Usenet Project - CSV Cleaning

Emerson Johnston

Maintainence

```
rm(list = ls())
knitr::opts_knit$set(root.dir = '/Users/emerson/Github/usenet_webpage')
```

```
# Load Libraries
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.1     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.1
## v purrr     1.0.2
## -- 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(dplyr)
library(readr)
library(syuzhet)

# Directories
output_directory <- "/Users/emerson/Github/usenet_webpage"
threads_directory <- file.path(output_directory, "CSV Files/Threads")
comments_directory <- file.path(output_directory, "CSV Files/Comments")

# Load the datasets
all_threads <- read.csv(file.path(threads_directory, "combined_threads.csv"))
all_comments <- read.csv(file.path(comments_directory, "combined_comments.csv"))
```

Dataset 1 - All Comments Cleaned

```
# Map newsgroups to IDs
newsgroup_ids <- c("netmed" = "NG01", "netmotss" = "NG02", "netnews" = "NG03",
                   "netpolitics" = "NG04", "netreligion" = "NG05", "netsingles" = "NG06")
```

```r
# Threads cleaning
all_threads <- all_threads %>%
  mutate(newsgroup_ID = factor(newsgroup, levels = names(newsgroup_ids), labels = newsgroup_ids),
         Unique_ThreadID = paste(newsgroup_ID, ThreadID, sep = "_")) %>%
  rename(NG_Relative_ThreadID = ThreadID) %>%
  select(Unique_ThreadID, newsgroup, newsgroup_ID, everything()) %>%
  mutate(Date = as.Date(Date, format = "%m/%d/%y"))

# Comments cleaning
all_comments <- all_comments %>%
  mutate(newsgroup_ID = factor(newsgroup, levels = names(newsgroup_ids), labels = newsgroup_ids),
         Unique_CommentID = paste(newsgroup_ID, Unique.Comment.ID, sep = "_"),
         NG_Relative_CommentID = Unique.Comment.ID,
         NG_Relative_ThreadID = Thread.ID,
         Thread.ID = paste(newsgroup_ID, Thread.ID, sep = "_")) %>%
  select(Unique_CommentID, newsgroup, newsgroup_ID, Thread.ID, NG_Relative_CommentID, NG_Relative_Threa
  mutate(Date.and.Time = as.POSIXct(gsub("[^[:alnum:] [:punct:]]", "", Date.and.Time), format = "%b %d,
         Hour = as.numeric(format(Date.and.Time, "%H")),
         Date = as.Date(Date.and.Time))

# Replace mentions of specific authors
all_comments <- all_comments %>%
  mutate(
    Author = case_when(
      Author == "SEVENER" ~ "Tim Sevener",
      Author == "The Polymath" ~ "Jerry Hollombe",
      TRUE ~ Author # Retain other authors as-is
    )
  )

# Add sentiment scores
all_comments <- all_comments %>%
  mutate(SentimentScore = get_sentiment(Full.Text, method = "afinn"))

# Remove Duplicates
all_comments <- all_comments %>%
  distinct(Full.Text, .keep_all = TRUE)

write.csv(all_threads, file.path(threads_directory, "dataset1_threads.csv"), row.names = FALSE)
write.csv(all_comments, file.path(comments_directory, "dataset1_comments.csv"), row.names = FALSE)
```

