STA 445 S24 Assignment 5

Mason Nabbefeld

2024-03-21

Problem 1

For the following regular expression, explain in words what it matches on. Then add test strings to demonstrate that it in fact does match on the pattern you claim it does. Do at least 4 tests. Make sure that your test set of strings has several examples that match as well as several that do not. Make sure to remove the `eval=FALSE` from the R-chunk options.

a. This regular expression matches: Any string with a lowercase 'a'.

```
strings <- c("cat", "cow", "fantastic", "MAN")
data.frame( string = strings ) %>%
mutate( result = str_detect(string, 'a') )
```

```
##      string result
## 1       cat   TRUE
## 2       cow  FALSE
## 3 fantastic   TRUE
## 4       MAN  FALSE
```

b. This regular expression matches: Any string with a lowercase 'ab' in that order.

```
strings <- c("abstract", "ab", "Ablaze", "ABLE")
data.frame( string = strings ) %>%
mutate( result = str_detect(string, 'ab') )
```

```
##     string result
## 1 abstract   TRUE
## 2       ab   TRUE
## 3   Ablaze  FALSE
## 4     ABLE  FALSE
```

c. This regular expression matches: Any string with a lowercase 'a' or a lowercase 'b' in any order.

```
strings <- c("after", "fig", "BAnker", "abduct")
data.frame( string = strings ) %>%
mutate( result = str_detect(string, '[ab]') )
```

```
##   string result
## 1  after   TRUE
## 2    fig  FALSE
## 3 BAnker  FALSE
## 4 abduct   TRUE
```

d. This regular expression matches: Any string that starts with a lowercase ‘a’ or ‘b’.

```
strings <- c("slab", "baste", "Bust", "avocado")
data.frame( string = strings ) %>%
mutate( result = str_detect(string, '^[ab]') )
```

```
##    string result
## 1    slab  FALSE
## 2   baste   TRUE
## 3    Bust  FALSE
## 4 avocado   TRUE
```

e. This regular expression matches: A string that starts with a number that can be repeated, then a single space, then a lower or uppercase ‘a’.

```
strings <- c("2 a", "22 A", " a 2", "2 sdfa","5a ", "55  aweroitu")
data.frame( string = strings ) %>%
mutate( result = str_detect(string, '\\d+\\s[aA]') )
```

```
##         string result
## 1          2 a   TRUE
## 2         22 A   TRUE
## 3          a 2  FALSE
## 4       2 sdfa  FALSE
## 5          5a   FALSE
## 6 55  aweroitu  FALSE
```

f. This regular expression matches: A string that starts with a number that can be repeated, then any number of spaces, then a lower or uppercase ‘a’.

```
strings <- c("2  a", "a 2", "2 sa", "22 a2 a", "55  aweroitu")
data.frame( string = strings ) %>%
mutate( result = str_detect(string, '\\d+\\s*[aA]') )
```

```
##         string result
## 1         2  a   TRUE
## 2          a 2  FALSE
## 3         2 sa  FALSE
## 4      22 a2 a   TRUE
## 5 55  aweroitu   TRUE
```

g. This regular expression matches: Anything, you cant get a false.

```
strings <- c(".", "", "...", "flow", "ww", "11111*", "
             ")
data.frame( string = strings ) %>%
mutate( result = str_detect(string, '.*') )
```

```
##            string result
## 1               .   TRUE
## 2                   TRUE
## 3             ...   TRUE
## 4            flow   TRUE
## 5              ww   TRUE
## 6          11111*   TRUE
## 7 \n                TRUE
```

h. This regular expression matches: Any string that starts with 2 of the same alphanumeric characters followed immediately by ‘bar’.

```
strings <- c("11bar", "1bar", "111bar", "88barerhs", "ssbar")
data.frame( string = strings ) %>%
mutate( result = str_detect(string, '^\\w{2}bar') )
```

```
##      string result
## 1     11bar   TRUE
## 2      1bar  FALSE
## 3    111bar  FALSE
## 4 88barerhs   TRUE
## 5     ssbar   TRUE
```

i. This regular expression matches: Any string that contains ‘foo.bar’ or any string that starts with 2 of the same alphanumeric characters followed immediately by ‘bar’.

```
strings <- c("foo\\.bar", "ssbar", "sgfoo.barse", "foo .bar")
data.frame( string = strings ) %>%
mutate( result = str_detect(string, '(foo\\.bar)|(^\\w{2}bar)') )
```

```
##        string result
## 1   foo\\.bar  FALSE
## 2       ssbar   TRUE
## 3 sgfoo.barse   TRUE
## 4    foo .bar  FALSE
```

Problem 2

The following file names were used in a camera trap study. The S number represents the site, P is the plot within a site, C is the camera number within the plot, the first string of numbers is the YearMonthDay and the second string of numbers is the HourMinuteSecond.

```
file.names <- c( 'S123.P2.C10_20120621_213422.jpg',
                 'S10.P1.C1_20120622_050148.jpg',
                 'S187.P2.C2_20120702_023501.jpg')
```

Produce a data frame with columns corresponding to the `site`, `plot`, `camera`, `year`, `month`, `day`, `hour`, `minute`, and `second` for these three file names. So we want to produce code that will create the data frame:

```
     Site Plot Camera Year Month Day Hour Minute Second
     S123   P2    C10 2012    06  21   21     34     22
      S10   P1     C1 2012    06  22   05     01     48
      S187   P2     C2 2012    07  02   02     35     01
```

```
camTrap <- data.frame(file.names) %>% mutate(
  Site = str_extract(file.names, "S\\d+"),
  Plot = str_extract(file.names, "P\\d"),
  Camera = str_extract(file.names, "C\\d+"),
  Year = str_extract(file.names, "\\d{4}"),
  Month = str_sub(file.names, -15, -14 ),
  Day = str_sub(file.names, -13, -12 ),
  Hour = str_sub(file.names, -10, -9 ),
  Minute = str_sub(file.names, -8, -7 ),
  Second = str_extract(file.names, "\\d{2}(?=\\.jpg)")) %>% select(-file.names)
camTrap
```

```
##   Site Plot Camera Year Month Day Hour Minute Second
## 1 S123   P2    C10 2012    06  21   21     34     22
## 2  S10   P1     C1 2012    06  22   05     01     48
## 3 S187   P2     C2 2012    07  02   02     35     01
```

3. The full text from Lincoln’s Gettysburg Address is given below. Calculate the mean word length *Note: consider ‘battle-field’ as one word with 11 letters*).

```
Gettysburg <- 'Four score and seven years ago our fathers brought forth on this continent, a new nation

GburgSplit <- Gettysburg %>% str_replace_all('--', ' ') %>% str_replace_all(',','') %>%
  str_replace_all('\\.','') %>% str_replace('-',' ') %>% str_replace_all('  ', '') %>%
  str_split(" ")

mean(str_length(GburgSplit[[1]]))
```

```
## [1] 4.224265
```