Course 3 Step 1.13 - Your turn

Jiaying Wu

```
# text analysis
library(tidyverse)
library(tidytext)
```

The Simpsons

Now, it's your turn to apply what you've learned to explore characters from the popular animated TV series, The Simpsons, their dialogue and sentiment, and more.

```
# URL of simpsons_script_lines data
path_simpsons_script_lines <- "https://raw.githubusercontent.com/datascienceprogram/ids_course_data/mas

# Read simpsons_script_lines
scripts <- read_csv(path_simpsons_script_lines)

# URL of simpsons_characters data
path_simpsons_characters <- "https://raw.githubusercontent.com/datascienceprogram/ids_course_data/maste

# Read simpsons_characters
chs <- read_csv(path_simpsons_characters)
```

Left join chs to scripts, then sort the characters based on their total number of dialogue.

```
# Left join simpsons character data to the script data
sc <- left_join(scripts, chs, by = c("character_id" = "id"))

# Count the total number of dialogues from each character and arrange
sc %>%
  count(name, sort = TRUE)
```

```
## # A tibble: 6,143 x 2
##    name                    n
##    <chr>               <int>
##  1 Homer Simpson       29945
##  2 <NA>                19661
##  3 Marge Simpson       14192
##  4 Bart Simpson        13894
##  5 Lisa Simpson        11573
##  6 C. Montgomery Burns  3196
##  7 Moe Szyslak          2853
##  8 Seymour Skinner      2437
##  9 Ned Flanders         2139
## 10 Grampa Simpson       1952
## # ... with 6,133 more rows
```

Explore variables

```
# Select 'text' variables and look at an extract of the data
sc %>%
  select(raw_text, raw_character_text, raw_location_text, spoken_words, normalized_text, name, normaliz
  head(n = 10)
```

