DataSci 306, Homework 7

Max Han, maxhan

```
knitr::opts_chunk$set(echo = TRUE)
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(lubridate)
library(harrypotter)
library(tidytext)
```

Problem 1: Harry Potter data analysis (4 points)

Unless specified otherwise, all matches are case insensitive.

In problem 1 we will perform sentiment analysis of the Harry Potter books. The file `afinn.RData` contains a sentiment score for a large number of words in the English language:

```
load(url("https://datasets.stats306.org/afinn.RData"))
head(afinn)
```

```
## # A tibble: 6 x 2
##   word       value
##   <chr>      <dbl>
## 1 abandon       -2
## 2 abandoned     -2
## 3 abandons      -2
## 4 abducted      -2
## 5 abduction     -2
## 6 abductions    -2
```

Negatively connoted words receive low scores, while positively connoted words receive high scores:

```
filter(afinn, word %in% c("death", "hurrah"))
```

```
## # A tibble: 2 x 2
##   word   value
##   <chr>  <dbl>
## 1 death     -2
## 2 hurrah     5
```

The `tidytext::unnest_tokens()` function can be used to break a chunk of text into "tokens" (words, sentences, etc.) It works as follows. Consider the following tibble, which contains all chapters of the first book in the Harry Potter series:

```
chamber_tbl <- tibble(chapter = seq_along(chamber_of_secrets),
                      text = chamber_of_secrets) |> print()
```

```
## # A tibble: 19 x 2
##    chapter text
##      <int> <chr>
##  1       1 "THE WORST BIRTHDAY    Not for the first time, an argument had broke~
##  2       2 "    DOBBY'S WARNING    arry managed not to shout out, but it was a ~
##  3       3 "THE BURROW    Ron.l\" breathed Harry, creeping to the window and pu~
##  4       4 "AT FLOVRR 11 $ HAND BLOTTS    ife at the Burrow was as different as~
##  5       5 "THE WHOMPING    WILLOW    he end of the summer vacation came too qu~
##  6       6 "GILDEROY LOCKHART    he next day, however, Harry barely grinned onc~
##  7       7 "Harry looked bemusedly at the photograph Colin was brandishing unde~
##  8       8 "    \"What are you talking about, Harry? Perhaps you're getting a l~
##  9       9 "THE WRTITING ON THE WALL    What's going on here? What's going on?\~
## 10      10 "    THE ROGUE BLUDGER    ince the disastrous episode of the pixies,~
## 11      11 "    THE D-KJEL]ING C-L-IJIB    Harry woke up on Sunday morning to f~
## 12      12 "    THE POLYJUICE POTION    hey stepped off the stone staircase at ~
## 13      13 "    Malfoy started taking pictures with an imaginary camera and did~
## 14      14 "still, heart-shaped confetti was falling from the pale blue ceiling~
## 15      15 "Dippet sank back, looking faintly disappointed.    \"You may go, To~
## 16      16 "    \"The appointment - or suspension - of the headmaster is a matt~
## 17      17 "stood, terrified, waiting. There was a strange rumbling noise and t~
## 18      18 "    \"Right,\" said Professor McGonagall, whose nostrils were flare~
## 19      19 "    \"What d'you mean, I won't be -?\"    \"I've waited a long time~
```

To perform sentiment analysis, we need to break each chapter into words so that we can join it to the afinn table.

This is what `unnest_tokens()` does:

