SDS 264 HW3

On Your Own: Harry Potter

The `potter_untidy` dataset includes the text of 7 books of the Harry Potter series by J.K. Rowling. For a brief overview of the books (or movies), see this quote from Wikipedia:

> Harry Potter is a series of seven fantasy novels written by British author J. K. Rowling. The novels chronicle the lives of a young wizard, Harry Potter, and his friends Hermione Granger and Ron Weasley, all of whom are students at Hogwarts School of Witchcraft and Wizardry. The main story arc concerns Harry's conflict with Lord Voldemort, a dark wizard who intends to become immortal, overthrow the wizard governing body known as the Ministry of Magic, and subjugate all wizards and Muggles (non-magical people).

A few analyses from SDS 164:

New stuff!

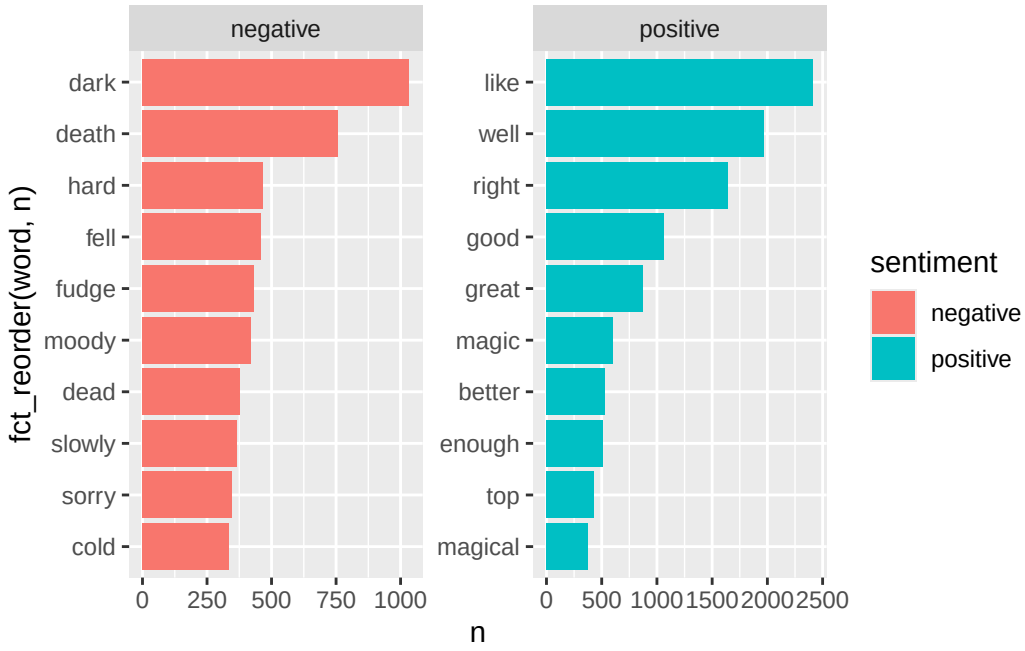
1. What words contribute the most to negative and positive sentiment scores? Show a faceted bar plot of the top 10 negative and the top 10 positive words (according to the "bing" lexicon) across the entire series.

```
bing_sentiments <- get_sentiments(lexicon = "bing")
# potter_tidy |>
#   inner_join(bing_sentiments, relationship = "many-to-many") |>
#   count(sentiment, word, sort = TRUE) |>
#   group_by(sentiment) |>
#   slice_max(n, n=10) |>
#   ungroup() |>
#   ggplot(aes(x = fct_reorder(word, n), y = n, fill = sentiment)) +
#   geom_col() +
#   coord_flip() +
```

```
#   facet_wrap(~ sentiment, scales = "free")

potter_tidy |>
  inner_join(bing_sentiments, relationship = "many-to-many") |>
  count(sentiment, word, sort = TRUE) |>
  group_by(sentiment) |>
  slice_max(n, n=10) |>
  ungroup() |>
  ggplot(aes(x = fct_reorder(word, n), y = n, fill = sentiment)) +
  geom_bar(stat = "identity") +
  facet_wrap(~ sentiment, scales = "free") +
  coord_flip()
```

```
Joining with `by = join_by(word)`
```