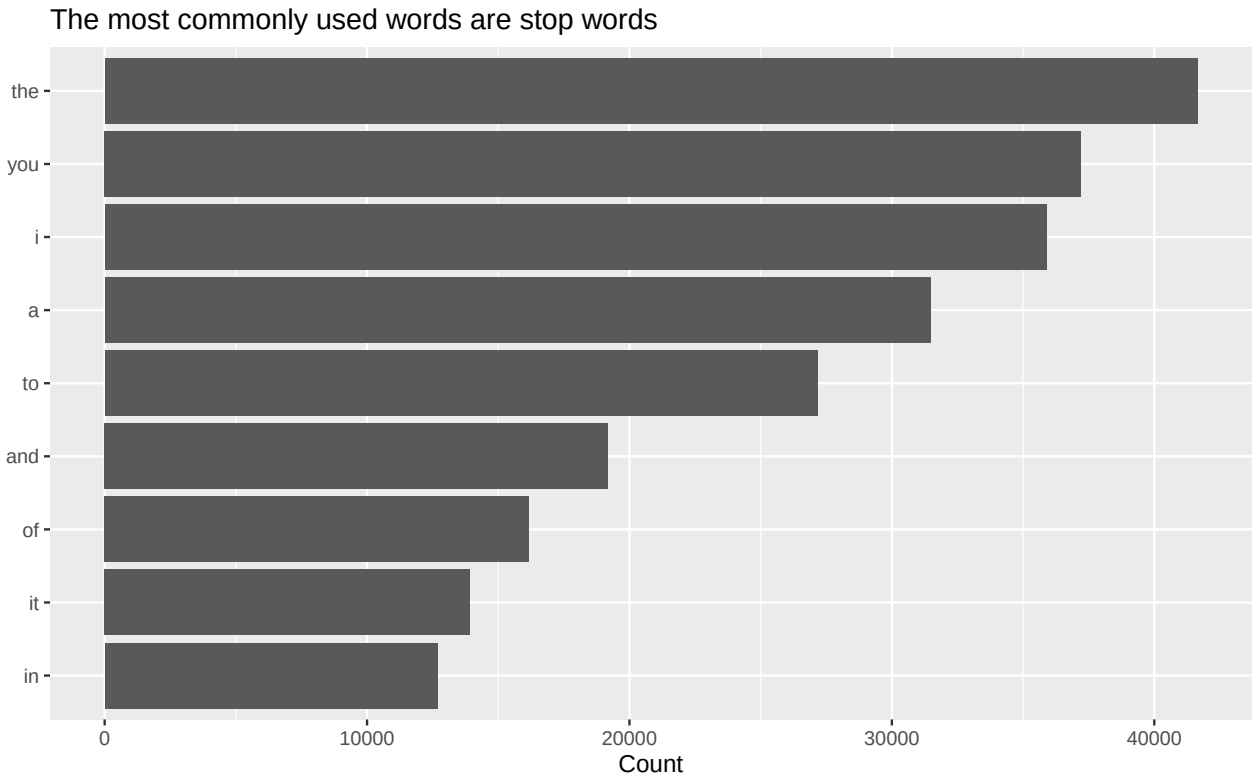
```
## # A tibble: 10 x 7
##    raw_text raw_character_t~ raw_location_te~ spoken_words normalized_text name
##    <chr>    <chr>            <chr>            <chr>        <chr>           <chr>
##  1 Miss Ho~ Miss Hoover      Springfield Ele~ No, actuall~ no actually it~ Miss~
##  2 Lisa Si~ Lisa Simpson     Springfield Ele~ Where's Mr.~ wheres mr berg~ Lisa~
##  3 Miss Ho~ Miss Hoover      Springfield Ele~ I don't kno~ i dont know al~ Miss~
##  4 Lisa Si~ Lisa Simpson     Springfield Ele~ That life i~ that life is w~ Lisa~
##  5 Edna Kr~ Edna Krabappel-~ Springfield Ele~ The polls w~ the polls will~ Edna~
##  6 Martin ~ Martin Prince    Springfield Ele~ I don't thi~ i dont think t~ Mart~
##  7 Edna Kr~ Edna Krabappel-~ Springfield Ele~ Bart?        bart            Edna~
##  8 Bart Si~ Bart Simpson     Springfield Ele~ Victory par~ victory party ~ Bart~
##  9 (Apartm~ <NA>             Apartment Build~ <NA>         <NA>            <NA>
## 10 Lisa Si~ Lisa Simpson     Apartment Build~ Mr. Bergstr~ mr bergstrom m~ Lisa~
## # ... with 1 more variable: normalized_name <chr>
```

Tokenise the variable

```
# Tokenise the clean dialogue variable and stored the tokenised text in word
sc <- sc %>%
  unnest_tokens(word, normalized_text)
```

The following bar chart describes the most commonly used words in the Simpsons.

```
sc %>%
  count(word, sort = TRUE) %>%
  top_n(10, n) %>%
  filter(!is.na(word)) %>%
  ggplot(aes(x = fct_reorder(word, n), y = n)) +
  geom_col() +
  coord_flip() +
  labs(y = "Count", title = "The most commonly used words are stop words") +
  theme(axis.title.y = element_blank())
```



1.What type of words are these?

Stop words

2.Do these words provide information about the context of the dialogue?

No they do not.

Stop words

Use the anti_join() function as well as the stop_words data frame from the tidytext package to remove stop words.

Once you have, count the number of times each word was used and arrange the words from most to least frequent.

```
# Remove stop words and then count and arrange words
sc_words <- anti_join(sc, stop_words, by = "word") %>%
  count(word, sort = TRUE)

# Look at sc_words
head(sc_words, n = 10)
```