Dataset 2 - AIDS-Related Comments (1982–1986)

```r
# Load libraries
library(dplyr)
library(stringr)

# Step 1: Define grouped keywords with weights
keyword_groups <- list(
  `10` = c("aids", "acquired immune deficiency syndrome", "human immunodeficiency virus",
           "grid", "gay-related immune deficiency", "gay plague"),
```

```
  `9` = c("hiv", "htlv", "human t-lymphotropic virus", "gay cancer", "kaposi's sarcoma"),
  `8` = c("slim disease", "pneumocystis pneumonia", "gay disease", "homosexual disease",
          "immune disease"),
  `7` = c("fag cancer", "homosexual cancer", "gay compromise syndrome", "aids hysteria"),
  `6` = c("gay fear", "fear of gay", "fear of homosexual", "gay panic", "queer disease",
          "pink disease"),
  `5` = c("sexual orientation disease", "sexual deviant disease", "patients",
          "victims", "carriers"),
  `4` = c("bisexual", "gay/bisexual", "gay men", "homosexuals", "gay sex", "homophobia"),
  `3` = c("virus", "syndrome", "outbreak", "pandemic", "blood disease", "blood test",
          "sexual transmission"),
  `2` = c("sodomy", "bathhouses", "promiscuity", "gay community", "homosexual acts",
          "infection", "epidemic"),
  `1` = c("sex", "queer", "gays", "lesbians", "lifestyle", "contagion",
          "unsafe practices")
)

# Combine all keywords into a single named vector with weights
keyword_weights <- unlist(lapply(names(keyword_groups), function(weight) {
  setNames(rep(as.numeric(weight), length(keyword_groups[[weight]])), keyword_groups[[weight]])
}))

# Define title keywords with weights (subset of `keyword_weights`)
title_keyword_weights <- keyword_weights[names(keyword_weights) %in% c(
  "aids", "htlv", "hiv", "acquired immune deficiency syndrome",
  "human immunodeficiency virus", "gay plague", "gay cancer",
  "kaposi's sarcoma", "pneumocystis pneumonia", "homosexual disease",
  "gay disease"
)]

# Step 2: Filter threads and comments for the desired period
relevant_threads <- all_threads %>%
  filter(Date >= as.Date("1981-12-01") & Date < as.Date("1987-03-01"))

# Step 3: Add relevancy scores to comments
match_keywords <- function(text, keywords, weights) {
  text <- tolower(text)
  sapply(names(keywords), function(kw) {
    if (str_detect(text, paste0("\\b", kw, "\\b"))) {
      return(weights[kw])
    } else {
      return(0)
    }
  }) %>% sum()
}

comments_with_relevancy <- all_comments %>%
  filter(Date >= as.Date("1981-12-01") & Date < as.Date("1987-03-01")) %>%
  rowwise() %>%
  mutate(Relevancy = match_keywords(Full.Text, keyword_weights, keyword_weights)) %>%
  filter(Relevancy > 0)

# Step 4: Identify relevant threads
```

```
relevant_threads <- relevant_threads %>%
  mutate(
    TitleHasKeyword = rowSums(sapply(names(title_keyword_weights), function(kw) {
      grepl(kw, tolower(Thread.Title), fixed = TRUE) * title_keyword_weights[kw]
    })) > 0
  ) %>%
  filter(Unique_ThreadID %in% comments_with_relevancy$Thread.ID | TitleHasKeyword)

# Step 5: Capture all comments in relevant threads
relevant_comments <- all_comments %>%
  filter(Thread.ID %in% relevant_threads$Unique_ThreadID) %>%
  mutate(
    Relevancy = rowSums(sapply(names(keyword_weights), function(kw) {
      grepl(kw, tolower(Full.Text), fixed = TRUE) * keyword_weights[kw]
    }))
  )

# Step 6: Calculate thread-level relevancy
thread_relevancy <- comments_with_relevancy %>%
  group_by(Thread.ID) %>%
  summarise(
    AvgCommentRelevancy = mean(Relevancy, na.rm = TRUE)  # Average score per relevant comment
  ) %>%
  mutate(
    TitleRelevancy = relevant_threads$TitleHasKeyword[match(Thread.ID, relevant_threads$Unique_ThreadID]
    ThreadRelevancyScore = ifelse(TitleRelevancy == 10, 10, AvgCommentRelevancy + TitleRelevancy)  # Pr
  )

# Step 7: Merge back thread relevancy into relevant_threads
relevant_threads <- relevant_threads %>%
  left_join(thread_relevancy, by = c("Unique_ThreadID" = "Thread.ID"))

# Step 8: Save the updated datasets
write.csv(relevant_threads, file.path(threads_directory, "dataset2_threads.csv"), row.names = FALSE)
write.csv(relevant_comments, file.path(comments_directory, "dataset2_comments.csv"), row.names = FALSE)
```

