ITA-2017 R-Tutorial

Hands-On Exercise 3

Task: In this last excercise, we well read in a text from a file, count the number of times, each word appears and output the result as a histogram of the most frequent words.

Download the file `test-sentences.txt` from the tutorial-page `www.smiffy.de/ita-2017` to your local computer.

In the next step, we read in the whole file in a vector of strings. Hint: In the setwd-comand you have to adapt the path to the location, where the downloaded file resides.

```
setwd('d:/Dropbox/ita-2017/tutorial')  # please adapt path
lines<-readLines('test-sentences.txt')
lines
str(lines)
```

The first line sets the working directory, in which the text file resides, while the second line reads in all the lines in a vector of strings. The last commands shows you the structure and content of the variable lines, which holds the text from the file.

In the next step, we concatenate the single lines, so that we have a long string, containing the whole text from the file. This is done with the command `paste(...)`:

```
text<-paste(lines, collapse=" ")
str(text)
```

The optional second parameter is responsible, that the lines are concatenated with an addituional space in between. Because we don't want to distinguish between 'The' and 'the', we change all the text to lowercase:

```
text<-tolower(text)
str(text)
```

Next we split the whole text into words. This can be done using the function `strsplit(...)`. `strsplit(...)` takes the string to split as the first parameter and the splitting character sequence as the second parameter. So in a first run we try, we split the text along the whitespace character:

```
words<-strsplit(text, ' ')
words
```

When inspecting the content of the Variable `words`, we realize, that the whitspace character is not sufficient as splittting sequence, because from time to time we also have punctation characters in our text (look for example the name „ishmael.“), which are wrongly assigned to the words. Fortunately, strsplit, also supports regular expressions, and so instead of specifying a single character or charactersequence, we can specify all characters, which are no letters as splitting character. This can be specified using the inverse character class of all letters (\W). The improved statement than looks as follows:

```
words<-strsplit(text,'\\W+')
words
```

The result is a vector of words. To count the number of times, the words appear inside the vector, we use the table-command:

```
occurence<-table(words)
occurence
str(occurence)
```

The result is a two dimensional datastructure of type table. The result can further be sorted in decreasing order with the following command:

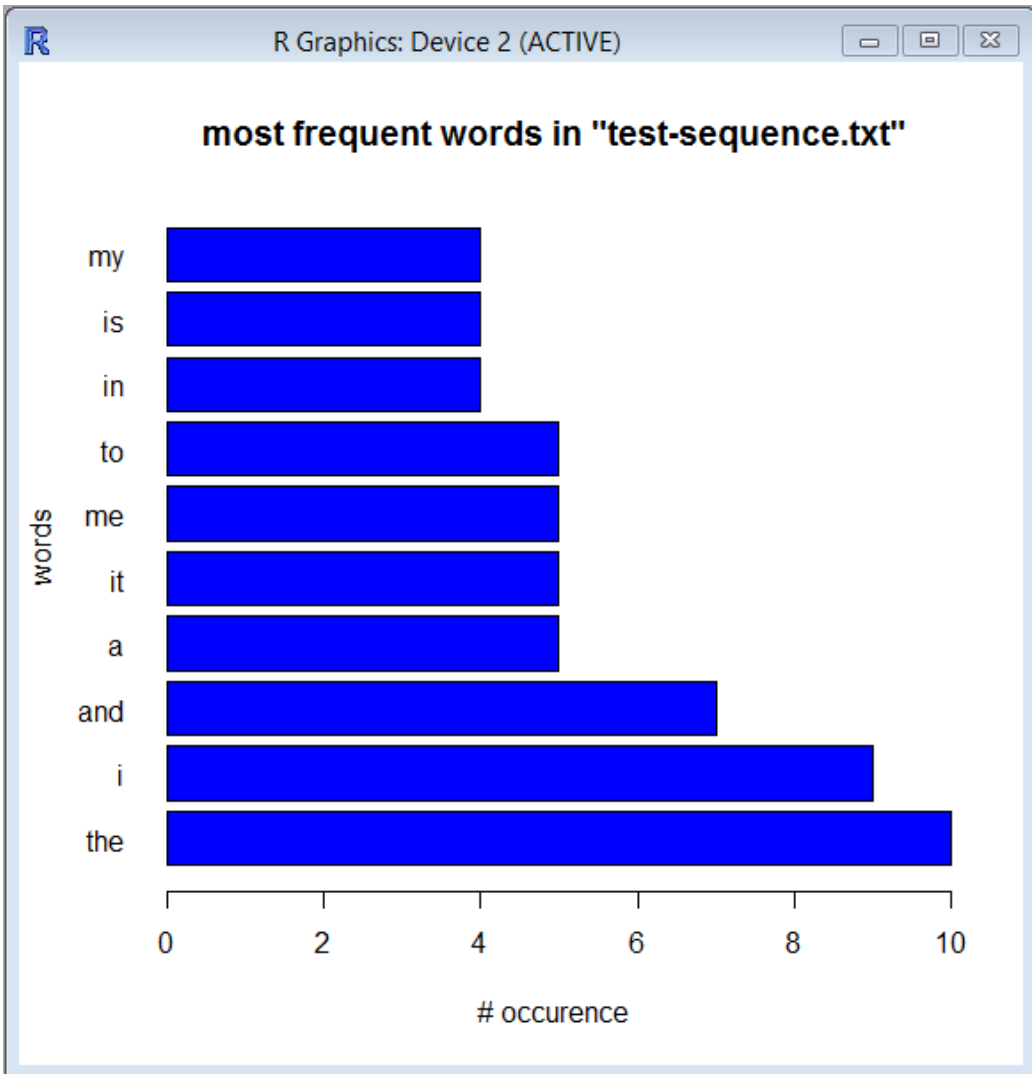
```
occurence<-sort(occurence, decreasing=T)
occurence[1:10]
```

The second line, limits the output to the ten most frequent words in the text. As a last step, we want to visualize this result in a barplot:

```
barplot(occurence[1:10])
```

Nice - isn’t it? To improve the visualisation of the barplot you can add a title, change the orientation of the words, add some axis description and so on. To get an imprssion of what can be done, type ?barplot into the R console and take a look at the manual-page. Try some of the possible extensions ...

Can you achieve the output from the next side?



As a last excercise, try to bundle all the previously used commands in a function, which takes the path to the file to inspect as the first parameter and the number of words to display as second parameter.

Example call:

```
path<-'d:/ita-2017/test-sentences.txt'
print_frequent_words(path, 15)
```