HW 5

Gwynnie Hayes

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:

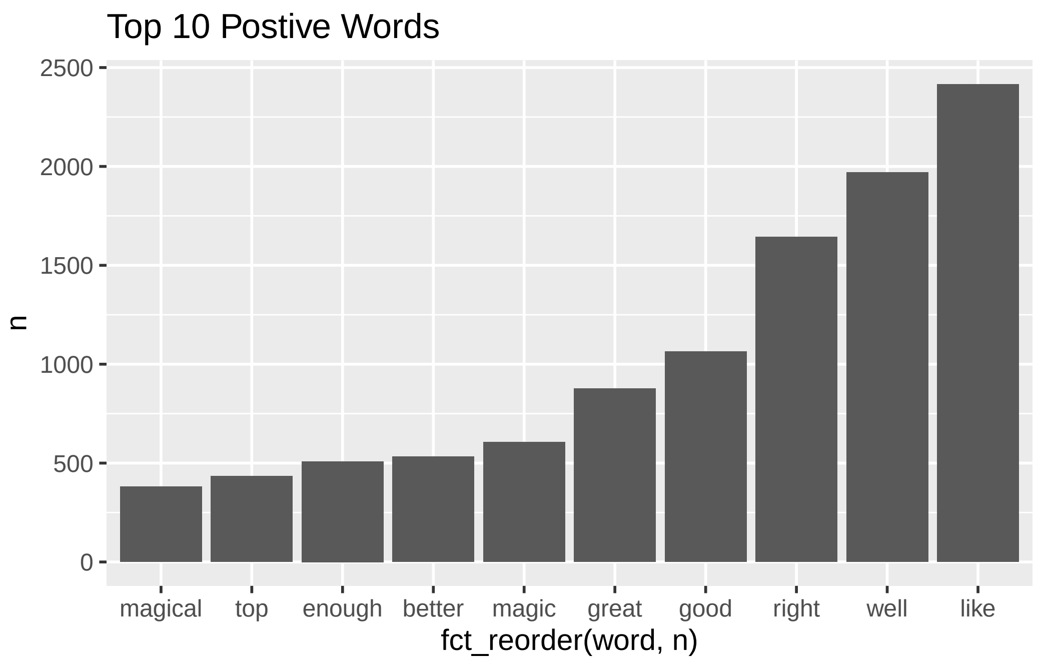
> Harry Potter is a series of seven fantasy novels written by British author J. K. Rowling. The novels chronicle the lives of a young wizard, Harry Potter, and his friends Hermione Granger and Ron Weasley, all of whom are students at Hogwarts School of Witchcraft and Wizardry. The main story arc concerns 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).

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.

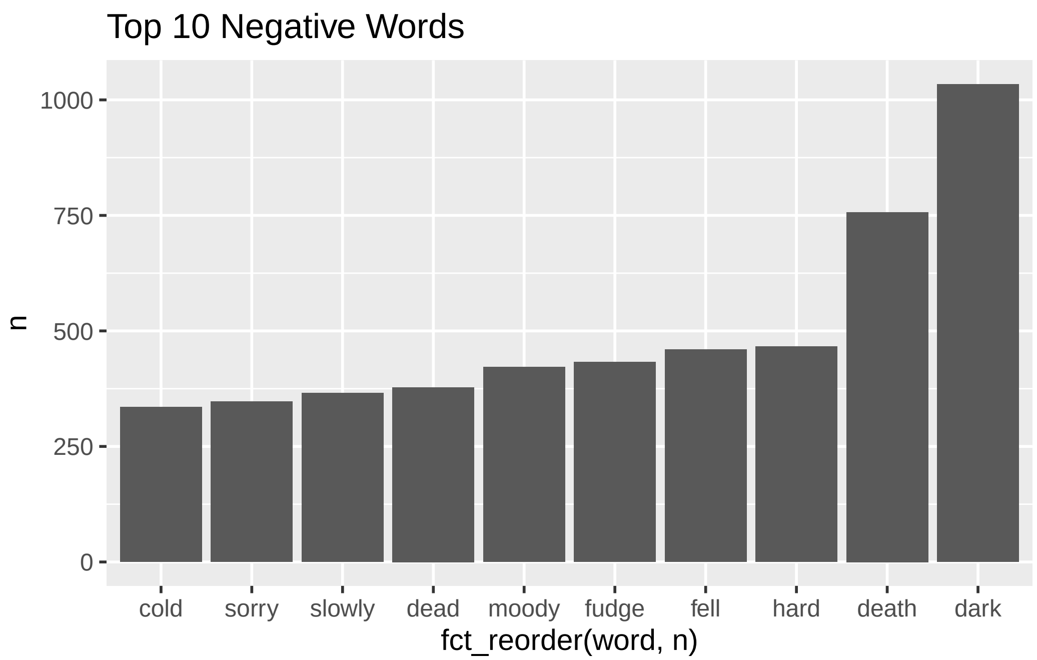
```r
bing_sentiment <- get_sentiments(lexicon = "bing")

potter_tidy |>
  inner_join(bing_sentiment, relationship = "many-to-many") |>
  filter(sentiment == "positive") |>
  count(word, sentiment) |>
  arrange(desc(n)) |>
  slice_max(n, n = 10) |>
  ggplot(aes(fct_reorder(word, n), n))+
    geom_col() +
  labs(title = "Top 10 Postive Words")
```