Dataset 3 - AIDS-Related Comments (1982–1986) - Filtered to Exclude Bottom Quartile of Relevancy Scores

```
# Load required libraries
library(dplyr)
library(ggplot2)
library(stringr)

# Step 1: Separate threads with TitleRelevancy == 10
high_title_relevance_threads <- relevant_threads %>%
  filter(TitleRelevancy == 10)

# Step 1: Refine Keyword Weights
# Adjust weights to prioritize AIDS-specific keywords
keyword_weights <- list(
```

```
  `10` = c("aids", "acquired immune deficiency syndrome", "hiv",
           "human immunodeficiency virus", "gay plague"),
  `9` = c("grid", "gay-related immune deficiency", "htlv",
          "gay cancer", "kaposi's sarcoma"),
  `8` = c("immune deficiency", "immune system", "opportunistic infection",
          "slim disease", "gay disease", "homosexual disease"),
  `7` = c("aids-related complex", "arc", "pneumocystis pneumonia",
          "aids hysteria", "gay compromise syndrome"),
  `6` = c("homophobia", "treatment", "transmission", "sexual transmission"),
  `5` = c("virus", "health", "disease", "infection", "blood test"),
  `3` = c("syndrome", "outbreak", "epidemic"),
  `1` = c("queer", "gay", "lesbian", "homosexuals", "unsafe practices")
)

# Flatten the keyword list into a named vector
keyword_weights <- unlist(lapply(names(keyword_weights), function(weight) {
  setNames(rep(as.numeric(weight), length(keyword_weights[[weight]])), keyword_weights[[weight]])
}))

# Step 2: Adjust Filtering for Comments
match_keywords <- function(text, keywords, weights) {
  text <- tolower(text)
  sapply(names(keywords), function(kw) {
    if (str_detect(text, paste0("\\b", kw, "\\b"))) {
      return(weights[kw])
    } else {
      return(0)
    }
  }) %>% sum()
}

filtered_relevant_comments <- relevant_comments %>%
  rowwise() %>%
  mutate(
    Relevancy = match_keywords(Full.Text, keyword_weights, keyword_weights),
    CompoundMatch = sum(str_detect(tolower(Full.Text), "\\b(aids|hiv)\\b")) > 0 &
                    sum(str_detect(tolower(Full.Text), "\\b(immune|treatment|virus|syndrome)\\b")) > 0
  ) %>%
  filter(Relevancy > 0 & CompoundMatch)

# Step 3: Filter Threads Based on Relevant Comments
filtered_relevant_threads <- relevant_threads %>%
  filter(Unique_ThreadID %in% filtered_relevant_comments$Thread.ID)

# Step 4: Apply Percentile-Based Filtering
percentiles <- quantile(filtered_relevant_threads$ThreadRelevancyScore, probs = seq(0, 1, by = 0.01), na
top_25th_percentile_cutoff <- quantile(filtered_relevant_threads$ThreadRelevancyScore, probs = 0.75, na

filtered_relevant_threads <- filtered_relevant_threads %>%
  filter(ThreadRelevancyScore >= top_25th_percentile_cutoff)

# Step 5: Add Back High-Title-Relevance Threads
filtered_relevant_threads <- bind_rows(filtered_relevant_threads,
```

```
                                        high_title_relevance_threads %>%
                                          filter(!Unique_ThreadID %in% filtered_relevant_threads$Unique_T

# Step 6: Filter Comments for Final Threads
filtered_relevant_comments <- relevant_comments %>%
  filter(Thread.ID %in% filtered_relevant_threads$Unique_ThreadID)

# Step 7: Save the Final Datasets
write.csv(filtered_relevant_threads,
          file.path(threads_directory, "dataset3_threads.csv"),
          row.names = FALSE)
write.csv(filtered_relevant_comments,
          file.path(comments_directory, "dataset3_comments.csv"),
          row.names = FALSE)

# Step 8: Print Summary Statistics
cat("Number of threads after enhanced filtering:", nrow(filtered_relevant_threads), "\n")
```

```
## Number of threads after enhanced filtering: 115
```

```
cat("Number of comments after enhanced filtering:", nrow(filtered_relevant_comments), "\n")
```

```
## Number of comments after enhanced filtering: 288
```

