STA 445 S24 Assignment 5

Levi Mault

The Date…

Problem 1

```
library(tidyverse)
library(stringr)   # tidyverse string functions, not loaded with tidyverse
library(refinr)    # fuzzy string matching
```

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:
   The letter 'a' with TRUE.

```
        exaple1 <- c("This is a String! OMG")
        exaple2 <- c("This string is Awesome!")
        exaple3 <- c("This string is Boring!")
        exaple4 <- c("AAAAAAAAAAAAAAAaaaaaaaaaaaAAAAAAAAAAaaaaaaaaa")

        strings <- c(exaple1, exaple2, exaple3, exaple4)

        data.frame( string = strings ) %>%
          mutate( result = str_detect(string, 'a') )
```

```
##                                           string result
## 1                          This is a String! OMG   TRUE
## 2                        This string is Awesome!  FALSE
## 3                         This string is Boring!  FALSE
## 4 AAAAAAAAAAAAAAAaaaaaaaaaaaAAAAAAAAAAaaaaaaaaa   TRUE
```

b. This regular expression matches:
   The letters 'ab' with TRUE.

```
        exaple1 <- c("This is the absolute String! OMG")
        exaple2 <- c("This is the Absolute String! OMG")
        exaple3 <- c("tHIs sTrinG mAke$ aBs0lUt1y NULL s3n5E!")
        exaple4 <- c("abcdefghijklmnopqrstuvwxyz")

        strings <- c(exaple1, exaple2, exaple3, exaple4)

        data.frame( string = strings ) %>%
          mutate( result = str_detect(string, 'ab') )
```

```
##                                     string result
## 1        This is the absolute String! OMG   TRUE
## 2        This is the Absolute String! OMG  FALSE
## 3 tHIs sTrinG mAke$ aBs0lUt1y NULL s3n5E!  FALSE
## 4              abcdefghijklmnopqrstuvwxyz   TRUE
```

c. This regular expression matches:
   Either the letter ‘a’ or ‘b’ with TRUE.

```
exaple1 <- c("This is the absolute String! OMG")
exaple2 <- c("What even is this string?")
exaple3 <- c("This string is boring!", "This string is Better!", "We are all Mad down here :)")
exaple4 <- c("ab", "ba", "Ab", "aB", "AB")

strings <- c(exaple1, exaple2, exaple3, exaple4)

data.frame( string = strings ) %>%
  mutate( result = str_detect(string, '[ab]') )
```

```
##                               string result
## 1  This is the absolute String! OMG   TRUE
## 2         What even is this string?   TRUE
## 3            This string is boring!   TRUE
## 4            This string is Better!  FALSE
## 5       We are all Mad down here :)   TRUE
## 6                                ab   TRUE
## 7                                ba   TRUE
## 8                                Ab   TRUE
## 9                                aB   TRUE
## 10                               AB  FALSE
```

d. This regular expression matches:
   Returns TRUE if a string begins with either ‘a’ or ‘b’.

```
exaple1 <- c("absolute String! OMG")
exaple2 <- c("What even is this string?")
exaple3 <- c("This string is boring!", "back off! That was a nice string")
exaple4 <- c("ab", "ba", "Ab", "aB", "AB")

strings <- c(exaple1, exaple2, exaple3, exaple4)

data.frame( string = strings ) %>%
  mutate( result = str_detect(string, '^[ab]') )
```

```
##                              string result
## 1              absolute String! OMG   TRUE
## 2         What even is this string?  FALSE
## 3            This string is boring!  FALSE
## 4 back off! That was a nice string   TRUE
## 5                                ab   TRUE
## 6                                ba   TRUE
## 7                                Ab  FALSE
## 8                                aB   TRUE
## 9                                AB  FALSE
```

e. This regular expression matches:
   TRUE for any string that begins with a digit, followed by one white space and the letter ‘a’ or ‘A’.

```
        exaple1 <- c("absolute String! OMG")
        exaple2 <- c("123", "4 a random numer")
        exaple3 <- c("134 answers?!", "5        answers?", "1 truth and 2 lies")
        exaple4 <- c("1 Answer for all Questions")

        strings <- c(exaple1, exaple2, exaple3, exaple4)

        data.frame( string = strings ) %>%
          mutate( result = str_detect(string, '\\d+\\s[aA]') )
```

```
##                         string result
## 1       absolute String! OMG  FALSE
## 2                        123  FALSE
## 3           4 a random numer   TRUE
## 4              134 answers?!   TRUE
## 5          5        answers?  FALSE
## 6         1 truth and 2 lies  FALSE
## 7 1 Answer for all Questions   TRUE
```