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



```
potter_tidy |>
  inner_join(bing_sentiment, relationship = "many-to-many") |>
  filter(sentiment == "negative") |>
  count(word, sentiment) |>
  arrange(desc(n)) |>
  slice_max(n, n = 10) |>
  ggplot(aes(fct_reorder(word, n), n))+
    geom_col()+
  labs(title = "Top 10 Negative Words")
```

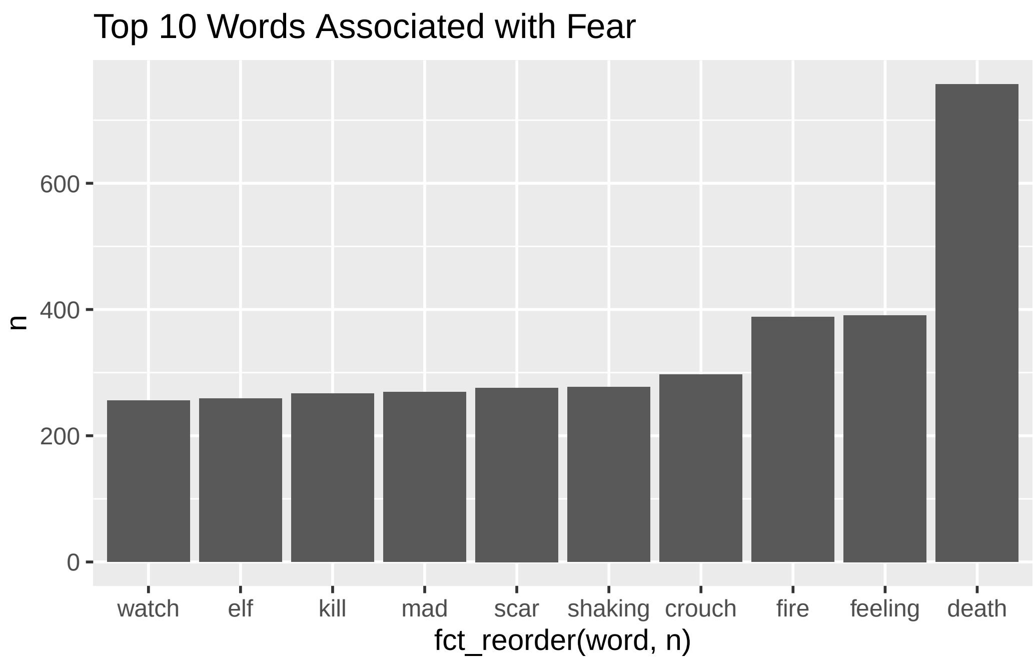
```
Joining with `by = join_by(word)`
```



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

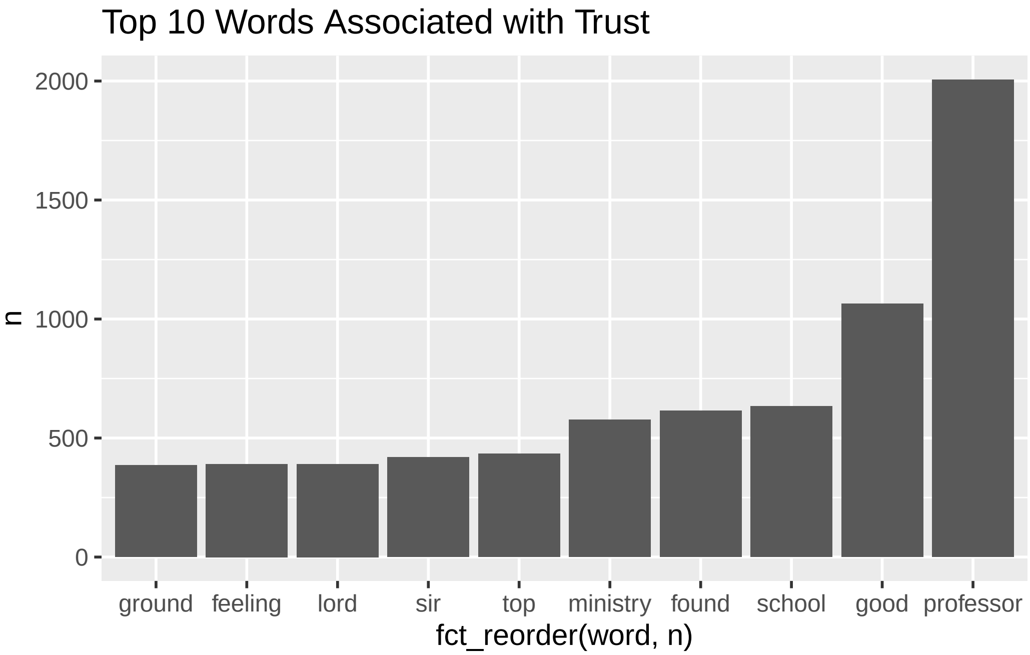
```
nrc_sentiment <- get_sentiments(lexicon = "nrc")

potter_tidy |>
  inner_join(nrc_sentiment, relationship = "many-to-many") |>
  filter(sentiment == "fear") |>
  count(word, sentiment) |>
  arrange(desc(n)) |>
  slice_max(n, n = 10) |>
  ggplot(aes(fct_reorder(word, n), n)) +
    geom_col () +
  labs(title = "Top 10 Words Associated with Fear")
```

