

Social Network Analysis über die subreddit Veranstaltung /r/Place2023

Mehmet Karaca

27.10.2023

Contents

Einleitung	1
Forschungsfragen:	2
Verwendete Librarys	2
Datenaufbereitung	2
Datenanalyse und Auswertung	3
Erster Überblick über den Datensatz	3
Untersuchung der Heatmap der Koordinatenpaare	9
Untersuchung der Hauptakteure (Stehen die Hauptakteure in Beziehung zu einander?)	14
Untersuchung der Beziehung der Top10 User im Kontext der Farben	14
Untersuchung der Beziehung der Top10 User im Kontext der Koordinaten	17
Interpretation der Ergebnisse und Beantwortung der Forschungsfrage	22
Interpretation der Ergebnisse	22
Beantwortung der Forschungsfrage	22
Fazit	22

Einleitung

Auf der Social Media “Reddit” gibt es sogenannte Subreddits, welche bestimmte Foren zu bestimmten Themen sind. Eines dieser Subreddits ist das Subreddit “Place”. Die Abkürzung /r/ steht als Abkürzung für ein Subreddit. Dieses bestimmte subreddit ist ein besonderes, da es nur für einen bestimmten Zeitraum existiert und jährlich stattfindet. Das erste Mal fand das Event 2017 statt und war als Aprilscherz gedacht. Am 1. April 2022 wurde das Event wiederholt und fand zum zweiten Mal statt. Die letzte Veranstaltung fand vom 20.07.2023 bis zum 25.07.2023 statt.

Des Weiteren gibt es eine weitere Besonderheit. In der Regel ist es in subreddits nur möglich Beiträge in Form von Bildern, Videos oder Texten zu posten. In diesem Subreddit ist es möglich Pixel auf einer Leinwand zu platzieren. Dabei können die Nutzer aus einer Palette an Farben auswählen. Allerdings können die Nutzer die Pixel nur nach Ablauf einer bestimmten Zeit setzen. Diese Zeit beträgt zwischen 5 und

20 Minuten. Dadurch ist schwierig als einzelner Nutzer ein Bild zu erstellen, da andere Nutzer die Pixel überschreiben können. Es ist Teamwork gefragt, um ein Bild zu erstellen. Dieses Subreddit ist sozusagen soziales Experiment, welches die Nutzer dazu anregen soll, gemeinsam etwas zu erstellen. Die Nutzer können sich absprechen und gemeinsam ein Bild erstellen. Dieses Bild kann ein Logo, eine Flagge oder ein Muster sein.

Für diese Arbeit wurde der Datensatz der letzten Veranstaltung aus dem Jahr 2023 verwendet. Die Daten wurden von der Webseite https://www.reddit.com/r/place/comments/15bjm5o/rplace_2023_data/ heruntergeladen.

Die Motivation für die Arbeit ist, dass es interessant ist zu untersuchen, wie die Nutzer sich verhalten und wie sie sich absprechen. Außerdem ist es interessant zu untersuchen, wie die Beziehungen zwischen den Nutzern sind und wie die Nutzer sich verhalten.

Forschungsfragen:

- Sind Muster zu erkennen und in welchen Bereichen der Leinwand sind diese Muster zu erkennen?
- Welche Bereiche der Leinwand wurden besonders oft gefüllt?
- Welche Nutzer waren am aktivsten und welche Farben haben sie genutzt?
- Welche Bereiche haben die aktivsten Nutzer besonders oft gefüllt?
- Gibt es generell Beziehungen zwischen den Nutzern?

Verwendete Librarys

```
#Installieren der benötigten Packages

#install.packages("tidyverse")
#install.packages("ggraph")
#install.packages("ggplot2")

#Laden der benötigten Packages
library(tidyverse)
library(ggraph)
library(ggplot2)
```

Datenaufbereitung

Wie bereits erwähnt, wurde der Datensatz von der Webseite auf Reddit bereitgestellt. Die Daten liegen als .csv Datei vor und sind sehr groß. Die Verarbeitung der Daten ist sehr aufwendig. Aufgrund der Größe wurde der Datensatz auch nicht in Github oder in Moodle hochgeladen. Daher wurde ein Kompromiss nach Absprache mit der Dozentin getroffen. Es wird ein Ausschnitt des Datensatzes verwendet, um die Größe des Datensatzes zu reduzieren. Darüber hinaus wurde der Datensatz vorverarbeitet, um die Daten zu vereinfachen und die Verarbeitung zu beschleunigen. Der verarbeitete Datensatz ist in der Abgabe beigelegt, sodass die Durchführung des RMarkdowns möglich ist. Die Vorverarbeitung wurde mit Python durchgeführt. Hierfür wurde ein Skript geschrieben, welches die Daten vorverarbeitet. Das Skript ist in der Abgabe beigelegt. Die Vorverarbeitung wurde mit Python durchgeführt, da die Verarbeitung mit R sehr lange gedauert hat. Darüber hinaus wurde die Library Pandas verwendet, welche sehr gut für die Verarbeitung von Daten geeignet ist.

Die Vorgehensweise der Vorverarbeitung ist wie folgt:

1. Es wurde eine Liste erstellt, welche die betroffenen Dateien enthält.

2. Mithilfe einer Schleife wurde jede Datei aus der Liste geöffnet und in ein Dataframe umgewandelt.
3. Jedes der Dataframes wurde mit der Funktion standardize_data vorverarbeitet.
4. In der Funktion wurde die Spalte timestamp angepasst, sodass es die Uhrzeit als int Wert enthält.
5. Die Datensätze wurden gefiltert, sodass nur die Daten zwischen 00:00:00 und 15:59:59 enthalten sind.
6. Die Spalte pixel_color (Hexadezimalzahl) wurde angepasst, sodass es die Farben als int Wert enthält.
7. Fehlerwerte in den Spalten x und y wurden entfernt.
8. Die Spalten x und y wurden angepasst, sodass die Koordinaten als int Wert enthalten.
9. Die Werte in der Spalte user wurden pseudonymisiert, sodass die Nutzer anonym bleiben.
10. Der verarbeitete Datensatz wird zurückgegeben und alle Dataframes werden in einem zusammenhängende Dataframe gespeichert.
11. Die Informationen zu dem neuen Dataframe werden ausgegeben
12. Der fertige Dataframe wird als neue .csv Datei als 2023_place_canvas_20072023.csv gespeichert.

Datenanalyse und Auswertung

Dieser Datensatz wird nun in R geladen und für die weitere Analyse verwendet. Die Daten werden in einem Dataframe gespeichert und anschließend werden die Daten untersucht. Es werden erste Tests durchgeführt, um einen Überblick über die Daten zu bekommen. Hierzu werden libraries wie tidyverse, igraph, ggplot2 und dplyr verwendet. Anschließend werden die Daten visualisiert, um weitere Erkenntnisse zu gewinnen.

Erster Überblick über den Datensatz

```
#Laden der Daten von /Users/karaca/src/Social_Network_Analysis/Datensatz/
#2023_place_canvas_20072023.csv
data <- read.csv("Datensatz/2023_place_canvas_20072023.csv", sep = ",")

#csv Datei in ein Dataframe umwandeln
data <- as.data.frame(data)

#Anzeigen der ersten 6 Zeilen zum testen
#head(data)

#Anzeigen der Struktur der Daten
#str(data)

#install.packages("igraph")
library(igraph)

## 
## Attaching package: 'igraph'

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

```

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

#install.packages("tidyverse")
library(tidyverse)

## -- Attaching core tidyverse packages ----- tidyverse 2.0.0 --
## v dplyr     1.1.3     v readr     2.1.4
## vforcats   1.0.0     v stringr   1.5.0
## v ggplot2   3.4.4     v tibble    3.2.1
## v lubridate 1.9.3     v tidyrr    1.3.0
## v purrr     1.0.2

## -- Conflicts ----- tidyverse_conflicts() --
## x lubridate::%--%()     masks igraph::%--%()
## x dplyr::as_data_frame() masks tibble::as_data_frame(), igraph::as_data_frame()
## x purrr::compose()      masks igraph::compose()
## x tidyrr::crossing()    masks igraph::crossing()
## x dplyr::filter()       masks stats::filter()
## x dplyr::lag()          masks stats::lag()
## x purrr::simplify()     masks igraph::simplify()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors

#install.packages(ggplot2)
#library(ggplot2)

#install.packages("dplyr")
#library(dplyr)

```

Da der Datensatz geladen worden ist, führen wir nun erste Tests durch, um einen Überblick über die Daten zu bekommen.

