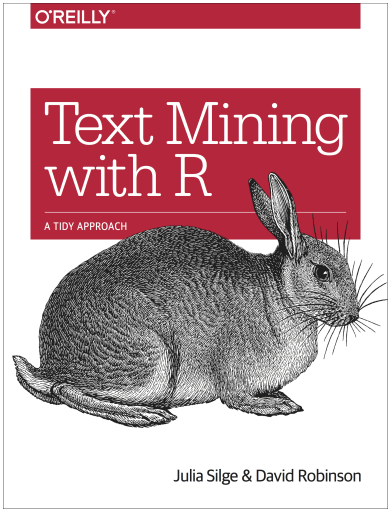
COMP824 2023 Text Mining - A brief overview

Sarah Marshall

Department of Mathematical Sciences
Auckland University of Technology

Required reading-Text Mining with R: A Tidy Approach[1]



[1] https://www.tidytextmining.com/

Tidy Data: Recap

Rules of Tidy Data

- Each variable must have its own column.
- Each observation must have its own row.
- Each value must have its own cell.

Text Data

“Because I could not stop for Death”, Emily Dickinson (1830 – 1886)

```
text <- c("Because I could not stop for Death -",
          "He kindly stopped for me -",
          "The Carriage held but just Ourselves -",
          "and Immortality")

text
```

```
[1] "Because I could not stop for Death -"
[2] "He kindly stopped for me -"
[3] "The Carriage held but just Ourselves -"
[4] "and Immortality"
```

Tidying Text - part 1 (create a dataframe)

```
text_df <- tibble(line = 1:4,
                  text = text)
text_df
```

```
# A tibble: 4 x 2
   line text
  <int> <chr>
1     1 Because I could not stop for Death -
2     2 He kindly stopped for me -
3     3 The Carriage held but just Ourselves -
4     4 and Immortality
```

Tidying Text - Tokens

A **token** is a meaningful unit of text, most often a word, that we are interested in using for further analysis, and tokenization is the process of splitting text into tokens.

Tidy text has: **one-token-per-document-per-row**.

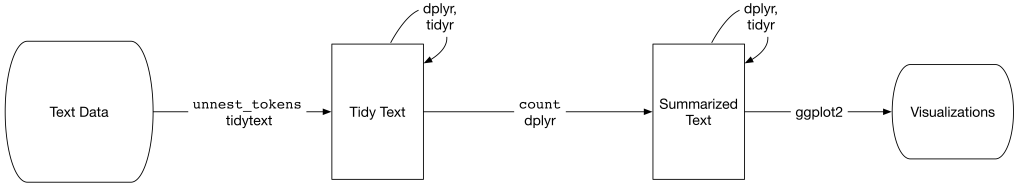
Tidying Text - part 2 (tokenization)

```
text_df %>%
  unnest_tokens(output = word, #new column name
                input = text #column name in df
                ) %>%
print(n = 4)
```

```
# A tibble: 20 x 2
   line word
  <int> <chr>
1     1 because
2     1 i
3     1 could
4     1 not
# i 16 more rows
```

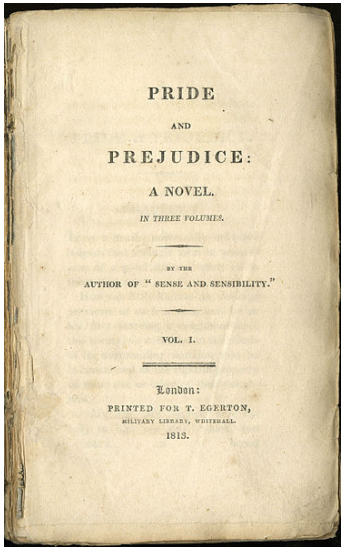
Notice: Line numbers, no punctuation, and lowercase

Tidying Text - the process[2]



[2] https://www.tidytextmining.com/tidytext.html

Text Data[3]



PRIDE

AND

PREJUDICE:

A NOVEL.

IN THREE VOLUMES.

BY THE

AUTHOR OF " SENSE AND SENSIBILITY."

VOL. I.

London:

PRINTED FOR T. EGERTON,

MILITARY LIBRARY, WHITEHALL.

