Advent of Code: Day 5

Keyi Long

2/19/2021

Reference

RegEx Cheatsheet: https://rstudio.com/wp-content/uploads/2016/09/RegExCheatsheet.pdf
Stringr Cheatsheet: https://evoldyn.gitlab.io/evomics-2018/ref-sheets/R_strings.pdf
Strings: https://jrnold.github.io/r4ds-exercise-solutions/strings.html

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(stringr)

# Read in dataset
raw <- read.csv("./input/input5.txt", header = F)
```

Functions

Detect Patterns.

- **str_detect(string, pattern = "")**, equivalent to `grepl(pattern, x)`.

Locate Patterns.

- **str_count(string, pattern = "")**, or return the length of all matches generated from `gregexpr()` or `stringr::str_locate_all()`

Patterns

- `\\1` pattern is called a backreference. It matches whatever the first group matched.
- **{n}** indicates exactly n matches. We use `\\1` instead of `{2}` to indicate **twice in a row**.
- Can use `.` to represent any character, or use "[a-z]" or [[:alpha:]].

- *: matches at least 0 times

Some examples

```
grepl("([a-z]){2}", "abccd")
```

```
## [1] TRUE
```

```
grepl("([a-z])\\1", "abccd")
```

```
## [1] TRUE
```

```
grepl("([a-z]){2}", "abcdc")
```

```
## [1] TRUE
```

```
grepl("([a-z])\\1", "abcdc")
```

```
## [1] FALSE
```

`(.).\\1.\\1`: A character followed by any character, the original character, any other character, the original character again. E.g. "abaca"
`(..)\\1`: Any two characters repeated. E.g. "a1a1", "chch"
`(.).*\\1`: A character repeated, with zero or more characters between them. E.g. "aa", "aba", "abca"

Part 1

Use 3 flags to indicate whether the given strings meet the corresponding criterion:
* Flag 1: contains at least three vowels.
* Flag 2: contains at least one letter that appears twice in a row.
* Flag 3: does not contain the strings `ab`, `cd`, `pq`, or `xy`.

```
data1 <- raw %>%
  # Flag 1: contains at least three vowels
  mutate(flag1 = (str_count(V1, "[aeiou]") >= 3)) %>%
  # Flag 2: contains at least one letter that appears twice IN A ROW
  mutate(flag2 = str_detect(V1, "(.)\\1")) %>%
  # Flag 3: does NOT contain the strings ab, cd, pq, or xy
  mutate(flag3 = !str_detect(V1, "ab|cd|pq|xy")) %>%
  filter(flag1*flag2*flag3 == TRUE)

nrow(data1)
```

```
## [1] 255
```

```
data2 <- raw %>%
  # Flag 1: contains at least three vowels
  rowwise() %>%
  # gregexpr: find starting position and length of all matches
  mutate(flag1 = (length(gregexpr('[aeiou]', V1)[[1]]) >= 3)) %>%
  # Flag 2: contains at least one letter that appears twice IN A ROW
  mutate(flag2 = grepl("([a-z])\\1", V1)) %>%
  # Flag 3: does NOT contain the strings ab, cd, pq, or xy
  mutate(flag3 = !grepl("ab|cd|pq|xy", V1)) %>%
  filter(flag1*flag2*flag3 == TRUE)

nrow(data2)
```

```
## [1] 255
```

Part 2

Use 2 flags to indicate whether the given strings meet the corresponding criterion:
* Flag 1: contains a pair of any two letters that appears at least twice without overlapping.
* Flag 2: contains at least one letter which repeats with exactly one letter between them

```
data3 <- raw %>%
  # Flag 1: contains a pair of any two letters that appears at least twice without overlapping
  mutate(flag1 = str_detect(V1, "(..).*\\1")) %>%
  # Flag 2: contains at least one letter which repeats with exactly one letter between them
  mutate(flag2 = str_detect(V1, "(.).\\1")) %>%
  filter(flag1 == TRUE & flag2 == TRUE)

nrow(data3)
```

```
## [1] 55
```

When using `grepl`, there is an issue when the repeated pair of characters are not the first two characters. But we do not need to worry about this when using `str_detect`.

```
grepl("(..).*\\1", "qwabxyzab") # 'ab' repeated but cannot be detected
```

```
## [1] FALSE
```

```
str_detect("qwabxyzab", "(..).*\\1") # 'ab' repeated and detected
```

```
## [1] TRUE
```

```
grepl(".*(..).*\\1", "qwabxyzab") # 'ab' detected
```

```
## [1] TRUE
```

```
data4 <- raw %>%
  # Flag 1: contains a pair of any two letters that appears at least twice without overlapping
  mutate(flag1 = grepl(".*(..).*\\1", V1)) %>%
  # Flag 2: contains at least one letter which repeats with exactly one letter between them
  mutate(flag2 = grepl("(.).\\1", V1)) %>%
  filter(flag1 == TRUE & flag2 == TRUE)

nrow(data4)
```

```
## [1] 55
```