Dataset 4 - AIDS-Related Comments (1982–1986) - Filtered to Exclude Bottom Quartiles of Relevancy Scores and for Only Influential Authors Comments and Threads

```
# Load required libraries
library(dplyr)
library(igraph)
```

```
##
## Attaching package: 'igraph'

## The following objects are masked from 'package:lubridate':
##
##     %--%, union

## The following objects are masked from 'package:dplyr':
##
##     as_data_frame, groups, union

## The following objects are masked from 'package:purrr':
##
##     compose, simplify

## The following object is masked from 'package:tidyr':
##
##     crossing
```

```
## The following object is masked from 'package:tibble':
##
##     as_data_frame

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

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

```r
# Step 1: Filter comments in the top 16th percentile threads
# Ensure `filtered_relevant_comments` only includes comments from top 16th percentile threads
filtered_relevant_comments <- filtered_relevant_comments %>%
  filter(Thread.ID %in% filtered_relevant_threads$Unique_ThreadID)

# Step 2: Create an author co-participation network
author_pairs <- filtered_relevant_comments %>%
  filter(!is.na(Thread.ID)) %>%
  group_by(Thread.ID) %>%
  summarise(
    Pairs = list(if (length(unique(Author)) > 1) {
      as.data.frame(t(combn(unique(Author), 2)))
    } else {
      NULL
    })
  ) %>%
  unnest(Pairs, keep_empty = TRUE) %>%
  rename(Author1 = V1, Author2 = V2) %>%
  count(Author1, Author2, name = "Weight") %>%
  filter(!is.na(Author1) & !is.na(Author2))

# Step 3: Create a graph from the author pairs
author_network <- graph_from_data_frame(author_pairs, directed = FALSE)

# Step 4: Identify influential authors using degree centrality
degree_centrality <- strength(author_network, mode = "all", weights = E(author_network)$Weight)

# Create a data frame of authors and their influence scores
influential_authors <- data.frame(
  Author = names(degree_centrality),
  InfluenceScore = degree_centrality
) %>%
  arrange(desc(InfluenceScore)) %>%
  head(20) # Select the top 20 influential authors

# Save the list of influential authors
write.csv(influential_authors, file.path(output_directory, "/CSV Files/influential_authors.csv"), row.na

# Print the top influential authors
print(influential_authors)
```

```
##                                          Author InfluenceScore
```

```
## Craig Werner                                Craig Werner            59
## Ron Rizzo                                      Ron Rizzo            57
## Steve Dyer                                    Steve Dyer            44
## John Gurian                                  John Gurian            25
## Rob Bernardo                                Rob Bernardo            24
## Bill Stoll                                    Bill Stoll            19
## Bill Tanenbaum                            Bill Tanenbaum            17
## USENET News Administration USENET News Administration            17
## ems                                                  ems            17
## Rod Williams                                Rod Williams            16
## pam pincha                                    pam pincha            16
## Alan J Rosenthal                        Alan J Rosenthal            15
## Andrew Klossner                          Andrew Klossner            15
## Brian Mavrogeorge                      Brian Mavrogeorge            15
## David Sher                                    David Sher            15
## Harold Ancell                              Harold Ancell            15
## Mike Leibensperger                    Mike Leibensperger            15
## Ron Natalie <ron>                      Ron Natalie <ron>            15
## Roy Smith                                      Roy Smith            15
## stephanie da silva                    stephanie da silva            15
```

```r
# Step 5: Filter comments authored by influential authors
influential_author_comments <- filtered_relevant_comments %>%
  filter(Author %in% influential_authors$Author)

# Step 6: Identify threads with at least one influential author
influential_threads <- filtered_relevant_threads %>%
  filter(Unique_ThreadID %in% influential_author_comments$Thread.ID)

# Step 7: Include all comments in threads with influential authors
all_comments_in_influential_threads <- filtered_relevant_comments %>%
  filter(Thread.ID %in% influential_threads$Unique_ThreadID)

# Step 8: Save the final datasets
write.csv(influential_threads,
          file.path(threads_directory, "dataset4_threads.csv"),
          row.names = FALSE)
write.csv(all_comments_in_influential_threads,
          file.path(comments_directory, "dataset4_comments_all.csv"),
          row.names = FALSE)
write.csv(influential_author_comments,
          file.path(comments_directory, "dataset4_comments_onlyinfluential.csv"),
          row.names = FALSE)