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



```
potter_tidy |>
  inner_join(nrc_sentiment, relationship = "many-to-many") |>
  filter(sentiment == "trust") |>
  count(word, sentiment) |>
  arrange(desc(n)) |>
  slice_max(n, n = 10) |>
  ggplot(aes(fct_reorder(word, n), n)) +
    geom_col () +
  labs(title = "Top 10 Words Associated with Trust")
```

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



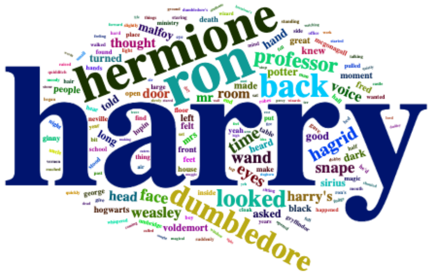
3. Make a wordcloud for the entire series after removing stop words using the "smart" source.

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

words_df <- potter_tidy |>
  anti_join(smart_stop) |>
  count(word) |>
  arrange(desc(n)) |>
  data.frame()
```

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

```
wordcloud2(
  words_df,
  size = 1,
  shape = 'circle'
)
```

```
file:////private/var/folders/mr/t2grrvgn50v_2prqk9lzyyt40000gn/T/RtmpDfUUwR/file1a41686ecf7c
```

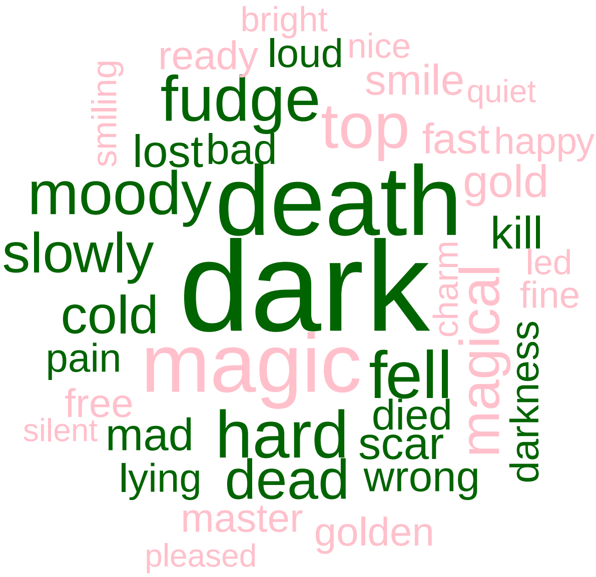

harry
hermione
ron
back
dumbledore
looked
professor
hagrid
time
wand
face
eyes
head
harry's
snape
thought
malfoy
door
room
voice
hand
weasley
voldemort
told
turned
long
mr
dark
good
sirius
black
asked
hogwarts
george
people
great
knew
potter
made
fred
moment
left
felt
floor
front
house
feet

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!)

