Text mining ChatGPT

Abishan, Josua, Lars, Luca

2023-05-18

This research aims to explore and gain insights into Chat GPT, an advanced language model, through text mining of tweets. The dataset provided for this study consists of tweets collected between November 30, 2022, and January 29, 2023. With nearly 180,000 tweets at our disposal, a comprehensive analysis will be conducted to uncover valuable information.

The initial phase of this study involves a general analysis of the dataset, where patterns, trends, and key features will be identified. Following that, dictionaries will be defined, and automated content analysis techniques will be applied. This step will enable us to categorize and extract meaningful information from the tweets efficiently.

In the final stage, a sentiment analysis will be performed using the "syuzhet" method. This approach will allow us to assess the emotional tone and sentiment expressed in the tweets, providing a deeper understanding of the public perception and attitudes towards Chat GPT during the specified time period.

By leveraging the power of text mining and analyzing a substantial amount of Twitter data, this research aims to contribute to the existing knowledge about Chat GPT and provide valuable insights into its reception among Twitter users.

Load packages and data

```
library (syuzhet)
library (stringr)
library (tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.2     v purrr     1.0.1
## v forcats   1.0.0     v readr     2.1.4
## v ggplot2   3.4.2     v tibble    3.2.1
## v lubridate 1.9.2     v tidyr     1.3.0
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library (ggplot2)
library(scales)
```

```
##
## Attaching package: 'scales'
##
```

```
## The following object is masked from ’package:purrr’:
##
##     discard
##
## The following object is masked from ’package:readr’:
##
##     col_factor
##
## The following object is masked from ’package:syuzhet’:
##
##     rescale
```

```
library(stringi)
library(lubridate)
library(dplyr)
options(scipen=999)

load("ChatGPT.rda")
```

1. Question: What can you tell us about the users that tweet about ChatGPT?

```
# Creating a copy of tweets
tweets_orig <- tweets

# take unique users
Users <- tweets[4:10]
Users = Users[!duplicated(Users$User),]

# Calculating average lenght of tweet
char_counts <- nchar(tweets$Tweet)
av_char_count <- mean(char_counts)
rounded_avg_char_count <- round(av_char_count, 2)

#create median
retweets_median = median(Users$Retweets)
retweets_mean = mean(Users$Retweets)

likes_median = median(Users$Likes)
likes_mean = mean(Users$Likes)

Friends_median = median(Users$UserFriends)
Friends_mean = mean(Users$UserFriends)

Followers_median = median(Users$UserFollowers)
Followers_mean = mean(Users$UserFollowers)

verified_median = median(Users$UserVerified)
verified_mean = mean(Users$UserVerified)

# Create a tibble with the values
my_table <- tibble(
  Statistic = c("Retweets", "Likes", "Friends", "Followers", "Verified"),
  Median = c(retweets_median, likes_median, Friends_median, Followers_median,
             verified_median),
  Average = c(retweets_mean, likes_mean, Friends_mean, Followers_mean,
              verified_mean)
)

print(my_table)
```

```
## # A tibble: 5 x 3
##   Statistic Median   Average
##   <chr>      <dbl>     <dbl>
## 1 Retweets       0    0.833
## 2 Likes          1    4.61
## 3 Friends      402 1142.
## 4 Followers    285 5134.
## 5 Verified       0    0.0226
```

In the initial step, our focus was on analyzing the users who engaged with Chat GPT on Twitter. To ensure the accuracy of our analysis, we began by selecting unique user data, as there was a possibility that the same user may have tweeted about Chat GPT multiple times.

The resulting table provided us with valuable statistics on user engagement. By analyzing these user metrics, we gain insights into the reach, influence, and engagement levels of users discussing Chat GPT on Twitter. This information sets the foundation for a deeper understanding of user behavior and its potential impact on the perception and adoption of Chat GPT.

```
# Count the number of tweets per user
tweet_counts <- as.data.frame(table(tweets$User))

# Rename the columns of the tweet_counts data frame
colnames(tweet_counts) <- c("User", "Tweet_Count")

# Create a new data frame with the number of tweets per user
multi_tweet_users <- tweet_counts[tweet_counts$Tweet_Count >= 1,]

# Create a table that shows the number of times each user has tweeted about Chat GPT
multi_tweet_table <- as.data.frame(table(multi_tweet_users$Tweet_Count))

# Rename the columns of the multi_tweet_table data frame
colnames(multi_tweet_table) <- c("Number_of_Tweets", "Number_of_Users")
```

The generated table reveals that the majority of users only tweeted once about the topic. Surprisingly, only 19.64% of individuals tweeted a second time about Chat GPT. However, upon closer examination at the end of the table, it becomes evident that there are a few users who tweeted about it frequently. One particular user stands out, having tweeted 1325 times about Chat GPT within a span of 61 days, which is quite remarkable.

The frequency at which this person tweeted raises eyebrows. It implies that, on average, they posted a tweet every 66 minutes. Such prolific tweeting activity prompts curiosity and warrants further investigation into the motivations and circumstances behind their extensive engagement with Chat GPT.

This finding highlights the wide variation in user behavior when it comes to discussing Chat GPT on Twitter. While the majority of users exhibit minimal involvement, a small group demonstrates extraordinary dedication and interest in the topic.

In the next step, I want to determine which user has the highest number of retweeted tweets. By analyzing the dataset, I will identify the user whose tweets have been retweeted the most frequently. This information will provide insights into the user's influence and the impact of their tweets within the community. By identifying the user with the highest retweet count, we can gain valuable insights into the most engaging and influential content in our dataset.

```
# Group the tweets by user and calculate the total retweets for each user
user_retweets <- tweets %>%
  group_by(User) %>%
  summarise(total_retweets = sum(Retweets)) %>%
  arrange(desc(total_retweets)) %>%
  top_n(10)
```

```
## Selecting by total_retweets
```

