1. **What regular expressions do you use to extract all the dates in this blurb:** [**http://bit.ly/regexexercise2**](http://bit.ly/regexexercise2) **and to put them into the following format YYYY-MM-DD?**

My answer can be seen in this link: <https://regex101.com/r/3Bf8KH/2>

Et billede, der indeholder tekst, elektronik, skærmbillede, software

Automatisk genereret beskrivelse

First, I made three different groups, each in its own parenthesis. Each parenthesis consists of “\d” which “matches any single digit”[[1]](#footnote-1). Then I outline the number of the digits in each group. To outline this, I use the curly brackets (Tuborgklammer), in which I write A) “1,2” for one or two digits, or B) “4” for four digits. The last step to do is managing the characters between the three groups. These characters are dots, commas, hyphens, and slashes. Spaces between the different parts are also included here. To solve that, I add a dot between the groups, outside the parentheses. This dot implicates that there is a character. According to the guide, the dot “matches any character”.[[2]](#footnote-2) In the two dates, “3.27,•1513” and “5/14,•1607”, there are spaces. To catch these, I add an asterisk because it matches “the preceding element zero or more times”.[[3]](#footnote-3) The last thing to do is to implement the right date-format, which is YYYY-DD-MM, in the other field (substitution, the above field on the site). To do that, I use dollarsign to arrange in the right order. The string is “$3-$1-$2” because the years are the last group, the months are the first group, and the days are the second group.

1. **Write a regular expression to convert the stopwordlist (list of most frequent Danish words) from Voyant in** [**http://bit.ly/regexexercise3**](http://bit.ly/regexexercise3) **into a neat stopword list for R (which comprises “words” separated by commas, such as** [**http://bit.ly/regexexercise4**](http://bit.ly/regexexercise4)**). Then take the stopwordlist from R** [**http://bit.ly/regexexercise4**](http://bit.ly/regexexercise4) **and convert it into a Voyant list (words on separate line without interpunction)**

First, I copied the stopwordlist from the Google Sheet. Then, I inserted the list in Regex101. Before this step, I chose “Substitution” under “Function” in the left side. I inserted the stopwordlist in the above field (called “Text String”). To convert the stopwordlist I used the regular regression “\n” which matches newline characters. I used this because the stopwordlist was in a sheet format. After that, I wrote the regression: “,” in the substitution field. This regression indicates that the different words shall be separated by commas. Furthermore, the words should be in quotation marks according to the example file in the task description.

This part can be seen here:

From Voyant to stopwordlist for R

<https://regex101.com/r/g3E5ew/1>

**Et billede, der indeholder tekst, skærmbillede, software, Computerikon

Automatisk genereret beskrivelse**

Then, I should take the stopwordlist from R and convert it to into a Voyant list which means that the words shall be on separate lines without interpunction. To solve this part of the task, I wrote the same regressions but in the reverse order. Therefor, I wrote: “,” to catch the way, in which the words were organized. Then, I wrote “\n” to indicate that the words should be on lines.

This can be seen here:

From stopwordlist for R to Voyant: successfully, hurraayyy!

<https://regex101.com/r/Fmn3Lu/1>

Et billede, der indeholder tekst, skærmbillede, software, Webside

Automatisk genereret beskrivelse

1. **In 250 words, answer the following question: “What are the basic principles for using spreadsheets for good data organization?”**

For the first, it is very important to be consistent! An example is writing dates. You can’t use two or several different date formats. A recommended date format could be following: YYYY-MM-DD. Furthermore, it is very important to treat the cells in a good way which means that you can’t leave them empty. You must keep one thing in a cell. A good structure for organizing the data would be a rectangle. That means subjects as the rows and the variables as the columns. It is also a very good idea to create a data dictionary which makes it easier for others to read, use and understand the datasheet. The datasheet must be readable, understandable, reproducible, and usable. Choosing good names for things is also very important. The names must be understandable and logical – not something random stuff! Color or highlighting as data is not preferable. Calculations in the raw data files either. Keep these in another file. One of the most important tips is about the treatment of your files. You must make several backups. After every crucial change of the file, you must make a

backup of it. Using data validation is important as well for avoiding data entry errors. The last important tip is that you must save the data in plan text files. To sum up, datasheets are very useful for organizing big quantities of data. Furthermore, you can generate graphs and several other kinds of charts and diagram – based on your datasheet in Excel or Google Sheets.

1. **Challenge (OPTIONAL)! Can you find all the instances of ‘Dis Manibus’ invocation in the EDH inscriptions in** [**https://bit.ly/regexexercise5**](https://bit.ly/regexexercise5)**? Beware of the six possible canonical versions of the Dis Manibus formula!.**

To solve this problem, I wrote this regular expression:

\b[Dd]is\s[Mm]anibus\b

The above regular regression consists of many different parts. “\b” matches a word boundary which means that the start of the word will be “D” or “d”. The square brackets “[Dd]” means that the word starts with capital or lowercase. Therefor it will be “Dis”. “\s” matches any space in the text. Again, “[Mm] matches the capital or lowercase letter. The last part, “\b” is a word boundary which means that the word “Manibus” ends with “s”.

Here is the link:

<https://regex101.com/r/Zl9t9O/1>

Et billede, der indeholder tekst, skærmbillede, nummer/tal, dokument

Automatisk genereret beskrivelse

1. Regular Expressions, <https://librarycarpentry.org/lc-data-intro/01-regular-expressions.html>, accessed 23 November 2023 [↑](#footnote-ref-1)
2. ibid. [↑](#footnote-ref-2)
3. ibid. [↑](#footnote-ref-3)