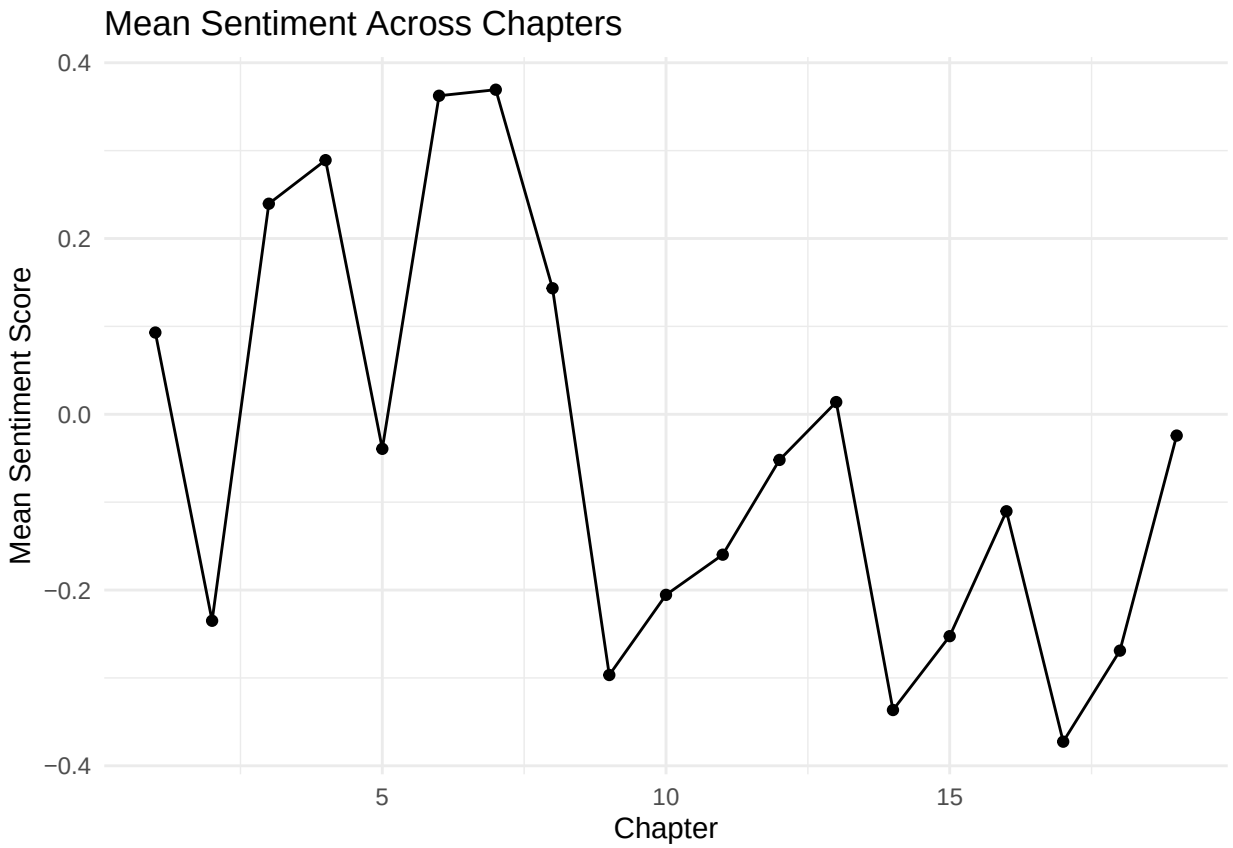
```
# break sentences into words
chamber_tok <- unnest_tokens(chamber_tbl, input = text, output = word) |> print()
```

```
## # A tibble: 85,408 x 2
##    chapter word
##      <int> <chr>
##  1       1 the
##  2       1 worst
##  3       1 birthday
##  4       1 not
##  5       1 for
##  6       1 the
##  7       1 first
##  8       1 time
##  9       1 an
## 10       1 argument
## # i 85,398 more rows
```

1(a) By joining this table to other tables containing text data and summarizing, we can generate scores of how positive or negative the text is. Using the table and afinn, we can assign sentiment scores to various portions of text. Generate a plot reflecting how the `mean` sentiment changes across all the chapters of the above book in the Harry Potter series. What conclusion can you draw from the plot? (1 point)

In the plot, most average value is under zero. Thus, we can conclude the overall sentiment is dark.

```
# your solution
chamber_tok|> left_join(afinn, join_by(word)) |> group_by(chapter) |> summarise(mean = mean(value, na.r
  ggplot(aes(x = chapter, y = mean)) +
  geom_line() +
  geom_point() +
  labs(title = "Mean Sentiment Across Chapters",
       x = "Chapter",
       y = "Mean Sentiment Score") +
  theme_minimal()
```



1(b) Some people say that the Harry Potter books became darker (more negative) over time. Use sentiment analysis to investigate this, and report your conclusion here. (2 points)

HINT: Run the following code to obtain a list of all the Harry Potter books under the `harrypotter` package.