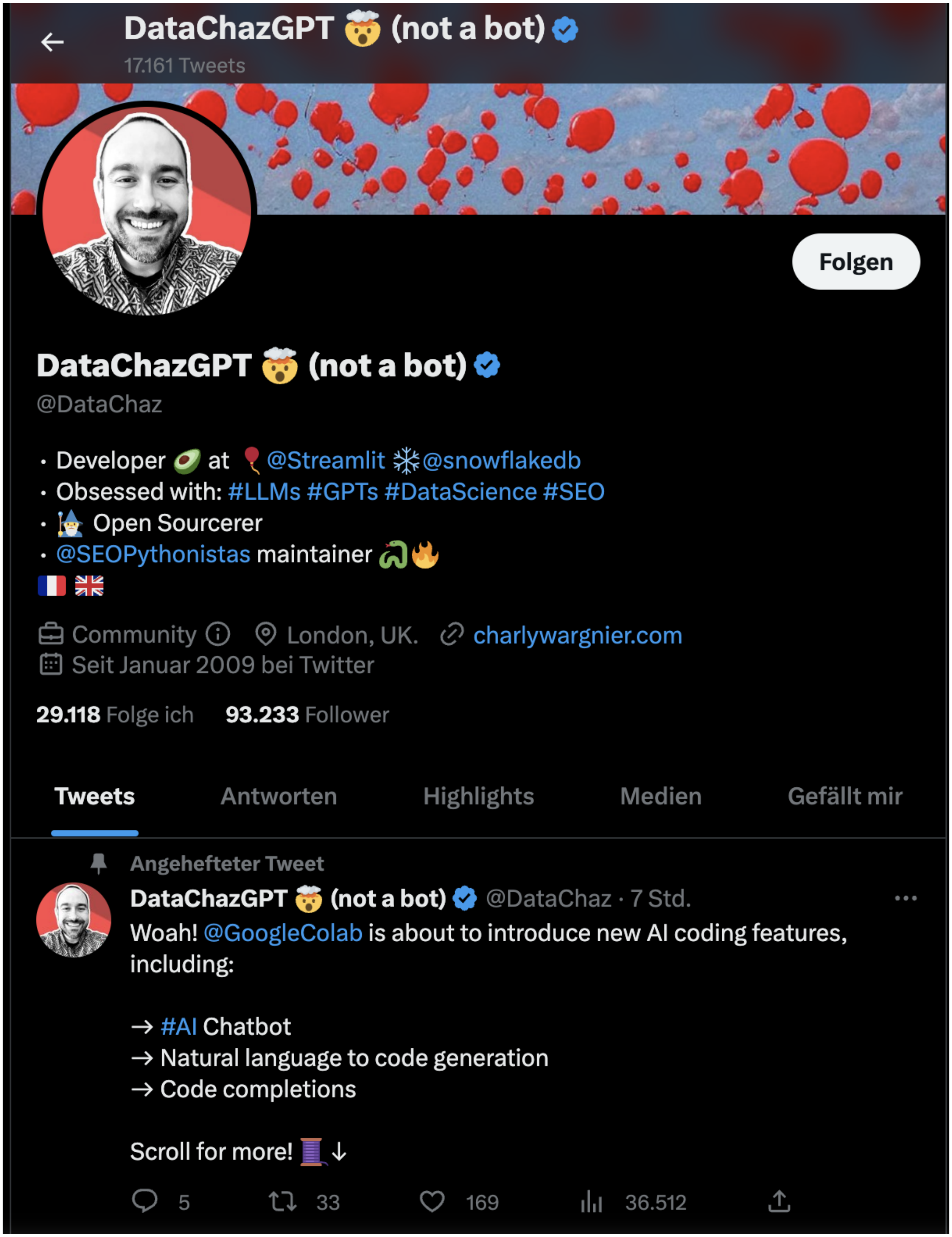
```
# Print the table
user_retweets
```

```
## # A tibble: 10 x 2
##    User            total_retweets
##    <chr>                    <int>
##  1 DataChaz                 21287
##  2 GuyP                      4295
##  3 johnvianny                3506
##  4 hopiumdefi                2955
##  5 StevenXGe                 2269
##  6 kevinschawinski           2031
##  7 PythonPr                  1824
##  8 VeilleCyber3              1686
##  9 Python_Dv                 1466
## 10 FeraSY1                   1403
```

It is evident that the tweets by DataChaz have been retweeted significantly more than others. DataChaz is a London-based developer who has been active on Twitter since 2009. He has a strong interest in Artificial Intelligence and regularly updates his 93,000 followers with valuable information in this domain. DataChaz often shares useful code snippets, allowing enthusiasts to explore various AI APIs and gain hands-on experience. The high retweet count indicates the relevance and impact of DataChaz's content, making him a prominent figure within the AI community on Twitter.

```
# Create a table with the top 10 users with the highest tweet counts
top_10_users <- tweet_counts %>%
  arrange(desc(Tweet_Count)) %>%
  head(10)

# Print the top 10 users table
print(top_10_users)
```

```
##              User Tweet_Count
## 1   MidJourneyAI_        1325
## 2   richardkimphd         898
## 3    VeilleCyber3         464
## 4  ChatGPTSpecial         357
```



DataChazGPT 🤯 (not a bot)
17.161 Tweets

DataChazGPT 🤯 (not a bot)
@DataChaz

- Developer 🥑 at 🎈@Streamlit ❄️@snowflakedb
- Obsessed with: #LLMs #GPTs #DataScience #SEO
- 🧙 Open Sourcerer
- @SEOPythonistas maintainer 🐍🔥

🇫🇷 🇬🇧

Community ⓘ London, UK. charlywargnier.com
Seit Januar 2009 bei Twitter

29.118 Folge ich 93.233 Follower

Angehefteter Tweet

DataChazGPT 🤯 (not a bot) @DataChaz · 7 Std.

Woah! @GoogleColab is about to introduce new AI coding features, including:

→ #AI Chatbot
→ Natural language to code generation
→ Code completions

Scroll for more! 🧵↓

Figure 1: Twitter account of DataChaz

]

```
## 5        DataChaz         350
## 6    BBacktesting         340
## 7  saadnajeebsaad         255
## 8       richgel999         232
## 9      GiolliLucio         225
## 10   EmergentMind         213
```

We manually examined the users in the dataset, and it was observed that "MidJourneyAI_" had the highest number of tweets. However, upon closer inspection, it became apparent that this user had been repeatedly posting the same tweet using a bot. This behavior raises questions about the authenticity and motives behind their actions.

In contrast, the user with the second-highest number of tweets, "richardkimphd," appears to be a genuine account. This user predominantly shares tweets related to AI, indicating a keen interest in the topic. Their account reflects a high level of activity, suggesting that they are an active and engaged Twitter user.

These findings underscore the importance of discerning between automated or inauthentic accounts and genuine user engagement when analyzing Twitter data. Understanding the characteristics and behaviors of different users can help shed light on the broader dynamics and perspectives within the online AI community.

We decided to delete all multiple tweets before analyzing the data further.

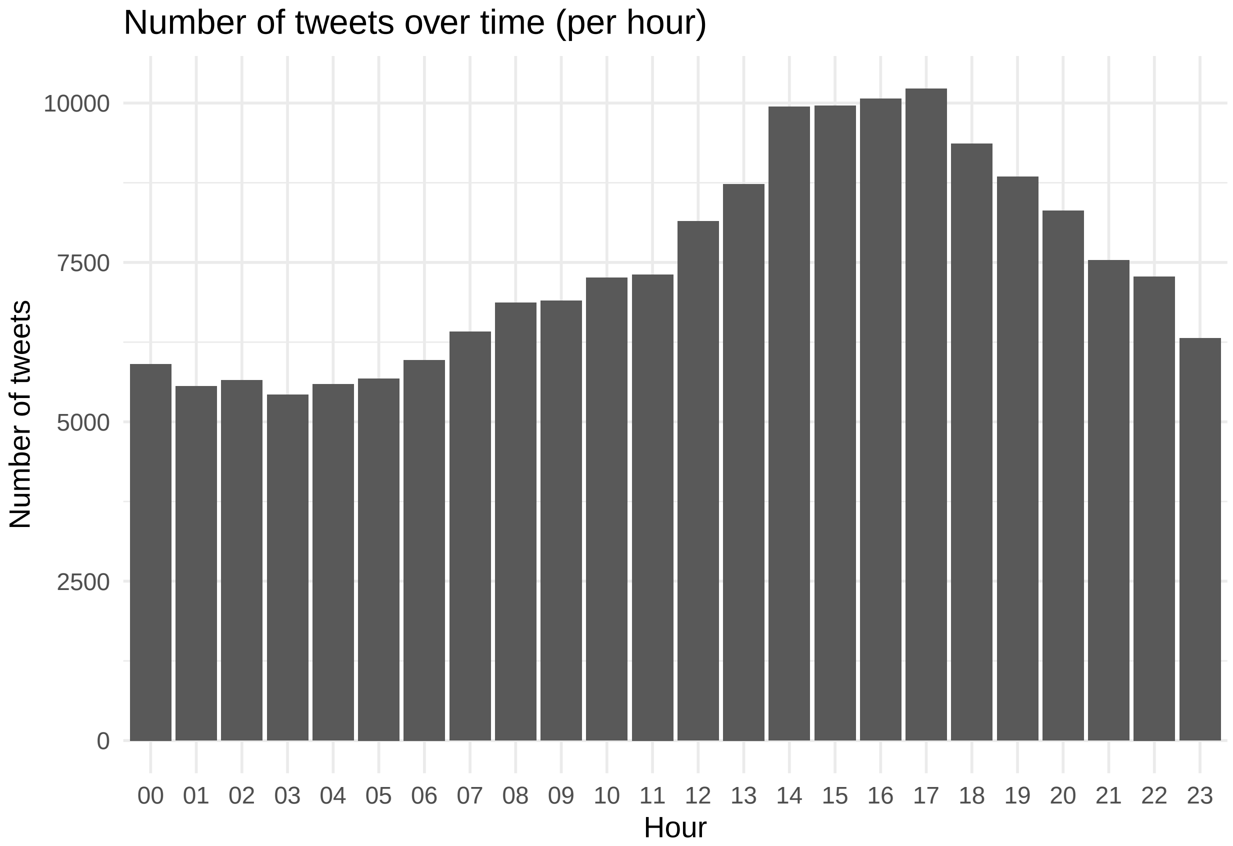
```
# Number of rows before removing duplicate tweets
before <- nrow(tweets)

# Remove rows with duplicate tweets
tweets_unique <- tweets[!duplicated(tweets$Tweet), ]

# Number of rows after removing duplicate tweets
after <- nrow(tweets_unique)

# Number of rows removed
removed <- before - after

# Display the number of removed rows
cat("The removal of duplicate tweets resulted in", removed,
    "rows being removed.")
```

```
## The removal of duplicate tweets resulted in 959 rows being removed.
```

Fortunately, only 959 tweets were removed, which is not a significant loss considering the dataset size of nearly 180,000 tweets.

