Week 1 - Regex/Tables

Emil Bæk Mogensen

2023-09-14

Packages

```
library(tidyverse)
```

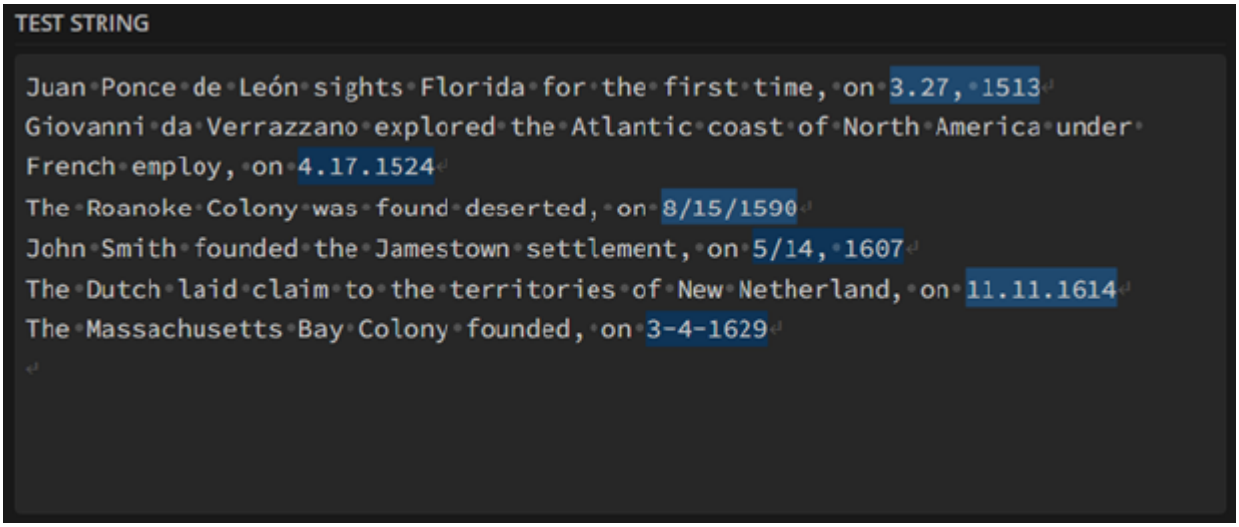
```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.3     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.0
## v ggplot2   3.4.3     v tibble    3.2.1
## v lubridate 1.9.2     v tidyr     1.3.0
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(tinytex)
library(knitr)
```

1a) What regular expressions do you use to extract all the dates and to put them into the following format YYYY-MM-DD?

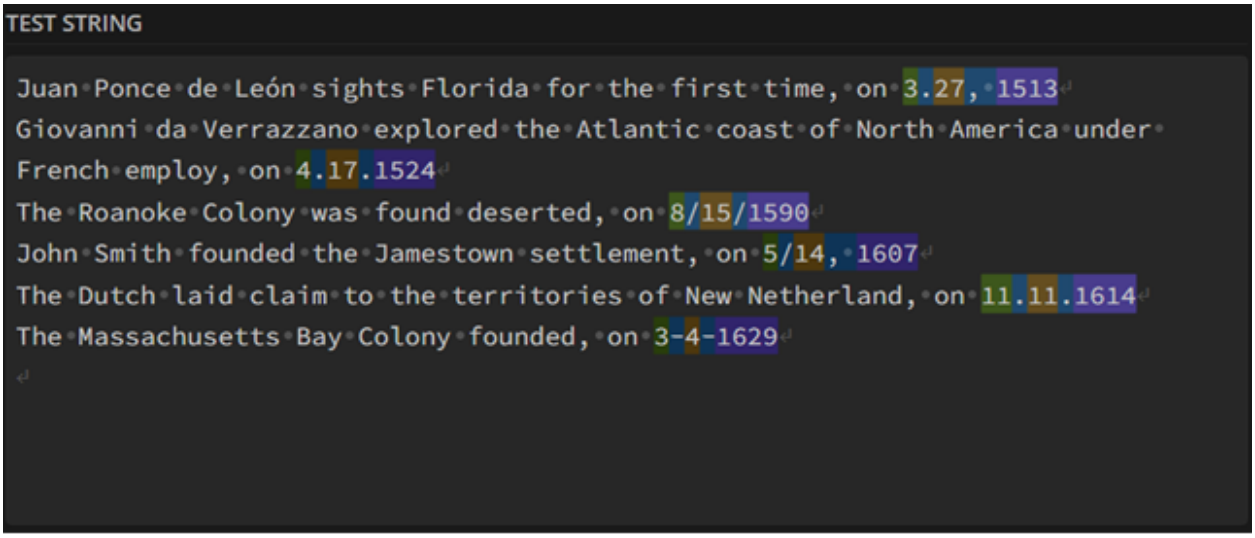
Firstly I wrote the following regex to match all the date formats in the text.

```
#Regex: \d{1,2}.\d{1,2}.*?\d{4}
```