# Step 9: Print summary statistics
cat("Number of threads involving influential authors:", nrow(influential_threads), "\n")
```

```
## Number of threads involving influential authors: 56
```

```r
cat("Number of comments in these threads (all comments):", nrow(all_comments_in_influential_threads), "
```

```
## Number of comments in these threads (all comments): 192
```

```r
cat("Number of comments by influential authors only:", nrow(influential_author_comments), "\n")
```

```
## Number of comments by influential authors only: 112
```

Descriptive Statistics Tables

Dataset 1 Descriptive Statistics

```r
library(sjPlot)

# Dataset 1: All Comments
dataset1_summary <- all_comments %>%
  group_by(newsgroup) %>%
  summarize(
    Threads = n_distinct(Thread.ID),
    Comments = n(),
    Authors = n_distinct(Author),
    Avg_Comments_Per_Thread = Comments / Threads,
    Avg_Sentiment_Score = mean(SentimentScore, na.rm = TRUE)
  )

# Add a totals row
dataset1_totals <- dataset1_summary %>%
  summarize(
    newsgroup = "Total",
    Threads = sum(Threads),
    Comments = sum(Comments),
    Authors = n_distinct(all_comments$Author),
    Avg_Comments_Per_Thread = sum(Comments) / sum(Threads),
    Avg_Sentiment_Score = mean(all_comments$SentimentScore, na.rm = TRUE)
  )

# Combine the summary with the totals row
dataset1_summary <- bind_rows(dataset1_summary, dataset1_totals)

# Save or print the summary
tab_df(dataset1_summary, file = paste0(output_directory, "/Images and Tables/Tables/dataset1_statistics
```

newsgroup

Threads

Comments

Authors

Avg_Comments_Per_Thread

Avg_Sentiment_Score

netmed

1439

3622

1365

2.52

-0.47

netmotss

1039

2394

674

2.30

0.69

netnews

2412

5219

1783

2.16

1.59

netpolitics

4059

13333

2477

3.28

-2.00

netreligion

2835

7016

1419

2.47

1.79

netsingles

3416

10290

2655

3.01

3.21

Total

15200

41874

7031

2.75

0.65

Dataset 2 Descriptive Statistics

```
dataset2_stats <- relevant_comments %>%
  group_by(newsgroup) %>%
  summarize(
    Related_Comments = n(),
    Related_Threads = n_distinct(Thread.ID),
    Total_Comments_in_Related_Threads = sum(n()),
    Unique_Authors_in_Related_Threads = n_distinct(Author),
    Percent_Comments_with_Keyword = mean(Relevancy > 0) * 100,
    Avg_Comment_Relevancy = mean(Relevancy, na.rm = TRUE),
    Avg_Thread_Relevancy = mean(relevant_threads$ThreadRelevancyScore[relevant_threads$newsgroup == firs
  )

# Add a totals row
dataset2_totals <- dataset2_stats %>%
  summarize(
    newsgroup = "Total",
    Related_Comments = sum(Related_Comments),
    Related_Threads = sum(Related_Threads),
    Total_Comments_in_Related_Threads = sum(Total_Comments_in_Related_Threads),
    Unique_Authors_in_Related_Threads = n_distinct(relevant_comments$Author),
    Percent_Comments_with_Keyword = mean(Percent_Comments_with_Keyword),
    Avg_Comment_Relevancy = mean(Avg_Comment_Relevancy),
    Avg_Thread_Relevancy = mean(Avg_Thread_Relevancy)
  )

dataset2_stats <- bind_rows(dataset2_stats, dataset2_totals)

# Save as HTML or CSV
tab_df(dataset2_stats, file = paste0(output_directory, "/Images and Tables/Tables/dataset2_statistics.ht
```

newsgroup

Related_Comments

Related_Threads

Total_Comments_in_Related_Threads

Unique_Authors_in_Related_Threads

Percent_Comments_with_Keyword

Avg_Comment_Relevancy

Avg_Thread_Relevancy

netmed

1752

403

1752

706

42.07

3.10

5.51

netmotss

1834

650

1834

568

78.30

4.53

5.81

netnews

358

44

358

209

16.48

0.80

3.58

netpolitics

3993

400

3993

1114

20.11

0.74

3.95

netreligion

1546

286

1546

473

35.19

1.02

2.94

netsingles

4896

713

4896

1545

36.85

0.72

1.95

Total

14379

2496

14379

3317

38.16

1.82

3.96

Dataset 3 Descriptive Statistics