The removal of these tweets has minimal impact on the overall analysis and findings. With such a vast collection of tweets, the remaining dataset still provides ample data for conducting meaningful analyses and drawing reliable conclusions.

```
# Average/ median of Hour when to tweet
# Create Histogramm for Tweettime
plot_dataHour <- tweets %>%
  group_by (timeofday_hour) %>%
  count()

ggplot (plot_dataHour,
        aes (x=timeofday_hour, y=n)) +
  geom_bar(stat = "identity")+
  theme_minimal () +
  ggtitle("Number of tweets over time (per hour)") +
  xlab("Hour") +
  ylab("Number of tweets")
```



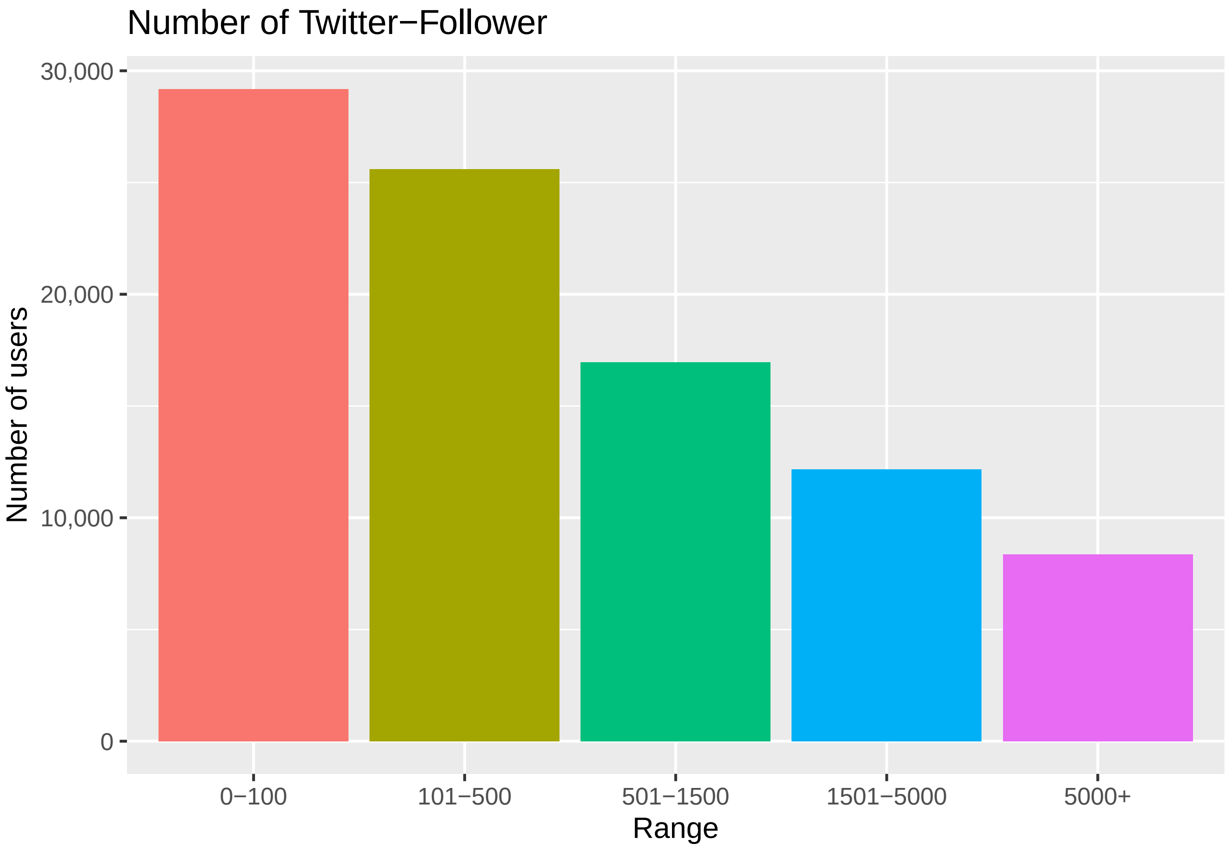
It is evident from the plot that the number of tweets steadily increases from around noon and peaks at around 5 PM. This pattern suggests that most people engage in tweeting about the topic during their work hours.

The surge in tweet activity during the afternoon hours aligns with the assumption that individuals are actively discussing and sharing their thoughts on Chat GPT while they are at work or engaged in related activities.

However, starting from 6 PM, there is a noticeable decline in the number of tweets. This decrease could be attributed to the assumption that individuals tend to finish their workday around 5 PM and gradually shift their focus to other activities or responsibilities outside of work.

```
#Number of follower
#range breaks
range_breaks <- c(0, 100, 500, 1500, 5000, 15000000)

#Appling cut() on follower-data
Users$range <- cut(Users$UserFollowers, breaks = range_breaks)

Users <- na.omit(Users)
```

```
# Creating Barplot
ggplot(Users, aes(x = range, fill = range)) +
  geom_bar() +
  labs(title = "Number of Twitter-Follower", x = "Range", y = "Number of users") +
  scale_x_discrete(labels = c("0-100", "101-500", "501-1500",
                              "1501-5000", "5000+")) +
  scale_y_continuous(labels = scales::comma_format()) +
  guides(fill = "none")
```



The graph depicting the number of followers per user reveals some interesting insights.

Firstly, it is noticeable that a significant portion of users, approximately 30,000 individuals, have a relatively low follower count ranging from 0 to 100. This suggests a large number of users with a limited reach or audience. Moving on, around 25,000 users fall into the range of 101 to 500 followers, indicating a moderate level of influence or popularity among this group.On the other hand, there is a smaller subset of approximately 8,000 users who boast a substantial following, surpassing 5,000 followers. These individuals likely possess a higher level of influence, attracting a significant number of people interested in their content or opinions.