According to the following plot, the books are not getting dark over time. They are just dark over all.

```
# help(package = "harrypotter")

phil_tbl <- tibble(chapter = seq_along(philosophers_stone),
                   text = philosophers_stone)
prisoner_tbl <- tibble(chapter = seq_along(prisoner_of_azkaban),
                   text = prisoner_of_azkaban)
goblet_tbl <- tibble(chapter = seq_along(goblet_of_fire),
                   text = goblet_of_fire)
phoenix_tbl <- tibble(chapter = seq_along(order_of_the_phoenix),
                   text = order_of_the_phoenix)
```

```r
prince_tbl <- tibble(chapter = seq_along(half_blood_prince),
                     text = half_blood_prince)
hallows_tbl <- tibble(chapter = seq_along(deathly_hallows),
                     text = deathly_hallows)

all_books <- bind_rows(
  phil_tbl |> mutate(book = "Philosopher's Stone"),
  chamber_tbl |> mutate(book = 'Chamber of Secrets'),
  prisoner_tbl |> mutate(book = "Prisoner of Azkaban"),
  goblet_tbl |> mutate(book = "Goblet of Fire"),
  phoenix_tbl |> mutate(book = "Order of the Phoenix"),
  prince_tbl |> mutate(book = "Half-Blood Prince"),
  hallows_tbl |> mutate(book = "Deathly Hallows")
) |> print()
```

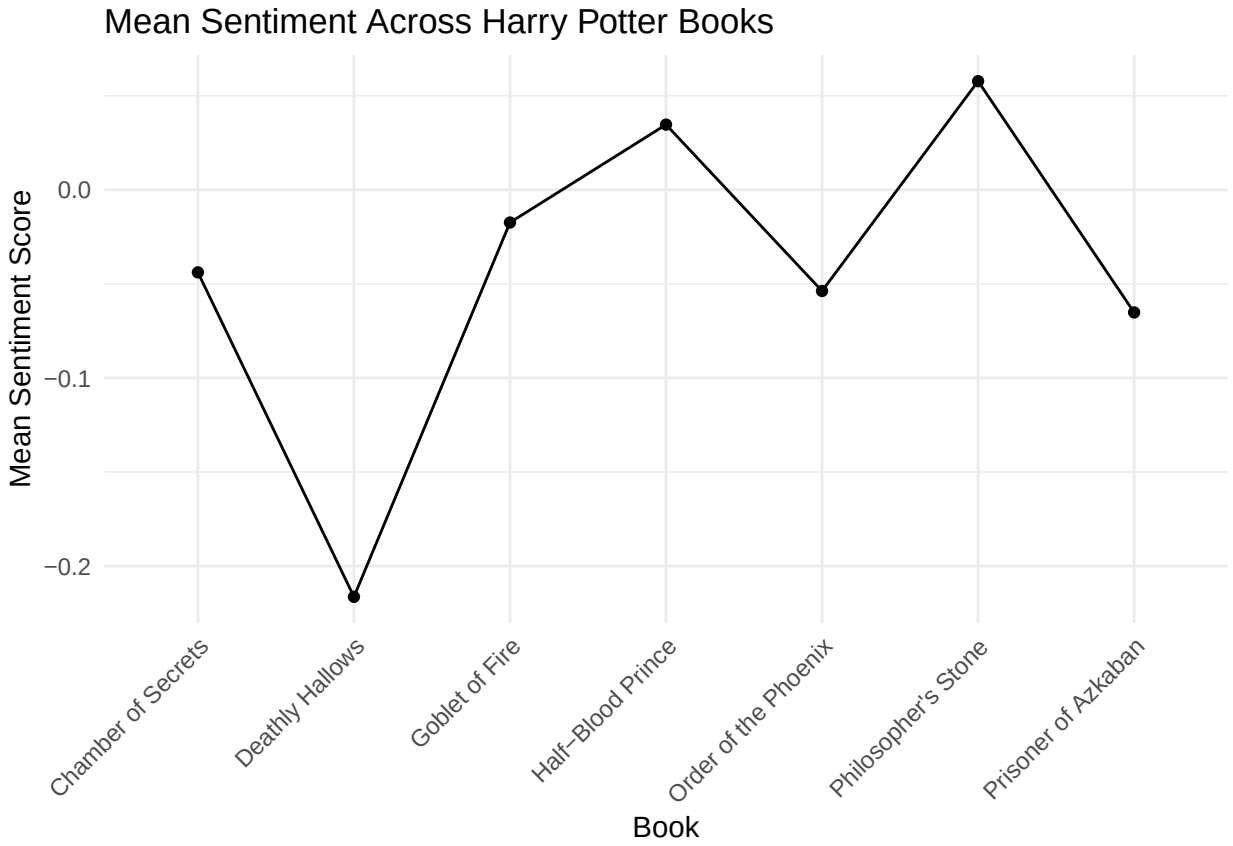
```
## # A tibble: 200 x 3
##    chapter text                                                      book
##      <int> <chr>                                                     <chr>
##  1       1 "THE BOY WHO LIVED    Mr. and Mrs. Dursley, of number four, Pr~ Phil~
##  2       2 "THE VANISHING GLASS    Nearly ten years had passed since the ~ Phil~
##  3       3 "THE LETTERS FROM NO ONE    The escape