

DATA607 HW3

Vanita Thompson

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/) [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

```
majors <- read.csv(url('https://raw.githubusercontent.com/fivethirtyeight/data/master/college-majors/majors.csv'), as.is=T, stringsAsFactors=F)
```

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

Majors containing data or statistics:

```
majors$Major[grepl("DATA", majors$Major)]
```

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

```
majors$Major[grepl("STATISTICS", majors$Major)]
```

```
## [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")
```

Enter list values

```
fruit_list <- '[1] "bell pepper" "bilberry" "blackberry" "blood orange"
[5] "blueberry" "cantaloupe" "chili pepper" "cloudberry"
[9] "elderberry" "lime" "lychee" "mulberry"
[13] "olive" "salal berry"'
```

```
##Load library
```

```
library(stringr)
```

unlist data

```
foods <- str_extract_all(fruit_list, '[a-z]+\s[a-z]+\s[a-z]+')
unlist(foods)
```

```
## [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`

The string `'\1\1'` and the character before it, as long as it isn't a new line

`"(.)\2\1"`

This could match anything like a 4 letter palindrome. one character, another character, the same as the 2nd character, the same as the first character, while surrounded by quotes.

`(..)\1`

This matches any two characters followed by the string `'\1'`

`"(.)\1\1"`

This matches any character, followed by any other character, then the first character again, then any other character, then the first character again, while surrounded by quotes.

`"(.)\1\1\1"`

This would match any three characters and any or no characters in between the first 3 characters reversed, while surrounded by quotes.

#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.)

`([a-z]{2})\1`

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

`[a-z]([a-z])[a-z]\1[a-z]`