Given these observations, our next step would involve delving deeper into the users with the highest number of followers.

```
top_users <- tweets %>%
  group_by(User) %>%
  summarise(Total_Tweets = n(),
            User_Followers = first(UserFollowers)) %>%
  arrange(desc(User_Followers)) %>%
  top_n(10)
```

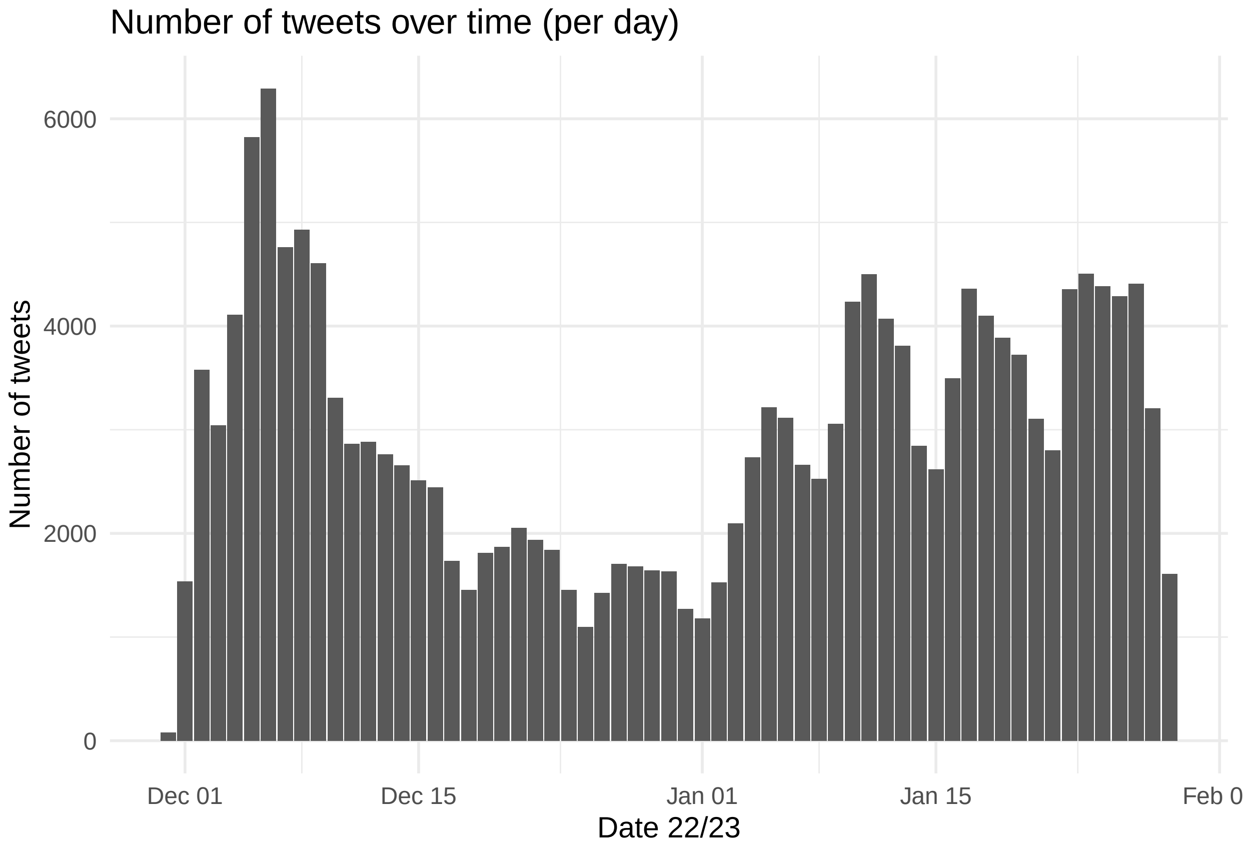
```
## Selecting by User_Followers
```

```
top_users
```

```
## # A tibble: 10 x 3
##    User            Total_Tweets User_Followers
##    <chr>                  <int>          <int>
##  1 timesofindia              15       14671776
##  2 TimesNow                   5       10322487
##  3 TechCrunch                 4       10288879
##  4 htTweets                  12        8695201
##  5 the_hindu                 15        7973030
##  6 IndiaToday                14        6242552
##  7 ZeeNewsEnglish            11        5652093
##  8 ARYNEWSOFFICIAL            2        5240147
##  9 CNNnews18                  3        4745068
## 10 rahulkanwal                1        4483530
```

It is interesting to observe that the majority of users in the top 10 list are from India. A few users originate from Pakistan, and one or two users represent the United States. This highlights the significant presence and influence of Indian users in the dataset, reflecting the popularity of these accounts among Twitter users in the region.

```
#Number of tweets over time
plot_data <- tweets %>%
  group_by (tweet_date) %>%
  count()

ggplot (plot_data,
        aes (x=tweet_date, y=n)) +
  geom_bar(stat = "identity")+
  theme_minimal () +
  ggtitle("Number of tweets over time (per day)") +
  xlab("Date 22/23") +
  ylab("Number of tweets")
```



Hier Erklärung Grafik einfügen. (Joshi) Es ist ein klarer Anstieg von Tweets zuerkennen vom 2. Dezember bis zum 9. Dezember. (noch übersetzten und erklären)

Ideen Abishan (muss noch kontrolliert werden) OpenAI has unveiled on the 3th of November the public beta launch of the DALL · E API, providing developers with the opportunity to incorporate DALL · E's image generation capabilities into their own applications and products. With DALL · E's remarkable adaptability, users can effortlessly produce and modify a diverse range of images, spanning from artistic renditions to convincingly lifelike representations.

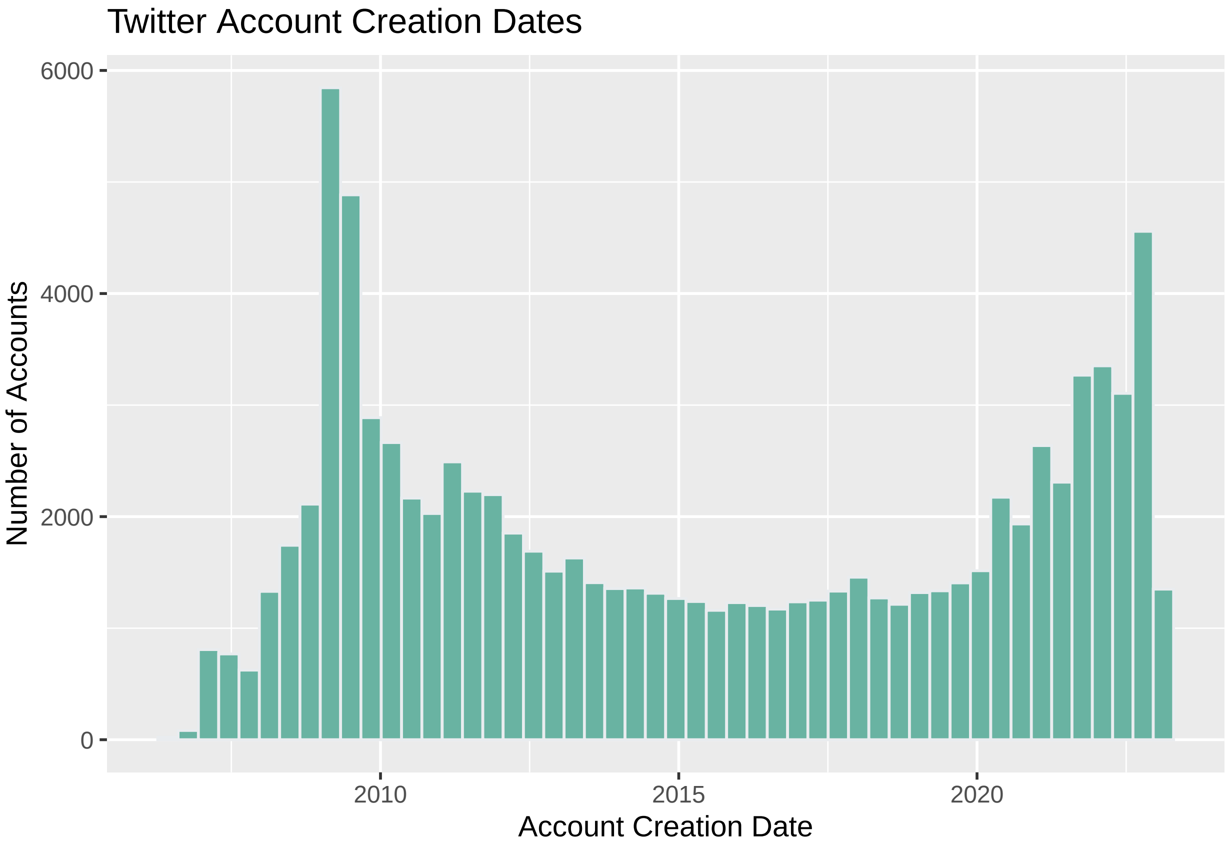
We suspect that the reason for the relative calm on Twitter in early December was likely due to the public beta release of the DALL · E API. It is possible that there were significant discussions and debates about Chat GPT during the period from early November to early December, which gradually subsided. However, on December 8, OpenAI published a new piece of content that once again sparked discussions surrounding Chat GPT.