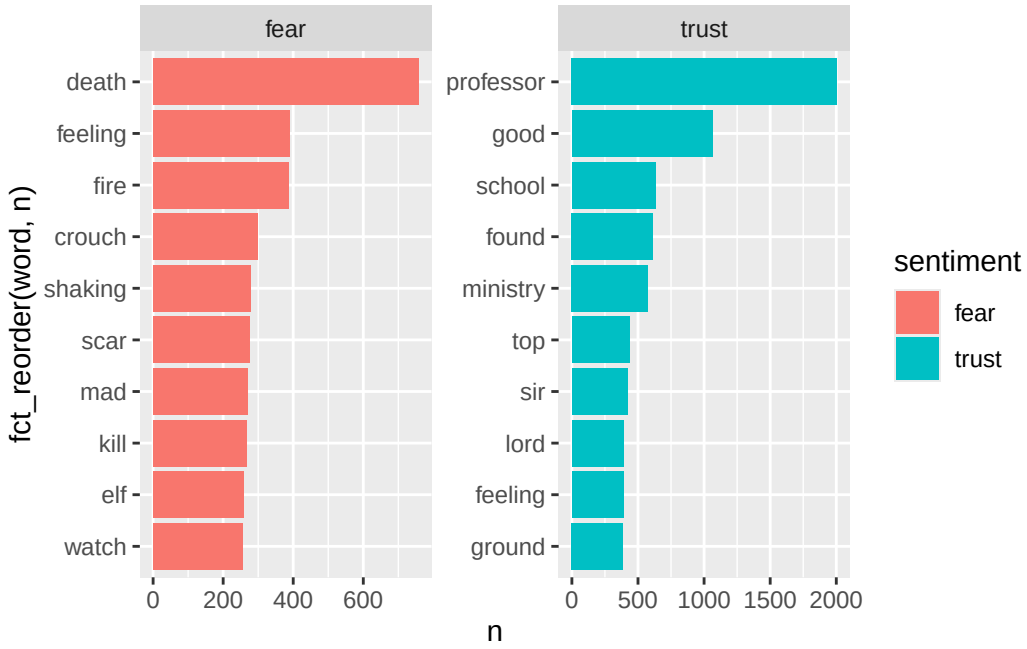
2. Find a list of the top 10 words associated with “fear” and with “trust” (according to the “nrc” lexicon) across the entire series.

```
nrc_sentiments <- get_sentiments(lexicon = "nrc")

nrc_sentiments |>
```

```
  filter(sentiment == "fear" | sentiment == "trust") |>
  inner_join(potter_tidy, relationship = "many-to-many") |>
  count(sentiment, word, sort = TRUE) |>
  group_by(sentiment) |>
  slice_max(n, n=10) |>
  ungroup() |>
  ggplot(aes(x = fct_reorder(word, n), y = n, fill = sentiment)) +
  geom_bar(stat = "identity") +
  facet_wrap(~ sentiment, scales = "free") +
  coord_flip()
```

```
Joining with `by = join_by(word)`
```



3. Make a wordcloud for the entire series after removing stop words using the "smart" source.

```
smart_stopwords <- get_stopwords(source = "smart")

potter_words <- potter_tidy |>
  anti_join(smart_stopwords) |>
  anti_join(potter_names, join_by(word == firstname)) |>
```

```
  anti_join(potter_names, join_by(word == lastname)) |>
  count(word) |>
  arrange(desc(n))
```

```
Joining with `by = join_by(word)`
```

```
set.seed(1234)
wordcloud(
    words = potter_words$word,
    freq = potter_words$n,
    scale=c(3,0.3),
    max.words = 100,
    random.order = FALSE,
    rot.per = 0,
    colors = brewer.pal(4, "Dark2"))
```