1813.

[3]https://commons.wikimedia.org/wiki/File:PrideAndPrejudiceTitlePage.jpg

Text Data (cont.)

```
janeaustenr::prideprejudice %>% head(12)
```

```
 [1] "PRIDE AND PREJUDICE"
 [2] ""
 [3] "By Jane Austen"
 [4] ""
 [5] ""
 [6] ""
 [7] "Chapter 1"
 [8] ""
 [9] ""
[10] "It is a truth universally acknowledged, that a single man in possession"
[11] "of a good fortune, must be in want of a wife."
[12] ""
```

Tidying Jane Austen's Works

```
original_books <- austen_books() %>%
  group_by(book) %>%
  mutate(linenumber = row_number(),
         chapter = cumsum(
                   str_detect(text,
                              regex("^chapter [\\divxlc]",
                                    ignore_case = TRUE)))) %>%
  ungroup()

original_books %>% print(n = 6)
```

```
# A tibble: 73,422 x 4
  text                    book  linenumber chapter
  <chr>                   <fct>      <int>   <int>
1 "SENSE AND SENSIBILITY" Sens~          1       0
2 ""                      Sens~          2       0
3 "by Jane Austen"        Sens~          3       0
4 ""                      Sens~          4       0
5 "(1811)"                Sens~          5       0
6 ""                      Sens~          6       0
# i 73,416 more rows
```

Tidying Jane Austen's Works (cont.)

```
tidy_books <- original_books %>%
  unnest_tokens(word, text, token = "words")
tidy_books
```

```
# A tibble: 725,055 x 4
   book                linenumber chapter word
   <fct>                    <int>   <int> <chr>
 1 Sense & Sensibility          1       0 sense
 2 Sense & Sensibility          1       0 and
 3 Sense & Sensibility          1       0 sensibi~
 4 Sense & Sensibility          3       0 by
 5 Sense & Sensibility          3       0 jane
 6 Sense & Sensibility          3       0 austen
 7 Sense & Sensibility          5       0 1811
 8 Sense & Sensibility         10       1 chapter
 9 Sense & Sensibility         10       1 1
10 Sense & Sensibility         13       1 the
# i 725,045 more rows
```

Stop words

Stop words are the **little** words like *the*, *of*, *to* etc.

```
data(stop_words)
stop_words
```

```
# A tibble: 1,149 x 2
   word        lexicon
   <chr>       <chr>
 1 a           SMART
 2 a's         SMART
 3 able        SMART
 4 about       SMART
 5 above       SMART
 6 according   SMART
 7 accordingly SMART
 8 across      SMART
 9 actually    SMART
10 after       SMART
# i 1,139 more rows
```

Removing the stop words

```
tidy_books <- tidy_books %>%
  anti_join(stop_words)
```

Preliminary Analysis

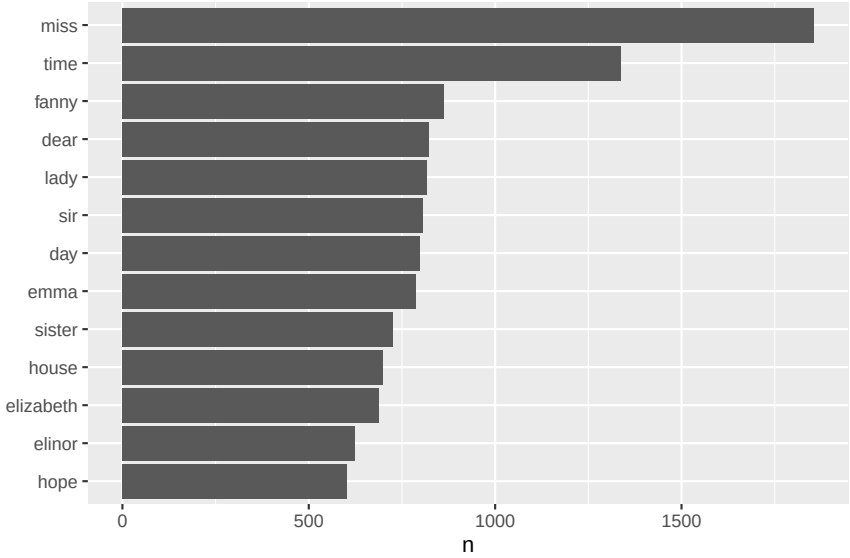
```
tidy_books %>%
  count(word, sort = TRUE)
```

```
# A tibble: 13,914 x 2
   word       n
   <chr>  <int>
 1 miss    1855
 2 time    1337
 3 fanny    862
 4 dear     822
 5 lady     817
 6 sir      806
 7 day      797
 8 emma     787
 9 sister   727
10 house    699
# i 13,904 more rows
```

