Week3 Assignment

Coffy Andrews-Guo

September 12, 2021

R Character Manipulation and Date Processing

Question 1 The 173 majors listed in fivethirtyeight.com's College Majors dataset [https://fivethirtyeight.com/features/the-economic-guide-to-picking-a-college-major/], was pulled from [https://github.com/fivethirtyeight/data/blob/master/college-majors/majors-list.csv]

```
library(readr)

urlfile = "https://raw.githubusercontent.com/fivethirtyeight/data/master/college-majors/majors-list.csv

majors <- read_csv(url(urlfile))
```

```
## Rows: 174 Columns: 3

## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (3): FOD1P, Major, Major_Category

##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
spec(majors)
```

```
## cols(
##   FOD1P = col_character(),
##   Major = col_character(),
##   Major_Category = col_character()
## )
```

Code identification of majors that contain either "DATA" or "STATISTICS"

```
library(stringr)

majors1 <- majors %>%
  filter(str_detect(majors$Major, "DATA") | str_detect(majors$Major, "STATISTICS"))
majors1
```

```
## # A tibble: 3 x 3
##   FOD1P Major                                          Major_Category
##   <chr> <chr>                                          <chr>
## 1 6212  MANAGEMENT INFORMATION SYSTEMS AND STATISTICS Business
## 2 2101  COMPUTER PROGRAMMING AND DATA PROCESSING      Computers & Mathematics
## 3 3702  STATISTICS AND DECISION SCIENCE               Computers & Mathematics
```

Question 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”)

```
x <- c("bell pepper", "bilberry", "blackberry", "blood orange", "blueberry", "cantaloupe", "chili peppe
print(x)
```

```
##  [1] "bell pepper"  "bilberry"     "blackberry"   "blood orange" "blueberry"
##  [6] "cantaloupe"   "chili pepper" "cloudberry"   "elderberry"   "lime"
## [11] "lychee"       "mulberry"     "olive"        "salal berry"
```

The two exercises below are taken from R for Data Science, 14.3.5.1

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

(.)\1\1 Response: This expression will match the same character appearing three times in a row.

“(.)(.)\2\1” Response: This expression will match a pair of characters with the same pair of characters in reversed order.

(..)\1 Response: This expression wil match any two characters that are repeated.

“(.).\1.\1” Response: This expression will match any character followed by any character, the original character, any character, then the original character.

"(.)(.)(.).*\3\2\1" Response: This expression will match any three characters, followed by zero, then followed by the same three characters in a reverse order.

Question 4

Construct regular expressions to match words that:

Start and end with the same character.

Response: "^(.)((.*\1)|
1?)"

```
library(stringr)
str_subset(words, "^(.)((.*\\1$)|\\1?$)")
```

```
##  [1] "a"          "america"    "area"       "dad"        "dead"
##  [6] "depend"     "educate"    "else"       "encourage"  "engine"
## [11] "europe"     "evidence"   "example"    "excuse"     "exercise"
## [16] "expense"    "experience" "eye"        "health"     "high"
## [21] "knock"      "level"      "local"      "nation"     "non"
## [26] "rather"     "refer"      "remember"   "serious"    "stairs"
## [31] "test"       "tonight"    "transport"  "treat"      "trust"
## [36] "window"     "yesterday"
```

Contain a repeated pair of letters (e.g. “church” contains “ch” repeated twice.) Response: "([A-Za-z][A-Za-z]).*\1"

```
str_subset(words, "([A-Za-z][A-Za-z]).*\\1")
```

```
##  [1] "appropriate" "church"      "condition"   "decide"      "environment"
##  [6] "london"      "paragraph"   "particular"  "photograph"  "prepare"
## [11] "pressure"    "remember"    "represent"   "require"     "sense"
## [16] "therefore"   "understand"  "whether"
```

Contain one letter repeated in at least three places (e.g. “eleven” contains three “e”s.) Response: “([a-z]).*\1.*\1”

```
str_subset(words, "([a-z]).*\\1.*\\1")
```

```
##  [1] "appropriate" "available"   "believe"     "between"     "business"
##  [6] "degree"      "difference"  "discuss"     "eleven"      "environment"
## [11] "evidence"    "exercise"    "expense"     "experience"  "individual"
## [16] "paragraph"   "receive"     "remember"    "represent"   "telephone"
## [21] "therefore"   "tomorrow"
```

Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.

RPubs

GitHub