On December 8, OpenAI released an interview featuring Christian Gibson, an engineer on the Supercomputing team at the company. In the interview, he shares insights into his journey into engineering and how he became a part of OpenAI. Christian also discusses the specific challenges he focuses on addressing, such as the complexities involved in AI workflows and overcoming bottlenecks when running codes on supercomputers. He highlights what sets working on supercomputing at OpenAI apart from other places, such as the immense scale of the operation, and provides a glimpse into his typical day at OpenAI.

This practical and informative publication likely led to renewed discussions and conversations about Chat GPT.

```r
# Convert UserCreated to datetime format
Users$UserCreated <- ymd_hms(Users$UserCreated)

# Create the plot
```

```r
ggplot(Users, aes(x = UserCreated)) +
  geom_histogram(bins = 50, fill = "#69b3a2", color = "#e9ecef") +
  labs(x = "Account Creation Date", y = "Number of Accounts") +
  ggtitle("Twitter Account Creation Dates")
```



The plotted data of Twitter account creation reveals two noticeable peaks. The first peak occurred around 2009, which corresponds to a significant surge in user growth for Twitter. During this period, Twitter experienced a substantial influx of new users, marking a pivotal moment in its expansion and popularity.

The second peak in the plot can be attributed to the acquisition of Twitter by Elon Musk, a prominent figure in the tech industry. This event likely generated significant attention and media coverage, leading to a surge in user account creations. Beyond these observations, the chart offers no other discernible insights or useful information.

2. What are the tweets about, what do users associated the new technology with (e.g. industries, specific applications, and also emotions)?

Pre processing

```r
# Define a function to preprocess the text
preprocess_text <- function(text) {

  # Convert text to lower case
  text <- tolower(text)
  # Remove emojis and emoticons
  text <- gsub("[\U0001F600-\U0001F64F\U0001F910-\U0001F96F\U0001F300-\U0001F5FF\U0001F680-\U0001F6FF\U0

  # Remove numbers
  text <- gsub("\\d+", "", text)

  # Remove punctuation
  text <- gsub("[[:punct:]]", "", text)

  # Remove whitespace
  text <- gsub("\\s+", " ", text)

  # Remove stopwords and other words to be removed
  words_to_remove <- c("the", "and", "in", "to", "a", "of")
  words_to_remove_pattern <- paste0("\\b(", paste(words_to_remove,
                                                   collapse = "|"), ")\\b")
  text <- gsub(words_to_remove_pattern, "", text, ignore.case = TRUE)

  # Return the preprocessed text
  return(text)
}

# Apply the preprocessing function to the Tweet column
tweets$preprocessed_text <- sapply(tweets$Tweet, preprocess_text)
```

In the steps seen above, we have performed preprocessing steps to clean and prepare the text data. The steps include converting the text to lowercase, removing emojis and emoticons, eliminating numbers, removing punctuation, reducing whitespace, and removing stopwords and other specified words. These preprocessing steps aim to standardize the text data and remove irrelevant elements, ensuring a cleaner and more consistent dataset for further analysis. Applying the preprocessing function to the “Tweet” column of the “tweets” dataset creates a new column called “preprocessed_text,” which contains the preprocessed text ready for the analysis.

Automated Content Analysis

```
# Select text column and create your custom dictionaries
# Dictionary 1: Education
tweets$dic1 <- str_count(tweets$preprocessed_text,
"education|learning|school|students|teachers|classroom|knowledge|
university|academic|skills|curriculum|research|study|online learning|
student loans")
tweets$dic1_occurrence <- ifelse(tweets$dic1 >= 2, 1, 0)

# Dictionary 2: Consulting Company, Digital Transformation
tweets$dic2 <- str_count(tweets$preprocessed_text,
"consulting|company|digitaltransformation|strategy|innovation|technology|
business|data analytics|automation|change management|customer experience|
digitalization|leadership|agile|disruption")
tweets$dic2_occurrence <- ifelse(tweets$dic2 >= 2, 1, 0)

# Dictionary 3: Industry, Factories
```

```
tweets$dic3 <- str_count(tweets$preprocessed_text,
"industry|factories|manufacturing|production|supply chain|logistics|machinery|
automation|quality control|assembly line|efficiency|process optimization|
industrial revolution|industrial engineering|industrialization")
tweets$dic3_occurrence <- ifelse(tweets$dic3 >= 2, 1, 0)

# Dictionary 4: Art
tweets$dic4 <- str_count(tweets$preprocessed_text,
"art|artists|painting|sculpture|photography|creativity|exhibition|museum|
aesthetics|expression|visual arts|contemporary art|performance art|art gallery|
artistic expression")
tweets$dic4_occurrence <- ifelse(tweets$dic4 >= 2, 1, 0)

# Dictionary 5: Finance, Banking
tweets$dic5 <- str_count(tweets$preprocessed_text,
"finance|banking|financial services|investment|money|loans|savings|credit cards|
wealth management|financial planning|insurance|retirement|mortgage|
asset management|banking sector")
tweets$dic5_occurrence <- ifelse(tweets$dic5 >= 2, 1, 0)

# Calculate the sums of occurrences for each dictionary
sum(tweets$dic1_occurrence)
```

```
## [1] 6286
```

```
sum(tweets$dic2_occurrence)
```

```
## [1] 1971
```

```
sum(tweets$dic3_occurrence)
```

```
## [1] 129
```

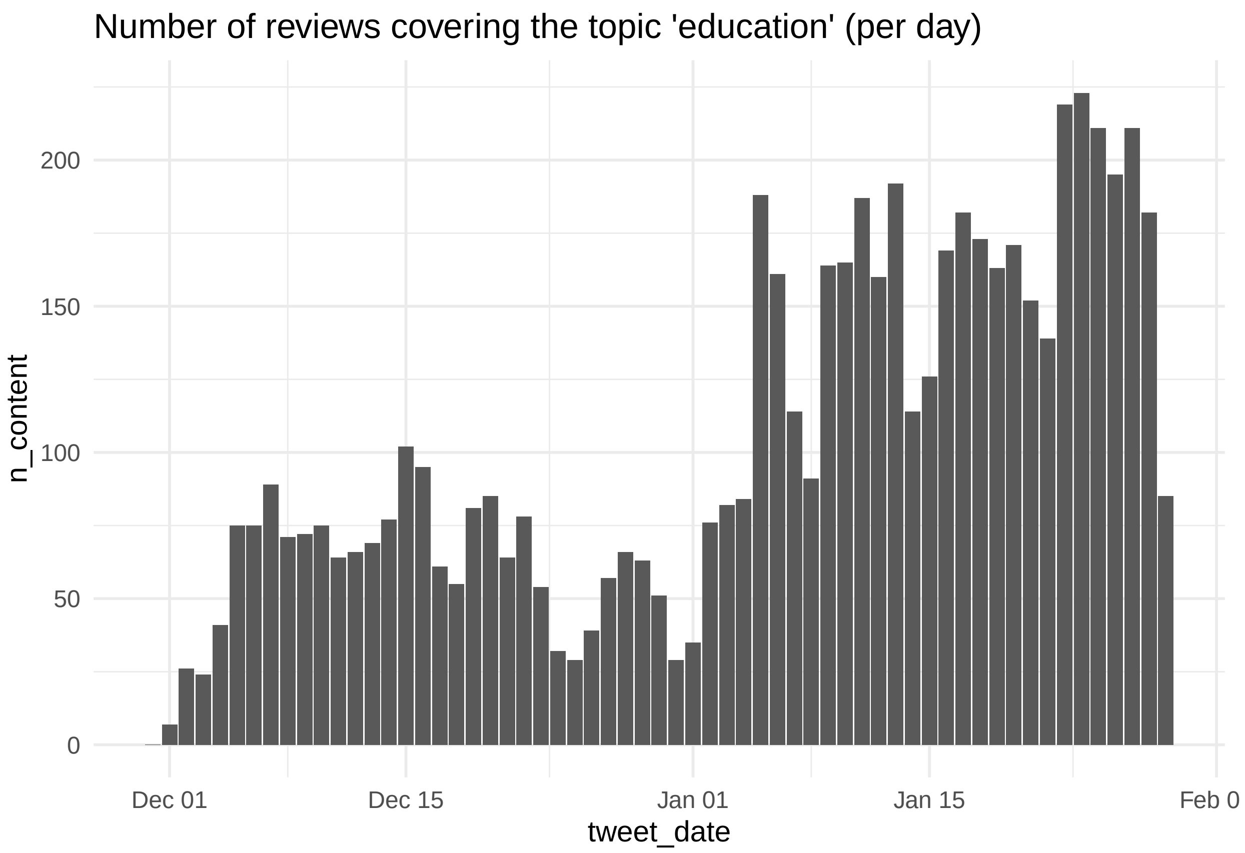
```
sum(tweets$dic4_occurrence)
```

```
## [1] 6045
```

```
sum(tweets$dic5_occurrence)
```