To make them the same format grouping is needed of the above explanation. The regex will then look like the following

```
#Regex: (\d{1,2}).(\d{1,2}).*?(\d{4})
```



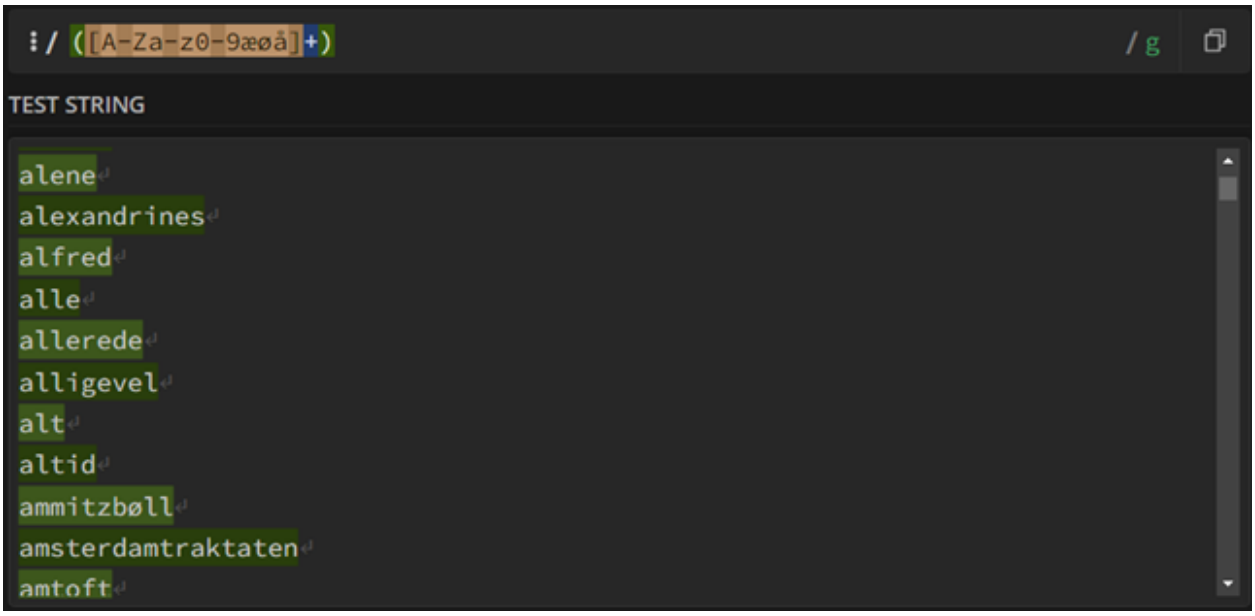
The substitution pattern formats the text into the same format:

```
#Substitute: $3-$2-$1
```

2a) Write a regular expression to convert the stopwordlist (list of most frequent Danish words) from Voyant into a neat stopword list for R (which comprises "words" separated by commas.