```r
potter_wordcloud <- potter_tidy |>
  anti_join(stop_words) |>
  inner_join(bing_sentiment, relationship = "many-to-many") |>
  count(word, sentiment) |>
  arrange(desc(n)) |>
  slice_max(n, n = 20, by = sentiment)|>
  mutate(color = ifelse(sentiment == "positive", "pink", "darkgreen"))
```

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

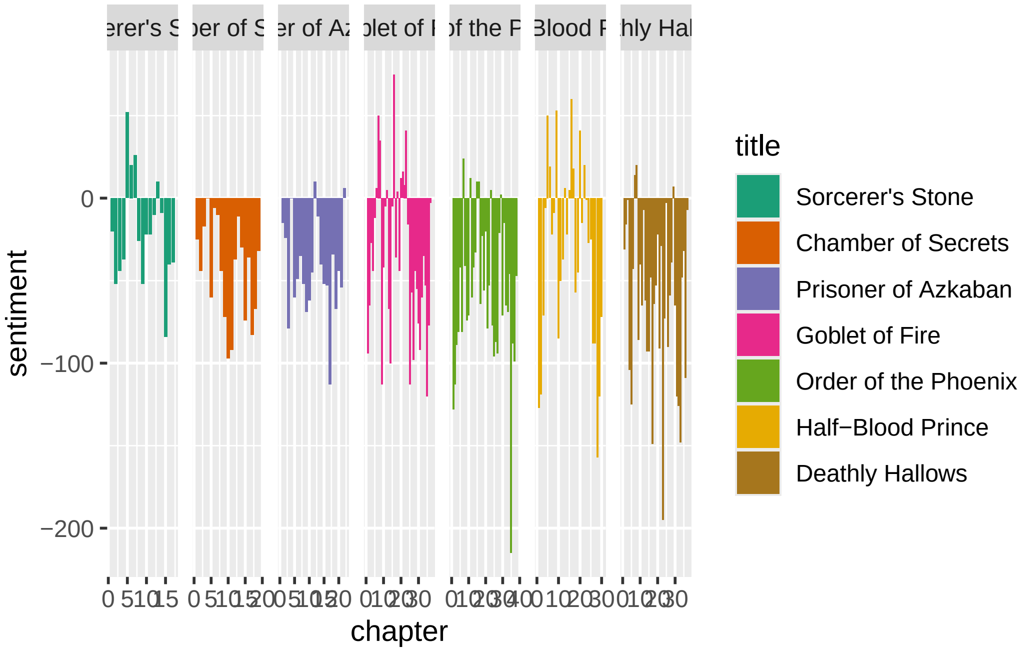
```r
wordcloud(
  words = potter_wordcloud$word,
  freq = potter_wordcloud$n,
  ordered.colors = TRUE,
  colors = potter_wordcloud$color,
  random.order = FALSE
)
```



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.

```r
potter_tidy |>
  inner_join(bing_sentiment, relationship = "many-to-many") |>
  count(title, chapter, sentiment) |>
  pivot_wider(names_from = sentiment, values_from = n, values_fill = 0) |>
  mutate(sentiment = positive - negative) |>
  group_by(title) |>
  ggplot(aes(x = chapter, y = sentiment, fill = title)) +
    geom_col() +
  facet_grid(~title, scales = "free") +
  scale_fill_brewer(palette = "Dark2")
```

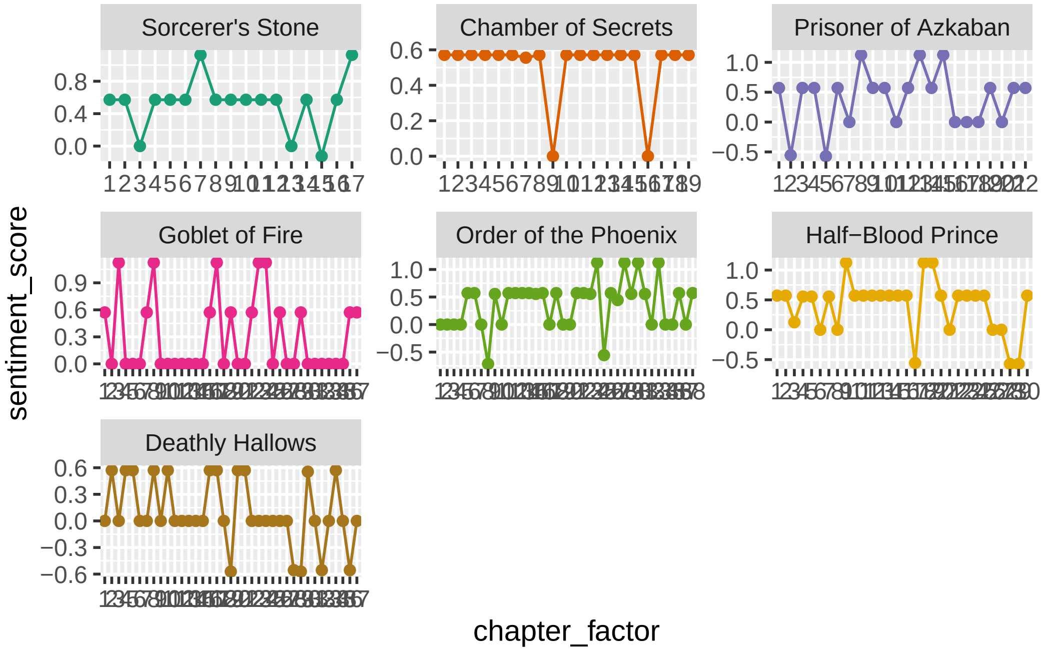
```
Joining with `by = join_by(word)`
```



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_sentiment <- get_sentiments(lexicon = "afinn")

potter_tidy |>
  inner_join(afinn_sentiment, relationship = "many-to-many") |>
  count(title, chapter, value) |>
  group_by(title, chapter) |>
  summarize(sentiment_score = mean(value), .groups = "drop") |>
  mutate(chapter_factor = factor(chapter)) |>
  ggplot(aes(x = chapter_factor, y = sentiment_score, group = title, color = title)) +
    geom_line(show.legend = FALSE) +
    geom_point(show.legend = FALSE) +
  facet_wrap(~title, scale = "free") +
  scale_color_brewer(palette = "Dark2")
```