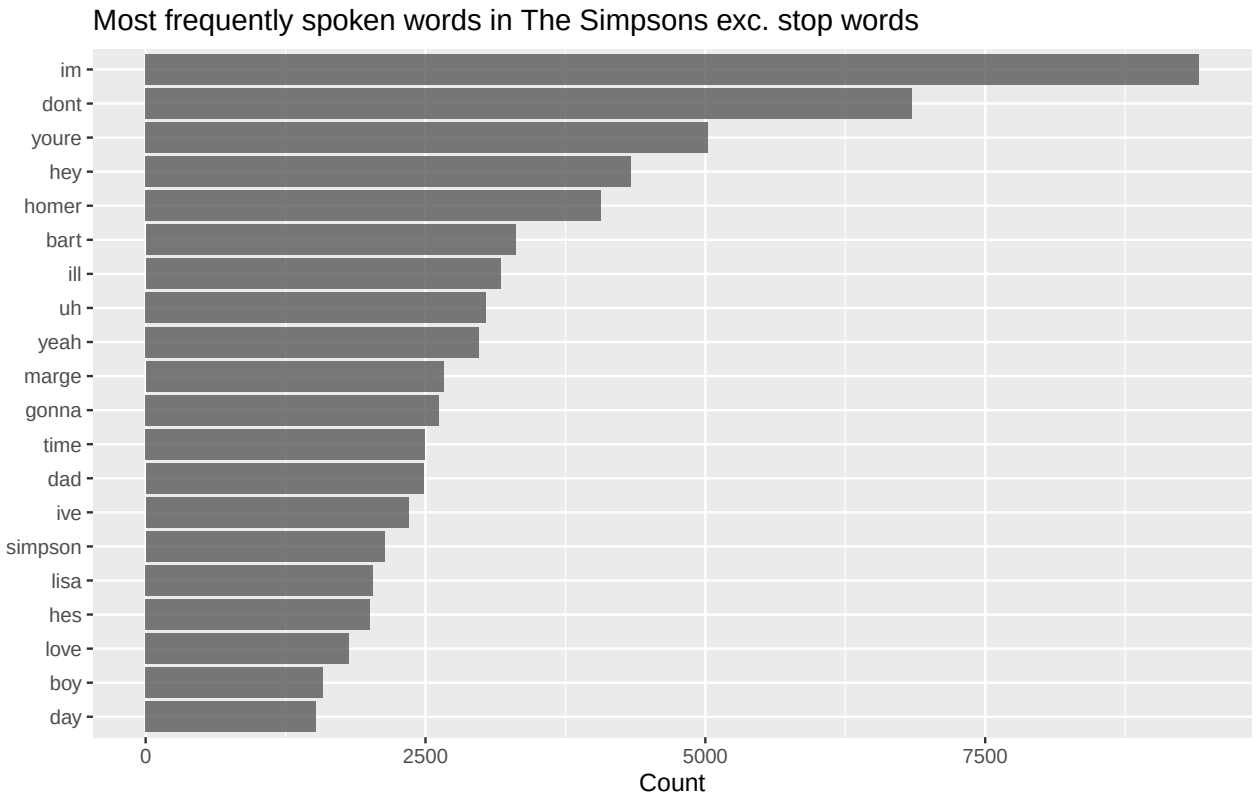
```
## # A tibble: 10 x 2
##    word      n
##    <chr> <int>
##  1 <NA>  26107
##  2 im     9412
```

```
##  3 dont   6843
##  4 youre  5021
##  5 hey    4336
##  6 homer  4068
##  7 bart   3305
##  8 ill    3170
##  9 uh     3039
## 10 yeah   2975
```

Notice that the most frequently used word is not a word but actually a missing value in the data. You will need to filter out the missing value from sc_words. Once you have, your final bar chart of the most frequently used words should appear like the following

```
# Filter our the missing value
sc_words <- sc_words %>%
  filter(!is.na(word))

# Bar chart of most frequently used words
sc_words %>%
  top_n(20) %>%
  ggplot(aes(x = fct_reorder(word, n), y = n)) +
  geom_col(alpha = 0.8) +
  labs(x = '', y = 'Count',
       title = 'Most frequently spoken words in The Simpsons exc. stop words') +
  coord_flip()
```



Sentiment analysis

Join the AFINN lexicon, which scores negative and positive word on a scale between -5 to 5, to sc_words.

```
# Join AFINN lexicon to sc_word
sc_word_scores <- sc_words %>%
  inner_join(get_sentiments("afinn"), by = "word")
```