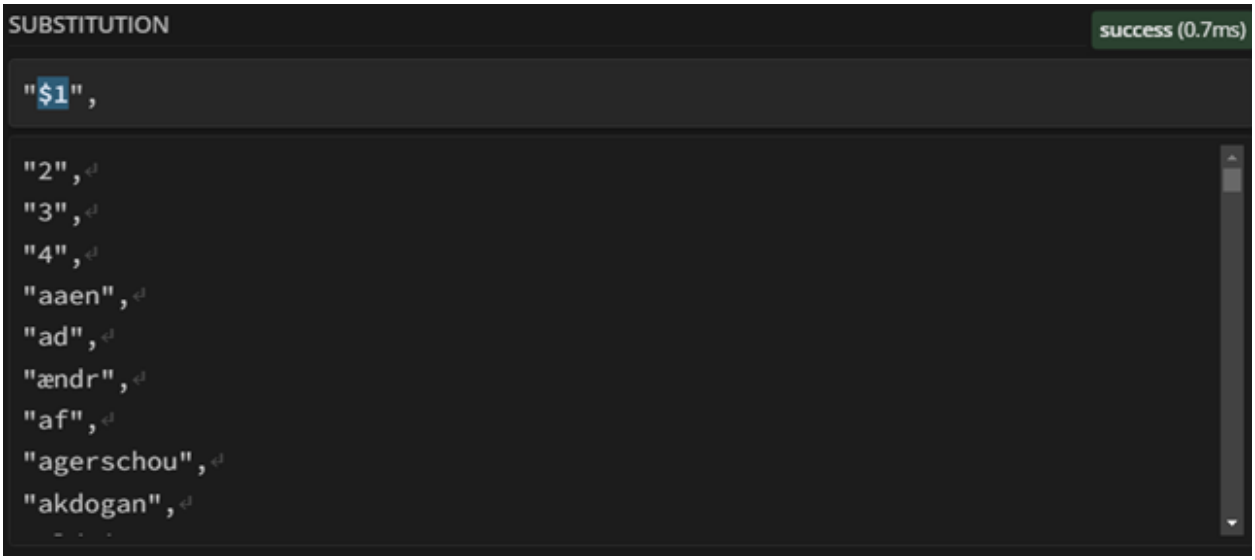
The following substitution makes the stopword list for R as in words separated by commas.

```
# Regex: ([A-Za-z0-9æøå]+)
```



The following substitution makes the stopword list for R as in words separated by commas.

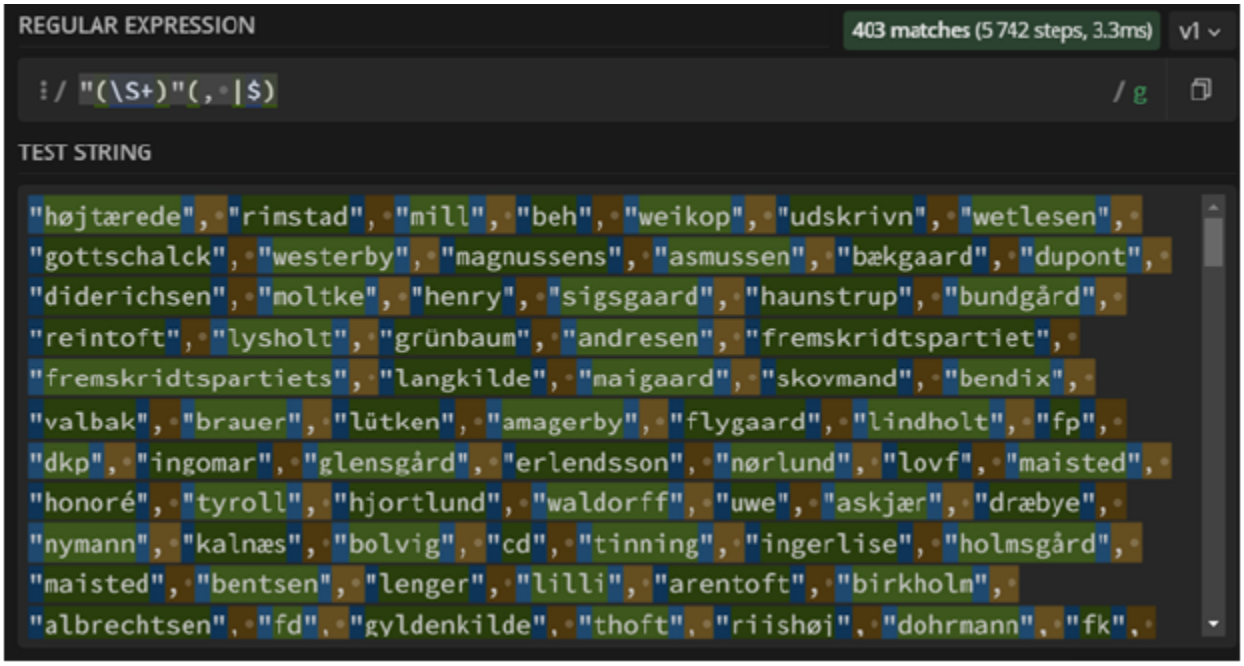
```
# Substitute: "$1",
```



2b) **Then take the stopwordlist and convert it into a Voyant list of words on separate line without interpunction) To convert the R stopwordlist into Voyant list:**