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



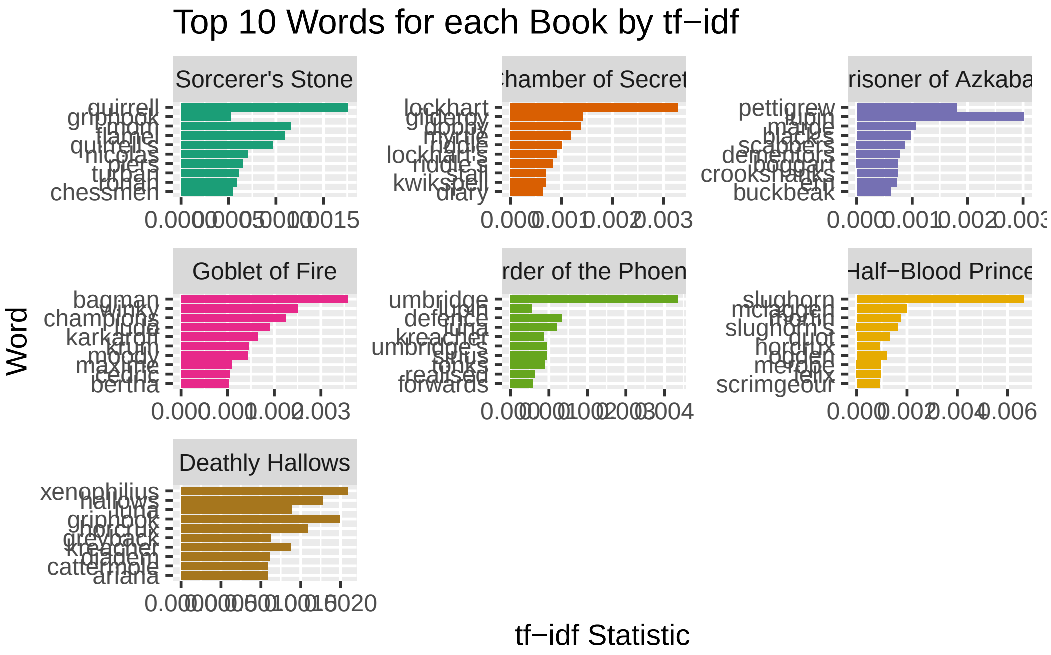
7. Make a faceted bar plot showing the top 10 words that distinguish each book according to the tf-idf statistic.

```
potter_tfidf <- potter_tidy |>
  anti_join(stop_words) |>
```

```r
  count(word, title) |>
  bind_tf_idf(word, title, n)
```

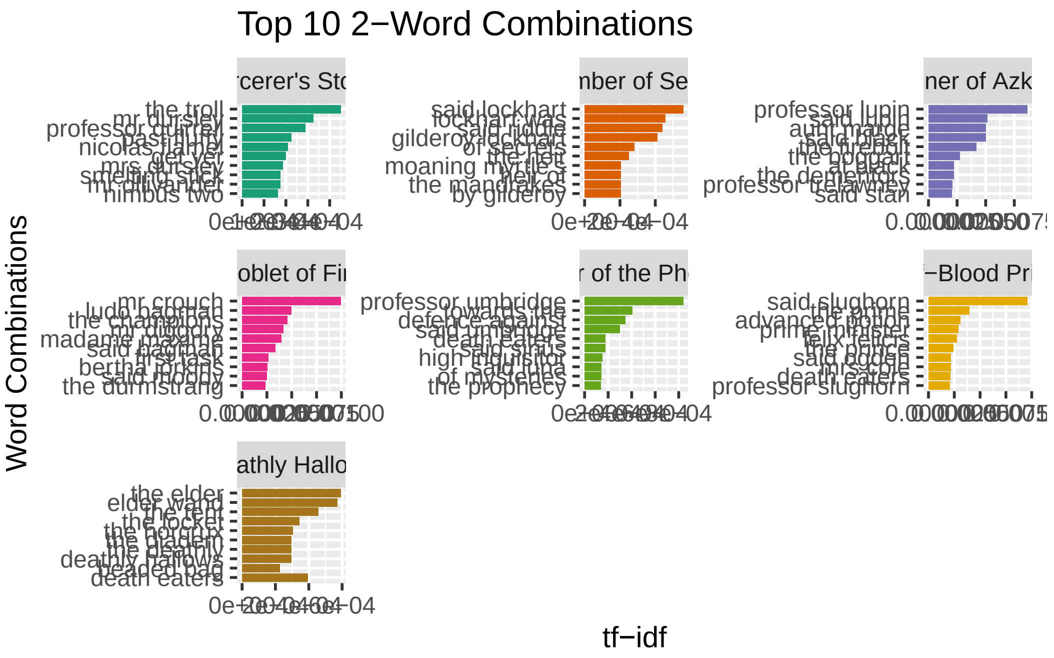
```
Joining with `by = join_by(word)`
```

```r
potter_tfidf |>
  group_by(title) |>
  arrange(desc(tf_idf)) |>
  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") +
  scale_fill_brewer(palette = "Dark2") +
  labs(title = "Top 10 Words for each Book by tf-idf", y = "tf-idf Statistic", x = "Word")
```