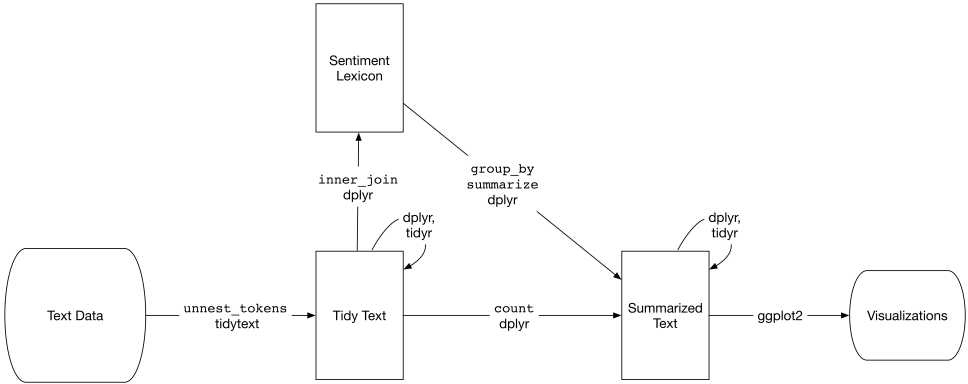
Preliminary Analysis - Plot

```
g <- tidy_books %>%
  count(word, sort = TRUE) %>%
  filter(n > 600) %>%
  mutate(word = reorder(word, n)) %>%
  ggplot(aes(word, n)) +
  geom_col() +
  xlab(NULL) +
  coord_flip()
```

Preliminary Analysis - Plot (cont.)



Sentiment Analysis[4]



- Several types of sentiment analysis.
- Today - individual words as representative of document

[4] https://www.tidytextmining.com/sentiment.html

Sentiment Analysis (cont.)

Three lexicons: *AFINN*, *bing*, *nrc*

```
get_sentiments("afinn") #-5 to +5
get_sentiments("nrc") # joy, fear etc
get_sentiments("bing") #positive, negative
```

Sentiment Analysis (cont.)

```
# Bing lexicon
sentiments
```

```
# A tibble: 6,786 x 2
   word        sentiment
   <chr>       <chr>
 1 2-faces     negative
 2 abnormal    negative
 3 abolish     negative
 4 abominable  negative
 5 abominably  negative
 6 abominate   negative
 7 abomination negative
 8 abort       negative
 9 aborted     negative
10 aborts      negative
# i 6,776 more rows
```

Extracting joy

```
tns <- getNamespace("textdata")
assignInNamespace(x = "printer", value = function(...) 1, ns = tns)
nrc_joy <- get_sentiments("nrc") %>%
  filter(sentiment == "joy")

nrc_joy
```

```
# A tibble: 687 x 2
   word          sentiment
   <chr>         <chr>
 1 absolution    joy
 2 abundance     joy
 3 abundant      joy
 4 accolade      joy
 5 accompaniment joy
 6 accomplish    joy
 7 accomplished  joy
 8 achieve       joy
 9 achievement   joy
10 acrobat       joy
# i 677 more rows
```

Sentiment Analysis

```
tidy_books %>%
  filter(book == "Emma") %>%
  inner_join(nrc_joy) %>%
  count(word, sort = TRUE) %>%
  print(n = 6)
```

```
# A tibble: 297 x 2
  word       n
  <chr>  <int>
1 friend   166
2 hope     143
3 happy    125
4 love     117
5 deal      92
6 found     92
# i 291 more rows
```

Notice: that this list may contain some words which may not be *joyful* e.g. found.

How does the sentiment change throughout the books?