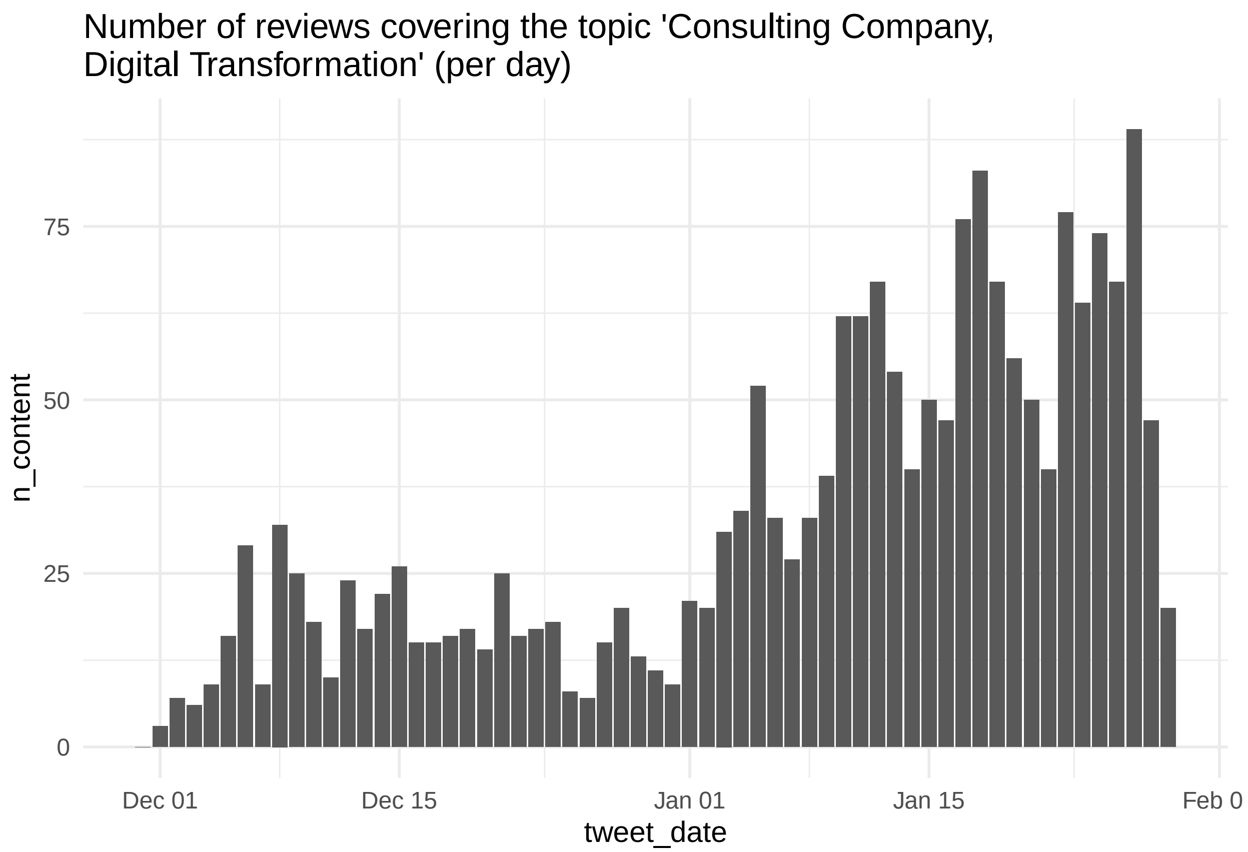
```
## [1] 548
```

```
## VISUALIZE RESULTS dictionary 1 (Education)
#sum of reviews that cover topic per day
plot_content_data1 <- tweets %>%
  group_by (tweet_date) %>%
  summarise(n_content=sum(dic1_occurrence))

ggplot (plot_content_data1, aes (x=tweet_date, y=n_content)) +
  geom_bar(stat = "identity")+ theme_minimal () +
  ggtitle("Number of reviews covering the topic 'education' (per day)")
```



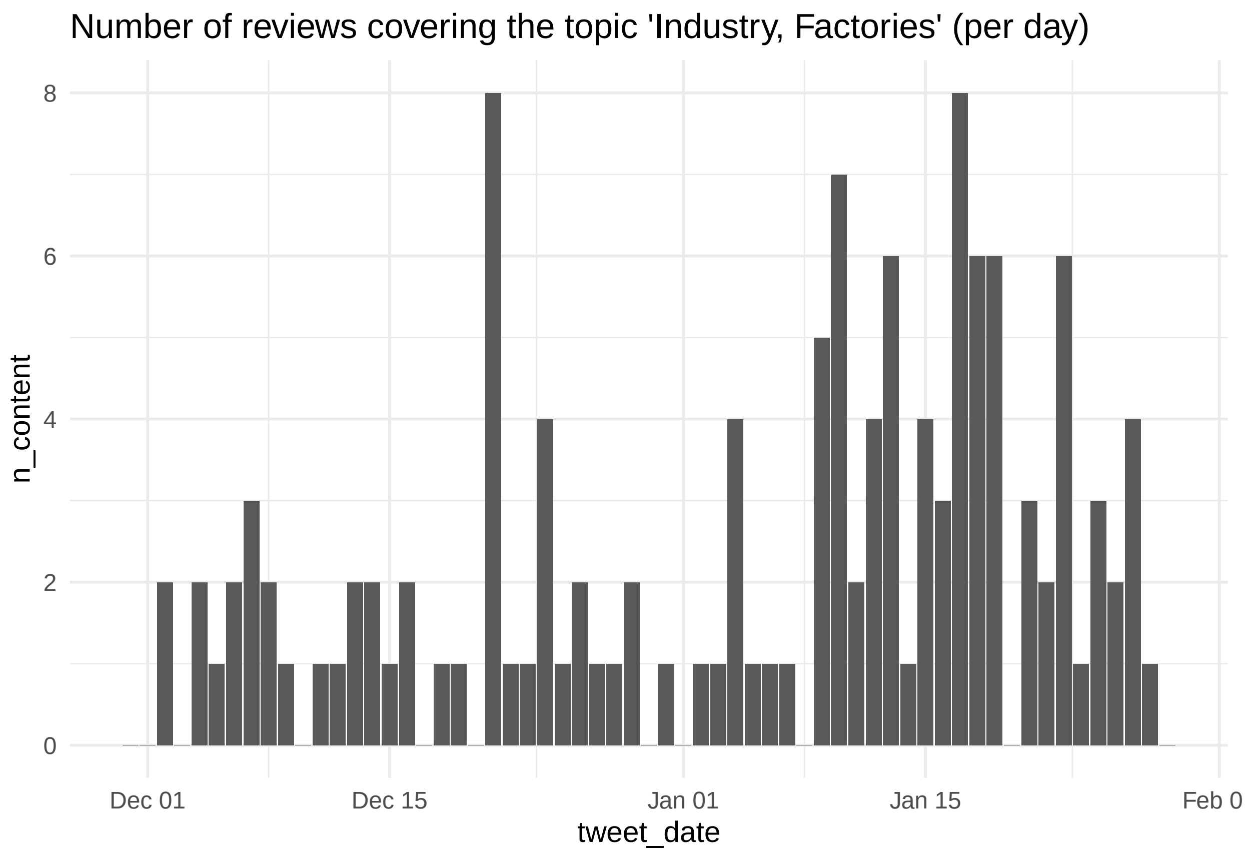
Erklärung einfügen Dictionary Education!!!

```
## VISUALIZE RESULTS dictionary 2 (Consulting Company, Digital Transformation)
#sum of reviews that cover topic per day
plot_content_data2 <- tweets %>%
  group_by (tweet_date) %>%
  summarise(n_content=sum(dic2_occurrence))

ggplot (plot_content_data2, aes (x=tweet_date, y=n_content)) +
  geom_bar(stat = "identity")+ theme_minimal () +
  ggtitle("Number of reviews covering the topic 'Consulting Company,
Digital Transformation' (per day)")
```