f. This regular expression matches:
   TRUE for any string that begins with a digit, followed by any number of white spaces and the letter ‘a’ or ‘A’.

```
        exaple1 <- c("absolute String! OMG")
        exaple2 <- c("123", "4 a random numer")
        exaple3 <- c("134 answers?!", "5        answers?", "1 truth and 2 lies")
        exaple4 <- c("1 Answer for all Questions")

        strings <- c(exaple1, exaple2, exaple3, exaple4)

        data.frame( string = strings ) %>%
          mutate( result = str_detect(string, '\\d+\\s*[aA]') )
```

```
##                         string result
## 1       absolute String! OMG  FALSE
## 2                        123  FALSE
## 3           4 a random numer   TRUE
## 4              134 answers?!   TRUE
## 5          5        answers?   TRUE
## 6         1 truth and 2 lies  FALSE
## 7 1 Answer for all Questions   TRUE
```

g. This regular expression matches:
   This will return true for anything, even empty strings.

```
        exaple1 <- c("absolute_String.OMG!")
        exaple2 <- c(".wierd", "Have you heard of papertoilet.com?")
        exaple3 <- c("", "5", "1 truth and 2 lies")
        exaple4 <- c("1 Answer for all Questions")

        strings <- c(exaple1, exaple2, exaple3, exaple4)

        data.frame( string = strings ) %>%
      mutate( result = str_detect(string, '.*') )
```

```
##                                 string result
## 1               absolute_String.OMG!   TRUE
## 2                             .wierd   TRUE
## 3 Have you heard of papertoilet.com?   TRUE
## 4                                      TRUE
## 5                                  5   TRUE
## 6                 1 truth and 2 lies   TRUE
## 7         1 Answer for all Questions   TRUE
```

h. This regular expression matches: This will return TRUE for any string that starts with any two alphanumeric symbols followed by the string ‘bar’. Anything can be written afterwards in the same string.

```
exaple1 <- c("absolute_String.OMG!")
exaple2 <- c("38 bar", "10457109485710bars")
exaple3 <- c("3bars", "55bar", "43bars afuafgho iufh USGFHAA EU uhf")
exaple4 <- c("Sorry I lost my... cool there for a sec...", "Watch this :)", "Rebarb Hook", "Wha

strings <- c(exaple1, exaple2, exaple3, exaple4)

data.frame( string = strings ) %>%
  mutate( result = str_detect(string, '^\\w{2}bar') )
```

```
##                                          string result
## 1                          absolute_String.OMG!  FALSE
## 2                                        38 bar  FALSE
## 3                            10457109485710bars  FALSE
## 4                                         3bars  FALSE
## 5                                         55bar   TRUE
## 6           43bars afuafgho iufh USGFHAA EU uhf   TRUE
## 7  Sorry I lost my... cool there for a sec...  FALSE
## 8                                 Watch this :)  FALSE
## 9                                   Rebarb Hook   TRUE
## 10                                    What? lol  FALSE
```

i. This regular expression matches: This will return TRUE for any string that starts with any two alphanumeric symbols followed by the string ‘bar’. Anything can be written afterwards in the same string. . . . . OR . . . . This string exactly ‘foo.bar’

```
exaple1 <- c("absolute_String.OMG!")
exaple2 <- c("foolish.bar", "10457109485710bars")
exaple3 <- c("foo .bar", "55bar", "43bars afuafgho iufh USGFHAA EU uhf")
exaple4 <- c("foolbar", "Rebarb Hook", "foo.bar")

strings <- c(exaple1, exaple2, exaple3, exaple4)

data.frame( string = strings ) %>%
  mutate( result = str_detect(string, '(foo\\.bar)|(^\\w{2}bar)') )
```

```
##                                  string result
## 1                  absolute_String.OMG!  FALSE
## 2                           foolish.bar  FALSE
## 3                    10457109485710bars  FALSE
## 4                              foo .bar  FALSE
## 5                                 55bar   TRUE
## 6   43bars afuafgho iufh USGFHAA EU uhf   TRUE
## 7                               foolbar  FALSE
## 8                         Rebarb Hook   TRUE
## 9                             foo.bar   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:

```
     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
```

```
file.df <- data.frame(file.names) %>%
  separate(file.names, into = c("Site", "Plot", "Camera", "Date", "Time"), sep = "[._]") %>%
  separate(Date, into = c("Year", "Month", "Day"), sep = c(4, 6, 8)) %>%
  separate(Time, into = c("Hour", "Minute", "Second"), sep = c(2, 4, 6))

print(file.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
```

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.'
```

```
Gettysburg_revised <- str_replace_all(Gettysburg, " -- ", " ")
Gettysburg_revised <- str_remove_all(Gettysburg_revised, '\\.|,|-')

words <- str_split(Gettysburg_revised, "\\s+")[[1]]

mean(str_length(words))
```

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