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

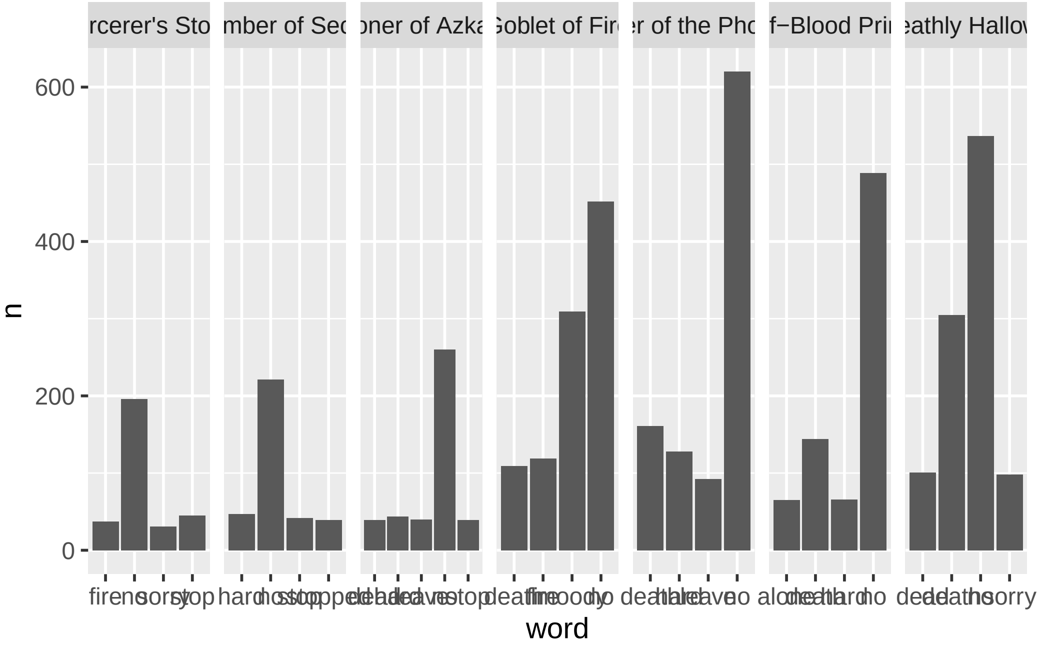
```
potter_ngram <- potter_untidy |>
  unnest_tokens(bigram, text, token = "ngrams", n = 2) |>
    filter(bigram != "NA") |>
  count(bigram, title) |>
  bind_tf_idf(bigram, title, n)

potter_ngram |>
  group_by(title) |>
  arrange(desc(tf_idf)) |>
  slice_max(tf_idf, n = 10) |>
  ungroup() |>
  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") +
  scale_fill_brewer(palette = "Dark2") +
  labs(title = "Top 10 2-Word Combinations", y = "tf-idf", x = "Word Combinations")
```



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.

```r
afinn_sentiment <- get_sentiments(lexicon = "afinn")

potter_tidy |>
  group_by(title, word) |>
  summarize(n = n()) |>
  arrange(desc(n)) |>
  inner_join(afinn_sentiment) |>
  filter(value < 0) |>
  arrange(value) |>
  slice_max(n, n = 4)|>
  ggplot(aes(x = word, y = n)) +
    geom_col() +
    facet_grid(~title, scales = "free")
```

