsempfehlungen für Berufseinsteiger und Unternehmen ab: Wo und wie sollten Fachkräfte ihre Karrieresuche fokussieren? Welche Unternehmen sollten strategische Investitionen in Gehälter und Talentförderung in Erwägung ziehen?

Ausblick: Weiterreichende Untersuchung des Zusammenhangs zwischen dem Einfluss von. Im Rahmen dieser Arbeit wurde bereits ein Schritt in diese Richtung getan(siehe Anhang)

## 6 Anhang

### 6.1 Geografische Betrachtung

Da bei der Betrachtung der Wettbewerbsstruktur die geografische Nähe von Unternehmen auch eine Rolle spielen kann, soll zunächst ein Netzwerk erstellt werden, das auf der geografischen Nähe von Unternehmen basiert. Diese Annahme beruht darauf, dass Unternehmen in derselben Region wahrscheinlich ähnliche Gehälter anbieten. Dies soll überprüft werden, um diese Arbeit um eine weitere Dimension zu erweitern.

#### 6.1.1 Erstellung eines Geografischen Netzwerkes

Die Gewichtung erfolgt linear, wobei jeder Standort eine Grundgröße von 3 hat, und für jedes Unternehmen an diesem Standort wird die Größe um 0.5 erhöht. Ab einer Größe von 4.5 wird die Farbe des Standorts geändert, um die Standorte mit mehreren Unternehmen hervorzuheben.

```
# Aus Gründen der Sichtbarkeit, werden bloß Locations mit mehr als einem
# Unternehmen dargestellt.

# Extract relevant columns for geographic visualization
edges_geo <- data %>%
  select(Company = `Company Name`, Location = `Location`) %>%
  distinct()

# Calculate the number of companies per location and filter for locations
# with more than one company
location_counts <- edges_geo %>%
  group_by(Location) %>%
  summarise(Company_Count = n()) %>%
  filter(Company_Count > 1) # Keep only locations with more than one company

# Filter edges to include only connections for locations with more than
# one company
filtered_edges <- edges_geo %>%