```

#Zuerst lassen wir uns den höchsten und niedrigsten Wert von timestamp anzeigen
#, um einen Überblick über die Zeit zu bekommen.
max(data$timestamp)

## [1] 155959

min(data$timestamp)

## [1] 130026

#Als Nächstes schauen wir uns an, wie viele einzigartige Werte es in der Spalte user gibt.
#Dies gibt uns die Anzahl an Usern an, da manche User mehrere Einträge haben.
length(unique(data$user))

## [1] 739675

```

```

#Der Datensatz wurde im Zeitraum von 13:00:26 bis 15:59:59 aufgenommen.
#In diesem Zeitraum haben 739675 User insgesamt 2411941 Pixel gesetzt.
#Die Pixel Color wurde als int Wert gespeichert, durch die Vorverarbeitung
#des Datensatzes. Die Zahlen haben einen bestimmten Schlüssel für die Farbe.
#Tabelle der Farben
#rot = #ff4500 = 1
#orange = #ffa800 = 2
#gelb = #ffd635 = 3
#grün = #00a368 = 4
#blau = #3690ea = 5
#lila = #b44ac0 = 6
#schwarz = #000000 = 7
#weiß = #ffffff = 8

```

```

#Diese Farben nutzen wir um eine Farbzuordnungstabelle zu erstellen
farbzuordnung <- c("#ff4500", "#ffa800", "#ffd635", "#00a368",
                     "#3690ea", "#b44ac0", "#000000", "#ffffff")

```

Als Nächstes zeigen wir uns die Maximal- und Minimalwerte von x und y anzeigen, um einen Eindruck des Koordinatensystems zu bekommen.

```
max(data$x)
```

```
## [1] 499
```

```
min(data$x)
```

```
## [1] -500
```

```
max(data$y)
```

```
## [1] 499
```

```
min(data$y)
```

```
## [1] -500
```

Das Koordinatensystem erstreckt sich von -500 bis 499 entlang sowohl der X-Achse als auch der Y-Achse. Wie bereits erwähnt ist die Leinwand 1000x1000 Pixel groß und passt somit zu dem Koordinatensystem. Die Koordinate 500|500 wurde nicht verwendet, daher ergeben sich als Max-Werte 499|499. Der Nullpunkt 0|0 befindet sich in der Mitte des Koordinatensystems.

Als nächsten Schritt wollen wir die Daten mit einem Plot visualisieren, um einen Eindruck der Daten zu bekommen. Hierfür erstellen wir ein Subset, welches nur aus den Koordinatenpaaren besteht.

```

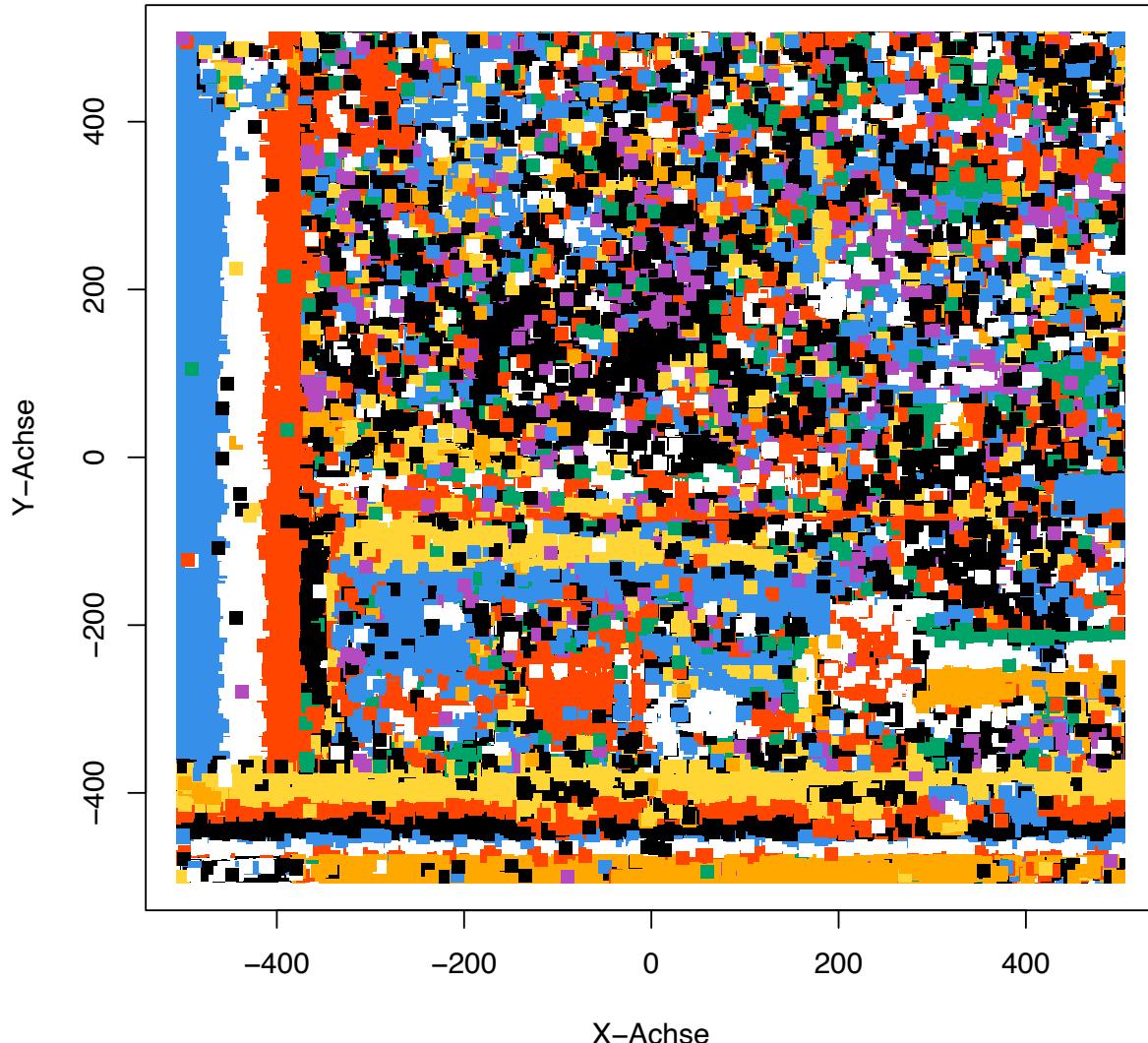
#Erstelle einen Subset, welcher nur aus den Koordinatenpaaren besteht
Koordinaten_subset_x_und_y <- data[,c("x", "y")]

#Diesen Subset wollen wir uns anschauen und plotten diesen in einem Scatterplot.
#plot(Koordinaten_subset_x_und_y$x, Koordinaten_subset_x_und_y$y, xlab = "X-Achse",
      #ylab = "Y-Achse")
#Aus Erkenntniss Gründen wurde der Plot auskommentiert, da dieser nur ein schwarzes
#Feld darstellt.

```

An dem Plot erkennt man, das beinahe alle Koordinaten mit einem Pixel belegt worden sind. Es gibt nur vereinzelte Ausnahmen, die nicht belegt worden sind. Aus dem Plot können keine tiefergehenden Erkenntnisse gezogen werden. Daher soll im nächsten Schritt die Punkte farbig visualisiert werden, um bestimmte Muster erkennen zu können. Hierzu erstellen wir einen neuen Subset, welcher aus den Koordinatenpaaren und dem pixelcolor besteht. Diesen subset wollen wir uns anschauen und plotten diesen auf einem Scatterplot mit der Farbe des pixel_color. Die Farben sind in der Tabelle der Farben oben beschrieben.

