R Character Manipulation and Date Processing

Subhalaxmi Rout

2/16/2020

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"

Answer

```
data <- read.csv("https://raw.githubusercontent.com/fivethirtyeight/data/master/college-majors/majors-li
head(data$Major)
```

```
## [1] GENERAL AGRICULTURE                    AGRICULTURE PRODUCTION AND MANAGEMENT
## [3] AGRICULTURAL ECONOMICS                 ANIMAL SCIENCES
## [5] FOOD SCIENCE                           PLANT SCIENCE AND AGRONOMY
## 174 Levels: ACCOUNTING ACTUARIAL SCIENCE ... ZOOLOGY
```

```
grep("DATA",data$Major, value = T)
```

```
## [1] "COMPUTER PROGRAMMING AND DATA PROCESSING"
```

```
grep("STATISTICS",data$Major, value = T)
```

```
## [1] "MANAGEMENT INFORMATION SYSTEMS AND STATISTICS"
## [2] "STATISTICS AND DECISION SCIENCE"
```

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")

Answer

```
library("stringr")
food <- '"bell pepper"  "bilberry"     "blackberry"   "blood orange"
 "blueberry"    "cantaloupe"   "chili pepper" "cloudberry"
 "elderberry"   "lime"         "lychee"       "mulberry"
 "olive"        "salal berry"'
food <- str_extract_all(food, '[a-z]+\\s[a-z]+|[a-z]+')
food
```

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

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

- (.)\1\1
- "(.)(.)\2\1"
- (..)\1
- "(.).\1.\1"
- "(.)(.)(.).*\3\2\1"

Answer

```
(.)\1\1: same charecter 3 times.
example: "sss"

"(.)(.)\\2\\1": 2 charecters repeat in reverse way.
Example: "bccb"

(..)\1: any 2 charecters repeated.
Example: "abab"

"(.).\\1.\\1": 1st charecter followed by another charecter,
             again 1st charecter followed by other charecter,repeated 3 times.
Example: "abacad"

"(.)(.)(.).*\\3\\2\\1": 3 charecter followed by zero or more charecter
                      then same 3 charecter but in reverse way.
Example: "abc12cba"
```

#4 Construct regular expressions to match words that:

- Start and end with the same character.
- Contain a repeated pair of letters (e.g. "church" contains "ch" repeated twice.)
- Contain one letter repeated in at least three places (e.g. "eleven" contains three "e"s.)

Answer

```r
# Start and end with the same character.
words = c("ruler", "salsa", "environmental")
str1 <- str_subset(words, "^(.)((.*\\1$)|\\1?$)")
str1
```

```
## [1] "ruler"
```

```r
# Contain a repeated pair of letters (e.g. "church" contains "ch" repeated twice.)
str2 <- str_subset(words, "([a-z]{2})[a-z].*\\1")
str2
```

```
## [1] "salsa"         "environmental"
```

```r
# Contain one letter repeated in at least three places (e.g. "eleven" contains three "e"s.)
str3 <- str_subset(words, "([a-z]).*\\1.*\\1")
str3
```

```
## [1] "environmental"
```