A glimpse of sc_word_scores shows that these sentiment scores are stored in the variable value.

```
# Look at sc_word_scores
glimpse(sc_word_scores)
```

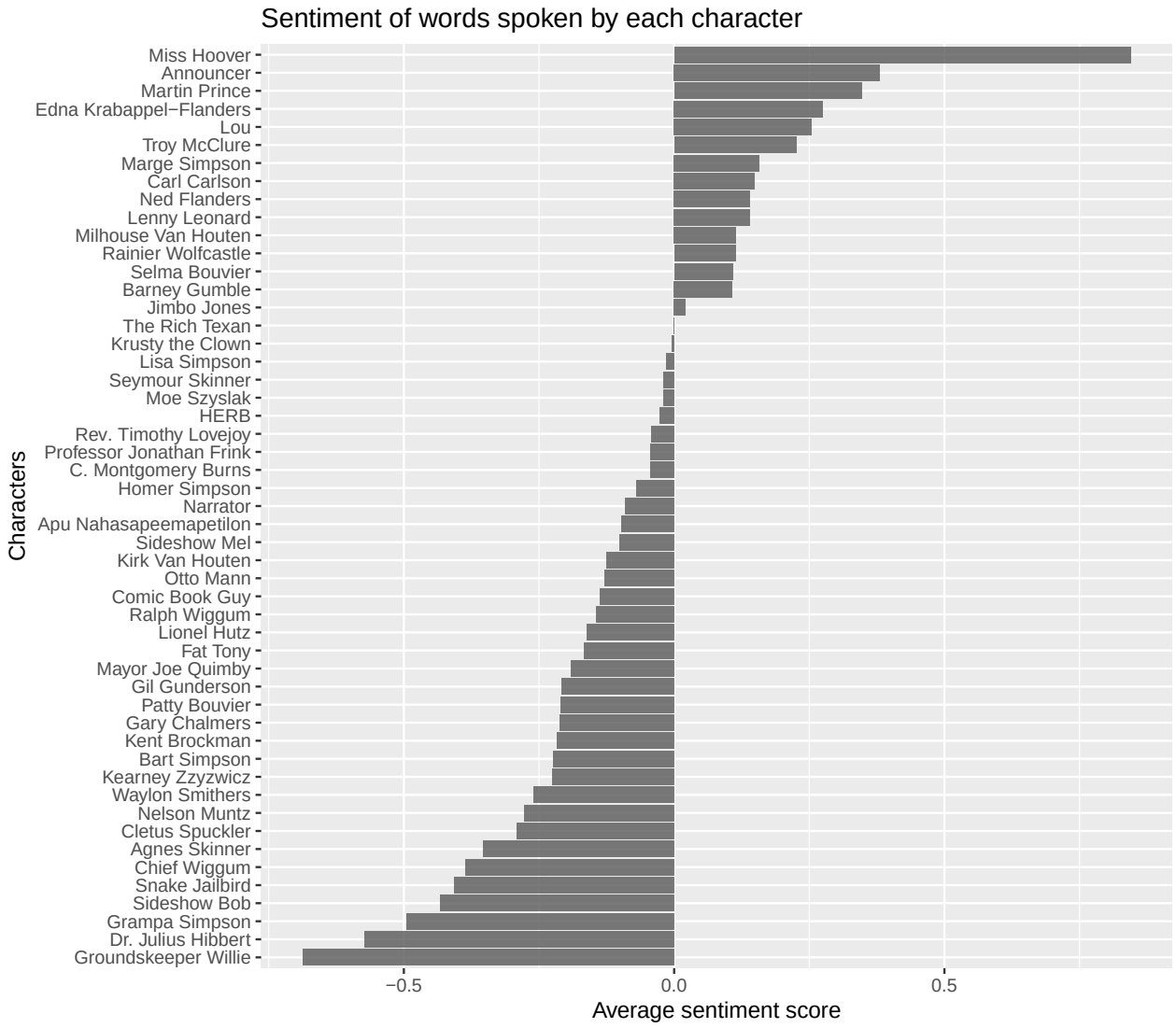
```
## Rows: 1,787
## Columns: 3
## $ word  <chr> "ill", "yeah", "love", "god", "stop", "bad", "wow", "hell", "...
## $ n     <int> 3170, 2975, 1816, 1258, 1198, 845, 790, 770, 734, 661, 657, 6...
## $ value <dbl> -2, 1, 3, 1, -1, -3, 4, -4, 3, 2, -2, 3, -3, 4, -2, -3, 1, 1,...
```

