data 607 lab3

Yina Qiao

2023-02-13

```
library(tidyverse)
```

```
## ── Attaching packages ─────────────────────────────────────── tidyverse 1.3.2 ──
## ✔ ggplot2 3.4.1     ✔ purrr   0.3.4
## ✔ tibble  3.1.7     ✔ dplyr   1.0.9
## ✔ tidyr   1.2.0     ✔ stringr 1.4.0
## ✔ readr   2.1.2     ✔ forcats 0.5.1
## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
```

1

```
df.college.majors = read.csv( url("https://raw.githubusercontent.com/fivethirtyeight/data/master/college-majors/majors-list.csv"))

vec.majors = df.college.majors$Major[grep("DATA|STATISTICS", df.college.majors$Major)]
print(vec.majors)
```

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

2

```
vec.text = c('[1] "bell pepper"  "bilberry"     "blackberry"   "blood orange"
[5] "blueberry"    "cantaloupe"   "chili pepper" "cloudberry"
[9] "elderberry"   "lime"         "lychee"       "mulberry"
[13] "olive"        "salal berry"')

vec.text.char = gsub("(\\n\\[\\d+\\])|(^\\[\\d+\\])", "", vec.text)
vec.text.char = strsplit(vec.text.char, '\\"')
vec.text.char = unlist(vec.text.char)
vec.text.char = vec.text.char[grep("[a-z]", vec.text.char)]
print(vec.text.char)
```

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

3

1> (.)\1\1. This will detect any three consecutive charactes in string format that has the same character.

```
str_detect("AAA", "(.)\1\1")
```

```
## [1] FALSE
```

```
str_detect("A\1\1", "(.)\1\1")
```

```
## [1] TRUE
```

2> “(.)(.)\2\1”. This matches any 4 consecutive characters in a string where the last 2 are the same as the first 2 characters in reverse order.

```
str_detect("eppe","(.)(.)\\2\\1")
```

```
## [1] TRUE
```

3> (..)\1 Regular expression not represented in string format that has two characters repeated twice in the same order

```
str_detect('nana', '(..)\1')
```

```
## [1] FALSE
```

4> “(.).\1.\1” This will match any five consecutive characters where character 1, 3, and 5 are the same

```
str_detect('abana',"(.).\\1.\\1")
```

```
## [1] TRUE
```

5> “(.)(.)(.).*\3\2\1” match any string that contains at least six characters with the last 3 characters as the same as the first 3 characters in reverse order

```
str_detect("123newyorkcity;321","(.)(.)(.).*\\3\\2\\1")
```

```
## [1] TRUE
```

4

Construct regular expressions to match words that: 1>Start and end with the same character.

```
#"^(.).*\\1$"
```

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

```
# "(..).*\\1"
```

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

```
# "(.).*\\1.*\\1"
```