Data science with R: tidyverse

III Data Wrangle: strings - factors (stringr & forcats)

Assignment

Create *R* script called *assignment_3.R*. From course sources download zip file called **corpus.zip**, extract its content in your **data** folder inside your *R*'s project folder.

Exercise 1

First import **corpus.txt** into R. Use function **readLines()** for import, assign imported corpus to R object.

If you if will have trouble running the code for this assignment (maybe it will run slow). You should use a smaller sample of corpus (code will be provided for sampling in the solution video and R script).

First some warming up:

- check number of lines in corpus
- check number of characters
- print first and last six lines

Now use **regular expressions** and tools provided by **stringr** library to finish given tasks:

- count how many lines include at least one punctuation
- show first 20 lines without any punctuation
- count how many lines include at least one number / digit
- inspect first 10 lines with digit present
- find string patterns that resemble phone numbers (search for patterns: *ddd-dddd* where d = digit 0-9)
- find string patterns that resemble dollar signs "$" (escaping needed)
- how many lines starts with word "The"?

Exercise 2

Continue with strings manipulations:

- use corpus and try to figure out which words usually come before comma ","
- if you consider first 5 letters at the beginning of each line, what are the top patterns that lines start with?
- find words where: a vowel is followed by a vowel
- find words where: a vowel is followed by two or more vowels
- find words where: 2 vowels are not followed by a vowel
- check occurrence of words "the", "be", "to", "of", "and" (*most common words counts*)
- *most common words counts*: sort words by their frequency!
- *most common words counts*: before pattern match convert corpus to lower case!
- inside *most common words counts* find *top 3 most common words*
- *top 3 most common words* check: number of lines only one word is present
- *top 3 most common words* check: number of lines 2 words are present
- *top 3 most common words* check: number of lines all three words are present
- *top 3 most common words* check: also add percentage % of lines for each given scenario!

Exercise 3

In this task the first thing you should accomplish is clean the corpus. When cleaning the corpus you should consider:

- convert all characters to lower case
- remove punctuation
- remove extra white spaces (more than one white space)
- replace tabs or new lines with a white space
- remove all digits
- assign clean corpus to object called **corpus.clean**

Now use clean corpus:

- and create a table called **corpus.words**
- **corpus.words** holds two columns: **word** - word from the corpus, **count** - frequency or counts how many time given word appears in the corpus
- to table **corpus.words** add column **% coverage**, which tells us the percentage of text covered by specific word
- $\%\ coverage = \frac{count}{sum(count)} \times 100$

Using **corpus.words** try to anwer the questions below:

- how many different words were found in corpus?
- how much text is covered with the most frequent word?
- how much text is covered with the top 10 most frequent words?
- how many words we need to cover 50%, or 70%, or 90% of corpus?

Exercise 4

In the last exercise we will create table **corpus.words.top100**:

- use table **corpus.words**
- first sample words out of top 100 most frequent words
- **HINT:** keep only first 100 rows of sorted table **corpus.words**
- repeat sampling 10000 times
- the probability for a word to be selected in sampling step is defined by column **count** ($prob = \frac{count}{sum(count)}$)
- **HINT:** for sampling rows from given table use **dpplyr**'s function **sample_n()**, with arguments: *size=10000*, *replace=T*, *weight=prob*
- when **corpus.words** is created convert column **word** to **factor** type
- extract unique factor levels using function from **forcats** package
- count each factor level occurrence using function from **forcats** package