Narrow your focus

Creates a data frame of characters with at least 1500 lines of dialogues.

```
# Data of the main characters based on number of dialogue
main_char <- sc %>%
  count(name, sort = TRUE) %>%
  filter(!is.na(name)) %>%
  filter(n >= 1500)

# Bar plot of sentiment of words spoken by each character
sc %>%
  anti_join(stop_words, by = "word") %>% # remove stop words
  filter(!is.na(word)) %>% # remove missing values
  inner_join(get_sentiments("afinn"), by = "word") %>%
  filter(name %in% main_char$name) %>% # this is like an anti_join()
  group_by(name) %>%
  # average sentiment score based on words spoken (duplicates are not counted as one word)
  summarise(m = mean(value)) %>%
  ungroup() %>%
  ggplot(aes(x = fct_reorder(name, m), y = m)) +
  geom_col(alpha = 0.8) +
  coord_flip() +
  labs(y = "Average sentiment score",
       x = "Characters",
       title = "Sentiment of words spoken by each character")
```



Bart Simpson

Explore how the sentiment of his words has changed over his life.

```
# Filter names containing Bart
sc %>%
  count(name, sort = TRUE) %>%
  filter(grepl("Bart", name)) %>%
  print(n = 50)
```

```
## # A tibble: 46 x 2
##    name                                                    n
##    <chr>                                               <int>
##  1 "Bart Simpson"                                     111522
##  2 "Bartender"                                           223
##  3 "Avatar Bart"                                         217
##  4 "Hamlet Bart"                                          98
##  5 "Bart-man"                                             74
```

```
##  6 "Swedish Bartender"                                                    39
##  7 "Old Bart"                                                             38
##  8 "Bart's Class"                                                         32
##  9 "Bart Puppy"                                                           25
## 10 "Bart's Congregation"                                                  25
## 11 "Bart's Voice"                                                         23
## 12 "Jerry Lewis Bart"                                                     23
## 13 "Australian Bartender"                                                 22
## 14 "Memory Bart"                                                          21
## 15 "St. Bartholomew"                                                      21
## 16 "Werewolf Bart"                                                        17
## 17 "2-Year-Old Bart"                                                      16
## 18 "1-Year-Old Bart"                                                      14
## 19 "Bart Snail"                                                           14
## 20 "Bart's Head"                                                          14
## 21 "One-eyed Bartender"                                                   13
## 22 "Hawaiian Bartender"                                                   11
## 23 "80-Year-Old Bart"                                                     10
## 24 "Bartholomé"                                                           10
## 25 "Lunchlady Bart"                                                       10
## 26 "Bart's Prince"                                                         9
## 27 "Thought Bubble Bart"                                                   9
## 28 "Baby Bart"                                                             7
## 29 "Bart-waitress"                                                         7
## 30 "Cartoon Bart"                                                          7
## 31 "Bart Head"                                                             6
## 32 "Bart Spider"                                                           6
## 33 "Bart on Tape"                                                          4
## 34 "Bart-Jack"                                                             4
## 35 "Spider Bart"                                                           4
## 36 "Bart's Fist"                                                           3
## 37 "Shorts\" Bart,shorts bart,\n6055,\"Shorts\" Homer,shorts homer,\n605~      3
## 38 "5-Year-Old Bart"                                                       2
## 39 "Bart's Friends"                                                        2
## 40 "Bart Bat"                                                              1
## 41 "Bart Butterfly"                                                        1
## 42 "Bartenders"                                                            1
## 43 "Barts"                                                                 1
## 44 "Just Stamp the Ticket\" Man,just stamp the ticket man,\n945,Linda Ro~      1
## 45 "Yeeeessss\" Man,yeeeessss man,\n4011,Bike Guy,bike guy,\n4012,Macho ~      1
## 46 "Yesss\" Man,yesss man,\n4079,Airline Employee,airline employee,\n408~      1
```

```
# Vector of Bart at various periods
bart_names <- c("Bart Simpson", "Baby Bart",
                "1-Year-Old Bart", "2-Year-Old Bart",
                "5-Year-Old Bart", "80-Year-Old Bart")

# Dialogue from Bart of various ages
bart <- sc %>%
  filter(name %in% bart_names)

# Tokenise dialogue from Bart of various ages
bart_word <- bart %>%
  filter(!word %in% stop_words$word) %>%
```

```
  count(name, word) %>%
  ungroup() %>%
  filter(!is.na(word))

# Join AFINN lexicon to bart_word
bart_s <- bart_word %>%
  inner_join(get_sentiments("afinn"), by = "word")

# Remaining Barts
bart_s %>%
  distinct(name)
```

```
## # A tibble: 3 x 1
##   name
##   <chr>
## 1 1-Year-Old Bart
## 2 5-Year-Old Bart
## 3 Bart Simpson
```