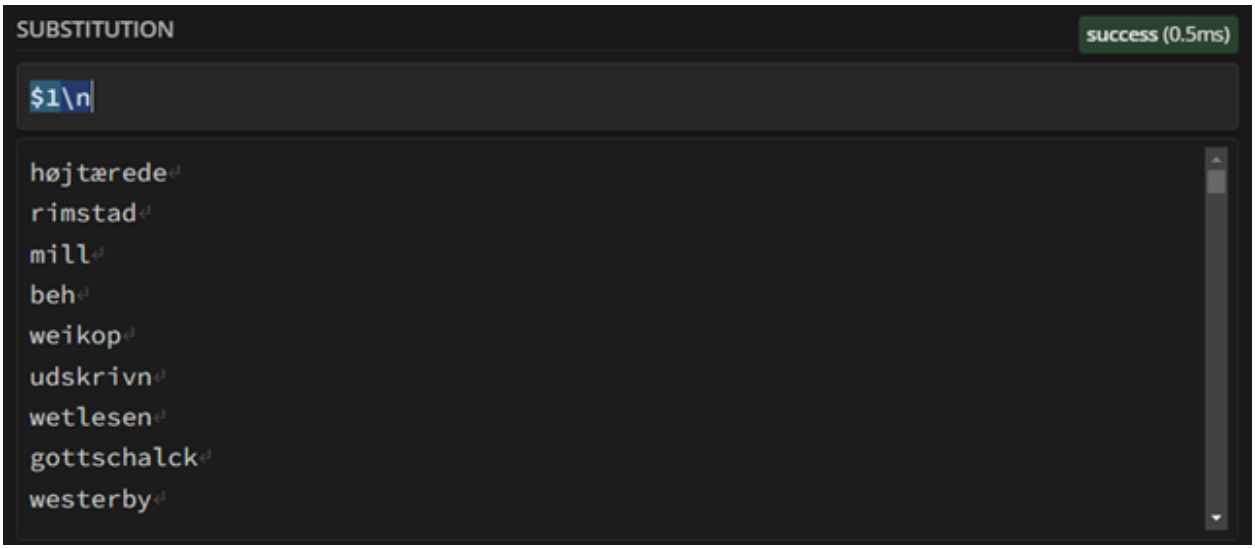
To convert the R stopwordlist into a Voyant list I used the expression:

```
# Regex: "(\S+)"(, |$)
```



Then i substituted with the following:

```
# Substitute: $1\n
```



3) **In 250 words, answer the following question: "What are the basic principles for using spreadsheets for good data organisation?**

When working with big datas in programs such as excel, it is important to do all tables in different sheets or different files. Multiple tables can draw false associations for the computer, but it also makes it easier as a researcher or reader to keep a clear overview. In extension, making backups and using any form of version control like git is crucial for backtracking and reproducing the science. It is all about standardization or, tidying the data. When one takes into account that 80 % of data analysis is spent on cleaning and preparing it. (Dasu and Johnson 2003) In tidy data each variable forms a column, each observation forms a row, and each observational unit forms a table (Wickham 2014). Furthermore, following a consistent structure inside

a rectangular sheet combined with the 'tidy' approach, which emprises a coherent structure and consistency in regard to variable names, subject identifiers, data layout, file names, avoiding special characters and using the same dating format such as the "ISO 8601" standard, further eliminates errors, mistakes and allows other researchers to work with the science more fluently. Although it is debated whether filling out empty cells with NA or leaving them blank is the best, what is key is following the same consistent structure as long as it does not interfere with the given program's calculations. Lastly, it is of the upmost importance that one never edits or works in the raw data, which should always be saved as a raw text file where editing is disabled before it is opened. Importing and working on the data in a different file ensures that the risk of human errors, i.e., unintentional manipulation of the raw data, is eliminated.

4) Challenge (OPTIONAL): Can you find all the instances of 'Dis Manibus' invocation?

```
# \b(?:D\s?M(?:\sS)?|Di[si]{1,2}\sM[ai]{1,2}nibus(?:\sSacrum)?)\b
```