Data607_HW3

Sean Connin

02/19/2021

1. College Majors Data Set

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

Approach:

1. Import dataset
2. Use str_detect() to identify strings containing “STATISTICS” or “DATA”
3. Subset rows with filter() retrieve these strings

```
majors <- read_csv("https://raw.githubusercontent.com/fivethirtyeight/data/master/college-major
s/majors-list.csv")
```

```
##
## -- Column specification --------------------------------------------------------
## cols(
##   FOD1P = col_character(),
##   Major = col_character(),
##   Major_Category = col_character()
## )
```

```
majors%<>%filter(str_detect(Major, "STATISTICS|DATA"))

majors%>%kbl()%>%kable_material(c("striped"))
```

FOD1P	Major	Major_Category
6212	MANAGEMENT INFORMATION SYSTEMS AND STATISTICS	Business
2101	COMPUTER PROGRAMMING AND DATA PROCESSING	Computers & Mathematics
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”)

Approach: adapted from https://bit.ly/2N4Q6zW (https://bit.ly/2N4Q6zW)

1. shQuote() - dbl quote a string to be passed into os shell, cmd is Windows default
2. paste() - create a character string and separate results with a comma via the collapse argument
3. cat() - outputs as a character vector

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


d<-(cat(paste(shQuote(d, type="cmd"), collapse=", ")))
```

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

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

(.)\1\1 - *missing quotes, no match, treated as a literal*

“(.)(.)\2\1” - *matches two single character capture groups followed by their reverse order.*

For example: if x= “bannnaannana” the match is “naan”

(..)\1 - *missing quotes, no match, treated as a literal*

“(.).\1.\1” - *matches a single character capture group followed by another character, this pattern repeats, and ends with the first capture group.*

For example: if x= “bannnaannantnan” the match is “nantn”

"(.)(.)(.).\3\2\1" - *matches a sequence of three single character capture groups followed by a character that repeats one or more time followed by a sequence of the three capture groups in reverse order.*

For example: if x= “banntaannantnan” the match is“anntaanna”

4. Construct regular expressions to match words that:

Start and end with the same character.

^(.).*\1$

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

(\w{2}).*?(\1) Adapted from: https://bit.ly/2Ng5lps (https://bit.ly/2Ng5lps)

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

```
(.).?(\1).?(\1).*?$
```