Notice that once the AFINN lexicon has been inner joined to bart_word, the 'Barts' of some periods have been removed, leaving only 1-year-old Bart, 5-year-old Bart and regular Bart Simpson remaining.

1.Why do you think this has happened?

This is possibly because the other Barts had little dialogue.

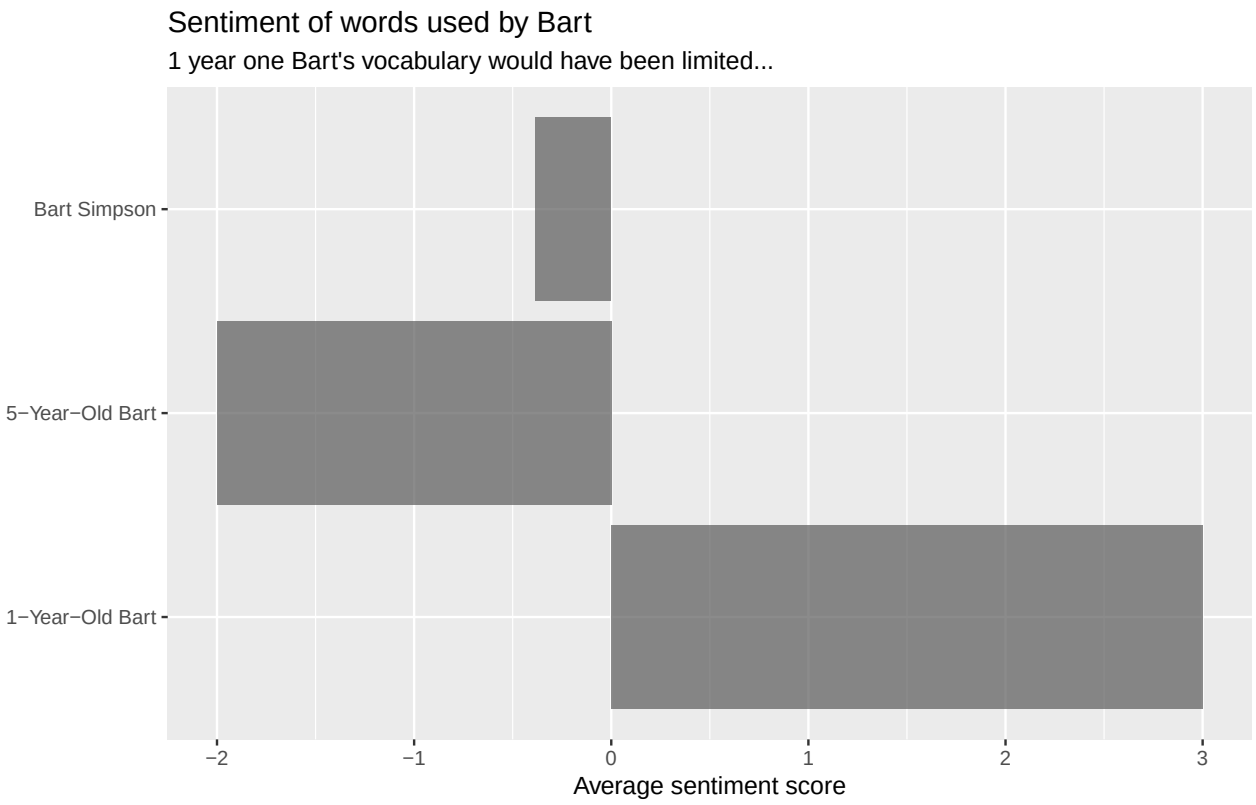
2.What problems could arise when analysing the words used by 1-year-old Bart and 5-year-old Bart?

