

# STA 445 S24 Assignment 5

Ryan Joel

March 26, 2024

```
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: the strings that contain the letter 'a'.

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

```
##   string result
## 1     a    TRUE
## 2    cat    TRUE
## 3    the   FALSE
## 4    dog   FALSE
```

- b. This regular expression matches: the strings that have letters 'a' and 'b' next to each other in a single string.

```
strings <- c("abba", "cd", "about", "boat")
data.frame( string = strings ) %>%
  mutate( result = str_detect(string, 'ab') )
```

```
##   string result
## 1   abba    TRUE
## 2     cd   FALSE
## 3  about    TRUE
## 4   boat   FALSE
```

- c. This regular expression matches: any string that contains the letters 'a' or 'b' within a single string.

```
strings <- c("Albuquerque", "New Mexico", "Portland", "Oregon")
data.frame( string = strings ) %>%
  mutate( result = str_detect(string, '[ab]') )
```

```
##           string result
## 1 Albuquerque    TRUE
## 2   New Mexico   FALSE
## 3   Portland    TRUE
## 4     Oregon   FALSE
```

- d. This regular expression matches: the strings that contain an 'a' or a 'b' at the beginning of the inputted string.

```
strings <- c("a", "dog", "be", "gone")
data.frame( string = strings ) %>%
  mutate( result = str_detect(string, '^[ab]') )
```

```
## string result
## 1      a    TRUE
## 2    dog FALSE
## 3     be    TRUE
## 4   gone FALSE
```

- e. This regular expression matches: the inputted strings that contain any digit, followed by one or more repetitions of white space and a lower/ uppercase 'a'.

```
strings <- c("11 a", "1 yvA", "32 A", "45tyj")
data.frame( string = strings ) %>%
  mutate( result = str_detect(string, '\\d+\\s[aA]') )
```

```
## string result
## 1  11 a    TRUE
## 2  1 yvA FALSE
## 3  32 A    TRUE
## 4 45tyj FALSE
```

- f. This regular expression matches: any of the inputted strings that contain any digit, followed by 0 or more repetitions of white space and a capital/ lowercase 'a'.

```
strings <- c("22Ath", "34 ayu", "67tij", "89 bvr")
data.frame( string = strings ) %>%
  mutate( result = str_detect(string, '\\d+\\s*[aA]') )
```

```
## string result
## 1 22Ath    TRUE
## 2 34 ayu   TRUE
## 3 67tij   FALSE
## 4 89 bvr   FALSE
```

- g. This regular expression matches: strings with any character with 0 or more repetitions. All inputs will yield true.

```
strings <- c("12we", "35", "abcd", "1645ry")
data.frame( string = strings ) %>%
  mutate( result = str_detect(string, '.*') )
```

```
## string result
## 1 12we    TRUE
## 2  35    TRUE
## 3 abcd    TRUE
## 4 1645ry  TRUE
```

- h. This regular expression matches: strings that begin with any alphanumeric character, 2 repetitions of that, followed by 'bar'.

```
strings <- c("22bar", "a4bar", "6bvar", "b75a9r")
data.frame( string = strings ) %>%
  mutate( result = str_detect(string, '^\\w{2}bar') )
```

```
## string result
```

```
## 1 22bar TRUE
## 2 a4bar TRUE
## 3 6bvar FALSE
## 4 b75a9r FALSE
```

- i. This regular expression matches: any string that either contains 'foo' followed by a '.' and 'bar' OR strings that begin with any alphanumeric character, 2 repetitions of that, followed by 'bar.'

```
strings <- c("22bar", "90yvbar", "foo.bar", "food.bar")
data.frame( string = strings ) %>%
  mutate( result = str_detect(string, '(foo\\.bar)|(^\\w{2}bar)') )
```

```
##      string result
## 1    22bar   TRUE
## 2   90yvbar FALSE
## 3   foo.bar   TRUE
## 4  food.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 = str_replace_all(file.names, pattern = "_", replacement = "\\.")
myphotos <- data.frame(file.names)
separate(myphotos, col=file.names, into= c("site", "plot", "camera", "a", "b", "c"), sep= "\\.")%>%
  mutate(year = str_sub(a, start=1, end=4), month = str_sub(a, start=5, end=6),
         day = str_sub(a, start=7, end=8), hour = str_sub(b, start=1, end=2),
         minute = str_sub(b, start=3, end=4), second = str_sub(b, start=5, end=6))%>%
  select(site, plot, camera, year, month, day, minute, second)
```

```
##   site plot camera year month day minute second
## 1 S123   P2    C10 2012    06 21    34    22
## 2 S10    P1     C1 2012    06 22     01    48
## 3 S187   P2     C2 2012    07 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.'

```
Gettysburg = str_replace_all(Gettysburg, pattern="\\. ", replacement = " ")
Gettysburg = str_replace_all(Gettysburg, pattern=",", replacement = " ")
Gettysburg = str_replace_all(Gettysburg, pattern="--", replacement = " ")
Gettysburg = str_replace_all(Gettysburg, pattern=" ", replacement = " ")
Gettysburg = str_replace_all(Gettysburg, pattern=" ", replacement = " ")
Gettysburg = str_replace_all(Gettysburg, pattern="\n", replacement = " ")
Gettysburg = str_replace(Gettysburg, pattern="-", replacement = "")
  Gettys2 <- str_split(Gettysburg, pattern = " ")[[1]] %>%
  str_length()
  mean(Gettys2[1:271])
```

```
## [1] 4.110701
```