Class12 Assignment

FedericaFusi

4/4/2021

Data

For this assignment, we are using the **state.name** and the **sentences** character vectors. They are both included in the tidyverse packages.

```
class(state.name)
class(sentences)

length(state.name)
length(sentences)
```

The scope of this assignment is to gain some practice with string manipulation and pattern matching. We won't care about data structure or cleaning, which we will address next week. The assignment consists of a set of exercises to be solved individually.

As last week, please submit your code but also a PDF file created with R markdown with your final results.

Exercises

Take advantage of the str_ functions to accomplish the following tasks. You are welcome to experiment with multiple syntaxes if you wish.

Part 1

Use the **state.name** dataset.

1. Extract the following states: Arizona, California, Illinois, Oregon
2. Extract all states that begin with the letter **a**. Use both **str_subset()** and **str_extract()** to see how the syntax differs. In both cases, extract the **full name** of all the states.
3. Extract all states whose names finish with **a OR e**.
4. Count how many state names are composed of **two words** (e.g. North Dakota)
5. Extract all states that contain at least one of these letters: **n, t, w, c** at any position within the name
6. Count how many states do **not** contain any one of these letters at any position: **w, z, y**
7. Extract all states that contain **at least one** of these letters: **c or i**. Exclude the initial letters from your count (e.g., do not include Colorado).

8. Identify which states are exactly 6-letter long (DO NOT use str_length here!!) **AND** states that are at least 6-letter long
9. Create a regular expression that will match any string in state.name vector
10. Find all states that start with two consonants.
11. Find all states that have two or more vowels in a row.
12. List the states that start with a vowel and end with a consonant

Part 2

Now use the Havard **sentences** vector. Start by transforming it into a tibble.

```
class(sentences)
length(sentences)

sentences_tbl <- as_tibble(sentences)

sentences_tbl
```

Create a few new columns to store the following information. Use **dplyr** functions, not the dollar sign framework.

1. The number of letters in each sentence. *What is the average number of letters across sentences?*
2. A dummy variable equal to 1 if the sentence contains a word ending with "ing", and 0 otherwise. Note that words not sentences should end in ing! *How many words with ing in the dataset?*
3. A dummy variable equal to 1 if the sentence contain a color such as red, orange, yellow, green, blue, purple, pink and 0 otherwise. Take advantage of the **paste0** function to answer this question (see slides). Pay attention not to have a too permissive pattern! *How many sentences contain colors?*
4. The number of words in each sentence. This requires a bit of creativity and there are likely multiple solutions! Give it a fair try but it won't negatively affect your grade if you cannot figure it out :)

A few hints:

- Have a look at the syntax for common sequences here (http://uc-r.github.io/regex_syntax), especially the first table in the "Sequences" section. Which symbols might help you to identify individual words? Play around with str_view_all to test your syntax.
- Note that I ended up using a loop here!