The words spoken may not be exactly as they are (baby/child language) in the dictionary, and so AFINN may overlook those words and not assign a score to them.

Plotting Bart's sentiment score

Generate a bar plot of Bart's sentiment score at various ages.

```
# Bar plot of sentiment score of Bart's dialogue at various ages
bart_s %>%
  group_by(name) %>%
  summarise(m = mean(value)) %>%
  ungroup() %>%
  ggplot(aes(x = name, y = m)) +
  geom_col(alpha = 0.7) +
  labs(x = "",
       y = "Average sentiment score",
       title = "Sentiment of words used by Bart",
       subtitle = "1 year one Bart's vocabulary would have been limited...") +
  coord_flip()
```



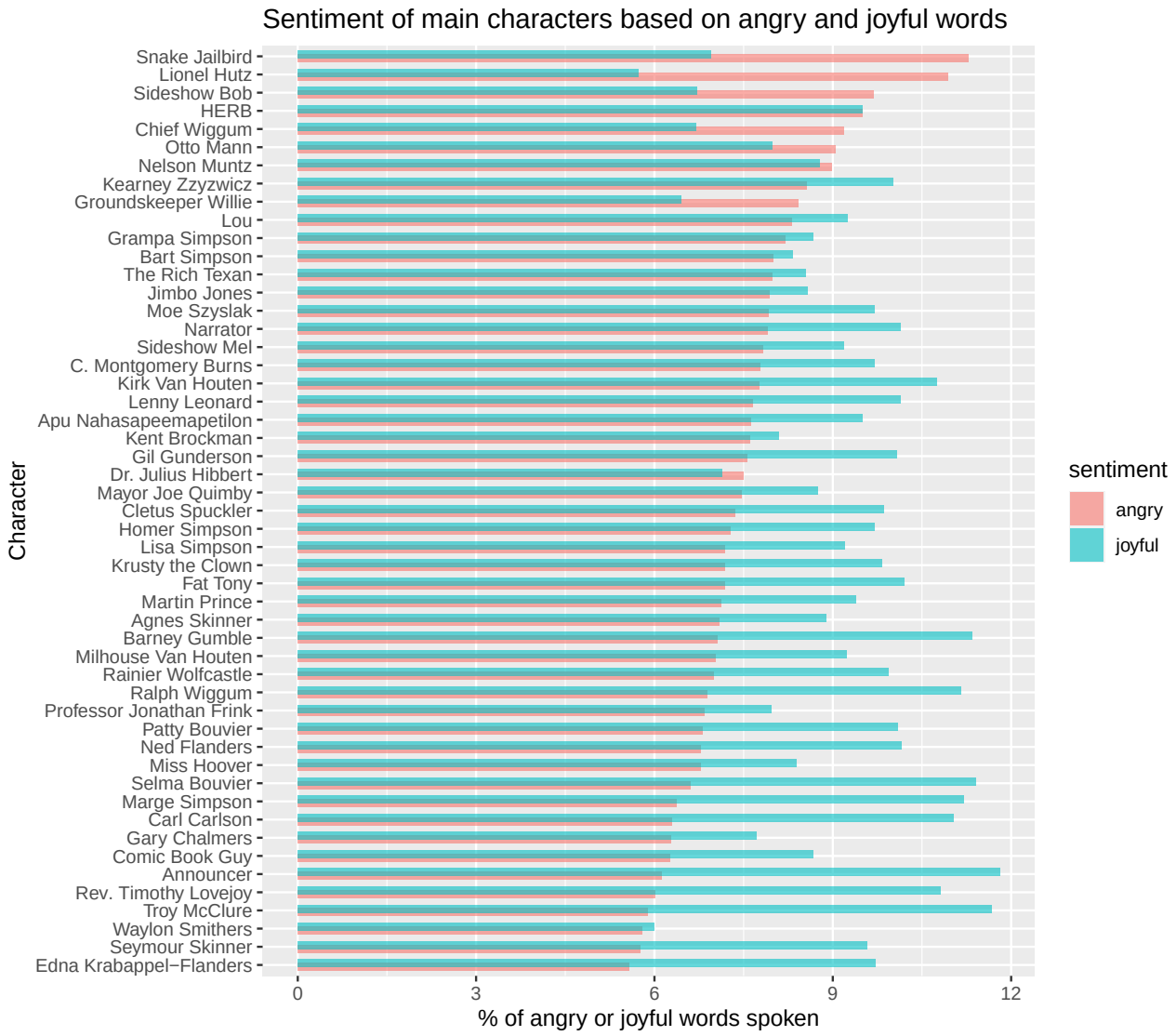
Angry and Joyful characters

Determine which characters are most 'angry' and 'joyful' based on the words they have spoken.

```
# Data of main char's words spoken with freq. and nrc sentiment
sc_senti_nrc <- sc %>%
  anti_join(stop_words, by = "word") %>% # remove stop words
  filter(!is.na(word)) %>% # remove missing values
  group_by(name, word) %>%
  summarise(n = n()) %>%
  ungroup() %>%
  inner_join(get_sentiments("nrc"), by = "word") %>%
  filter(name %in% main_char$name)

# Bar plot of angry and joyful sentiment
sc_senti_nrc %>%
  group_by(name) %>%
  summarise(angry = 100*sum(n[sentiment == "anger"]/sum(n)),