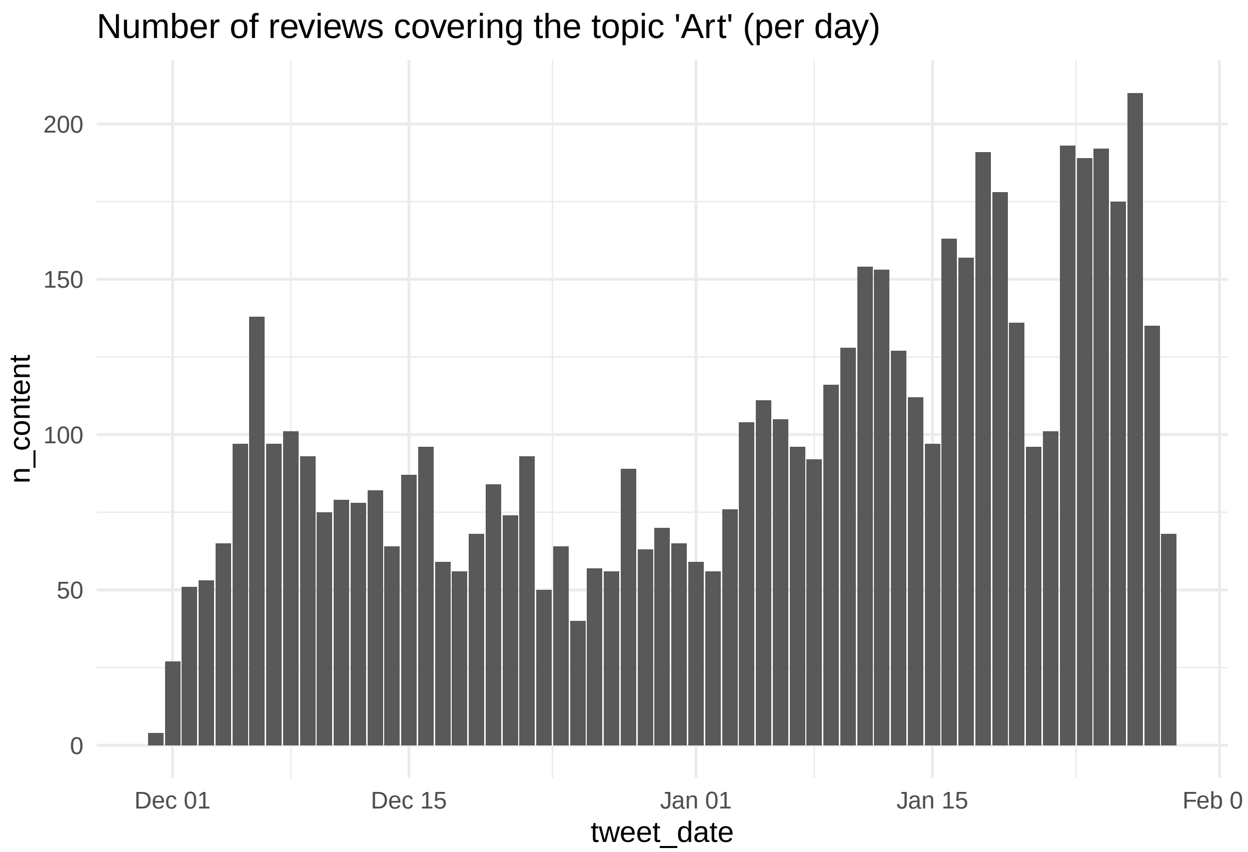
Erklärung einfügen Dictionary Consulting Company, Digital Transformation!!!

```
## VISUALIZE RESULTS dictionary 3 (Industry, Factories)
#sum of reviews that cover topic per day
plot_content_data3 <- tweets %>%
  group_by (tweet_date) %>%
  summarise(n_content=sum(dic3_occurrence))

ggplot (plot_content_data3, aes (x=tweet_date, y=n_content)) +
  geom_bar(stat = "identity")+ theme_minimal () +
  ggtitle("Number of reviews covering the topic 'Industry, Factories' (per day)")
```



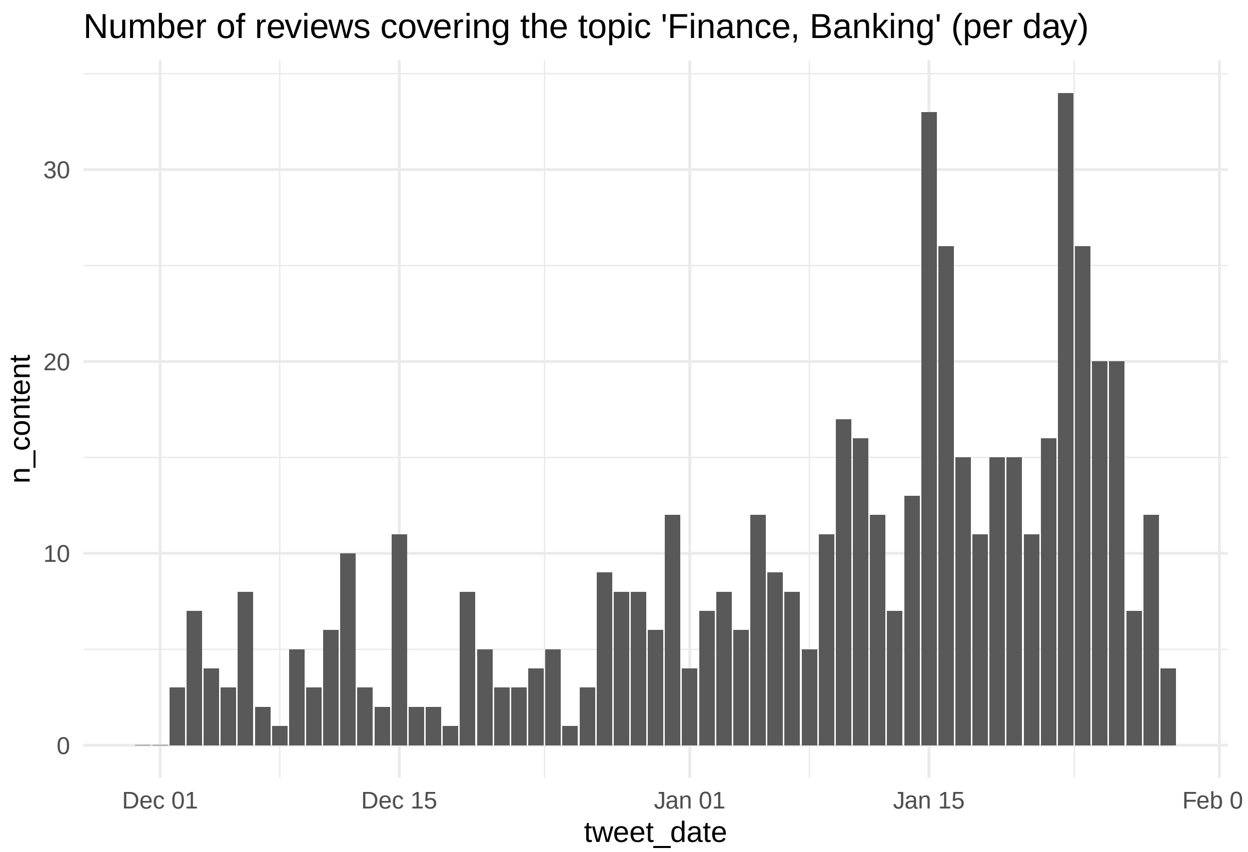
Erklärung einfügen Dictionary Industry, Factories!!!

```
## VISUALIZE RESULTS dictionary 4 (Art)
#sum of reviews that cover topic per day
plot_content_data4 <- tweets %>%
  group_by (tweet_date) %>%
  summarise(n_content=sum(dic4_occurrence))

ggplot (plot_content_data4, aes (x=tweet_date, y=n_content)) +
  geom_bar(stat = "identity")+ theme_minimal () +
  ggtitle("Number of reviews covering the topic 'Art' (per day)")
```