```
dataset3_stats <- filtered_relevant_comments %>%
  group_by(newsgroup) %>%
  summarize(
    Related_Comments = n(),
    Related_Threads = n_distinct(Thread.ID),
    Total_Comments_in_Related_Threads = sum(n()),
    Unique_Authors_in_Related_Threads = n_distinct(Author),
    Percent_Comments_with_Keyword = mean(Relevancy > 0) * 100,
    Avg_Comment_Relevancy = mean(Relevancy, na.rm = TRUE),
    Avg_Thread_Relevancy = mean(filtered_relevant_threads$ThreadRelevancyScore[filtered_relevant_threads
  )

# Add a totals row
dataset3_totals <- dataset3_stats %>%
  summarize(
    newsgroup = "Total",
    Related_Comments = sum(Related_Comments),
    Related_Threads = sum(Related_Threads),
    Total_Comments_in_Related_Threads = sum(Total_Comments_in_Related_Threads),
    Unique_Authors_in_Related_Threads = n_distinct(filtered_relevant_comments$Author),
    Percent_Comments_with_Keyword = mean(Percent_Comments_with_Keyword),
```

```
    Avg_Comment_Relevancy = mean(Avg_Comment_Relevancy),
    Avg_Thread_Relevancy = mean(Avg_Thread_Relevancy)
  )

dataset3_stats <- bind_rows(dataset3_stats, dataset3_totals)

# Save as HTML or CSV
tab_df(dataset3_stats, file = paste0(output_directory, "/Images and Tables/Tables/dataset3_statistics.h
```

newsgroup

Related_Comments

Related_Threads

Total_Comments_in_Related_Threads

Unique_Authors_in_Related_Threads

Percent_Comments_with_Keyword

Avg_Comment_Relevancy

Avg_Thread_Relevancy

netmed

167

51

167

88

83.23

14.61

11.55

netmotss

101

55

101

64

92.08

17.75

14.19

netpolitics

6

1

6

6

16.67

2.17

13.00

netreligion

2

2

2

2

100.00

15.00

10.00

netsingles

12

6

12

10

83.33

16.08

10.67

Total

288

115

288

159

75.06

13.12

11.88

Dataset 4 Descriptive Statistics

```
library(dplyr)

# Generate descriptive statistics for Dataset 4
dataset4_stats <- all_comments_in_influential_threads %>%
  group_by(newsgroup) %>%
  summarize(
    Unique_Threads = n_distinct(Thread.ID),  # Unique threads with influential authors
    Total_Comments_All = n(),  # Total comments in relevant threads
    Influential_Comments = sum(Author %in% influential_authors$Author),  # Comments by influential auth
    Non_Influential_Comments = Total_Comments_All - Influential_Comments,  # Comments by non-influentia
    Avg_Comment_Relevancy_All = mean(Relevancy, na.rm = TRUE),  # Avg relevancy for all comments
```

```
    Avg_Comment_Relevancy_Influential = mean(Relevancy[Author %in% influential_authors$Author], na.rm =
    Avg_Comment_Relevancy_Non_Influential = mean(Relevancy[!(Author %in% influential_authors$Author)], r
    Percent_Comments_With_Keyword_All = mean(Relevancy > 0) * 100,  # Percent of all comments with keyw
    Percent_Comments_With_Keyword_Influential = mean(Relevancy[Author %in% influential_authors$Author] >
    Percent_Comments_With_Keyword_Non_Influential = mean(Relevancy[!(Author %in% influential_authors$Au
    Unique_Authors_in_Threads = n_distinct(Author),  # Unique authors in threads
    Avg_Thread_Relevancy = mean(influential_threads$ThreadRelevancyScore[influential_threads$threads$new
  )