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

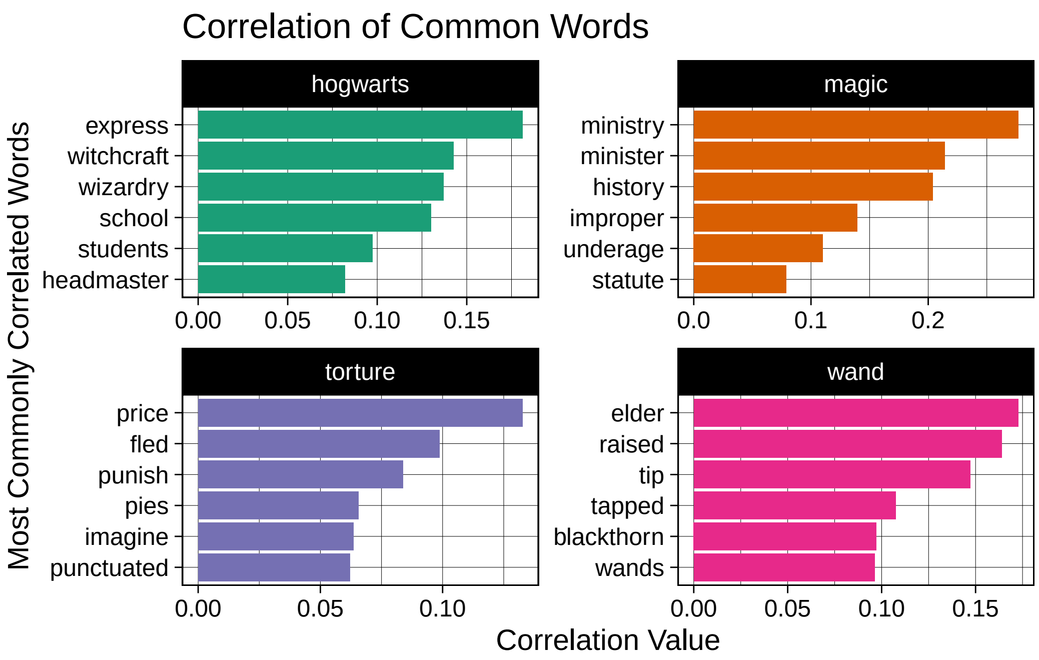


10. Select a set of 4 “interesting” terms and then use the Phi coefficient to find and plot the 6 words most correlated with each of your “interesting” words. Start by dividing `potter_tidy` into 80-word sections and then remove names and spells and stop words.

```r
potter <- potter_tidy |>
  mutate(section = row_number() %/% 80) |>
  anti_join(stop_words) |>
  anti_join(potter_spells, join_by(word == first_word)) |>
  anti_join(potter_spells, join_by(word == second_word)) |>
  anti_join(potter_names, join_by(word == firstname)) |>
  anti_join(potter_names, join_by(word == lastname))
```

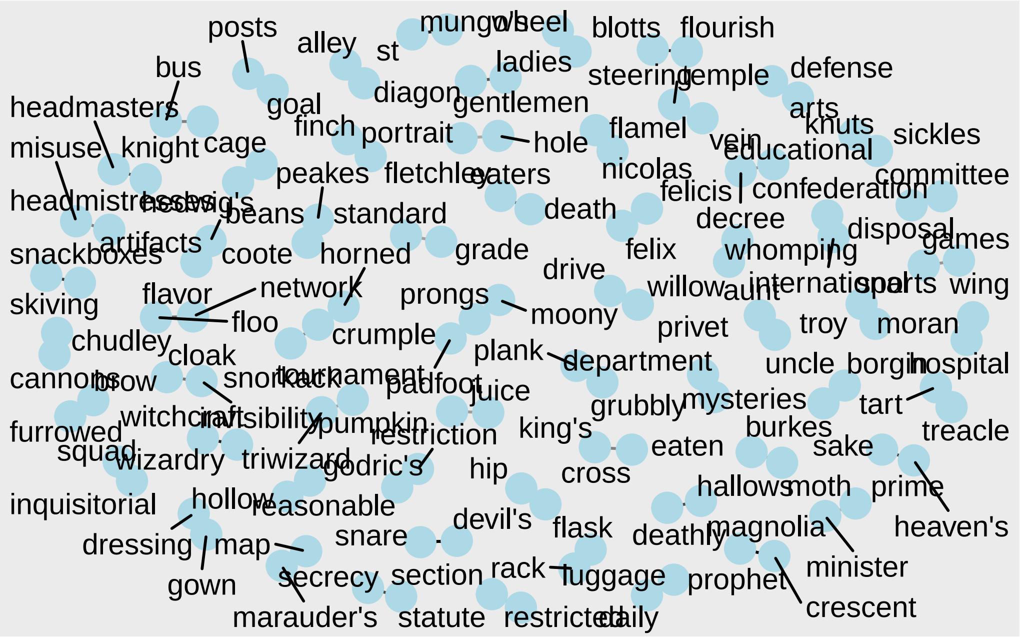
```
Joining with `by = join_by(word)`
```

```r
word_pairs <- potter |>
  pairwise_count(word, section, sort = TRUE)

word_cors <- potter |>
  group_by(word) |>
  filter(n() >= 10) |>
  pairwise_cor(word, section, sort = TRUE)


word_cors |>
  filter(item1 %in% c("torture", "hogwarts", "magic", "wand")) |>
  group_by(item1) |>
  slice_max(correlation, n = 6, with_ties = FALSE) |>
  ungroup() |>
  mutate(item2 = reorder(item2, correlation)) |>
  ggplot(aes(item2, correlation, fill = item1)) +
    geom_bar(stat = "identity", show.legend = FALSE) +
    facet_wrap(~ item1, scales = "free") +
    coord_flip() +
  theme_linedraw() +
  scale_fill_brewer(palette = "Dark2") +
  labs(title = "Correlation of Common Words", x = "Most Commonly Correlated Words", y = "Cor
```



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

```
set.seed(2016)

word_cors |>
  filter(correlation > .5) |>
  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)
```