```
# Erstellen eines neuen Subsets,  
#welches aus den Koordinatenpaaren und dem pixelcolor besteht.  
Koordinaten_subset_x_y_pixelcolor <- data.frame(x=data$x, y=data$y,  
    pixel_color=data$pixel_color)  
  
#Diesen Subset wollen wir uns anschauen und plotten diesen auf einem Scatterplot  
#mit der Farbe des pixelcolors  
plot(Koordinaten_subset_x_y_pixelcolor$x, Koordinaten_subset_x_y_pixelcolor$y,  
    col = farbzuordnung[Koordinaten_subset_x_y_pixelcolor$pixel_color] ,pch=15,  
    xlab = "X-Achse", ylab = "Y-Achse")
```

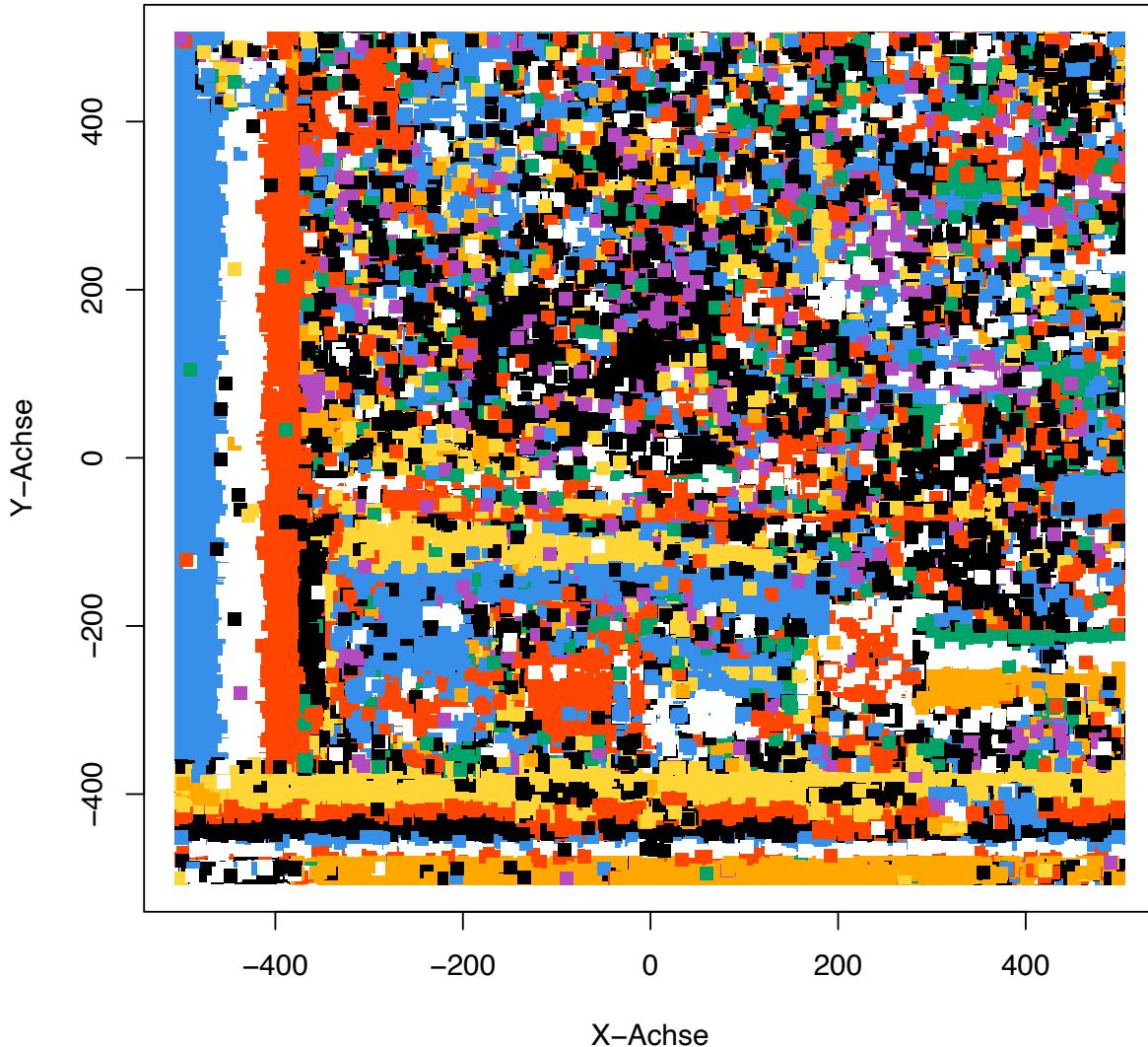


Durch diese Darstellung ist nun zu erkennen, dass es bestimmte Farbmuster gibt. Beispielsweise sind erste Flaggen zu erkennen wie die Französische (links), die Deutsche (links unten), türkische, indische, italienische. Allerdings haben wir ein Problem bei dieser Visualisierung, da viele Punkte doppelt vorkommen. Dies liegt daran, dass auf einem Pixel mehrere Pixel gesetzt worden sind. Deswegen möchten wir nur die letzten Pixel setzen lassen. Hierfür erstellen wir einen neuen Subset. In diesem subset gibt es keine doppelten Koordinatenpaare.

```
# Entfernen von doppelten Koordinatenpaaren, um nur die letzten Pixel zu behalten
Koordinaten_subset_eindeutig <-
Koordinaten_subset_x_y_pixelcolor[!duplicated(Koordinaten_subset_x_y_pixelcolor[, 
  c("x", "y")], fromLast = TRUE), ]

# Plot der eindeutigen Koordinatenpaare mit der Farbe des letzten pixel_color
plot(Koordinaten_subset_eindeutig$x, Koordinaten_subset_eindeutig$y,
```

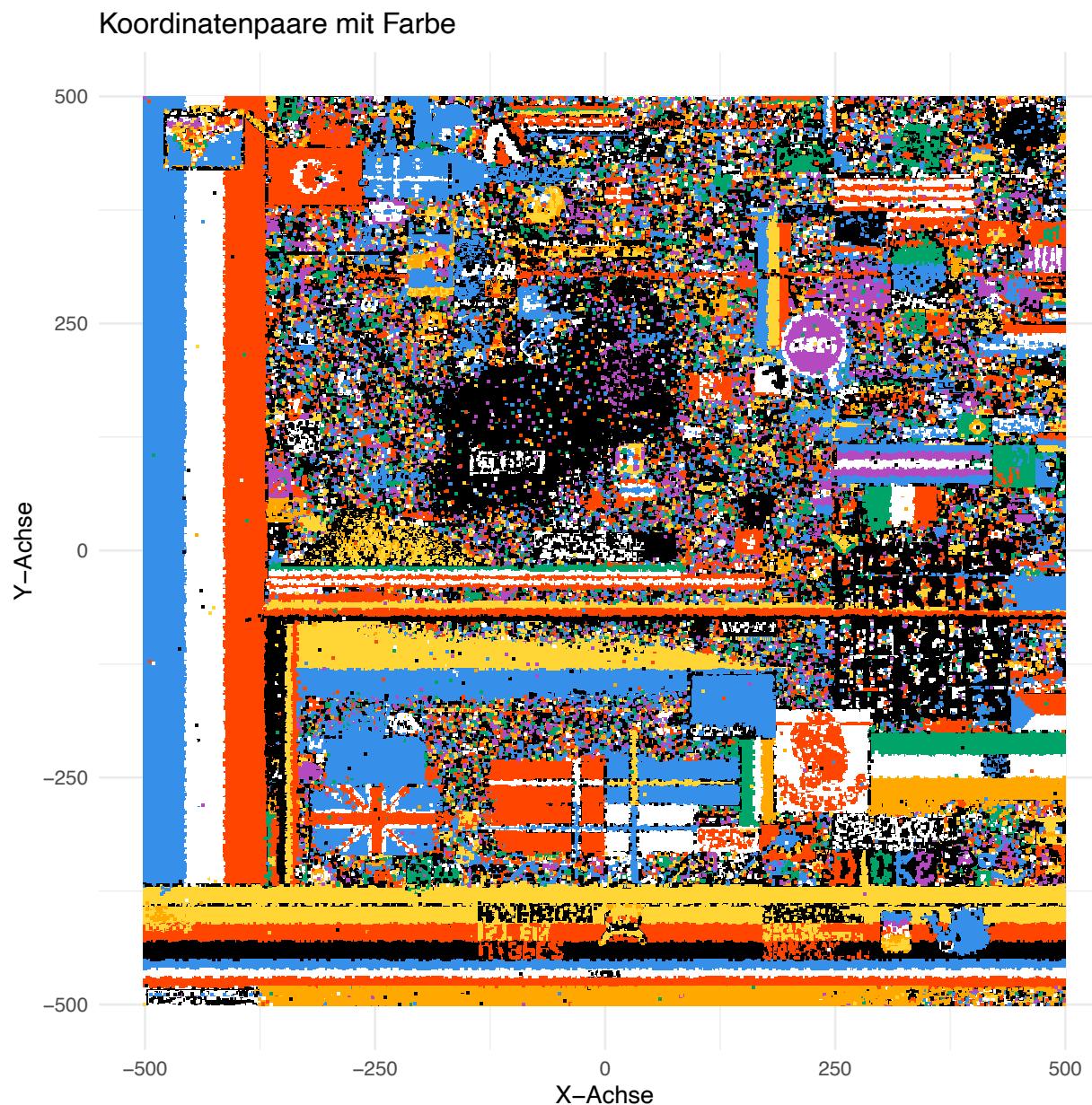
```
col = farbzuordnung[Koordinaten_subset_eindeutig$pixel_color], pch = 15,
xlab = "X-Achse", ylab = "Y-Achse")
```



Die Genauigkeit und Präzision der Grafik sind nun deutlich besser. Es sind keine doppelten Koordinatenpaare mehr vorhanden. Allerdings ist dies noch nicht perfekt. Beim Ausprobieren mit ggplot2 hat sich gezeigt, dass bei der Verwendung von ggplot2 die Grafik noch präziser wird. Daher wird im nächsten Schritt die Grafik mit ggplot2 geplottet.