of the Brazilian boa co~ Phil~
##  4       4 "THE KEEPER OF THE KEYS    BOOM. They knocked again. Dudley je~ Phil~
##  5       5 "DIAGON ALLEY    Harry woke early the next morning. Although h~ Phil~
##  6       6 "THE JOURNEY FROM PLATFORM NINE AND THREE-QUARTERS    Harry's ~ Phil~
##  7       7 "THE SORTING HAT    The door swung open at once. A tall, black~ Phil~
##  8       8 "THE POTIONS MASTER    There, look.\"    \"Where?\"    \"Next ~ Phil~
##  9       9 "THE MIDNIGHT DUEL    Harry had never believed he would meet a~ Phil~
## 10      10 "HALLOWEEN    Malfoy couldn't believe his eyes when he saw tha~ Phil~
## # i 190 more rows
```

```r
# your solution
all_books_tok <- unnest_tokens(all_books, input = text, output = word)

all_books_sentiment <- all_books_tok %>%
  inner_join(afinn, by = "word")

book_sentiment <- all_books_sentiment %>%
  group_by(book) %>%
  summarize(mean_sentiment = mean(value, na.rm = TRUE)) %>%
  arrange(match(book, c("Philosopher's Stone", "Chamber of Secrets", "Prisoner of Azkaban",
                        "Goblet of Fire", "Order of the Phoenix", "Half-Blood Prince", "Deathly Hallows

ggplot(book_sentiment, aes(x = book, y = mean_sentiment)) +
  geom_line(group = 1) +
  geom_point() +
  labs(title = "Mean Sentiment Across Harry Potter Books",
       x = "Book",
       y = "Mean Sentiment Score") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



1(c) Extract the proper nouns from chapter 1 of the philosophers_stone and display frequency of these names in descending order. To keep things simple, we will extract all words that start with upper case and is at least 5 characters long (1 point)