# Add a totals row for all newsgroups combined
dataset4_totals <- dataset4_stats %>%
  summarize(
    newsgroup = "Total",
    Unique_Threads = sum(Unique_Threads),
    Total_Comments_All = sum(Total_Comments_All),
    Influential_Comments = sum(Influential_Comments),
    Non_Influential_Comments = sum(Non_Influential_Comments),
    Avg_Comment_Relevancy_All = mean(Avg_Comment_Relevancy_All, na.rm = TRUE),
    Avg_Comment_Relevancy_Influential = mean(Avg_Comment_Relevancy_Influential, na.rm = TRUE),
    Avg_Comment_Relevancy_Non_Influential = mean(Avg_Comment_Relevancy_Non_Influential, na.rm = TRUE),
    Percent_Comments_With_Keyword_All = mean(Percent_Comments_With_Keyword_All, na.rm = TRUE),
    Percent_Comments_With_Keyword_Influential = mean(Percent_Comments_With_Keyword_Influential, na.rm =
    Percent_Comments_With_Keyword_Non_Influential = mean(Percent_Comments_With_Keyword_Non_Influential,
    Unique_Authors_in_Threads = sum(Unique_Authors_in_Threads),
    Avg_Thread_Relevancy = mean(Avg_Thread_Relevancy, na.rm = TRUE)
  )

# Combine statistics and totals
dataset4_stats <- bind_rows(dataset4_stats, dataset4_totals)

# Save or display the table
tab_df(dataset4_stats, file = paste0(output_directory, "/Images and Tables/Tables/dataset4_statistics.h
```

newsgroup

Unique_Threads

Total_Comments_All

Influential_Comments

Non_Influential_Comments

Avg_Comment_Relevancy_All

Avg_Comment_Relevancy_Influential

Avg_Comment_Relevancy_Non_Influential

Percent_Comments_With_Keyword_All

Percent_Comments_With_Keyword_Influential

Percent_Comments_With_Keyword_Non_Influential

Unique_Authors_in_Threads

Avg_Thread_Relevancy

netmed

35

143

85

58

14.57

17.27

10.60

80.42

83.53

75.86

68

NaN

netmotss

21

49

27

22

17.37

18.81

15.59

93.88

96.30

90.91

26

NaN

Total

56

192

112

80

15.97

18.04

13.10

87.15

89.91

83.39

94

NaN

```
# Print the table
print(dataset4_stats)
```

```
## # A tibble: 3 x 13
##   newsgroup Unique_Threads Total_Comments_All Influential_Comments
##   <chr>              <int>              <int>                <int>
## 1 netmed                35                143                   85
## 2 netmotss              21                 49                   27
## 3 Total                 56                192                  112
## # i 9 more variables: Non_Influential_Comments <int>,
## #   Avg_Comment_Relevancy_All <dbl>, Avg_Comment_Relevancy_Influential <dbl>,
## #   Avg_Comment_Relevancy_Non_Influential <dbl>,
## #   Percent_Comments_With_Keyword_All <dbl>,
## #   Percent_Comments_With_Keyword_Influential <dbl>,
## #   Percent_Comments_With_Keyword_Non_Influential <dbl>,
## #   Unique_Authors_in_Threads <int>, Avg_Thread_Relevancy <dbl>
```

Dataset 4 Author Statistics

```
# Filter influential authors' participation in Dataset Three
dataset3_influential_comments <- filtered_relevant_comments %>%
  filter(Author %in% influential_authors$Author)

dataset3_influential_threads <- filtered_relevant_threads %>%
  filter(Unique_ThreadID %in% dataset3_influential_comments$Thread.ID)

# Add descriptive statistics for each influential author
author_stats <- dataset3_influential_comments %>%
  group_by(Author) %>%
  summarise(
    Influence = unique(influential_authors$Influence[influential_authors$Author == Author]),
    Num_Comments = n_distinct(NG_Relative_CommentID),  # Number of comments authored
    Num_Threads = n_distinct(Thread.ID),  # Number of threads participated in
    Threads_Started = sum(Comment.ID == "CM00001", na.rm = TRUE),  # Threads started
    Avg_Sentiment = mean(SentimentScore, na.rm = TRUE),  # Average sentiment score
    Avg_Relevancy = mean(Relevancy, na.rm = TRUE)  # Average relevancy score
  )
```