```
library (ggplot2)
#Plotten der Grafik in ggplot
ggplot(Koordinaten_subset_eindeutig, aes(x, y, col = farbzuordnung[pixel_color])) +
  geom_point(shape = 15, size = 0.4) +
  scale_color_identity()
```

```
theme_minimal() +
  labs(title = "Koordinatenpaare mit Farbe", x = "X-Achse", y = "Y-Achse", col = "Farbe")
```



Diese Darstellung ist die präziseste Darstellung, die wir erstellen konnten. Es ist zu erkennen, dass die Flaggen deutlicher zu erkennen sind. Außerdem sind Sätze, Wörter und Logos zu erkennen. Aber auch Andeutung von Bildern wie z.B. ein blauer Elefant und ein "Pikachu" mit Sonnenbrille (Ein Pokemon aus dem gleichnamigen Spiel) sind der unteren rechten Ecke ist zu erkennen.

Untersuchung der Heatmap der Koordinatenpaare

Im folgenden Schritt soll die Quantität untersucht werden. Hierzu soll herausgefunden werden, welche Bereiche am meisten Pixel gesetzt bekommen haben. Es soll eine Heatmap erstellt werden, welcher die

Bereiche mit den meisten Färbungen anzeigt. Hierfür erstellen wir einen neuen Subset, welcher aus den Koordinatenpaaren besteht.

```
#Erstellen eines neuen Subsets, welches aus den Koordinatenpaaren besteht
Koordinaten_subset_x_y_heatmap <- data.frame(x=data$x, y=data$y)

#Füge eine neue Spalte hinzu, welcher die Koordinatenpaare zusammenfasst
Koordinaten_subset_x_y_heatmap <- transform(Koordinaten_subset_x_y_heatmap,
  x_y = paste(x, y, sep = "_"))

#Die doppelten Werte von x_y werden zusammengefasst und gezählt
#und in der Spalte count gespeichert und anschließend
#nach der Anzahl der Koordinatenpaare absteigend sortiert
Koordinaten_subset_x_y_heatmap <- Koordinaten_subset_x_y_heatmap %>% group_by(x_y) %>%
  summarise(count = n()) %>% arrange(desc(count))

#Nun sollen doppelte Koordinatenpaare x_y entfernt werden, um nur die eindeutigen
#Koordinatenpaare zu behalten
Koordinaten_subset_x_y_heatmap <- Koordinaten_subset_x_y_heatmap[
  !duplicated(Koordinaten_subset_x_y_heatmap$x_y), ]

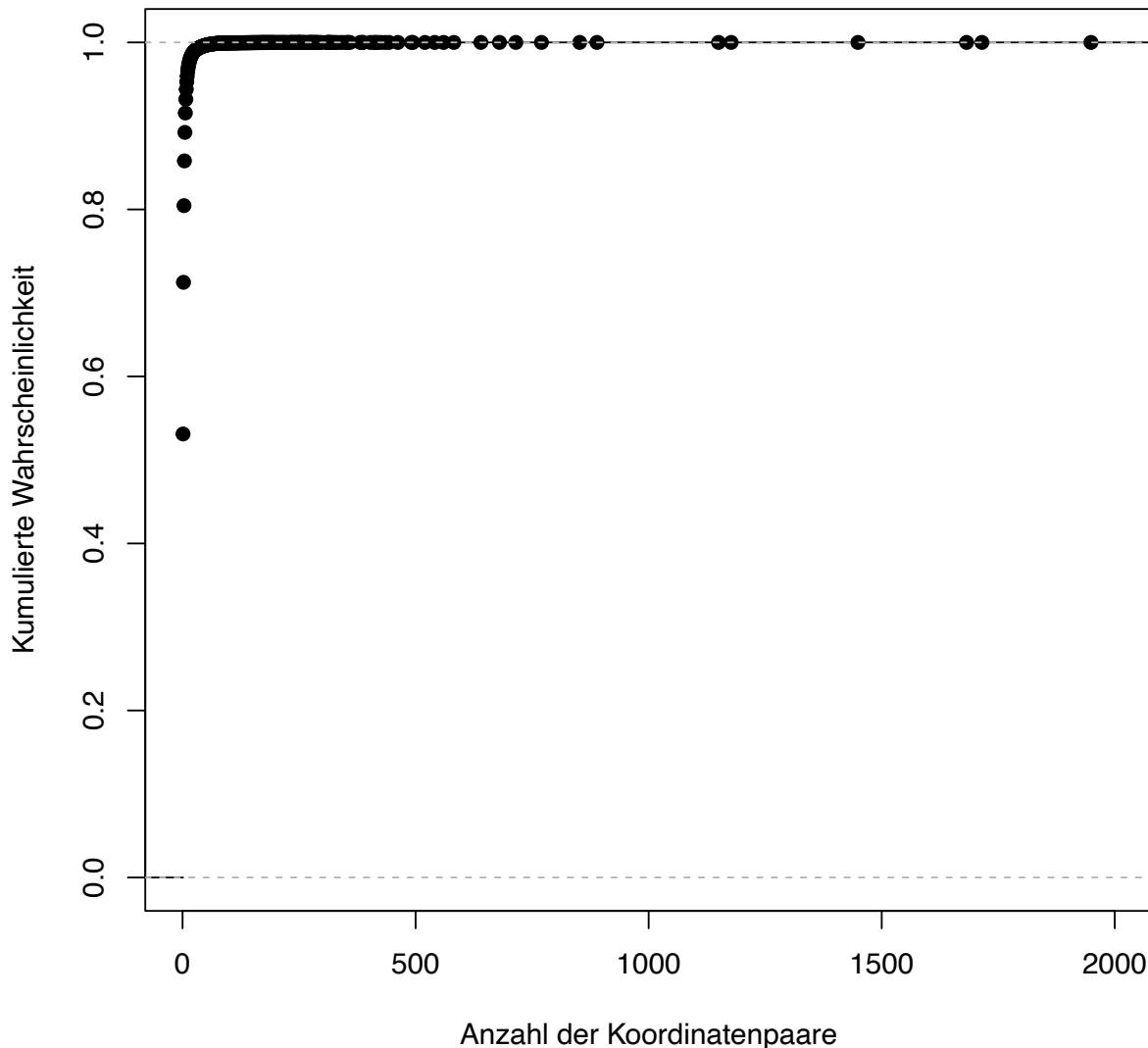
#Zeige die ersten 5 Zeilen des Subsets an zum testen
#head(Koordinaten_subset_x_y_heatmap)

# Zeige den größten und kleinsten Wert von count an
#max(Koordinaten_subset_x_y_heatmap$count)
#min(Koordinaten_subset_x_y_heatmap$count)

# Erstelle die Dichtefunktion
density_function <- ecdf(Koordinaten_subset_x_y_heatmap$count)

# Plotte die Dichtefunktion
plot(density_function, xlim=c(0, 2000), ylim=c(0, 1),
  xlab="Anzahl der Koordinatenpaare",
  ylab="Kumulierte Wahrscheinlichkeit",
  main="Kumulierte Dichtefunktion der Koordinatenpaare")
```

Kumulierte Dichtefunktion der Koordinatenpaare



Die kumulierte Dichtefunktion der Koordinatenpaare wurde benötigt, da vorher nicht ersichtlich war, wie die Verteilung der Koordinatenpaare ist. Ohne diese Erkenntnis war es durchaus schwierig eine passende Farbpalette zu erstellen, da die Verteilung sehr ungleichmäßig ist. Diese ungleichmäßige Verteilung mit einer gleichmäßigen Farbpalette darzustellen, würde zu einer falschen Darstellung führen. Beim Ausprobieren mit einer gleichmäßigen Farbpalette war die Visualisierung nicht aussagekräftig. Diese Erkenntnis hat sich im Verlauf der Untersuchung herausgestellt, weshalb Anpassungen vorgenommen wurden. Nun ist es möglich eine aussagekräftige Heatmap zu erstellen.