Count the positive/negative words in each set of 80 lines

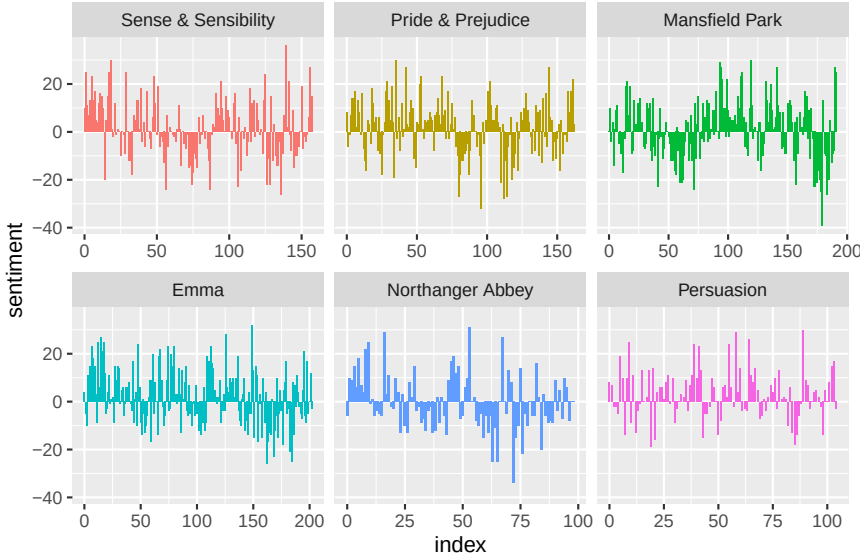
```
jane_austen_sentiment <- tidy_books %>%
  inner_join(get_sentiments("bing")) %>%
  count(book, index = linenumber %/% 80, sentiment) %>%
  spread(sentiment, n, fill = 0) %>%
  mutate(sentiment = positive - negative)

jane_austen_sentiment %>% print(n = 6)
```

```
# A tibble: 920 x 5
  book            index negative positive sentiment
  <fct>           <dbl>    <dbl>    <dbl>     <dbl>
1 Sense & Sensi~      0       16       26        10
2 Sense & Sensi~      1       19       44        25
3 Sense & Sensi~      2       12       23        11
4 Sense & Sensi~      3       15       22         7
5 Sense & Sensi~      4       16       29        13
6 Sense & Sensi~      5       16       39        23
# i 914 more rows
```

How does the sentiment change throughout the books? (cont.)

```
g <- ggplot(jane_austen_sentiment, aes(index, sentiment, fill = book)) +
  geom_col(show.legend = FALSE) +
  facet_wrap(~book, nrow = 2, scales = "free_x")
```

How does the sentiment change throughout the books? (cont.)



What was happening in Pride and Prejudice – index 80–90?

```
n_lines <- 80
original_books %>%
  filter(book == "Pride & Prejudice",
         between(linenumber, n_lines*80, n_lines*90 )) %>%
    print(n=10)
```

```
# A tibble: 801 x 4
   text                    book  linenumber chapter
   <chr>                   <fct>      <int>   <int>
 1 ""                      Prid~       6400      34
 2 "\"In such cases as t~  Prid~       6401      34
 3 "express a sense of o~  Prid~       6402      34
 4 "unequally they may b~  Prid~       6403      34
 5 "be felt, and if I co~  Prid~       6404      34
 6 "cannot--I have never~  Prid~       6405      34
 7 "bestowed it most unw~  Prid~       6406      34
 8 "anyone. It has been ~  Prid~       6407      34
 9 "of short duration. T~  Prid~       6408      34
10 "the acknowledgment o~  Prid~       6409      34
# i 791 more rows
```

What was happening in Pride and Prejudice – index 80–90? (cont.)

