# Week 3 | DATA 607

Jhalak Das

9/12/2021

Please deliver links to an R Markdown file (in GitHub and rpubs.com) with solutions to the problems below. You may work in a small group, but please submit separately with names of all group participants in your submission.

#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"

```
majorsWithStatOrData <-
read.csv("https://raw.githubusercontent.com/fivethirtyeight/data/master/colle
ge-majors/majors-list.csv",header = TRUE, sep = ",")

grep(pattern = 'STATISTICS|DATA', majorsWithStatOrData$Major, value = TRUE,
ignore.case = TRUE)
```

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

There is one major wih 'data' in the major name. There are two majors with 'statistics' in the major name.

#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_orig <- '[1] "bell pepper"  "bilberry"     "blackberry"   "blood
orange"

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

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

```
[13] "olive"        "salal berry"'
fruits_orig
```

```
## [1] "[1] \"bell pepper\"  \"bilberry\"     \"blackberry\"   \"blood
orange\"\n\n[5] \"blueberry\"    \"cantaloupe\"   \"chili pepper\"
\"cloudberry\"  \n\n[9] \"elderberry\"  \"lime\"         \"lychee\"
\"mulberry\"    \n\n[13] \"olive\"       \"salal berry\""
```

The two exercises below are taken from R for Data Science, 14.3.5.1 in the on-line version:

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

```
1. (.)\1\1
This will match characters that are three times in a row.

2. "(.)(.)\\2\\1"
This will match any four characters that read the same forward and backward.

3. (..)\1
This will match two characters repeated.

4. "(.).\\1.\\1"
This will match five characters where the first, third and fifth are the same
and the second and fourth can
be anything.

5. "(.)(.)(.).*\\3\\2\\1"
This will match six or more characters where the first three characters are
the same as the last three
in reverse order.
```

#4 Construct regular expressions to match words that:

i. Start and end with the same character.

(.)[a-z]*\1

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

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

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

a-z[a-z]\1[a-z]\1[a-z]