4. Create a wordcloud with the top 20 negative words and the top 20 positive words in the Harry Potter series according to the bing lexicon. The words should be sized by their respective counts and colored based on whether their sentiment is positive or negative. (Feel free to be resourceful and creative to color words by a third variable!)

```
sent_potter <- bing_sentiments |>
  inner_join(potter_tidy, relationship = "many-to-many") |>
  count(sentiment, word, sort = TRUE) |>
  group_by(sentiment) |>
  slice_max(n, n=20) |>
  mutate(
    sentiment = ifelse(sentiment == "positive", "green", "red")
  )
```

```
Joining with `by = join_by(word)`
```

```
wordcloud(
    words = sent_potter$word,
    freq = sent_potter$n,
    random.order = FALSE,
    max.words = 100,
    ordered.colors = TRUE,
    color = sent_potter$sentiment)
```


```
# sent_potter <- potter_tidy |>
#   inner_join(bing_sentiments, relationship = "many-to-many") |>
```

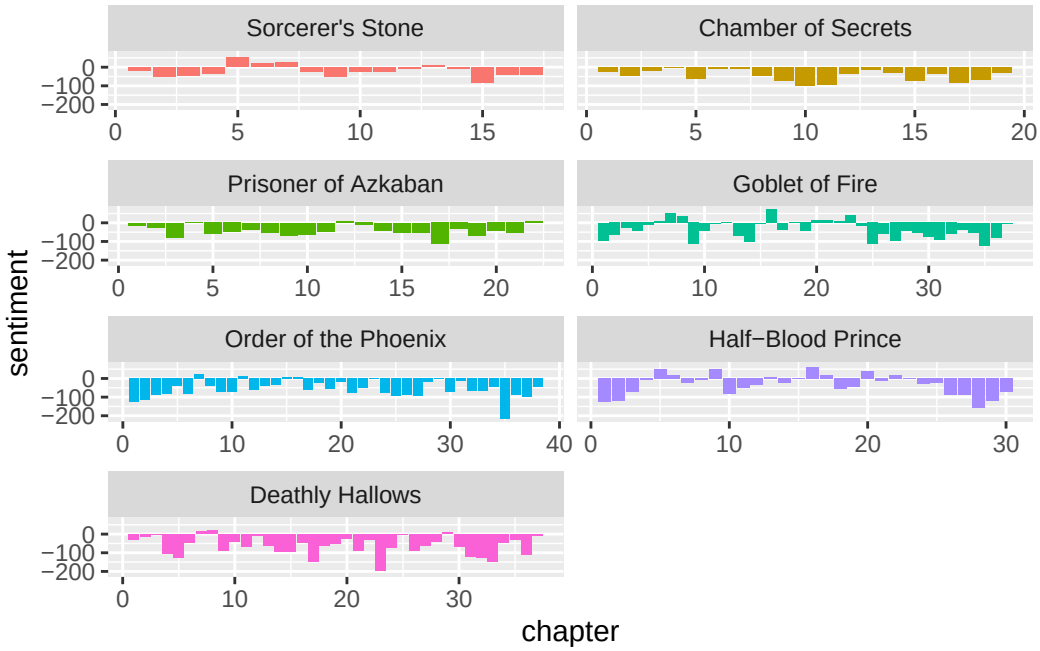
```
#   count(sentiment, word, sort = TRUE) |>
#   group_by(sentiment) |>
#   slice_max(n, n=20)
#
# set.seed(1234)
# wordcloud(
#     words = sent_potter$word,
#     freq = sent_potter$n,
#     random.order = FALSE,
#     ordered.colors = TRUE,
#     color = brewer.pal(4, "Dark2")[factor(sent_potter$sentiment)])
```

5. Make a faceted bar chart to compare the positive/negative sentiment trajectory over the 7 Harry Potter books. You should have one bar per chapter (thus chapter becomes the index), and the bar should extend up from 0 if there are more positive than negative words in a chapter (according to the "bing" lexicon), and it will extend down from 0 if there are more negative than positive words.

```
bing_sentiments <- get_sentiments(lexicon = "bing")

bing_sentiments |>
  inner_join(potter_tidy, relationship = "many-to-many") |>
  count(title, chapter, sentiment, sort = TRUE) |>
  pivot_wider(names_from = sentiment, values_from = n, values_fill = 0) |>
  mutate(sentiment = positive - negative) |>
  ggplot(aes(x = chapter, y = sentiment, fill = title)) +
  geom_col(show.legend = FALSE) +
    facet_wrap(~title, ncol = 2, scales = "free_x")
```

```
Joining with `by = join_by(word)`
```