https://www.youtube.com/watch?v=JF3ueHjUc3k

Most positive/negative words

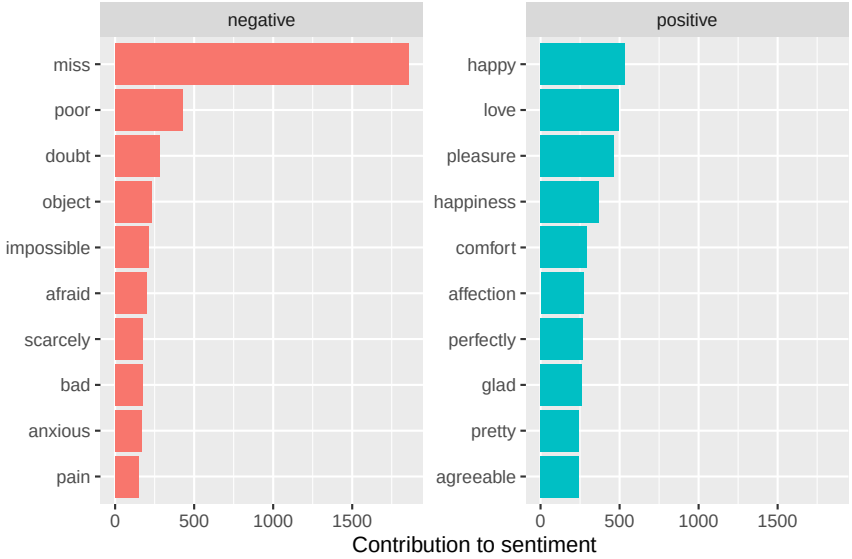
```
bing_word_counts <- tidy_books %>%
    inner_join(get_sentiments("bing")) %>%
    count(word, sentiment, sort = TRUE) %>%
    ungroup()

bing_word_counts %>% print(n=8)
```

```
# A tibble: 2,555 x 3
  word       sentiment     n
  <chr>      <chr>     <int>
1 miss       negative   1855
2 happy      positive    534
3 love       positive    495
4 pleasure   positive    462
5 poor       negative    424
6 happiness  positive    369
7 comfort    positive    292
8 doubt      negative    281
# i 2,547 more rows
```

Most positive/negative words (cont.)

```
g <- bing_word_counts %>%
  group_by(sentiment) %>%
  top_n(10) %>%
  ungroup() %>%
  mutate(word = reorder(word, n)) %>%
  ggplot(aes(word, n, fill = sentiment)) +
  geom_col(show.legend = FALSE) +
  facet_wrap(~sentiment, scales = "free_y") +
  labs(y = "Contribution to sentiment",
       x = NULL) +
  coord_flip()
```

Spot any anomalies?



Custom stop words

```
custom_stop_words <- bind_rows(tibble(word = c("miss"),
                                      lexicon = c("custom")),
                               stop_words)

custom_stop_words
```

```
# A tibble: 1,150 x 2
   word        lexicon
   <chr>       <chr>
 1 miss        custom
 2 a           SMART
 3 a's         SMART
 4 able        SMART
 5 about       SMART
 6 above       SMART
 7 according   SMART
 8 accordingly SMART
 9 across      SMART
10 actually    SMART
# i 1,140 more rows
```

Custom stop words (cont.)

```
tidy_books %>%
  anti_join(custom_stop_words)
```

Word Clouds

```
tidy_books %>%
  count(word) %>%
  with(wordcloud(word, n, max.words = 100))
```

Word Clouds - with colours (cont.)

```
pal <- brewer.pal(9,"BuGn")
pal <- pal[-(1:4)]  #remove light colours

tidy_books %>%
  count(word) %>%
  with(wordcloud(word, n, max.words = 100, colors = pal,
      rot.per = 0, fixed.asp = FALSE))
```

Word Clouds - with categories

```
tidy_books %>%
  inner_join(get_sentiments("bing"), by = "word") %>%
  count(word, sentiment, sort = TRUE) %>%
  acast(word ~ sentiment, value.var = "n", fill = 0) %>%
  comparison.cloud(colors = c("darkred", "darkgreen"),
                   title.colors=c("darkred", "darkgreen"),
                   max.words = 100, title.size = 2)
```

n-grams: A group of n words

```
austen_bigrams <- austen_books() %>%
  unnest_tokens(bigram, text,
                token = "ngrams", n = 2) %>%
  filter(!is.na(bigram)) # remove NAs
austen_bigrams
```

```
# A tibble: 662,783 x 2
   book                bigram
   <fct>               <chr>
 1 Sense & Sensibility sense and
 2 Sense & Sensibility and sensibility
 3 Sense & Sensibility by jane
 4 Sense & Sensibility jane austen
 5 Sense & Sensibility chapter 1
 6 Sense & Sensibility the family
 7 Sense & Sensibility family of
 8 Sense & Sensibility of dashwood
 9 Sense & Sensibility dashwood had
10 Sense & Sensibility had long
# i 662,773 more rows
```

