Assignment 3

2023-09-12

#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"

```
url <- "https://raw.githubusercontent.com/fivethirtyeight/data/master/college-majors/majors-list.csv"

majors <- read.csv(file = url, header = T)

str_subset(majors$Major, "DATA|STATISTICS", negate = F)
```

```
## [1] "MANAGEMENT INFORMATION SYSTEMS AND STATISTICS"
## [2] "COMPUTER PROGRAMMING AND DATA PROCESSING"
## [3] "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"

Note: I realized the purpose of this question after I finished questions 3 & 4, so that helped with making the str_extract code and regex

```
berry_str <- '"bell pepper"  "bilberry"     "blackberry"   "blood orange" "blueberry"    "cantaloupe"

berry_str
```

```
## [1] "\"bell pepper\"  \"bilberry\"     \"blackberry\"   \"blood orange\" \"blueberry\"    \"cantaloup
```

```
berries <- unlist(str_extract_all(berry_str, "\\w+\\s*\\w+"))

berries
```

```
##  [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"

To do this, I tested a few dataframes in the rstrings package, and then I made my own, consisting of multiple letters and combinations of those letters

```
strings <- c("aaa", "bbb", "ccc", "ddd", "abc", "abcd", "abcde",
             "aaaabbbbccccddddeeee", "alpha", "beta")

long_words <- c("pneumonoultramicroscopicsilicovolcanoconiosis",
                "Pseudopseudohypoparathyroidism", "Floccinaucinihilipilification", "Antidisestablishmen
                "Supercalifragilisticexpialidocious",
                "Incomprehensibilities", "Euouae",
                "Unimaginatively", "Honorificabilitudinitatibus",
                "Tsktsk", "Sesquipedalianism")
```

Testing the regex expressions, I determined that: (.)\1\1 returns strings consisting of three letters repeating in a row

(.)(.)\2\1 returns strings containing two characters that are followed by the same characters in reverse

(..)\1 Returns strings containing two letters that are repeated in the same order

I had to experiment with (.).\1.\1 a bit, but it seems to return strings containing a letter followed by a different letter, followed by the same letter as the first letter

(.)(.)(.).*\3\2\1 returns strings containing three letters followed by any other letters, and ending with the first three letters in (I believe) reverse order

```
str_view(strings, "(.)\\1\\1")
```

```
## [1] | <aaa>
## [2] | <bbb>
## [3] | <ccc>
## [4] | <ddd>
## [8] | <aaa>a<bbb>b<ccc>c<ddd>d<eee>e
```

```
str_view(fruit, "(.)(.)\\2\\1", match = T)
```

```
##  [5] | bell p<eppe>r
## [17] | chili p<eppe>r
```

```
str_view(words, "(..)\\1", match = T)
```

```
## [696] | r<emem>ber
```

```
str_view(long_words,"(.).\\1.\\1", match = T)
```

```
## [3] | Floccinauc<inihi>l<ipili>fication
## [6] | Incomprehens<ibili>ties
```

```
str_view(long_words, "(.)(.)(.).*\\3\\2\\1", match = T)
```

```
## [3] | Floccinaucinih<ilipili>fication
```

#3 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.)

Note: I used the following website for help with regards to understanding regex syntax: https://www3.ntu.edu.sg/home/ehchua/programming/howto/Regexe.html

```
##Start and end with the same character
str_view(strings, "^(.).*\\1$")
```

```
## [1] | <aaa>
## [2] | <bbb>
## [3] | <ccc>
## [4] | <ddd>
## [9] | <alpha>
```

```
##Contain a repeated pair of letters (e.g. "church" contains "ch" repeated twice.)

str_view(berries, "(\\w+\\w).*\\1")
```

```
##  [1] | bell <peppe>r
##  [7] | chili <peppe>r
##  [9] | eld<erber>ry
## [14] | s<alal> berry
```

```
##Contain one letter repeated in at least three places (e.g. "eleven" contains three "e"s.)

str_view(berries, "(\\w).*\\1.*\\1")
```

```
## [1] | b<ell peppe>r
## [4] | bl<ood o>range
## [7] | chili <pepp>er
## [9] | <elderbe>rry
```