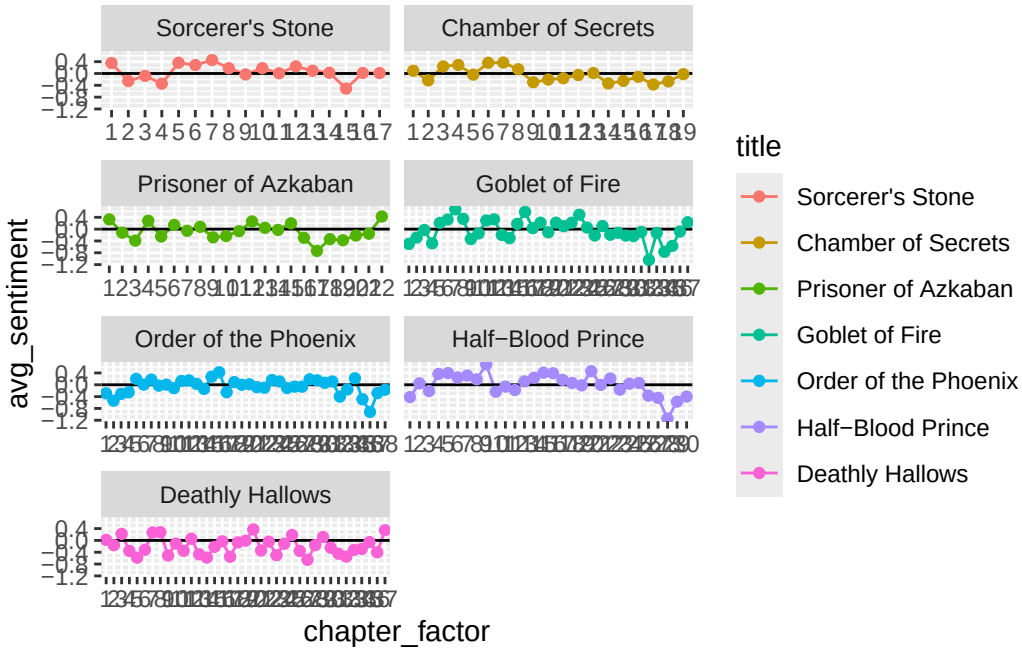


6. Repeat (5) using a faceted scatterplot to show the average sentiment score according to the "afinn" lexicon for each chapter. (Hint: use `mutate(chapter_factor = factor(chapter))` to treat chapter as a factor variable.)

```
afinn_sentiments <- get_sentiments("afinn")

afinn_sentiments |>
  inner_join(potter_tidy, relationship = "many-to-many") |>
  group_by(title, chapter) |>
  summarize(avg_sentiment = mean(value)) |>
  mutate(chapter_factor = factor(chapter)) |>
  ggplot(aes(x = chapter_factor, y = avg_sentiment, color = title, group = title)) +
  geom_line() +
  geom_hline(yintercept = 0) +
  geom_point() +
    facet_wrap(~title, ncol = 2, scales = "free_x")
```

```
Joining with `by = join_by(word)`
`summarise()` has grouped output by 'title'. You can override using the
`.groups` argument.
```



7. Make a faceted bar plot showing the top 10 words that distinguish each book according to the tf-idf statistic.

```
potter_tfidf <- potter_tidy |>
  group_by(title) |>
  count(word) |>
  bind_tf_idf(word, title, n) |>
  arrange(-tf_idf)
```

```
potter_tfidf |>
  group_by(title) |>
  slice_max(tf_idf, n = 10) |>
  ungroup() |>
  ggplot(aes(x = fct_reorder(word, tf_idf), y = tf_idf, fill = title)) +
    geom_col(show.legend = FALSE) +
    coord_flip() +
    facet_wrap(~title, scales = "free")
```