```
#Erstellen einer Tabelle, welche die Häufigkeit der Koordinatenpaare zählt  
Koordinaten_subset_x_y_heatmap_tabelle <- data.frame(x=data$x, y=data$y)
```

```
# Zähle die Häufigkeit der Koordinatenpaare und erstelle eine neue Tabelle  
häufigkeit_tabelle <- as.data.frame(table(Koordinaten_subset_x_y_heatmap_tabelle))
```

```

# Benenne die Spalten um
colnames(häufigkeit_tabelle) <- c("X", "Y", "Anzahl")

#Füge eine neue Spalte hinzu, welche den Koordinatenpaare Farben zuweist,
#je nach Anzahl der Koordinatenpaare
#Die Farbbereiche werden durch eine passende Auswahl definiert.
häufigkeit_tabelle$Farbe <- ifelse(häufigkeit_tabelle$Anzahl == 0, "white",
ifelse(häufigkeit_tabelle$Anzahl < 10, "purple",
ifelse(häufigkeit_tabelle$Anzahl < 50, "blue",
ifelse(häufigkeit_tabelle$Anzahl < 350, "orange", "red"))))

# Zeige einen Ausschnitt der Tabelle an zum testen
#print(head(häufigkeit_tabelle))

#Anzahl der Zeilen der Tabelle
nrow(häufigkeit_tabelle)

#Legende erstellen für die Heatmap
Legenden_Daten <- data.frame(Farbe = unique(häufigkeit_tabelle$Farbe),
Bedeutung = c(">350", "<350", "<50", "<10", "0"))

```

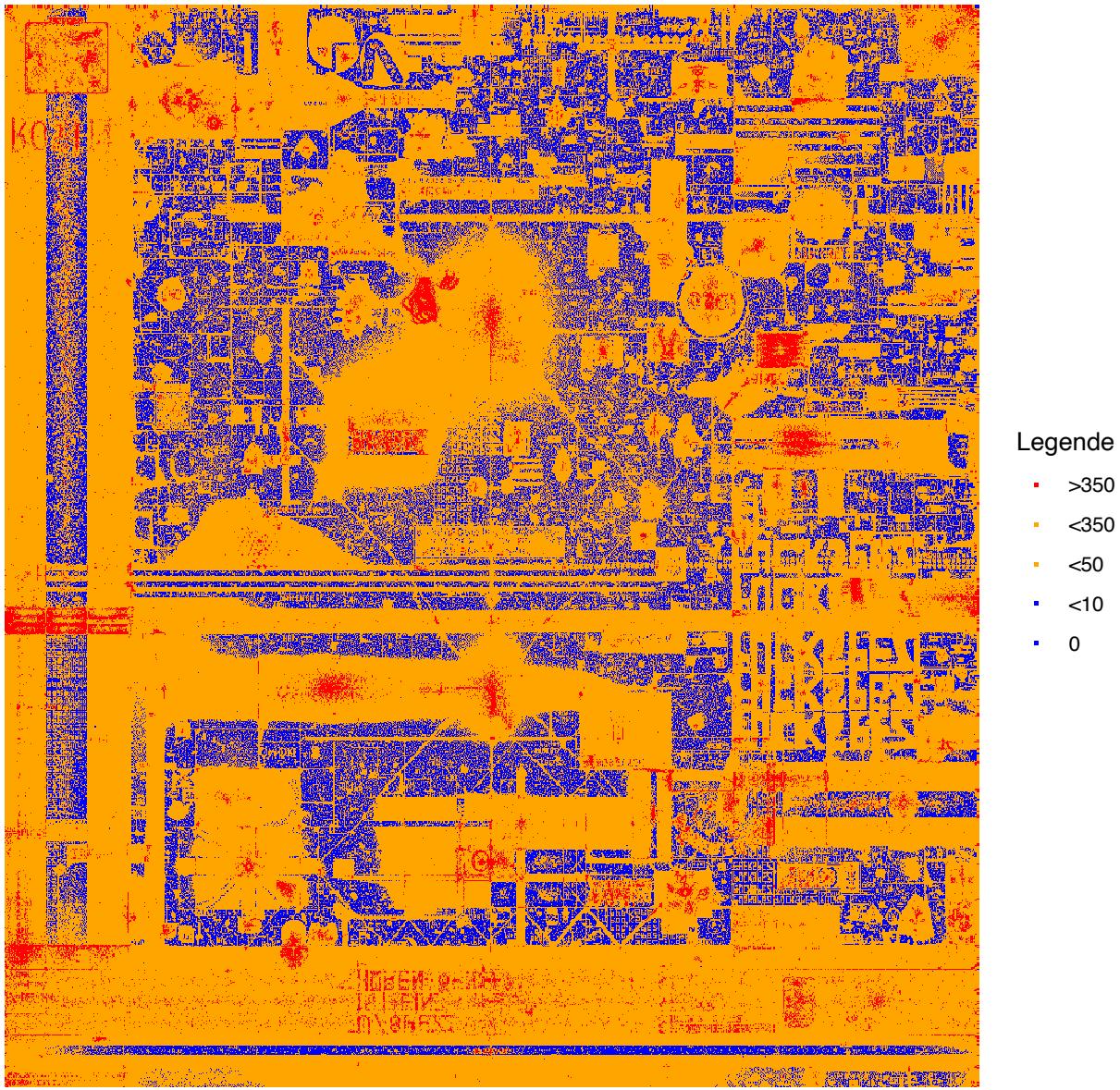
```

#Nun erstellen wir einen Plot, welcher die X-Achse und Y-Achse
#der Koordinatenpaare mit ihrer Farbe darstellt.
Koordinaten_subset_x_y_heatmap_tabelle_grafik <- ggplot(
  häufigkeit_tabelle,aes(X, Y, col = Farbe)) +
  geom_point(shape = 15, size = 0.5) + scale_color_manual(values = as.character(häufigkeit_tabelle$Farbe),
  labels = Legenden_Daten$Bedeutung) + theme_minimal() +
  theme(axis.title = element_blank(), axis.text = element_blank(),
  axis.line = element_blank(), axis.ticks = element_blank()) +
  labs(title = "Heatmap Anzahl Platzierungen", col = "Farbe") +
  guides(col = guide_legend(title = "Legende"))

#Anzeigen der Grafik
Koordinaten_subset_x_y_heatmap_tabelle_grafik

```

Heatmap Anzahl Platzierungen



Durch die Heatmap erkennt man sehr schön, welche Bereiche am meisten Pixel gesetzt bekommen haben. Die roten Bereiche wurden relativ oft gesetzt. Es gibt wenige Bereiche die rote sind. Es gibt Anhäufungen in den Ecken des Koordinatensystems mit Ausnahme der rechten unteren Ecke. Außerdem gibt es in der Mitte des Koordinatensystems kleinere Anhäufungen von Pixeln. In diesen roten Bereichen kann es durchaus sein, dass die Teilnehmer an der Veranstaltungen einander überboten haben. Die Annahme, dass die roten Bereiche entstanden sind, um bestimmte Bilder zu erzeugen und gemeinsam zu erstellen, kann widerlegt werden durch die Betrachtung der blauen Bereiche der Heatmap. In den blauen Bereichen sind zusammenhängende Bereiche und Muster zu erkennen. Diese Bereiche sind relativ groß und wurde daher nicht versucht zu zerstören oder zu überschreiben. Es sind vereinzelt auch Buchstaben erkenntlich, welche in den blauen Bereichen entstanden sind. Besonders auffällig ist der vertikale blaue Streifen auf der linken Seite der Heatmap. Zudem ist ein weiterer horizontaler blauer Streifen in der unteren Seite der Heatmap zu erkennen.

Untersuchung der Hauptakteure (Stehen die Hauptakteure in Beziehung zu einander?)

Da nun ein grober Überblick der gesamten Daten und des Koordinatensystems vorhanden ist, soll nun untersucht werden, welche User die Hauptakteure sind. Hierfür erstellen wir einen neuen Subset, welcher aus den Koordinatenpaaren, dem pixelcolor, dem timestamp und dem user besteht. Diesen Subset filtern wir nach den Top 10 Usern, welche die meisten Pixel gesetzt haben. Dieser Subset ist die Grundlage für die weiteren Untersuchungen zu den Pixel_Farben und den Koordinatenpaaren.