  filter(Location %in% location_counts$Location)

# Create an igraph object for geographic visualization
network_geo <- graph_from_data_frame(filtered_edges, directed = FALSE)

# Set vertex colors based on whether the node is a company or a location
company_colors <- "blue"
location_colors <- rainbow(nrow(location_counts))

# Set vertex size based on the number of companies at each location
vertex_sizes <- ifelse(V(network_geo)$name %in% location_counts$Location,
  3 + location_counts$Company_Count[
    match(V(network_geo)$name, location_counts$Location)
  ] * 0.5, # Linear scaling factor with minimum size 3
  3) # Default size for companies

# Assign colors and sizes to vertices
V(network_geo)$size <- vertex_sizes
V(network_geo)$color <- ifelse(V(network_geo)$name %in% location_counts$Location &
  vertex_sizes > 4.5,
  location_colors[match(V(network_geo)$name, location_counts$Location)],
  "grey")
```

```

# Plot the network
plot(network_geo,
      vertex.label = NA, # Remove labels from the plot
      vertex.size = V(network_geo)$size,
      vertex.color = V(network_geo)$color,
      edge.arrow.size = 0.3,
      layout = layout_with_fr,
)

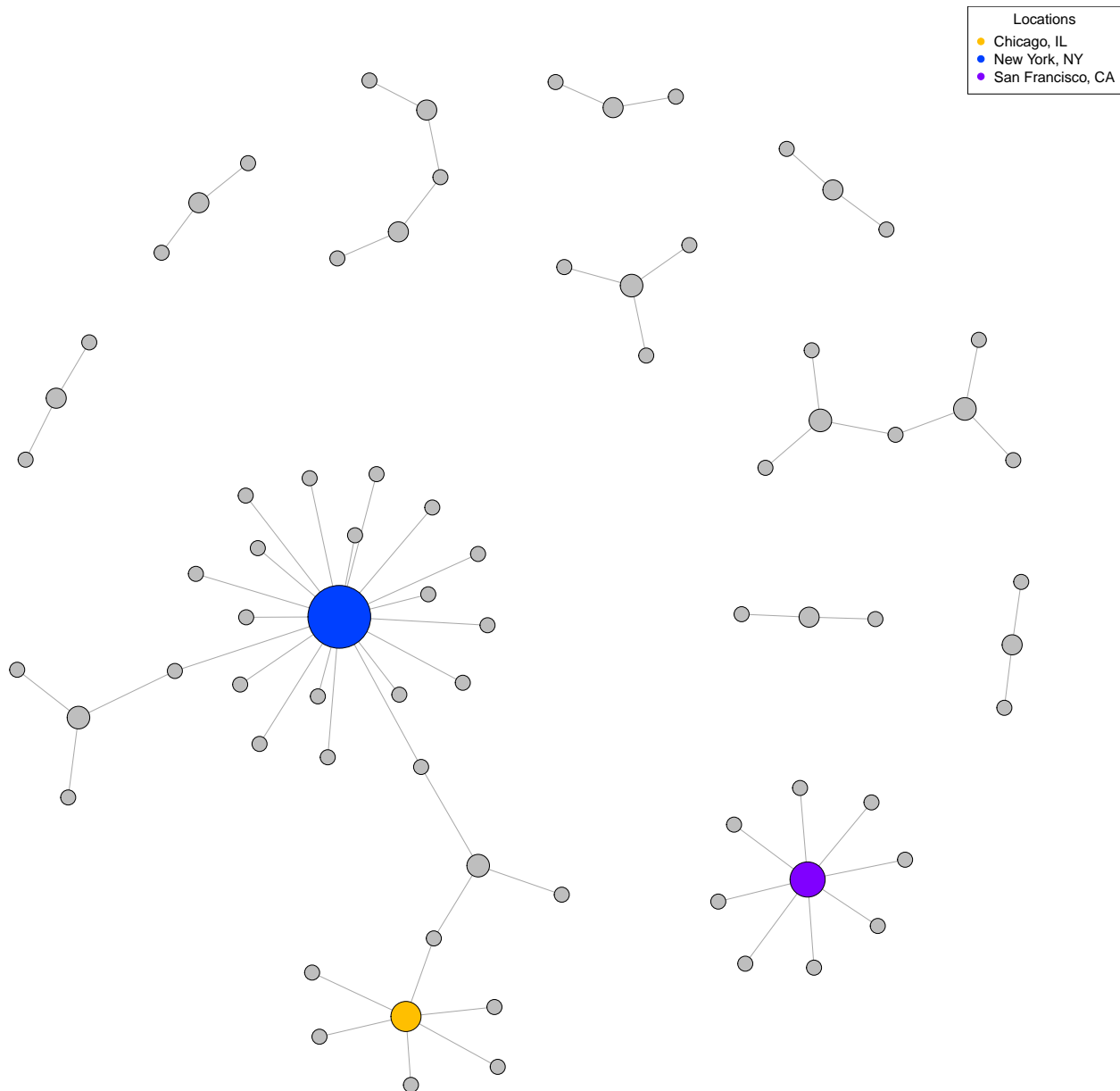
# Add legend for locations with size > 4.5
location_indices <- match(location_counts$Location, V(network_geo)$name)
large_locations <- location_counts$Location[vertex_sizes[location_indices] > 4.5]

large_location_colors <- location_colors[
  match(large_locations, location_counts$Location)
]
legend("topright",
      legend = large_locations,
      col = large_location_colors,
      pch = 19,
      title = "Locations")

title(main = "Geografisches Netzwerk basierend auf Unternehmensstandorten", cex.main = 2)

