Data 607 - Week 3 R Character Manipulation and Date Processing

Shamecca Marshall

2023-08-31

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"

Load data from GitHub

```
majors = read.csv(file="https://raw.githubusercontent.com/fivethirtyeight/data/master/college-majors/ma
str(majors)
```

```
## 'data.frame':    174 obs. of  3 variables:
##  $ FOD1P         : chr  "1100" "1101" "1102" "1103" ...
##  $ Major         : chr  "GENERAL AGRICULTURE" "AGRICULTURE PRODUCTION AND MANAGEMENT" "AGRICULTURAL E
##  $ Major_Category: chr  "Agriculture & Natural Resources" "Agriculture & Natural Resources" "Agricult
```

Provide code that identifies the majors that contain either "DATA" or "STATISTICS"

```
grep(pattern = 'data|statistics',majors$Major, value = TRUE, ignore.case = TRUE)
```

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_raw = '[1] "bell pepper"  "bilberry"     "blackberry"   "blood orange"

[5] "blueberry"    "cantaloupe"   "chili pepper" "cloudberry"

[9] "elderberry"   "lime"         "lychee"       "mulberry"

[13] "olive"        "salal berry"'


fruits_clean = c(scan(text=fruits_raw, what="character", quiet=TRUE))
fruits_clean = Filter(function(x) !any(grepl("\\[", x)), fruits_clean)
```

```
fruits_clean
```

```
##  [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 - Matches string with the same character repeated three times ex. 1215-2999 it will match 999

(..)\1 - Matches string format that has two characters repeated twice in the same order ex. 211414 it will match 1414

“(.).\1.\1” - Matches a character with the first character followed by the first character, followed by any other character, followed by the first character e.g. in string ex. 212329549 will match 21232

“(.)(.)(.).*\3\2\1” - Matches three characters that are following by zero or more characters and then have the pattern in reverse order. ex 214feb1994pink1215 it will match feb1994pink

4. Construct regular expressions to match words that:

Start and end with the same character.

```
s<- c("tweet", "tomorrow", "Mississippi", "appropriate", "educate", "dazed", "eleven", "error", "nanny"
```

```
str_view(s, "^(.)((.*\\1$)|\\1$)")
```

```
## [1] | <tweet>
## [5] | <educate>
## [6] | <dazed>
```

Contain a repeated pair of letters (e.g. “church” contains “ch” repeated twice.)

```
str_view(s, "([A-Za-z][A-Za-z]).*\\1")
```

```
##  [3] | M<issis>sippi
##  [4] | ap<propr>iate
## [10] | <church>
```

Contain one letter repeated in at least three places (e.g. “eleven” contains three “e”s.)

```
str_view(s, "([A-Za-z]).*\\1.*\\1")
```

```
## [2] | t<omorro>w
## [3] | M<ississippi>
## [4] | a<pprop>riate
## [7] | <eleve>n
## [8] | e<rror>
## [9] | <nann>y
```