```
chapter1 = philosophers_stone[1]
target_words <- str_extract_all(chapter1, "\\b[A-Z]\\w{4,}")
tibble(words = unlist(target_words)) |> group_by(words) |> summarise(count = n())
```

```
## # A tibble: 81 x 2
##    words    count
##    <chr>    <int>
##  1 About        2
##  2 After        2
##  3 Albus        3
##  4 Although     1
##  5 Black        1
##  6 Bonfire      1
##  7 Borrowed     1
##  8 Bristol      1
##  9 Britain      1
## 10 Could        2
## # i 71 more rows
```

Problem 2: Reddit dataset (6 points)

The file reddit_xmas_2017.RData contains 100,000 comments posted to Reddit on Christmas Day, 2017. Unless specified otherwise, all matches are case insensitive.

```
load(url('https://datasets.stats306.org/reddit_xmas_2017.RData'))
reddit |> print()
```

```
## # A tibble: 100,000 x 3
##    author              body                                  created_utc
##    <chr>               <chr>                                 <dttm>
##  1 br_shadow           "Thank you for this, there is a pers~ 2017-12-25 15:49:08
##  2 Ksalol              "They are not to quick actually. It'~ 2017-12-25 17:42:50
##  3 itscool83           "tell her you guys should hang out w~ 2017-12-25 18:54:13
##  4 Glu7enFree          "Autism is a high honor in the tech ~ 2017-12-25 07:48:17
##  5 Theotheogreato      "You thought a cat was your son?! "   2017-12-25 20:58:08
##  6 Shadrac121          "Hopfully she takes wat people say i~ 2017-12-25 22:27:31
##  7 1fzUjhemoSB1QV7zI7  "Si ce propui sa facem cu toata piel~ 2017-12-25 07:41:31
##  8 MinisterOfEducation "I don't mean to be impolite, but if~ 2017-12-25 19:28:35
##  9 AabidS10            "i dont have a 720p x265 of it, sorr~ 2017-12-25 13:20:32
## 10 S3RG10              "I'm dying to try Guatemalan sandals~ 2017-12-25 00:48:46
## # i 99,990 more rows
```

2(a) What are other people wishing? Count the first occurrence of the string `Happy <word>` or `Merry <word>` (case insensitive) in the comment body, if any, count the matches. To keep things interesting, do not include phrases matching (happy|merry) (to|with|for|about|and|that|if|i|you|when). (2 points)

Print a table containing the top 10 matches; a few of the rows are:

greeting	n
merry christmas	2040
happy holidays	-

```
greetings <- reddit |>
  mutate(
    greeting = str_extract(body, regex("\\b(happy|merry)\\s+\\w+", ignore_case = TRUE))
  ) |>
  filter(!str_detect(greeting, regex("\\b(happy|merry)\\s+(to|with|for|about|and|that|if|i|you|when)\\b
  mutate(greeting = str_to_lower(greeting)) |>
  count(greeting, sort = TRUE)

top_10_greetings <- greetings |>
  slice_head(n = 10)

print(top_10_greetings)
```

```
## # A tibble: 10 x 2
##    greeting            n
##    <chr>           <int>
##  1 merry christmas  2040
##  2 happy holidays    477
##  3 merry xmas         93
##  4 happy christmas    44
##  5 happy cake         28
##  6 happy birthday     23
##  7 happy new          15
##  8 happy holiday      13
##  9 merry x             7
## 10 happy cakeday       6
```

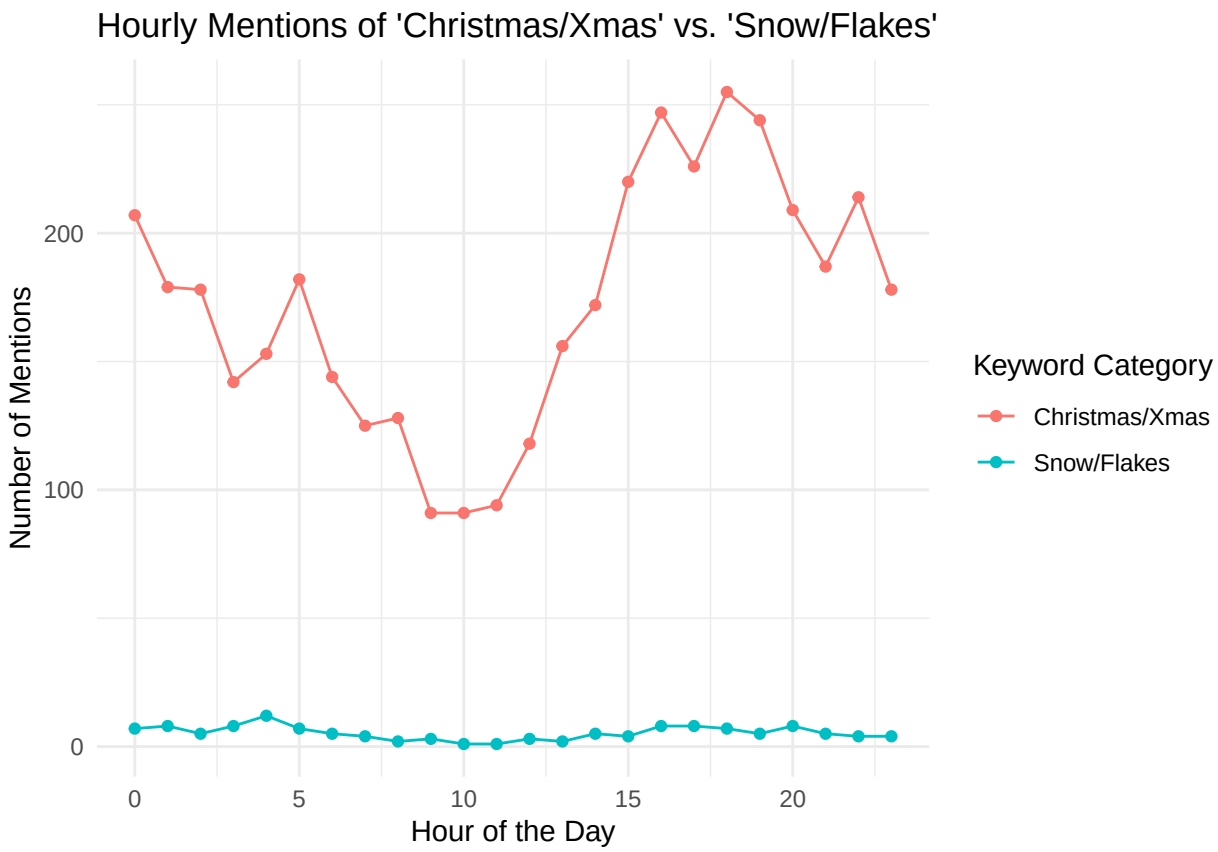
2(b) Find the number of times christmas or xmas is mentioned each hour. Similarly, find the number of mentions per hour of `snow` or `flakes`. Draw a plot comparing these two time series. (2 points)