8. Repeat (7) to show the top 10 2-word combinations that distinguish each book.

```
potter_bigram <- potter_untidy |>
  unnest_tokens(bigram, text, token = "ngrams", n = 2) |>
  filter(bigram != "NA")

potter_bigram_filtered <- potter_bigram |>
  separate(bigram, c("word1", "word2"), sep = " ") |>
  filter(!word1 %in% stop_words$word,
         !word2 %in% stop_words$word) |>
  count(word1, word2, sort = TRUE) |>
  unite(bigram, word1, word2, sep = " ")

potter_bigram_filtered
```

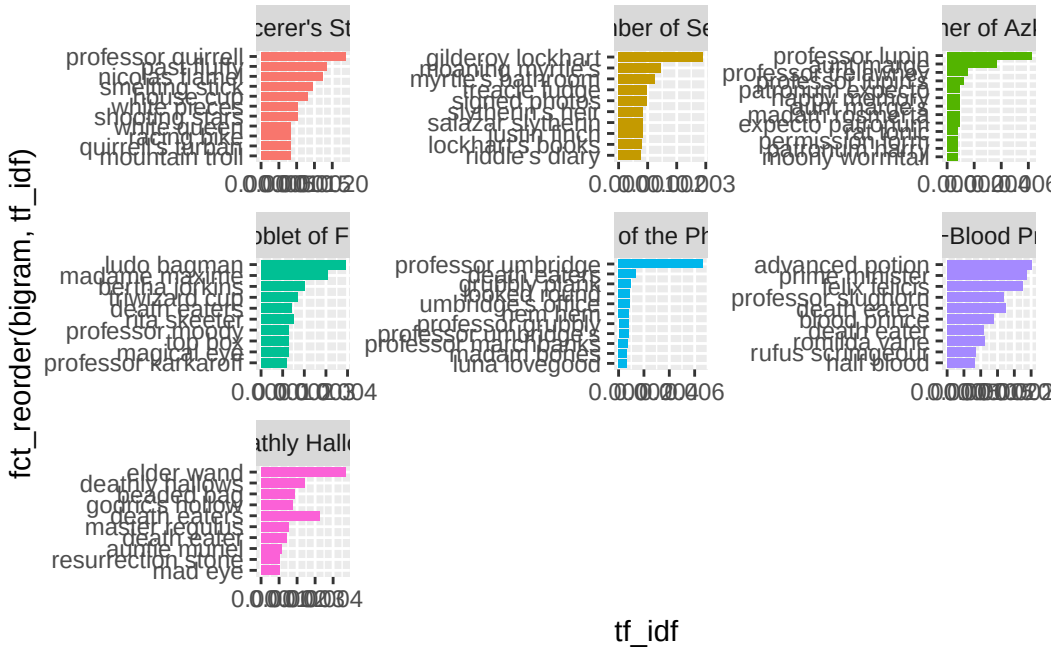
```
# A tibble: 89,258 x 2
   bigram                    n
   <chr>                 <int>
 1 professor mcgonagall    578
 2 uncle vernon            386
 3 harry potter            349
 4 death eaters            346
```

```
 5 harry looked           316
 6 harry ron              302
 7 aunt petunia           206
 8 invisibility cloak     192
 9 professor trelawney    177
10 dark arts              176
# i 89,248 more rows
```

```
potter_goodgrams <- potter_bigram_filtered |>
  left_join(potter_bigram) |>
  count(title, bigram) |>
  bind_tf_idf(bigram, title, n) |>
  arrange(desc(tf_idf))
```

```
Joining with `by = join_by(bigram)`
```

```
potter_goodgrams |>
  group_by(title) |>
  arrange(desc(tf_idf)) |>
  slice_max(tf_idf, n = 10) |>
  ggplot(aes(x = fct_reorder(bigram, tf_idf), y = tf_idf, fill = title)) +
    geom_col(show.legend = FALSE) +
    coord_flip() +
    facet_wrap(~title, scales = "free")
```



9. Find which words contributed most in the "wrong" direction using the afinn sentiment combined with how often a word appears among all 7 books. Come up with a list of 4 negation words, and for each negation word, illustrate the words associated with the largest "wrong" contributions in a faceted bar plot.