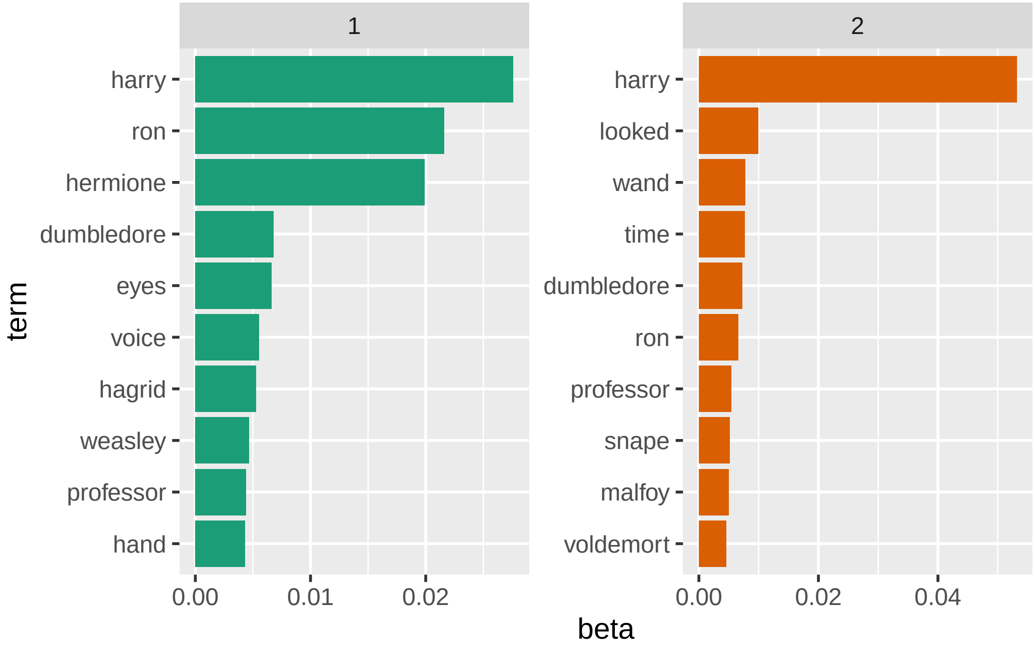
12. Use LDA to fit a 2-topic model to all 7 Harry Potter books. Be sure to remove names, spells, and stop words before running your topic models. (a) Make a plot to illustrate words with greatest difference between two topics, using log ratio. (b) Print a table with the gamma variable for each document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

```
potter_dtm <- potter_tidy |>
  count(word, title) |>
  anti_join(stop_words) |>
  cast_dtm(title, word, n)
```

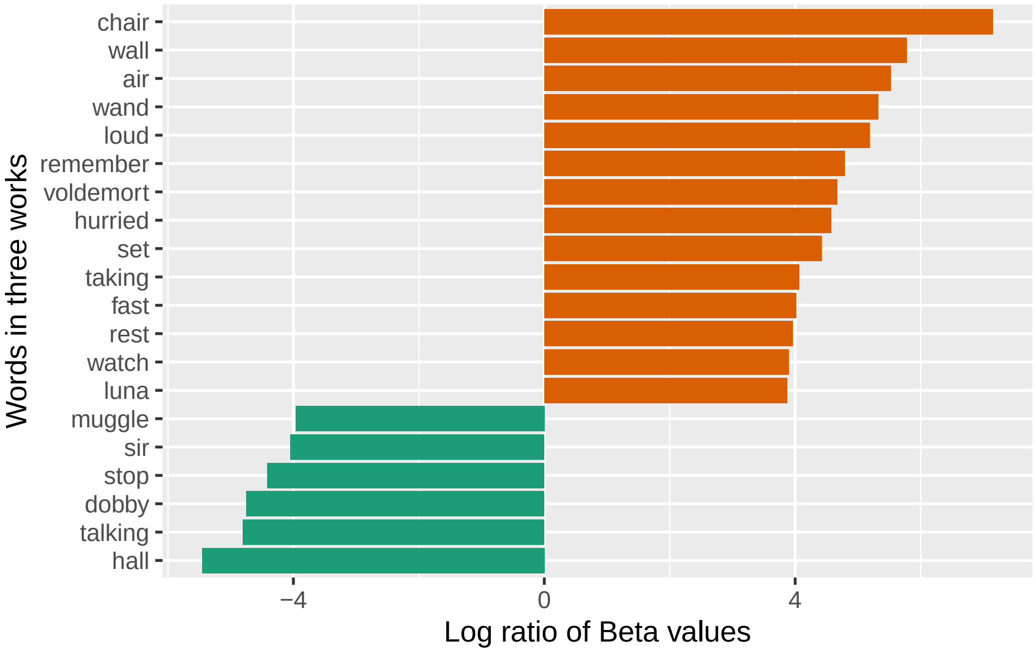
```
Joining with `by = join_by(word)`
```

```
potter_lda <- LDA(potter_dtm, k = 2, control = list(seed = 1234))

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

potter_terms <- potter_topics |>
  group_by(topic) |>
  slice_max(beta, n = 10) |>
  ungroup() |>
  arrange(topic, -beta)
```

```r
potter_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() +
    scale_fill_brewer(palette = "Dark2")
```



```r
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))

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 three works") +
  scale_fill_brewer(palette = "Dark2")
```



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