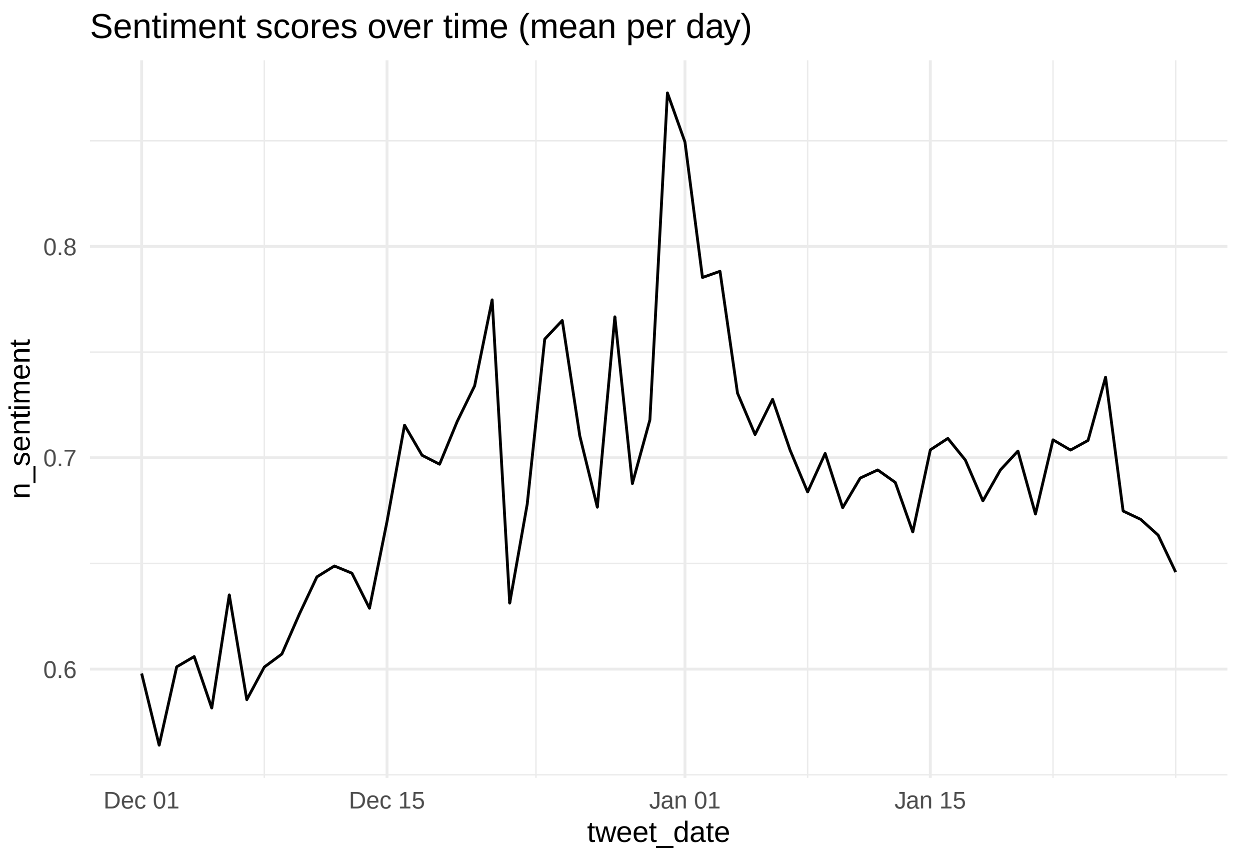
Erklärung einfügen Dictionary Art!!!

```
## VISUALIZE RESULTS dictionary 5 (Finance, Banking)
#sum of reviews that cover topic per day
plot_content_data5 <- tweets %>%
  group_by (tweet_date) %>%
  summarise(n_content=sum(dic5_occurrence))

ggplot (plot_content_data5, aes (x=tweet_date, y=n_content)) +
  geom_bar(stat = "identity")+ theme_minimal () +
  ggtitle("Number of reviews covering the topic 'Finance, Banking' (per day)")
```



Erklärung einfügen Dictionary Finance, Banking!!!

```
# Perform sentiment analysis
tweets$sentiment <- "NA"
tweets$sentiment <- get_sentiment(tweets$preprocessed_text,
                                  method="syuzhet", lang="english")

# Filter data based on the specified date
filtered_tweets <- tweets %>%
  filter(tweet_date >= as.POSIXct("2022-12-01 00:00:00"))

# Calculate the mean sentiment over time
plot_sentiment_data <- filtered_tweets %>%
  group_by(tweet_date) %>%
  summarise(n_sentiment = mean(sentiment))

# Plot the analysis
ggplot (plot_sentiment_data, aes (x=tweet_date, y=n_sentiment)) +
  geom_line()+ theme_minimal () +
  ggtitle("Sentiment scores over time (mean per day)")
```



We have chosen to select data starting from December 1, 2022, for the sentiment analysis plot. This decision was made because in the initial days of the dataset, there were relatively few users tweeting. Including these few users would heavily skew the overall sentiment analysis. The plot reveals a gradual increase in the n_sentiment from 0.6, eventually leading to a prominent peak around the end of 2022, reaching approximately 0.875. From there, until the end of the dataset in late January 2023, the n_sentiment remained consistently around 0.7. This trend suggests a notable shift in sentiment towards more positive or favorable expressions related to the topic of discussion captured in the tweets during the analyzed period.

Additional analysis

```
#correlation of sentiment with custom dictionary
cor(tweets$dic1, tweets$Likes, method = "pearson")
```

```
## [1] 0.004209948
```

```
cor(tweets$dic1, tweets$sentiment, method = "pearson")
```

```
## [1] 0.09936286
```

```
cor(tweets$dic2, tweets$Likes, method = "pearson")
```

```
## [1] -0.004210256
```

```
cor(tweets$dic2, tweets$sentiment, method = "pearson")
```

```
## [1] 0.09845637
```

```
cor(tweets$dic3, tweets$Likes, method = "pearson")
```

```
## [1] -0.001440939
```

```
cor(tweets$dic3, tweets$sentiment, method = "pearson")
```

```
## [1] 0.05057156
```

```
cor(tweets$dic4, tweets$Likes, method = "pearson")
```

```
## [1] -0.0005019979
```

```
cor(tweets$dic4, tweets$sentiment, method = "pearson")
```

```
## [1] 0.1073908
```

```
cor(tweets$dic5, tweets$Likes, method = "pearson")
```

```
## [1] -0.002258519
```

```
cor(tweets$dic5, tweets$sentiment, method = "pearson")
```

```
## [1] 0.03540213
```