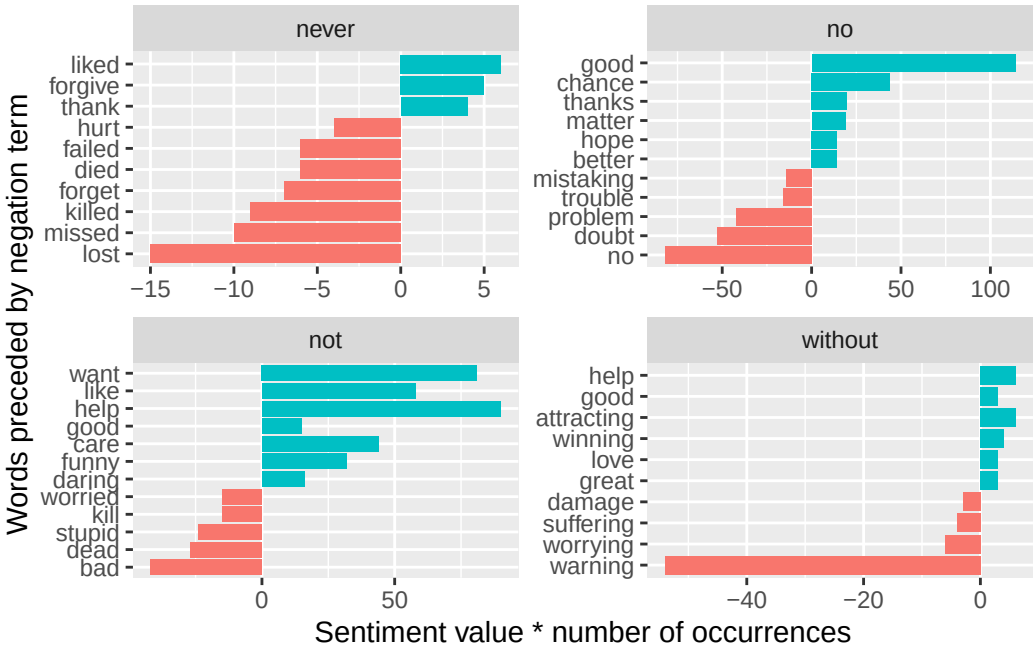
```
# An example of expanding the list of negation words
negation_words <- c("not", "no", "never", "without")

potter_bigram_separated <- potter_bigram |>
  separate(bigram, c("word1", "word2"), sep = " ") |>
  count(word1, word2, sort = TRUE) |>
  filter(!is.na(word1) & !is.na(word2))

negated_words <- potter_bigram_separated |>
  filter(word1 %in% negation_words) |>
  inner_join(afinn_sentiments, by = c(word2 = "word")) |>
  arrange(desc(n))

negated_words |>
  mutate(contribution = n * value) |>
  arrange(desc(abs(contribution))) |>
  group_by(word1) |>
  slice_max(abs(contribution), n = 10) |>
```

```
  ungroup() |>
  mutate(word2 = reorder(word2, contribution)) |>
  ggplot(aes(n * value, word2, fill = n * value > 0)) +
    geom_col(show.legend = FALSE) +
    facet_wrap(~ word1, scales = "free") +
    labs(x = "Sentiment value * number of occurrences",
         y = "Words preceded by negation term")
```



10. Select a set of 4 "interesting" terms and then use the Phi coefficient to find and plot the 6 words most correlated with each of your "interesting" words. Start by dividing `potter_tidy` into 80-word sections and then remove names and spells and stop words.

