Exercises on regular expressions

Contents

Exercise 1

- Create a variable called `text1` and populate it with the value "The current year is 2017"
- Create a variable called `my_pattern` and implement the required pattern for finding *any digit* in the variable `text1`.
- Use function `grepl` to verify if there is a digit in the string variable.

```
> text1 <- "The current year is 2017"
> my_pattern <- "[A-z]*[0-9]+[A-z]*"
> grepl(my_pattern,text1)
```

```
[1] TRUE
```

Exercise 2

- Use function `gregexpr` to find all the positions in `text1` where there is a digit.
- Place the results in a variable called *string_position*
- Can you obtain the same result using a function from the `stringr` package?

```
> string_position <- gregexpr(my_pattern,text1)
> string_position[[1]][1:length(string_position[[1]])]
```

```
[1] 21
```

```
> require(stringr)
> str_locate(text1, my_pattern)
```

```
     start end
[1,]    21  24
```

Exercise 3

- Create a variable called `my_pattern` and implement the required pattern for finding **one digit** and **one uppercase alphanumeric** character, in variable `text1`. HINT: combine predefined classes in the regex pattern.
- Use function `grepl` or its `stringr` equivalent to verify if the searched pattern exists on the string.

```
> my_pattern <- "[[:upper:][:digit:]]"
> grepl(my_pattern,text1)
```

```
[1] TRUE
```

```
> str_locate_all(text1, my_pattern)
```

```
[[1]]
     start end
[1,]     1   1
[2,]    21  21
[3,]    22  22
[4,]    23  23
[5,]    24  24
```

Exercise 4

- Use function `regexpr` to find the position of the first space in `text1`.
- Place the results in a variable called `first_space` and Use function `grepl` or its `stringr` equivalent to verify if the searched pattern exists on the string.

```
> my_pattern <- "[[:blank:]]"
> first_space <- regexpr(my_pattern,text1)
> first_space[[1]][1]
```

```
[1] 4
```

```
> str_locate(text1, my_pattern)
```

```
     start end
[1,]     4   4
```

Exercise 5

- Create a pattern that checks in `text1` if there is a lowercase character, followed by any character and then by a digit.

```
> my_pattern <- "[[:lower:]].[[:digit:]]"
> grepl(my_pattern,text1)
```

```
[1] TRUE
```

```
> str_detect(text1, my_pattern)
```

```
[1] TRUE
```

Exercise 6

- Find the starting position of the above string. Place the results in a variable called `string_pos2`

```
> string_pos2 <- str_locate(text1, my_pattern)
> string_pos2[1]
```

```
[1] 19
```

```
> string_pos2 <- gregexpr(my_pattern,text1)[[1]][1]
> string_pos2
```

```
[1] 19
```

Exercise 7

- Find the following pattern: one space followed by two lowercase letters and one more space.
- Use a function that returns the starting point of the found string and place its result in `string_pos3`.

```
> my_pattern <- "\\s[a-z][a-z]\\s"
> string_pos3 <- str_locate(text1, my_pattern)
> string_pos3
```

```
     start end
[1,]    17  20
```

```
> string_pos3 <- gregexpr(my_pattern,text1)[[1]][1]
> string_pos3
```

```
[1] 17
```

Exercise 8

- Using the sub function, replace the pattern found on the previous exercice by the string " is not “”
- Place the resulting string in `text2` variable.

```
> text2 <- sub(my_pattern," is not ",text1)
> text2
```

```
[1] "The current year is not 2017"
```

```
> text2 <- str_replace(text1, my_pattern," is not ")
> text2
```

```
[1] "The current year is not 2017"
```

Exercise 9

- Find in `text2` the following pattern: Four digits starting at the end of the string.
- Use a function that returns the starting point of the found string and place its result in `string_pos4`.

```
> my_pattern <- "\\d{4}$"
> string_pos4 <- gregexpr(my_pattern,text2)[[1]][1]
> string_pos4
```

```
[1] 25
```

```
> string_pos4 <- str_locate(text2, my_pattern)
> string_pos4
```

```
     start end
[1,]    25  28
```

Exercise 10

- Using the `substr` function, and according to the position ofthe string found in the previous excercise, extract the first two digits found at the end of `text2`.

```
> substr(text2,start = string_pos4,string_pos4+1)
```

```
[1] "20"
```