Strings Lab

David Gerard

2019-10-30

Learning Objectives

- Practice `stringr`, `dplyr`, and `ggplot2`.
- **WARNING**: Do **not** use `str_view()` or `str_view_all()` on these data. It will stall your computer. The data aren't *that* large, but `str_view()` and `str_view_all()` are inefficient with medium data.

Exercise 1: Scrabble Words

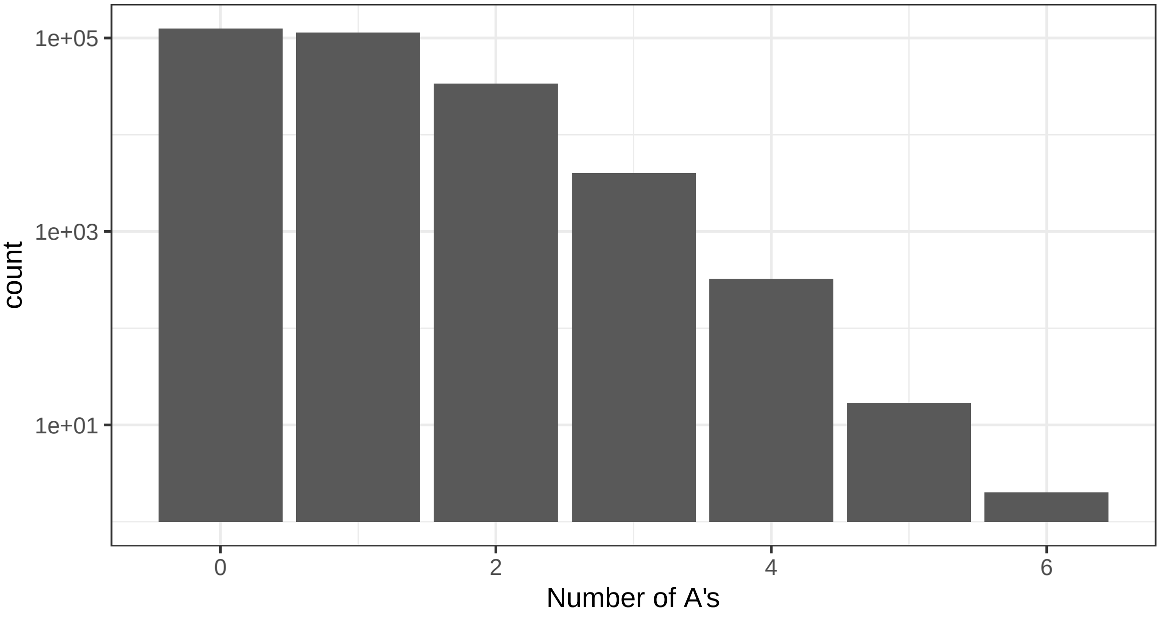
For this exercise, we are using the Collins Scrabble Words, which is most commonly used outside of the United States. The dictionary most often used in the United States is the Tournament Word List.

1. Load the 2015 list of Collins Scrabble Words into R from https://dcgerard.github.io/stat_412_612/data/words.txt (note: "NA" is an official Scrabble word).

2. There is a mnemonic rule in English: "i before e except after c". This states that if you are unsure of the sequence of "ei" or "ie", then you should choose "ie" unless this sequence follows "c". Use regex and stringr to determine how many scrabble words violate this rule and how many scrabble words satisfy this rule. Hint: check for "EITHER" in your word list.

3. Suppose you switch the "E" and the "I" in any word that contains an "EI" or "IE" pair. How many of these resulting strings still valid scrabble words? What are the shortest and longest strings that are still words? Hint: read the help file of `str_replace_all()`. You can pass a function to the `replacement` argument.

4. Let's look at the distribution of vowels the english language. For each vowel (`A`, `E`, `I`, `O`, and `U`), count the number of occurances in each word. Then make this plot:



5. Challenge (you can skip this one if you want). A *palindrome* is a word that is the same read backward as forward. For example *racecar* and *madam*. How many palindromes are there? What is the longest palindrome? You might need a `for` loop here.

Exercise 2: From RDS:

1. Replace all forward slashes in a string with backslashes. Test it out on the following string:

   ```r
   x <- "hello\\////how//are////you////"
   ```

2. Construct regular expressions to match words that:

   a. Start and end with the same character. A word of length 1 should be matched. Test it out on `"A"`, `"AB"`, and `"ABA"`.

   b. Contain a repeated pair of letters (e.g. "church" contains "ch" repeated twice.) Test it out on `"AAA"`, `"AAAA"`, and `"AABAA"`.

   c. Contain one letter repeated in at least three places (e.g. "eleven" contains three "e"s.) Test it out on `"AAA"`, `"AAB"`, and `"AABA"`.

Exercise 3: Jane Austen

For this exercise, we will consider the works of Jane Austen as stored in the janeaustenr package.

```r
library(janeaustenr)
bookdf <- austen_books()
```

`bookdf` (which we created above) is a data frame that contains a line of text and the book from which that line belongs.

1. Populate `bookdf` with line numbers (so at the start of each book, the line number begins at 1).

2. Add a column to `bookdf` called `new_chapter` that is `TRUE` if the line begins a new chapter and `FALSE` otherwise.

3. Read about the `cumsum()` function. Try it out on the following vector:

   ```r
   c(FALSE, FALSE, FALSE, TRUE, FALSE, FALSE, TRUE, TRUE, FALSE)
   ```

   What do you think `cumsum()` does when evaluated with a logical vector?

4. Use the `cumsum()` function (as well as other functions) to find the chapter number of each line of text. Add this as a new column to `bookdf` called `chapter`.

5. Apply this code to get one word per row in the data frame `janedf`:

   ```r
   library(tidytext)
   bookdf %>%
     unnest_tokens(word, text) ->
     janedf
   ```

6. Use stringr and regular expressions to create shortened titles for the books that contain just the first characters of each word. For example, `"Sense & Sensibility"` should change to `"S&S"` while `"Emma"` should change to just `"E"`. Add these shortened titles to the `janedf` data frame.

   Hints: I used `str_replace_all()` for this question. Try making the regex changes on this data frame then use joining.

   ```
   book_title <- tibble(book = levels(janedf$book))
   ```

7. Is there an association between word length and book? First, calculate the proportion of words of each length for each book.

   Now use `geom_line()` to plot the word length against proportion of words, color coding by book.

8. From the `bookdf` data frame, create a data frame with two columns, `book` and `text`. There should be only six rows, and each element in `text` should contain the entire text from the book in `book`.

9. Create a function that will take as input a string and return another string where the name of any Bennet sister mentioned is preceded with "mecha". The Bennet sisters are Elizabeth (Eliza or Lizzy), Mary, Kitty (Catherine), Lydia, and Jane.

   For example, in my implementation, `mechabennet()`, I have the following outputs:

   ```
   text <- "Elizabeth passed quietly out of the room, Jane and Kitty followed"
   mechabennet(text)
   ```

   ```
   ## [1] "mecha-Elizabeth passed quietly out of the room, mecha-Jane and mecha-Kitty followed"
   ```