Data607-Week03-Assignment-Character Manipulation And Date Processing

Vinayak Kamath

2/12/2020

1. Using the 173 majors listed in fivethirtyeight.com's College Majors dataset [https://fivethirtyeight.com/features/the-economic-guide-to-picking-a-college-major/], provide code that identifies the majors that contain either "DATA" or "STATISTICS":

```
theUrl <- "https://raw.githubusercontent.com/fivethirtyeight/data/master/college-majors/majors-list.csv
majors.df <- read.csv(file = theUrl, header = T , sep = ',' )
str(majors.df)
```

```
## 'data.frame':    174 obs. of  3 variables:
##  $ FOD1P         : Factor w/ 174 levels "1100","1101",..: 1 2 3 4 5 6 7 8 10 11 ...
##  $ Major         : Factor w/ 174 levels "ACCOUNTING","ACTUARIAL SCIENCE",..: 70 6 5 7 67 151 164 117
##  $ Major_Category: Factor w/ 16 levels "Agriculture & Natural Resources",..: 1 1 1 1 1 1 1 1 1 ...
```

```
major.pattern <- c("DATA" ,  "STATISTICS")

majors.df[grep("DATA|STATISTICS", majors.df$Major, value = F, ignore.case = T), ]
```

```
##    FOD1P                                          Major          Major_Category
## 44  6212 MANAGEMENT INFORMATION SYSTEMS AND STATISTICS                Business
## 52  2101      COMPUTER PROGRAMMING AND DATA PROCESSING Computers & Mathematics
## 59  3702               STATISTICS AND DECISION SCIENCE Computers & Mathematics
```

---

2. Write code that transforms the data below:

[1] “bell pepper” “bilberry” “blackberry” “blood orange”

[5] “blueberry” “cantaloupe” “chili pepper” “cloudberry”

[9] “elderberry” “lime” “lychee” “mulberry”

[13] “olive” “salal berry”

Into a format like this:

c(“bell pepper”, “bilberry”, “blackberry”, “blood orange”, “blueberry”, “cantaloupe”, “chili pepper”, “cloudberry”, “elderberry”, “lime”, “lychee”, “mulberry”, “olive”, “salal berry”)

```
fruits.data <- c("bell pepper", "bilberry", "blackberry", "blood orange", "blueberry", "cantaloupe", "cl
fruits.data
```

```
##  [1] "bell pepper"  "bilberry"     "blackberry"   "blood orange" "blueberry"
##  [6] "cantaloupe"   "chili pepper" "cloudberry"   "elderberry"   "lime"
## [11] "lychee"       "mulberry"     "olive"        "salal berry"
```

```
#toString(fruits.data)
#final.string <- toString(gsub(pattern = ", ",replacement ='",  "',toString(fruits.data)))
#final.string
dput(as.character(fruits.data))
```

```
## c("bell pepper", "bilberry", "blackberry", "blood orange", "blueberry",
## "cantaloupe", "chili pepper", "cloudberry", "elderberry", "lime",
## "lychee", "mulberry", "olive", "salal berry")
```

The two exercises below are taken from R for Data Science, 14.3.5.1 in the on-line version:

3. Describe, in words, what these expressions will match:

```
test.str <-  c('aaa','aab','aba','baa','bab','bba','bbb','ccc','ccb','cbc','cbb','cab','caa','abc','abb
```

(.)\1\1
Answer: Matches occurrence of a character repeating three times in the input string; Sample below:

```
pattern <- '(.)\\1\\1'
test.str %>%
  str_subset(pattern)
```

```
##  [1] "aaa"  "bbb"  "ccc"  "aaaa" "aaab" "baaa" "bbba" "bbbb" "aaac" "cccc"
```

"(.)(.)\2\1"
Answer: Matches a palindrome of 4 characters anywhere in the input string; Sample below:

```
pattern <- "(.)(.)\\2\\1"
test.str %>%
  str_subset(pattern)
```

```
##  [1] "aaaa"     "baab"     "bbbb"     "cccc"     "cbbc"     "redder"
##  [7] "hannah"   "abbabaab" "abbaabba" "abba"
```

(..)\1
Answer: Matches occurrence of two characters followed by the same two characters in the input string; Sample below.

```
pattern <- "(..)\\1"
test.str %>%
  str_subset(pattern)
```

```
## [1] "aaaa"     "baba"     "bbbb"     "cccc"     "abbabaab" "abab"
```

"(.).\1.\1"
Answer: Matches occurrence of a 5 character string where the first, third and 5th character are all same, and the second and fourth character can be any other character (same or different) in the input string; Sample below.

```
pattern <- "(.).\\1.\\1"
test.str %>%
  str_subset(pattern)
```

```
## [1] "abaCat" "abaca"
```

"(.)(.)(.).*\3\2\1"
Answer: Matches occurrence of any 6 or more character string where the first and last character are same, the second and second last character are same, the third and third last character are same and between them zero or more occurrence of any characters; sample as below:

```
pattern <- "(.)(.)(.).*\\3\\2\\1"
test.str %>%
  str_subset(pattern)
```

```
## [1] "redder"    "hannah"    "tatfortat" "abbaabba"
```

---

4. Construct regular expressions to match words that:

```
test.str <- c("TIME","CHURCH","ELEVEN","POP","SEVENTEEN","BANANA" ,"EAGLE")
```

Start and end with the same character.
Answer: ^(.).*\1$; sample as below:

```
pattern <- "^(.).*\\1$"
test.str %>%
  str_subset(pattern)
```

```
## [1] "POP"   "EAGLE"
```

Contain a repeated pair of letters (e.g. "church" contains "ch" repeated twice.)
Answer: (..).*\1 ; sample as below:

```
pattern <- "(..).*\\1"
test.str %>%
  str_subset(pattern)
```

```
## [1] "CHURCH"    "SEVENTEEN" "BANANA"
```

Contain one letter repeated in at least three places (e.g. "eleven" contains three "e"s.)
Answer: (.).*\1.*\1 ; sample as below:

```
pattern <- "(.).*\\1.*\\1"
test.str %>%
  str_subset(pattern)
```

```
## [1] "ELEVEN"    "SEVENTEEN" "BANANA"
```