```

## Geografisches Netzwerk basierend auf Unternehmensstandorten



Wie zu erwarten war, sind die meisten Unternehmen in Ballungszentren wie New York, Chicago und San Francisco angesiedelt.

### 6.1.2 Vergleich der Gehälter zwischen den Hotspot- und den anderen Regionen

```
# Ausgabe der farbigen Standorte  
print(large_locations)
```

```
## [1] "Chicago, IL"      "New York, NY"     "San Francisco, CA"
```

```
# Filterung der Daten für die Hotspot-Regionen  
data_hotspots <- data %>%  
  filter(`Location` %in% large_locations)
```

```

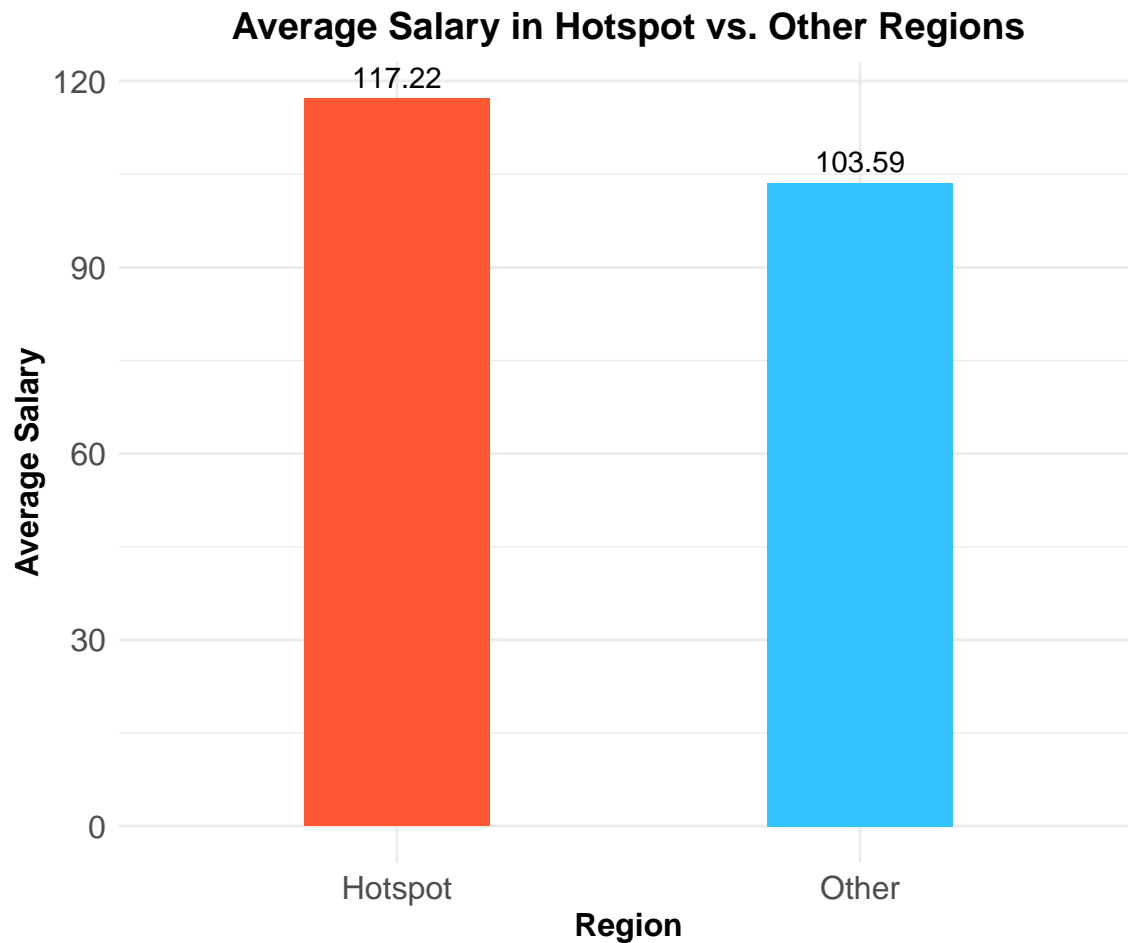
# Filterung der Daten für die anderen Regionen
data_other <- data %>%
  filter(!`Location` %in% large_locations)

# Durchschnittsgehalt in den Hotspot-Regionen
avg_salary_hotspots <- mean(data_hotspots$`Salary Estimate`, na.rm = TRUE)

# Durchschnittsgehalt in den anderen Regionen
avg_salary_other <- mean(data_other$`Salary Estimate`, na.rm = TRUE)

# Erstellung eines Balkendiagramms
ggplot(data = data.frame(Region = c("Hotspot", "Other"),
                          Average_Salary = c(avg_salary_hotspots,
                                              avg_salary_other)),
       aes(x = Region, y = Average_Salary, fill = Region)) +
  geom_bar(stat = "identity", width = 0.4) +
  scale_fill_manual(values = c("Hotspot" = "#FF5733", "Other" = "#33C3FF")) +
  labs(title = "Average Salary in Hotspot vs. Other Regions",
       x = "Region",
       y = "Average Salary") +
  theme_minimal() +
  theme(
    plot.title = element_text(hjust = 0.5, size = 14, face = "bold"),
    axis.title.x = element_text(size = 12, face = "bold"),
    axis.title.y = element_text(size = 12, face = "bold"),
    axis.text.x = element_text(size = 12),
    axis.text.y = element_text(size = 12),
    legend.position = "none"
  ) +
  geom_text(aes(label = round(Average_Salary, 2)), vjust = -0.5, size = 4)

```



```
# Berechnung der Gehaltsunterschiede
salary_diff <- avg_salary_hotspots - avg_salary_other

# Ausgabe der Gehaltsunterschiede
print(paste("Durchschnittsgehalt in Hotspot-Regionen:", avg_salary_hotspots))

## [1] "Durchschnittsgehalt in Hotspot-Regionen: 117.223880597015"
print(paste("Durchschnittsgehalt in anderen Regionen:", avg_salary_other))

## [1] "Durchschnittsgehalt in anderen Regionen: 103.59"
print(paste("Durchschnittlicher Gehaltsunterschied:", salary_diff))

## [1] "Durchschnittlicher Gehaltsunterschied: 13.6338805970149"
```

Das Ergebnis zeigt, dass entsprechend der vorher getroffenen Annahme, die Gehälter in den Hotspot-Regionen im Durchschnitt höher sind als in anderen Regionen. Dies impliziert eine Korrelation zwischen geografischer Nähe und Gehaltsniveau. Eine mögliche Erklärung hierfür könnte die höhere Lebenshaltungskosten in Ballungszentren sein, die höhere Gehälter erforderlich machen.



## 7 Literaturverzeichnis

Davenport, Thomas H.; Patil, D. J. 2012. »Data Scientist: The Sexiest Job of the 21st Century«, in Harvard Business Review vom 1. Oktober 2012. <https://hbr.org/2012/10/data-scientist-the-sexiest-job-of-the-21st-century> (Zugriff vom 30.10.2024).

Google Trends, <https://trends.google.com/trends/explore?date=all&q=%22data%20science%22,%22data%20scientist%22> (Zugriff vom 30.10.2024).