```
#Erstellen eines neuen Subsets, welches aus den Koordinatenpaaren, dem pixelcolor,
#dem timestamp und dem user besteht
Top10User <- data.frame(x=data$x, y=data$y, pixel_color=data$pixel_color,
                         timestamp=data$timestamp, user=data$user)

# Filtern des Subsets nach den Top 10 Usern,
#welche die meisten Pixel gesetzt haben
Top10User <- data %>% group_by(user) %>% summarise(count = n()) %>%
  arrange(desc(count)) %>% head(10)

# Filtern des ursprünglichen Dataframes nach den Top 20 Usern
Top10User <- data %>% filter(user %in% Top10User$user)

#Anzeigen der Spalten von Top10User
#colnames(Top10User)
#Top10User

#printe die unique User zum testen
#unique(Top10User$user)

#printe die unique Farben zum testen
#unique(Top10User$pixel_color)
```

Untersuchung der Beziehung der Top10 User im Kontext der Farben

Als nächstes wollen wir untersuchen, welche Farben die Top 10 User gesetzt haben und wie diese in Verbindung stehen. Hierfür passen wir den Subset Top10User an, sodass dieser nur noch aus den Spalten user und pixel_color besteht.

```
#Die doppelten Werte von user und pixel_color werden zusammengefasst und gezählt
#und in der Spalte count gespeichert und anschließend nach der Anzahl
#absteigend sortiert. Der Dataframe benötigt user und pixel_color als Vektoren
#for count als Attribut(Gewichtung) der Kanten.
Top10UserFarben <- Top10User %>% group_by(user, pixel_color) %>% summarise(
  count = n()) %>% arrange(desc(count))
```

```
## 'summarise()' has grouped output by 'user'. You can override using the
## '.groups' argument.
```

```
#Ausschreiben der Farben in pixel_color als Wörter statt Zahlen
#for die spätere Verwendung. Der Hintergrund des Plots ist weiß,
#weshalb die Farbe Weiß nicht verwendet werden kann.
#Daher wird die Farbe Pink verwendet.
```

```

Top10UserFarben$pixel_color <- ifelse(Top10UserFarben$pixel_color == 1, "red",
ifelse(Top10UserFarben$pixel_color == 2, "orange",
ifelse(Top10UserFarben$pixel_color == 3, "yellow",
ifelse(Top10UserFarben$pixel_color == 4, "green",
ifelse(Top10UserFarben$pixel_color == 5, "blue",
ifelse(Top10UserFarben$pixel_color == 6, "purple",
ifelse(Top10UserFarben$pixel_color == 7, "black", "pink"))))))))

#Anzeigen von Top10User$pixel_color zum testen
#Top10User$pixel_color

#Erstellen von Vektoren für die Knoten und Kanten des Graphen
usernamen <- c(Top10UserFarben$user)
pixel_color <- c(Top10UserFarben$pixel_color)
gewichtung <- c(Top10UserFarben$count)

#Anzahl der Einträge in den Vektoren zum testen.
#length(usernamen)
#length(pixel_color)
#length(gewichtung)
#usernamen
#gewichtung
#pixel_color1

#Erstellen einer Matrix aus den Vektoren
Top10UserFarben_Matrix <- cbind(usernamen, pixel_color)

#Funktionen zum testen
#Top10User_namenUndFarben
#class(Top10User_namenUndFarben)

#Erstellen eines Graphen aus dem Dataframe (Wie in der Vorlesung gezeigt)
Top10UserFarben_Netzwerk <- graph_from_data_frame(Top10UserFarben_Matrix, directed = FALSE)

#Testen ob der Graph erstellt worden ist
#Top10User_Netzwerk

#Hinzufügen von Attributen zu den Knoten und Kanten
Top10UserFarben_Netzwerk <- set_edge_attr(Top10UserFarben_Netzwerk,
"gewichtung", value = gewichtung)

Top10UserFarben_Netzwerk <- set_edge_attr(Top10UserFarben_Netzwerk,
"pixel_color", value = pixel_color)

#Testen ob die Attribute hinzugefügt worden sind
#Top10User_Netzwerk_gewichtet

#Die Knotenfarbe wird hier bestimmt und soll
#sich von den Farben der Kanten unterscheiden.
vertex_colors <- rep("beige", vcount(Top10UserFarben_Netzwerk))

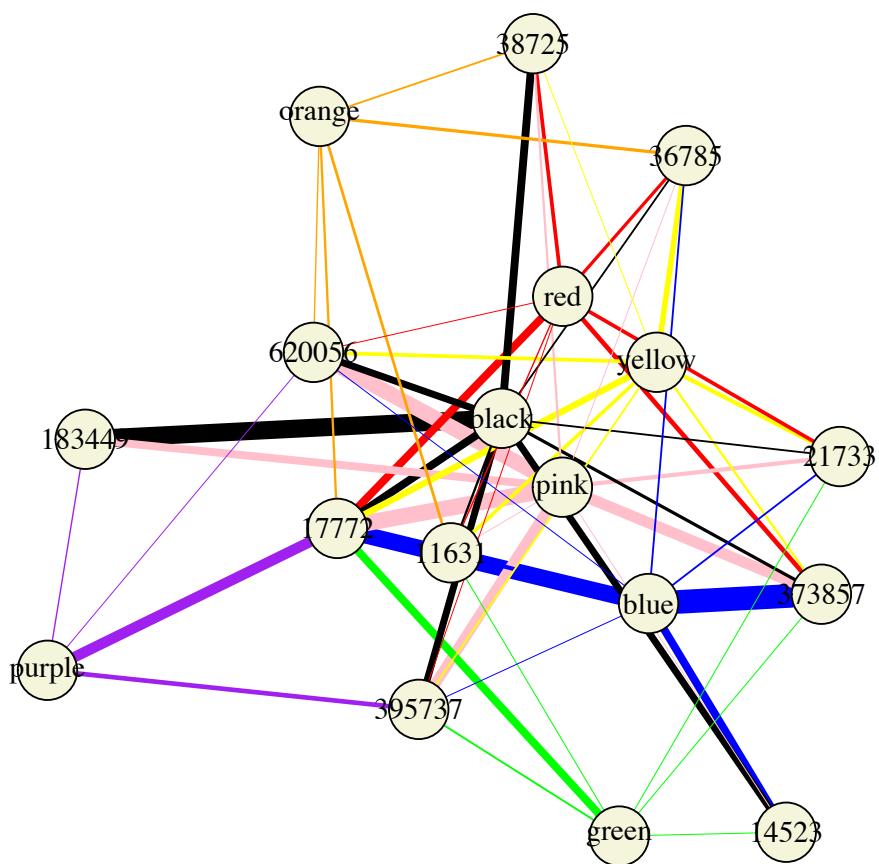
#Plotten des Graphen

```

```

plot(Top10UserFarben_Netzwerk,
      #Layout des Graphen
      layout = layout.fruchterman.reingold,
      #Kanten Dicke und Farbe der Kanten bestimmen
      edge.width = E(Top10UserFarben_Netzwerk)$gewichtung/4,
      edge.color = E(Top10UserFarben_Netzwerk)$pixel_color,
      #Knoten Farbe und Größe bestimmen
      vertex.frame.color = "black",
      vertex.label.color = "black",
      vertex.color = vertex_colors)

```



Bei der Betrachtung des Netzwerks fallen einige Farben besonders auf. Die Knoten Blau, Schwarz und Weiß(In der Abbildung als Pink dargestellt) sind besonders ausgeprägt und bilden in der Mitte ein Dreieck. Der Knoten Blau ist mit den Usern 14523, 373857 und 17772 sehr stark verbunden und hat breite Kanten.

Gleichzeitig hat der User 14523 eine zweite Kante zum Knoten Schwarz, welches auch auffällig ist. Der Knoten Schwarz hat insgesamt 4 stark ausgeprägt Kanten, wobei die stärkste Kante zum User 183449 führt. Der User 183449 hat meistens die Farben Weiß und Schwarz genutzt, wobei es auch zur Nutzung von Violett kam. Auch der Knoten Weiß hat 4 stark ausgeprägte Kanten. Einer dieser Kanten führt zum User 17772. Dieser User hat eine sehr breite Auswahl an Farben und hat gleichzeitig sehr ausgeprägte Kanten. Der User hat viele verschiedene Farben häufig genutzt zu haben. Dieser User fällt damit sehr auf und ist im Vergleich zu den anderen Usern in der Mitte zu verorten. Eine weitere Auffälligkeit ist der Knoten Orange. Dieser Knoten hat vier Kanten, aber diese sind dünn.

Untersuchung der Beziehung der Top10 User im Kontext der Koordinaten

Im nächsten Schritt sollen die Top10 User auf ihre Beziehungen untersucht werden. Es soll herausgefunden werden, ob es Auffälligkeiten gibt, zwischen den Nutzern und den Koordinaten, die sie ausgewählt haben. Hierfür passen wir den Subset Top10User an, sodass dieser nur noch aus den Spalten user und die Koordinaten besteht. Da die Koordinaten aus den zwei Spalten x und y besteht, werden diese Spalten zusammengefügt.

