STA 445 S24 Assignment 5

Paige Hawkinson

2024-03-26

```
library(tidyverse)
```

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: *if the word in strings obtains an a, regardless of placement*

```
strings <- c("pal", "loop", "ball", "one")
  data.frame( string = strings ) %>%
  mutate( result = str_detect(string, 'a') )
```

```
##   string result
## 1    pal   TRUE
## 2   loop  FALSE
## 3   ball   TRUE
## 4    one  FALSE
```

b. This regular expression matches: *if the word in strings obtains ab in that specific order, regardless of placement*

```
strings <- c("abs", "apple", "absolute", "ball", "cab")
  data.frame( string = strings ) %>%
  mutate( result = str_detect(string, 'ab') )
```

```
##     string result
## 1      abs   TRUE
## 2    apple  FALSE
## 3 absolute   TRUE
## 4     ball  FALSE
## 5      cab   TRUE
```

c. This regular expression matches: *If the word in the string contains only a or b*

```r
strings <- c("cat", "pal", "pen", "dent")
  data.frame( string = strings ) %>%
  mutate( result = str_detect(string, '[ab]') )
```

```
##   string result
## 1    cat   TRUE
## 2    pal   TRUE
## 3    pen  FALSE
## 4   dent  FALSE
```

d. This regular expression matches: *If the word in the string contains only a or b at the beginning of the string*

```r
strings <- c("abs", "ball", "cats", "stem")
  data.frame( string = strings ) %>%
  mutate( result = str_detect(string, '^[ab]') )
```

```
##   string result
## 1    abs   TRUE
## 2   ball   TRUE
## 3   cats  FALSE
## 4   stem  FALSE
```

e. This regular expression matches: *If the word in the string contains a digit that repeats one or more times, a white space, and has only a or A*

```r
strings <- c("111 aA", "Apple", "22 aAaA", "aaAAAHHHH")
  data.frame( string = strings ) %>%
  mutate( result = str_detect(string, '\\d+\\s[aA]') )
```

```
##      string result
## 1    111 aA   TRUE
## 2     Apple  FALSE
## 3   22 aAaA   TRUE
## 4 aaAAAHHHH  FALSE
```

f. This regular expression matches: *If the word in the string contains a digit that repeats one or more times, a white space, zero or more repetitions of the white space, and has only a or A*

```r
strings <- c("11a ", "aA", "22 A", "379 bcd")
  data.frame( string = strings ) %>%
  mutate( result = str_detect(string, '\\d+\\s*[aA]') )
```

```
##    string result
## 1    11a    TRUE
## 2      aA  FALSE
## 3    22 A   TRUE
## 4 379 bcd  FALSE
```

g. This regular expression matches: *Any character with zero or more repetitions*

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

```
##   string result
## 1  three   TRUE
## 2      c   TRUE
```

h. This regular expression matches: *Any alphanumeric character at the beginning of the string, followed by 2 repetitions and bar*

```
strings <- c("yebar", "poobar", "2pbar", "peebar")
  data.frame( string = strings ) %>%
  mutate( result = str_detect(string, '^\\w{2}bar') )
```

```
##   string result
## 1  yebar   TRUE
## 2 poobar  FALSE
## 3  2pbar   TRUE
## 4 peebar  FALSE
```

i. This regular expression matches: *foo is in the string followed by .bar or any alphanumeric character at the beginning of the string, followed by 2 repetitions and bar*

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

```
##              string result
## 1           foo.bar   TRUE
## 2           poopbar  FALSE
## 3 peepsforeasterbar  FALSE
## 4             yebar   TRUE
```

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:

```
three.files <- data.frame(
   file.names = c( 'S123.P2.C10_20120621_213422.jpg',
                   'S10.P1.C1_20120622_050148.jpg',
                 'S187.P2.C2_20120702_023501.jpg'))
separate(three.files, col = "file.names", into = c("site", "plot", "camera", "date", "time", "jpg"), se
  mutate(year = str_sub(date, start = 1, end = 4),
         month = str_sub(date, start = 5, end = 6),
         day = str_sub(date, start = 7, end = 8),
         hour = str_sub(time, start = 1, end = 2),
         minute = str_sub(time, start = 3, end = 4),
         second = str_sub(time, start = 5, end = 6)) %>%
  select("site", "plot", "camera", "year", "month", "day", "hour", "minute", "second")
```

```
##   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*). 4.224ish is the answer if it is done right

```
Gettysburg <- 'Four score and seven years ago our fathers brought forth on this
continent, a new nation, conceived in Liberty, and dedicated to the proposition
that all men are created equal. Now we are engaged in a great civil war, testing
whether that nation, or any nation so conceived and so dedicated, can long
endure. We are met on a great battle-field of that war. We have come to dedicate
a portion of that field, as a final resting place for those who here gave their
lives that that nation might live. It is altogether fitting and proper that we
should do this. But, in a larger sense, we can not dedicate -- we can not
consecrate -- we can not hallow -- this ground. The brave men, living and dead,
who struggled here, have consecrated it, far above our poor power to add or
detract. The world will little note, nor long remember what we say here, but it
can never forget what they did here. It is for us the living, rather, to be
dedicated here to the unfinished work which they who fought here have thus far
so nobly advanced. It is rather for us to be here dedicated to the great task
remaining before us -- that from these honored dead we take increased devotion
to that cause for which they gave the last full measure of devotion -- that we
here highly resolve that these dead shall not have died in vain -- that this
nation, under God, shall have a new birth of freedom -- and that government of
the people, by the people, for the people, shall not perish from the earth.'
```

```
Gettysburg.2 <- str_replace_all(Gettysburg, pattern = "\\.", replacement = " ")
Gettysburg.3 <- str_replace_all(Gettysburg.2, pattern = "\\--", replacement = " ")
Gettysburg.4 <- str_replace_all(Gettysburg.3, pattern = "\\-", replacement = "")
Gettysburg.5 <- str_replace_all(Gettysburg.4, pattern = "\\,", replacement = " ")
Gettysburg.6 <- str_split(Gettysburg.5, pattern = "\\s+" )
Gettysburg.7 <- str_length(Gettysburg.6[[1]])
mean(Gettysburg.7)
```

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