Jane Austen - bigrams

```
austen_bigrams %>%
  count(bigram, sort = TRUE)
```

```
# A tibble: 193,209 x 2
   bigram       n
   <chr>    <int>
 1 of the    2853
 2 to be     2670
 3 in the    2221
 4 it was    1691
 5 i am      1485
 6 she had   1405
 7 of her    1363
 8 to the    1315
 9 she was   1309
10 had been  1206
# i 193,199 more rows
```

Removing bigrams with stop words

```
bigrams_separated <- austen_bigrams %>%
  separate(bigram, c("word1", "word2"), sep = " ")

bigrams_filtered <- bigrams_separated %>%
  filter(!word1 %in% stop_words$word) %>%
  filter(!word2 %in% stop_words$word)

# new bigram counts:
bigram_counts <- bigrams_filtered %>%
  count(word1, word2, sort = TRUE)

bigram_counts %>% print(n = 4)
```

```
# A tibble: 28,974 x 3
  word1   word2          n
  <chr>   <chr>      <int>
1 sir     thomas       266
2 miss    crawford     196
3 captain wentworth    143
4 miss    woodhouse    143
# i 28,970 more rows
```

Visualising bigrams

```
bigrams_united <- bigrams_filtered %>%
  unite(bigram, word1, word2, sep = " ")

bigrams_united %>%
  count(bigram) %>%
  with(wordcloud(bigram, n, max.words = 100,
       rot.per = 0, fixed.asp = FALSE))
```

Twitter data - an application

Download and load the data

```
load("twitter_data_2021.Rdata")

#twitter_data_2021.Rdata should be stored in the same
directory
```

```
rt_archie_vs_don <- bind_rows(list(archie = rt_archie,
                                   don = rt_don), .id = "id")
```

Tidy the data

```
(tidy_tweets <- rt_archie_vs_don %>%
  unnest_tokens(word, text, token = "words") %>%
  anti_join(custom_stop_words, by = "word") %>%
     select(word, id) %>%
  mutate(word = str_replace_all(word, "[^a-z0-9]", "")) %>%
  select(word, id))
```

```
# A tibble: 32,233 x 2
   word     id
   <chr>    <chr>
 1 harry    archie
 2 amp      archie
 3 meghan   archie
 4 duke     archie
 5 amp      archie
 6 duchess  archie
 7 sussex   archie
 8 archie   archie
 9 generous archie
10 donation archie
# i 32,223 more rows
```

Sentiment

```
tweet_sentiment <- tidy_tweets %>%
  inner_join(get_sentiments("nrc"), by = "word")
tweet_sentiment
```

```
# A tibble: 13,648 x 3
   word     id      sentiment
   <chr>    <chr>   <chr>
 1 harry    archie anger
 2 harry    archie negative
 3 harry    archie sadness
 4 duke     archie positive
 5 generous archie joy
 6 generous archie positive
 7 generous archie trust
 8 donation archie positive
 9 harry    archie anger
10 harry    archie negative
# i 13,638 more rows
```

Sentiment (cont.)