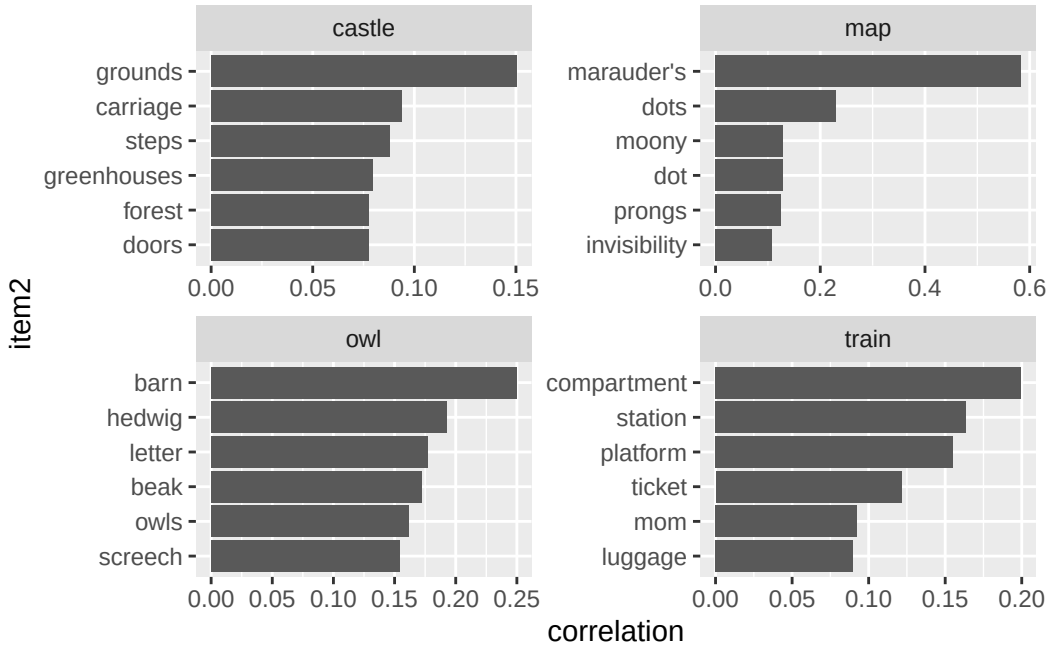
```
potter_section <- potter_tidy |>
  mutate(section = 1 + row_number() %/% 80) |>
  filter(!word %in% stop_words$word,
         !word %in% potter_names$firstname,
         !word %in% potter_names$lastname,
         !word %in% potter_spells$first_word,
         !word %in% potter_spells$second_word,
         !is.na(word))

potter_section |>
  pairwise_count(word, section, sort = TRUE)
```

```
# A tibble: 5,339,950 x 3
   item1     item2         n
   <chr>     <chr>     <dbl>
 1 eaters    death       301
 2 death     eaters      301
 3 looked    eyes        295
 4 eyes      looked      295
 5 looked    time        241
 6 time      looked      241
 7 harry's   looked      227
 8 looked    harry's     227
 9 professor looked      224
10 looked    professor   224
# i 5,339,940 more rows
```

```
potter_cors <- potter_section |>
  group_by(word) |>
  filter(n() >= 10) |>
  pairwise_cor(word, section, sort = TRUE)

potter_cors |>
  filter(item1 %in% c("castle", "train", "map", "owl")) |>
  group_by(item1) |>
  slice_max(correlation, n = 6) |>
  ungroup() |>
  mutate(item2 = reorder(item2, correlation)) |>
  ggplot(aes(item2, correlation)) +
    geom_bar(stat = "identity") +
    facet_wrap(~ item1, scales = "free") +
    coord_flip()
```



11. Create a network graph to visualize the correlations and clusters of words that were found by the `widyr` package in (10).

```
# for a correlation over .45
potter_cors |>
  filter(correlation > .45) |>
  graph_from_data_frame() |>
  ggraph(layout = "fr") +
    geom_edge_link(aes(edge_alpha = correlation), show.legend = FALSE) +
    geom_node_point(color = "lightblue", size = 5) +
    geom_node_text(aes(label = name), repel = TRUE) +
    theme_void()
```

```
Warning: ggrepel: 1 unlabeled data points (too many overlaps). Consider
increasing max.overlaps
```