```
#Neuer Subset auf Grundlage von Top10User
Top10UserKoordinaten <- Top10User

#Füge eine neue Spalte hinzu, welcher die Koordinatenpaare zusammenfasst
Top10UserKoordinaten <- transform(Top10UserKoordinaten, x_y = paste(x, y, sep = "_"))

#Die doppelten Werte von x_y werden zusammengefasst und gezählt und in der
#Spalte count gespeichert und anschließend nach der Anzahl der
#Koordinatenpaare absteigend sortiert.
Top10UserKoordinaten <- Top10UserKoordinaten %>% group_by(user, x_y) %>%
  summarise(count = n()) %>% arrange(desc(count))

## `summarise()` has grouped output by 'user'. You can override using the
## `.` argument.

#Anzeigen der Spalten von Top10UserKoordinaten zum testen
#colnames(Top10UserKoordinaten)
#Top10UserKoordinaten

#Erstellen von Vektoren für die Knoten und Kanten des Graphen
usernamen2 <- c(Top10UserKoordinaten$user)
koordinaten <- c(Top10UserKoordinaten$x_y)
gewichtung2 <- c(Top10UserKoordinaten$count)

#Erstellen einer Matrix aus den Vektoren
Top10UserKoordinaten_Matrix <- cbind(usernamen2, koordinaten)

#Testen ob die Matrix erstellt worden ist
#Top10UserKoordinaten_Matrix

#Erstellen eines Graphen aus dem Dataframe (Wie in der Vorlesung gezeigt)
Top10UserKoordinaten_Netzwerk <- graph_from_data_frame(Top10UserKoordinaten_Matrix,
  directed = FALSE)

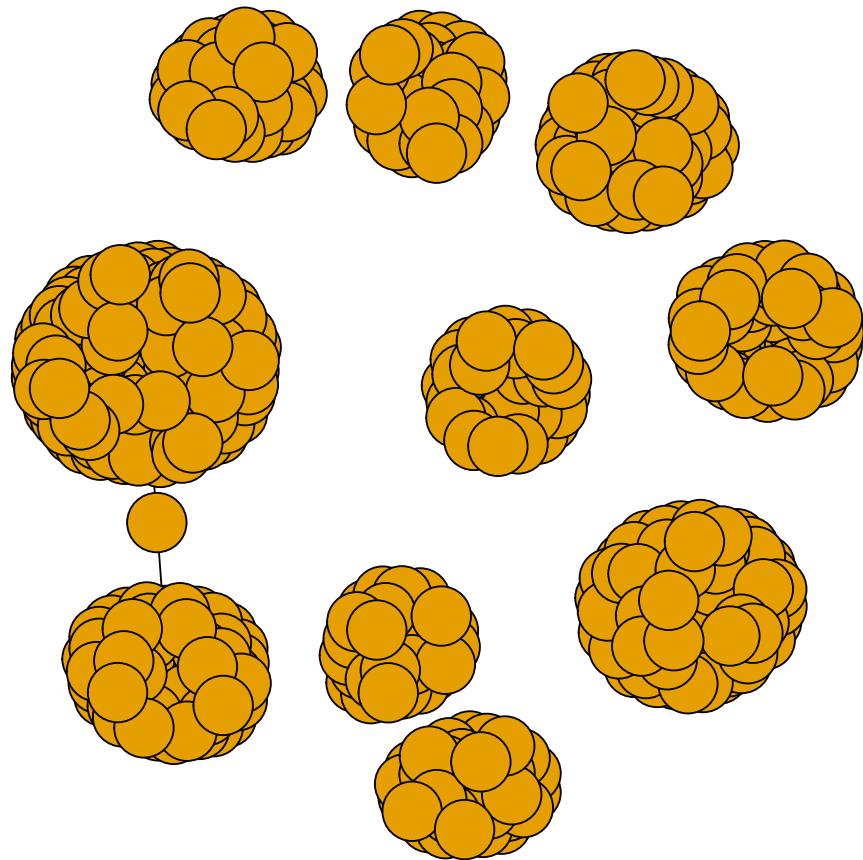
#Hinzufügen von Attributen zu den Knoten und Kanten
```

```

Top10UserKoordinaten_Netzwerk <- set_edge_attr(Top10UserKoordinaten_Netzwerk,
  "gewichtung", value = gewichtung2)

#Plotten des Graphen
plot(Top10UserKoordinaten_Netzwerk,
  #Layout des Graphen
  layout = layout.fruchterman.reingold,
  #Kanten Dicke und Farbe der Kanten bestimmen
  edge.width = E(Top10UserKoordinaten_Netzwerk)$gewichtung,
  edge.color = "black",
  vertex.label =NA)

```



Aus dem Graphen ist erkenntlich, dass es nur einen Knoten gibt, welcher zwei User verbindet. Die restlichen Knoten sind alle mit nur einem User verbunden. Daher sollten die Bereiche des Koordinatensystems in

Bereiche eingeteilt werden. Hierzu wird das Koordinatensystem in 16 Bereiche eingeteilt. Die X-Achse und die Y-Achse werden jeweils in 4 Bereiche eingeteilt, sodass $4 \times 4 = 16$ Bereiche entstehen. Auf der X-Achse geht ein Bereich von -500 bis -250, -250 bis 0, 0 bis 250 und 250 bis 500. Auf der Y-Achse geht ein Bereich von -500 bis -250, -250 bis 0, 0 bis 250 und 250 bis 500. Die Bereiche werden nummeriert mit 1 bis 16 beginnend von links unten nach rechts oben.

```
#Neuer Subset auf Grundlage von Top10User
Top10UserKoordinaten <- Top10User

#Füge eine neue Spalte hinzu, welcher die Koordinatenpaare zusammenfasst
Top10UserKoordinaten <- transform(Top10UserKoordinaten, x_y = paste(x, y, sep = "_"))

#Nun werden die Bereiche anhand der x und y Werte bestimmt.
#Hierfür wird eine neue Spalte erstellt, welche die Bereiche enthält.
Top10UserKoordinaten$Bereich <- ifelse(Top10UserKoordinaten$x < -250 &
  Top10UserKoordinaten$y < -250, 1,
  ifelse(Top10UserKoordinaten$x < 0 & Top10UserKoordinaten$y < -250, 2,
  ifelse(Top10UserKoordinaten$x < 250 & Top10UserKoordinaten$y < -250, 3,
  ifelse(Top10UserKoordinaten$x < 500 & Top10UserKoordinaten$y < -250, 4,
  ifelse(Top10UserKoordinaten$x < -250 & Top10UserKoordinaten$y < 0, 5,
  ifelse(Top10UserKoordinaten$x < 0 & Top10UserKoordinaten$y < 0, 6,
  ifelse(Top10UserKoordinaten$x < 250 & Top10UserKoordinaten$y < 0, 7,
  ifelse(Top10UserKoordinaten$x < 500 & Top10UserKoordinaten$y < 0, 8,
  ifelse(Top10UserKoordinaten$x < -250 & Top10UserKoordinaten$y < 250, 9,
  ifelse(Top10UserKoordinaten$x < 0 & Top10UserKoordinaten$y < 250, 10,
  ifelse(Top10UserKoordinaten$x < 250 & Top10UserKoordinaten$y < 250, 11,
  ifelse(Top10UserKoordinaten$x < 500 & Top10UserKoordinaten$y < 250, 12,
  ifelse(Top10UserKoordinaten$x < -250 & Top10UserKoordinaten$y < 500, 13,
  ifelse(Top10UserKoordinaten$x < 0 & Top10UserKoordinaten$y < 500, 14,
  ifelse(Top10UserKoordinaten$x < 250 & Top10UserKoordinaten$y < 500, 15,
  ifelse(Top10UserKoordinaten$x < 500 & Top10UserKoordinaten$y < 500, 16, 0))))))))))))))

#Anzeigen der Spalten von Top10UserKoordinaten zum testen
#colnames(Top10UserKoordinaten)

