STA 445 S24 Assignment 5

Graceson Mule

03/26/2024

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: *Insert your answer here...*

It checks if the string contains the letter 'a'.

```
library(tidyverse)

        strings <- c("apple", "banana", "fog", "student", "data", "science")
        data.frame( string = strings ) %>%
          mutate( result = str_detect(string, 'a') )
```

```
##    string result
## 1   apple   TRUE
## 2  banana   TRUE
## 3     fog  FALSE
## 4 student  FALSE
## 5    data   TRUE
## 6 science  FALSE
```

b. This regular expression matches: *Insert your answer here...*

Checks if the string contains 'ab'.

```
        strings <- c("watermelon", "pineapple", "abra", "kadabra")
        data.frame( string = strings ) %>%
          mutate( result = str_detect(string, 'ab') )
```

```
##       string result
## 1 watermelon  FALSE
## 2  pineapple  FALSE
## 3       abra   TRUE
## 4    kadabra   TRUE
```

c. This regular expression matches: *Insert your answer here...*

Checks if string contains either 'a' or 'b' or both

```
        strings <- c("apple", "cranberry", "three", "blind", "mice", "ab")
        data.frame( string = strings ) %>%
          mutate( result = str_detect(string, '[ab]') )
```

```
##      string result
## 1     apple   TRUE
## 2 cranberry   TRUE
## 3     three  FALSE
## 4     blind   TRUE
## 5      mice  FALSE
## 6        ab   TRUE
```

d. This regular expression matches: *Insert your answer here...*

Checks if beginning of string contains either an 'a', or 'b'.

```
        strings <- c("apple", "three", "blind", "mice", "abra", "kadabra")
        data.frame( string = strings ) %>%
          mutate( result = str_detect(string, '^[ab]') )
```

```
##    string result
## 1   apple   TRUE
## 2   three  FALSE
## 3   blind   TRUE
## 4    mice  FALSE
## 5    abra   TRUE
## 6 kadabra  FALSE
```

e. This regular expression matches: *Insert your answer here...*

Checks if string contains any amount of digits followed by any white space and either 'a', 'A'.

```
        strings <- c("8965567 robot", "675658 apple", "3896\nAbra", " 12331 kadabra")
        data.frame( string = strings ) %>%
          mutate( result = str_detect(string, '\\d+\\s[aA]') )
```

```
##           string result
## 1  8965567 robot  FALSE
## 2   675658 apple   TRUE
## 3     3896\nAbra   TRUE
## 4  12331 kadabra  FALSE
```

f. This regular expression matches: *Insert your answer here...*

Checks if string contains any amount of digits followed by any amount of white space (including 0) and either 'a', or 'A'.

```
        strings <- c("8965567 robot", "675658apple", "3896\n\nAbra", " 12331   kadabra")
        data.frame( string = strings ) %>%
          mutate( result = str_detect(string, '\\d+\\s*[aA]') )
```

```
##             string result
## 1    8965567 robot  FALSE
## 2      675658apple   TRUE
## 3     3896\n\nAbra   TRUE
## 4  12331   kadabra  FALSE
```

g. This regular expression matches: *Insert your answer here...*

Check for any character repeated any amount of times (including 0).

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

```
##   string result
## 1 banana   TRUE
## 2  apple   TRUE
## 3  three   TRUE
## 4  blind   TRUE
## 5   mice   TRUE
## 6          TRUE
```

h. This regular expression matches: *Insert your answer here...*

Checks beginning of string for any alphanumeric character repeated twice followed by 'bar'.

```
        strings <- c("$$bar", "apple", "121bar", "zzbar", "11bar", "rAAbar")
        data.frame( string = strings ) %>%
          mutate( result = str_detect(string, '^\\w{2}bar') )
```

```
##   string result
## 1  $$bar  FALSE
## 2  apple  FALSE
## 3 121bar  FALSE
## 4  zzbar   TRUE
## 5  11bar   TRUE
## 6 rAAbar  FALSE
```

i. This regular expression matches: *Insert your answer here...*

Checks if string contains 'foo.bar' or if beginning of string starts with any alphanumeric number repeated twice followed by 'bar' and captures the group.

```
        strings <- c("2029foo.bar1242", "32foo..bar", "12barvnms", "1$bar")
        data.frame( string = strings ) %>%
          mutate( result = str_detect(string, '(foo\\.bar)|(^\\w{2}bar)') )
```

```
##             string result
## 1 2029foo.bar1242   TRUE
## 2      32foo..bar  FALSE
## 3       12barvnms   TRUE
## 4           1$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:

```
file.names.df <- data.frame(files = file.names)
file.names.df <- separate(file.names.df, col=files,
         into = c("Site", "Plot", "Camera", "Date", "Time"), sep = "_|\\.") %>%
  mutate(Year = str_sub(Date, start = 1, end = 4)) %>%
  mutate(Month = str_sub(Date, start = 5, end = 6)) %>%
  mutate(Day = str_sub(Date , start = 7, end = 8)) %>%
  mutate(Hour = str_sub(Time, start = 1, end = 2)) %>%
  mutate(Minute = str_sub(Time, start = 3, end = 4)) %>%
  mutate(Second = str_sub(Time, start = 5, end = 6))
file.names.df <- subset(file.names.df, select = -c(4,5))
file.names.df
```

```
##   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
```

```
     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
```

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, 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.'
```

```
# format string
Gettysburg <- str_replace_all(Gettysburg, pattern = '-', replacement = '') %>%
  str_replace_all(pattern = '\\W+', replacement = ' ') %>%
  str_trim('right')

str_split(Gettysburg, pattern = " ")[[1]] %>%
  str_length() %>%
  mean()
```

```
## [1] 4.239852
```