12. Use LDA to fit a 2-topic model to all 7 Harry Potter books. Be sure to remove names, spells, and stop words before running your topic models. (a) Make a plot to illustrate words with greatest difference between two topics, using log ratio. (b) Print a table with the gamma variable for each document and topic. Based on (a) and (b), can you interpret what the two topics represent?

```
potter_dtm <- potter_tidy |>
  filter(!word %in% stop_words$word,
         !word %in% potter_names$firstname,
         !word %in% potter_names$lastname,
         !word %in% potter_spells$first_word,
         !word %in% potter_spells$second_word,
         !is.na(word)) |>
  group_by(title) |>
  count(word) |>
  cast_dtm(title, word, n)

# set a seed so that the output of the model is predictable
potter_lda <- LDA(potter_dtm, k = 2, control = list(seed = 1234))

potter_topics <- tidy(potter_lda, matrix = "beta")
potter_topics
```

```
# A tibble: 46,898 x 3
   topic term        beta
   <int> <chr>      <dbl>
 1     1 0     0.0000150
 2     2 0     0.0000248
 3     1 1     0.000241
 4     2 1     0.000276
 5     1 1473  0.00000409
 6     2 1473  0.00000183
 7     1 1637  0.00000202
 8     2 1637  0.00000365
 9     1 17    0.00000285
10     2 17    0.00000834
# i 46,888 more rows
```

```
# # Find the most common words within each topic
# potter_top_terms <- potter_topics |>
#   group_by(topic) |>
#   slice_max(beta, n = 10) |>
#   ungroup() |>
#   arrange(topic, -beta)

# potter_top_terms |>
#   mutate(term = reorder_within(term, beta, topic)) |>
#   ggplot(aes(beta, term, fill = factor(topic))) +
#     geom_col(show.legend = FALSE) +
#     facet_wrap(~ topic, scales = "free") +
#     scale_y_reordered()

# Find words with greatest difference between two topics, using log ratio
potter_beta_wide <- potter_topics |>
  mutate(topic = paste0("topic", topic)) |>
  pivot_wider(names_from = topic, values_from = beta) |>
  filter(topic1 > .001 | topic2 > .001) |>
  mutate(log_ratio = log2(topic2 / topic1))

potter_beta_wide
```

```
# A tibble: 200 x 4
   term       topic1    topic2 log_ratio
   <chr>       <dbl>     <dbl>     <dbl>
 1 air      0.00165  0.00149      -0.150
```

```
 2 answer   0.000295 0.00108       1.87
 3 appeared 0.000490 0.00121       1.31
 4 arm      0.00170  0.000407     -2.06
 5 aunt     0.00199  0.0000977    -4.35
 6 bed      0.000836 0.00187       1.16
 7 bit      0.00111  0.00204       0.882
 8 boy      0.000513 0.00237       2.20
 9 burst    0.00102  0.000266     -1.94
10 called   0.000422 0.00173       2.04
# i 190 more rows
```

```
potter_beta_wide |>
  arrange(desc(abs(log_ratio))) |>
  slice_max(abs(log_ratio), n = 20) |>
  mutate(term = reorder(term, log_ratio)) |>
  ggplot(aes(log_ratio, term, fill = log_ratio > 0)) +
    geom_col(show.legend = FALSE) +
    labs(x = "Log ratio of Beta values",
         y = "Words in Potter Books")
```


```
potter_documents <- tidy(potter_lda, matrix = "gamma")
```

```r
# Find documents for each topic
potter_documents |>
  group_by(topic) |>
  slice_max(gamma, n = 10) |>
  ungroup() |>
  arrange(topic, -gamma)
```

```
# A tibble: 14 x 3
   document             topic gamma
   <chr>                <int> <dbl>
 1 Chamber of Secrets       1 0.519
 2 Prisoner of Azkaban      1 0.501
 3 Half-Blood Prince        1 0.484
 4 Sorcerer's Stone         1 0.481
 5 Order of the Phoenix     1 0.457
 6 Goblet of Fire           1 0.456
 7 Deathly Hallows          1 0.444
 8 Deathly Hallows          2 0.556
 9 Goblet of Fire           2 0.544
10 Order of the Phoenix     2 0.543
11 Sorcerer's Stone         2 0.519
12 Half-Blood Prince        2 0.516
13 Prisoner of Azkaban      2 0.499
14 Chamber of Secrets       2 0.481
```

Topic 1: Chamber of Secrests and Prisoner of Azkaban, Topic 2: Half-Blood Prince, Sorcerer's Stone, Order of the Phoenix, Goblet of Fire and Deathly Hallows. Topic 1 is Book 2 and 3, they're kind of related in what is going on in the series so the words and characters are similar.