#Die doppelten Werte von Bereich und user werden zusammengefasst und gezählt
#und in der Spalte count gespeichert und anschließend nach der
# Anzahl der Koordinatenpaare absteigend sortiert.
Top10UserKoordinaten <- Top10UserKoordinaten %>% group_by(user, Bereich) %>%
  summarise(count = n()) %>% arrange(desc(count))

## `summarise()` has grouped output by 'user'. You can override using the
## `.` argument.

#Testen ob die Spalte Bereich hinzugefügt worden ist
#Top10UserKoordinaten

#Erstellen von Vektoren für die Knoten und Kanten des Graphen
usernamen3 <- c(Top10UserKoordinaten$user)
bereich <- c(Top10UserKoordinaten$Bereich)
gewichtung3 <- c(Top10UserKoordinaten$count)
```

```

#Erstellen einer Matrix aus den Vektoren
Top10UserKoordinaten_Matrix <- cbind(usernamen3, bereich)

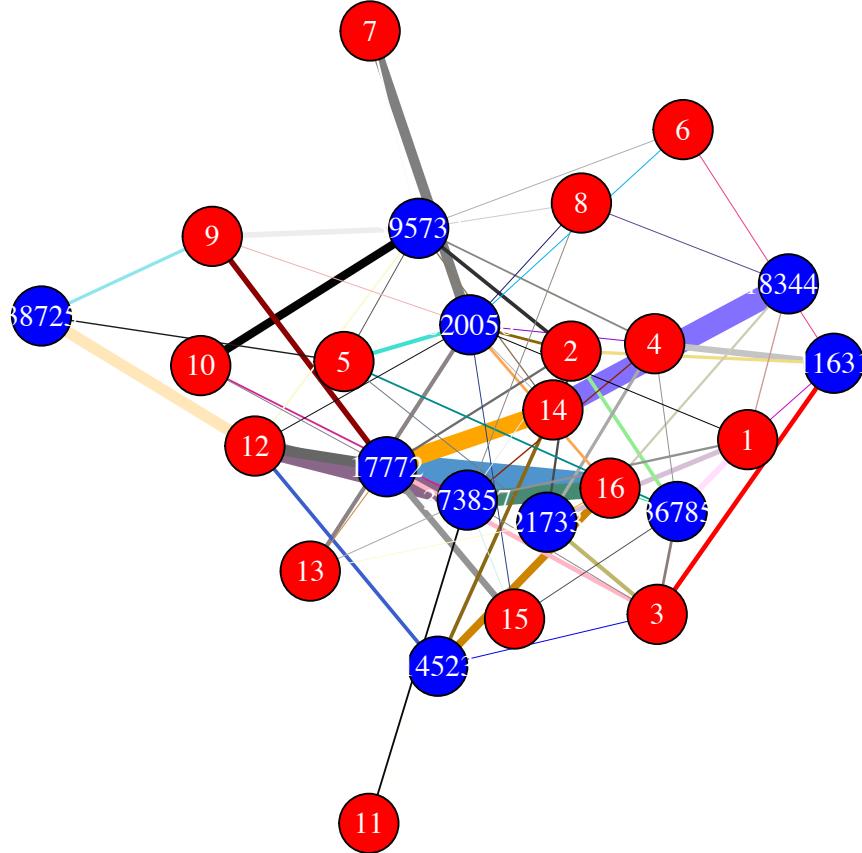
#Top10UserKoordinaten_Matrix

#Erstellen eines Graphen aus dem Dataframe (Wie in der Vorlesung gezeigt)
Top10UserKoordinaten_Netzwerk <- graph_from_data_frame(Top10UserKoordinaten_Matrix,
  directed = FALSE)

#Hinzufügen von Attributen zu den Knoten und Kanten
Top10UserKoordinaten_Netzwerk <- set_edge_attr(Top10UserKoordinaten_Netzwerk,
  "gewichtung", value = gewichtung3)

#Plotten des Graphen
plot(Top10UserKoordinaten_Netzwerk,
  #Layout des Graphen
  layout = layout.auto,
  #Kanten Dicke und Farbe der Kanten bestimmen
  edge.width = E(Top10UserKoordinaten_Netzwerk)$gewichtung/4,
  #Zufällige Farben für die Edges auswählen, um die Kanten besser zu unterscheiden
  edge.color = sample(colors(), ecount(Top10UserKoordinaten_Netzwerk)),
  vertex.color = ifelse(as.numeric(V(Top10UserKoordinaten_Netzwerk)$name)
    <= 16, "red", "blue"),
  vertex.label.color = "white",
  vertex.label.size = 0.5)

```



Dieses neue Netzwerk zeigt deutlich besser die Beziehungen der User zu den Bereichen. Der User 183449 hat sehr starke Beziehungen zum Bereich 14. Des Weiteren ist der Bereich 14 auch sehr stark mit User 17772 verknüpft. Das bedeutet, dass diese beiden User höchstwahrscheinlich gegeneinander konkurriert oder zusammengearbeitet haben. Eine weitere Auffälligkeit ist, dass der User 17772 weitere sehr starke Beziehungen zu den Bereichen 12 und 16, neben dem Bereich 14 hat. Es wurde bereits vorher ermittelt, dass dieser User die meisten Pixel gefärbt hat. Außerdem gibt es manche Bereiche, die nur von einem User alleine gefärbt worden sind. Hierzu kann der Bereich 11 und 14523 betrachtet werden. Der Bereich 7 wurde nur von einem einzigen User (395737) gefärbt. Dieser User hatte also nicht so viel User-Interaktionen, wie die anderen User. In der Mitte befinden sich 4 weitere User die relativ nah aneinander sind und sehr viele kleinere Verbindungen zu den Bereichen haben. Diese User haben wohl sich auf viele unterschiedliche Bereiche verteilt und nicht nur einen bestimmten Bereich gefärbt. Daraus könnte man vermuten, dass diese User als Unterstützer tätig waren. Zusätzlich könnte man vermuten, dass die User mit starken Beziehungen in einem Bereich als Gruppenführer agiert haben.

Interpretation der Ergebnisse und Beantwortung der Forschungsfrage

Interpretation der Ergebnisse

Es wurde ein guter Eindruck geschaffen über die Verteilung der Pixel und die Verteilung der Farben. Generell sind viele Indizien für Beziehungen und Abhängigkeiten zwischen den Usern zu erkennen. Allerdings ist die Bestätigung dieser Indizien nicht möglich, da die User pseudonymisiert sind und unbekannt. Daher können nur Vermutungen angestellt werden.

Beantwortung der Forschungsfrage

Die zu Beginn gestellten Forschungsfragen sollen nun beantwortet werden.

- Sind Muster zu erkennen und in welchen Bereichen der Leinwand sind diese Muster zu erkennen?

Auf der Leinwand sind viele Muster zu erkennen. Das soziale Experiment hat gezeigt, dass die Teilnehmer einander beeinflussen, aber dennoch Ideen von Bildern und Mustern umgesetzt werden können. Diese Muster sind auf der Leinwand zu erkennen.

- Welche Bereiche der Leinwand wurden besonders oft gefüllt?

Die Visualisierung mit der Heatmap veranschaulicht sehr gut die Bereiche, welche sehr oft gefüllt worden sind. Diese Bereiche werden in rot dargestellt.

- Welche Farben haben die aktivsten Nutzer besonders oft genutzt?

Die Farben der aktivsten Nutzer sind sehr unterschiedlich, aber es gibt einige Farben, welche besonders oft genutzt worden sind. Die Farbe Blau, Schwarz und Weiß (In der Abbildung als Pink dargestellt) sind besonders oft genutzt worden.

- Welche Bereiche haben die aktivsten Nutzer besonders oft gefüllt?

Die Bereiche, welche die aktivsten Nutzer besonders oft gefüllt haben, sind die Bereiche 14, 12 und 16.

- Gibt es generell Beziehungen zwischen den Nutzern?

Generell können Beziehungen zwischen den Nutzern erkannt werden, allerdings sind dies nur Indizien. Eine eindeutige Bestätigung ist nicht möglich.

Fazit

Die Social Network Analysis ist eine sehr interessante Methode, um Beziehungen zwischen Usern zu untersuchen. Es wurden ein besonders interessanter Datensatz ausgewählt, welcher sehr viele Möglichkeiten zur Untersuchung bietet. Aus den gegebenen Daten konnte eine gute Analyse erstellt werden, welche viele Indizien für Beziehungen zwischen den Usern aufzeigt. Weitere Visualisierungen und Untersuchungen sind möglich, aber würden den Rahmen der Arbeit übersteigen. Dies ist ein kurzer Einblick in die Social Network Analysis und die Möglichkeiten, welche diese bietet.