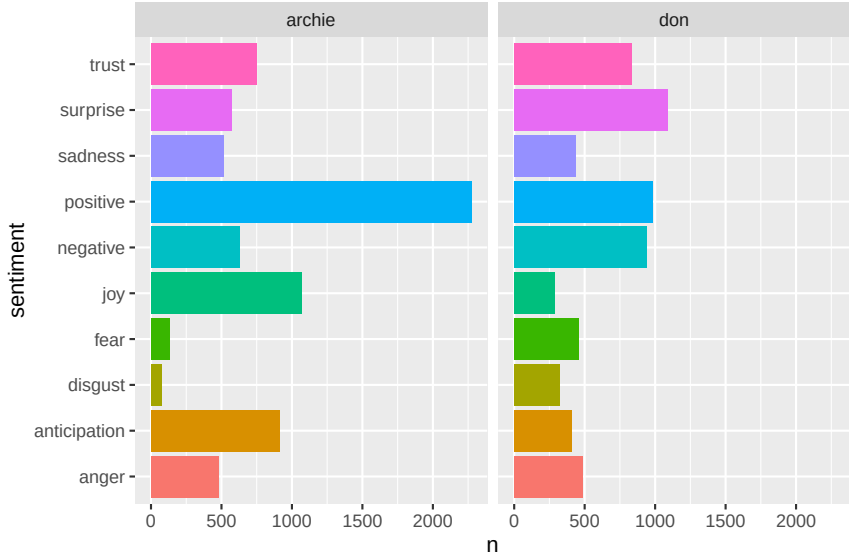
```
(tweet_sentiment_summary <- tweet_sentiment %>%
    count(id, sentiment))
```

```
# A tibble: 20 x 3
   id      sentiment        n
   <chr>   <chr>        <int>
 1 archie  anger          480
 2 archie  anticipation   912
 3 archie  disgust         76
 4 archie  fear           134
 5 archie  joy           1068
 6 archie  negative       630
 7 archie  positive      2272
 8 archie  sadness        517
 9 archie  surprise       569
10 archie  trust          750
11 don     anger          486
12 don     anticipation   407
13 don     disgust        322
14 don     fear           457
15 don     joy            287
16 don     negative       940
17 don     positive       981
18 don     sadness        438
19 don     surprise      1088
20 don     trust          834
```

Sentiment - Plot 1

```
g <- ggplot(tweet_sentiment_summary) +
    geom_col(mapping = aes(x = sentiment,
                           y = n,
                           fill = sentiment)) +
    facet_wrap(~id) +
    coord_flip() +
    theme(legend.position = "none")
```

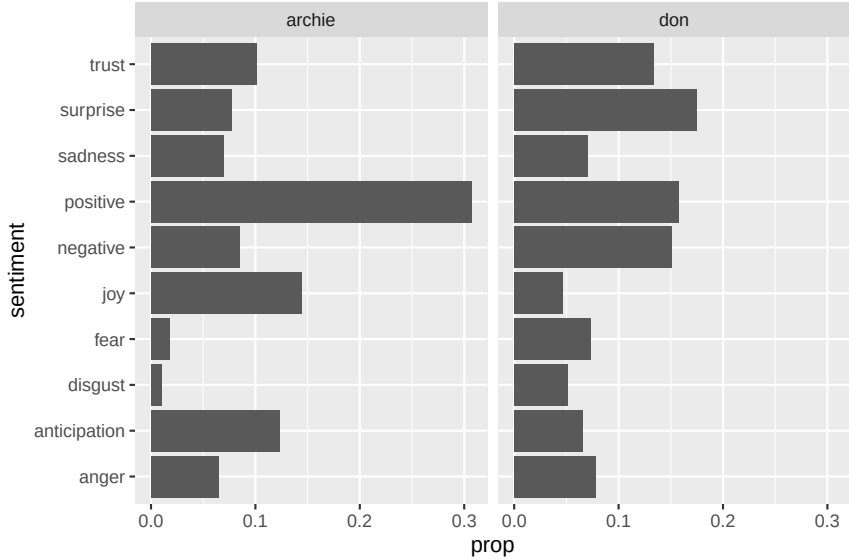
Sentiment - Plot 1 (cont.)



Sentiment - Plot 2

```
g <-  ggplot(tweet_sentiment) +
    geom_bar(mapping = aes(x = sentiment,
                           y = ..prop.., group = 1
             ),
             stat="count") +
    facet_wrap(~id) +
    coord_flip()
```

Sentiment - Plot 2 (cont.)



Positive vs Negative sentiment

```
pal <- brewer.pal(4,"Dark2")
tidy_tweets %>%
  inner_join(get_sentiments("bing"), by = "word") %>%
  count(word, sentiment, id, sort = TRUE) %>%
  acast(word ~ sentiment + id, value.var = "n", fill = 0) %>%
  comparison.cloud(colors = pal, title.colors=pal,  title.size=2)
```