```
reddit <- reddit |>
  mutate(timestamp = as_datetime(created_utc))

mentions <- reddit |>
  mutate(hour = hour(timestamp)) |>
  filter(str_detect(body, regex("\\b(christmas|xmas)\\b", ignore_case = TRUE)) |
         str_detect(body, regex("\\b(snow|flakes)\\b", ignore_case = TRUE))) |>
  mutate(
    keyword = case_when(
      str_detect(body, regex("\\b(christmas|xmas)\\b", ignore_case = TRUE)) ~ "Christmas/Xmas",
      str_detect(body, regex("\\b(snow|flakes)\\b", ignore_case = TRUE)) ~ "Snow/Flakes",
      TRUE ~ NA_character_
    )
  ) |>
  filter(!is.na(keyword))

hourly_mentions <- mentions |>
  group_by(hour, keyword) |>
  summarize(count = n(), .groups = "drop")

ggplot(hourly_mentions, aes(x = hour, y = count, color = keyword)) +
  geom_line() +
  geom_point() +
  labs(
    title = "Hourly Mentions of 'Christmas/Xmas' vs. 'Snow/Flakes'",
    x = "Hour of the Day",
    y = "Number of Mentions",
    color = "Keyword Category"
  ) +
  theme_minimal()
```



2(c) Using `afinn` dataset, calculate the average sentiment scores of reddit comments for each hour. When is the most positive time in Christmas Day? (2 points)

```
reddit <- reddit |>
  mutate(timestamp = as_datetime(created_utc),
         hour = hour(created_utc))

reddit_sentiment <- reddit |>
  unnest_tokens(word, body) |>
  inner_join(afinn, by = "word") |>
  group_by(hour) |>
  summarize(avg_sentiment = mean(value, na.rm = TRUE), .groups = "drop")

most_positive_hour <- reddit_sentiment |>
  filter(avg_sentiment == max(avg_sentiment))

print(reddit_sentiment)
```

```
## # A tibble: 24 x 2
##     hour avg_sentiment
##    <int>         <dbl>
##  1     0         0.932
##  2     1         0.913
##  3     2         0.911
##  4     3         0.929
##  5     4         0.988
##  6     5         0.966
##  7     6         0.910
```

```
##  8     7         0.917
##  9     8         0.908
## 10     9         0.868
## # i 14 more rows
```

```
print(most_positive_hour)
```

```
## # A tibble: 1 x 2
##    hour avg_sentiment
##   <int>         <dbl>
## 1     4         0.988
```

```
ggplot(reddit_sentiment, aes(x = hour, y = avg_sentiment)) +
  geom_line(color = "blue") +
  geom_point() +
  labs(
    title = "Average Sentiment Score of Reddit Comments by Hour on Christmas Day",
    x = "Hour of the Day",
    y = "Average Sentiment Score"
  ) +
  theme_minimal()
```