```
## Warning: There were 4 warnings in `summarise()`.
## The first warning was:
## i In argument: `Influence = unique(...)`.
## i In group 6: `Author = "Craig Werner"`.
## Caused by warning in `influential_authors$Author == Author`:
## ! longer object length is not a multiple of shorter object length
## i Run `dplyr::last_dplyr_warnings()` to see the 3 remaining warnings.
```

```
## Warning: Returning more (or less) than 1 row per `summarise()` group was deprecated in
```

```
## dplyr 1.1.0.
## i Please use ‘reframe()‘ instead.
## i When switching from ‘summarise()‘ to ‘reframe()‘, remember that ‘reframe()‘
##   always returns an ungrouped data frame and adjust accordingly.
## Call ‘lifecycle::last_lifecycle_warnings()‘ to see where this warning was
## generated.

## ‘summarise()‘ has grouped output by ’Author’. You can override using the
## ‘.groups‘ argument.
```

```
# Filter out authors with missing Influence scores and sort by Influence
author_stats <- author_stats %>%
  filter(!is.na(Influence)) %>%
  arrange(desc(Influence))

# Save the table as an HTML file
tab_df(author_stats, file = paste0(output_directory, "/Images and Tables/Tables/dataset4_author_statisti
```

Author

Influence

Num_Comments

Num_Threads

Threads_Started

Avg_Sentiment

Avg_Relevancy

Craig Werner

59

26

24

17

-3.31

21.35

Ron Rizzo

57

28

21

14

-7.25

23.79

Steve Dyer

44

14

14

4

-3.00

17.07

John Gurian

25

4

2

0

1.00

6.25

Rob Bernardo

24

10

8

3

-4.20

14.10

Bill Stoll

19

4

3

2

-13.50

15.75

Bill Tanenbaum

17

4

2

0

-5.50

10.00

USENET News Administration

17

2

2

1

-4.00

5.00

ems

17

5

2

0

-4.60

4.80

Rod Williams

16

3

3

2

0.00

13.67

pam pincha

16

2

2

0

-2.00

17.50

Alan J Rosenthal

15

1

1

0

-3.00

10.00

Andrew Klossner

15

1

1

0

-7.00

18.00

Brian Mavrogeorge

15

2

1

0

-4.50

16.50

David Sher

15

1

1

0

-9.00

10.00

Harold Ancell

15

1

1

0

-6.00

24.00

Mike Leibensperger

15

1

1

0

9.00

10.00

Ron Natalie

15

1

1

0

-11.00

12.00

Roy Smith

15

1

1

0

4.00

10.00

stephanie da silva

15

1

1

0

0.00

10.00

```
# Display the table
print(author_stats)
```

```
## # A tibble: 20 x 7
## # Groups:   Author [20]
##    Author       Influence Num_Comments Num_Threads Threads_Started Avg_Sentiment
##    <chr>            <dbl>        <int>       <int>           <int>         <dbl>
##  1 Craig Werner        59           26          24              17         -3.31
##  2 Ron Rizzo           57           28          21              14         -7.25
##  3 Steve Dyer          44           14          14               4         -3
##  4 John Gurian         25            4           2               0          1
##  5 Rob Bernardo        24           10           8               3         -4.2
##  6 Bill Stoll          19            4           3               2        -13.5
##  7 Bill Tanenb~        17            4           2               0         -5.5
##  8 USENET News~        17            2           2               1         -4
##  9 ems                 17            5           2               0         -4.6
## 10 Rod Williams        16            3           3               2          0
## 11 pam pincha          16            2           2               0         -2
## 12 Alan J Rose~        15            1           1               0         -3
## 13 Andrew Klos~        15            1           1               0         -7
## 14 Brian Mavro~        15            2           1               0         -4.5
## 15 David Sher          15            1           1               0         -9
## 16 Harold Ance~        15            1           1               0         -6
## 17 Mike Leiben~        15            1           1               0          9
## 18 Ron Natalie~        15            1           1               0        -11
## 19 Roy Smith           15            1           1               0          4
## 20 stephanie d~        15            1           1               0          0
## # i 1 more variable: Avg_Relevancy <dbl>
```