            joyful = 100*sum(n[sentiment == "joy"]/sum(n))) %>%
  ungroup() %>%
  mutate(name = fct_reorder(name, angry)) %>%
  gather("sentiment", "score", 2:3) %>%
  ggplot(aes(x = name, y = score, fill = sentiment)) +
  geom_bar(stat = "identity", alpha = 0.6, position = position_dodge(width = 0.4)) +
  coord_flip() +
  labs(title = "Sentiment of main characters based on angry and joyful words",
       y = "% of angry or joyful words spoken",
```

```
        x = "Character")
```


Sentiment of main characters based on angry and joyful words
Character
Snake Jailbird
Lionel Hutz
Sideshow Bob
HERB
Chief Wiggum
Otto Mann
Nelson Muntz
Kearney Zzyzwicz
Groundskeeper Willie
Lou
Grampa Simpson
Bart Simpson
The Rich Texan
Jimbo Jones
Moe Szyslak
Narrator
Sideshow Mel
C. Montgomery Burns
Kirk Van Houten
Lenny Leonard
Apu Nahasapeemapetilon
Kent Brockman
Gil Gunderson
Dr. Julius Hibbert
Mayor Joe Quimby
Cletus Spuckler
Homer Simpson
Lisa Simpson
Krusty the Clown
Fat Tony
Martin Prince
Agnes Skinner
Barney Gumble
Milhouse Van Houten
Rainier Wolfcastle
Ralph Wiggum
Professor Jonathan Frink
Patty Bouvier
Ned Flanders
Miss Hoover
Selma Bouvier
Marge Simpson
Carl Carlson
Gary Chalmers
Comic Book Guy
Announcer
Rev. Timothy Lovejoy
Troy McClure
Waylon Smithers
Seymour Skinner
Edna Krabappel−Flanders
0
3
6
9
12
% of angry or joyful words spoken
sentiment
angry
joyful