# STA 445 S24 Assignment 5

Sofia Mendoza

03/27/2024

```r
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
library(stringr)
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v forcats   1.0.0     v readr     2.1.5
## v ggplot2   3.4.4     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.0
## v purrr     1.0.2

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become error
```

## 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 that contains at least one lowercase a

```
string <- c("apple", "grape", "kiwi", "blueberry")
        strings <- c("apple", "grape", "kiwi", "blueberry")
        data.frame( string = strings ) %>%
          mutate( result = str_detect(string, 'a') )
```

```
##      string result
## 1     apple   TRUE
## 2     grape   TRUE
## 3      kiwi  FALSE
## 4 blueberry  FALSE
```

b. This regular expression matches: Any strings that contain the substring ab

```
string <- c("abc", "cab", "acb", "xyz")
        strings <- c("abc", "cab", "acb", "xyz")
        data.frame( string = strings ) %>%
          mutate( result = str_detect(string, 'ab') )
```

```
##   string result
## 1    abc   TRUE
## 2    cab   TRUE
## 3    acb  FALSE
## 4    xyz  FALSE
```

c. This regular expression matches: Any strings that contain either a or b anywhere within them

```
strings <- c("cab", "acb", "xyz", "foo")
        strings <- c("cab", "acb", "xyz", "foo")
        data.frame( string = strings ) %>%
          mutate( result = str_detect(string, '[ab]') )
```

```
##   string result
## 1    cab   TRUE
## 2    acb   TRUE
## 3    xyz  FALSE
## 4    foo  FALSE
```

d. This regular expression matches: Any strings that start with either a or b

```
strings <- c("banana", "cherry", "apple", "grape")
        strings <- c("banana", "cherry", "apple", "grape")
        data.frame( string = strings ) %>%
          mutate( result = str_detect(string, '^[ab]') )
```

```
##   string result
## 1 banana   TRUE
## 2 cherry  FALSE
## 3  apple   TRUE
## 4  grape  FALSE
```

e. This regular expression matches: Any strings that contain one or more digits

```
strings <- c("123 a", "4 A", "lol", "abc")
        strings <- c("123 a", "4 A", "lol", "abc")
        data.frame( string = strings ) %>%
          mutate( result = str_detect(string, '\\d+\\s[aA]') )
```

```
##   string result
## 1  123 a   TRUE
## 2    4 A   TRUE
## 3    lol  FALSE
## 4    abc  FALSE
```

f. This regular expression matches: Any strings that contain one or more digits, optionally followed by any number of whitespace characters and then either the letter a or A

```
strings <- c("123a", "456 A", "0A", "lal3")
        strings <- c("123a", "456 A", "0A", "lal3")
        data.frame( string = strings ) %>%
          mutate( result = str_detect(string, '\\d+\\s*[aA]') )
```

```
##   string result
## 1   123a   TRUE
## 2  456 A   TRUE
## 3     0A   TRUE
## 4   lal3  FALSE
```

g. This regular expression matches: Any string, including an empty string, matches zero or more occurrences of any character

```
string <- c("Good morning, world!","","12345","$%^&*()_+","lol")
        strings <- c("Good morning, world!","","12345","$%^&*()_+","lol")
        data.frame( string = strings ) %>%
      mutate( result = str_detect(string, '.*') )
```

```
##                  string result
## 1 Good morning, world!   TRUE
## 2                        TRUE
## 3                 12345   TRUE
## 4             $%^&*()_+   TRUE
## 5                   lol   TRUE
```

h. This regular expression matches: strings that start with exactly two word characters letters, digits, or underscores, followed directly by bar

```
string<- c("abbar","12bar","_9bar","abar","abcbar")
        strings <- c("abbar","12bar","_9bar","abar","abcbar")
        data.frame( string = strings ) %>%
          mutate( result = str_detect(string, '^\\w{2}bar') )
```

```
##   string result
## 1  abbar   TRUE
## 2  12bar   TRUE
## 3  _9bar   TRUE
## 4   abar  FALSE
## 5 abcbar  FALSE
```

i. This regular expression matches: Any strings that either contain foo.bar exactly as it appears, or start with exactly two word characters like letters, digits, or underscores followed directly by “bar”.

```
string <- c("foo.bar","abbar","12bar","foo bar","foobar")
        strings <- c("foo.bar","abbar","12bar","foo bar","foobar")
        data.frame( string = strings ) %>%
          mutate( result = str_detect(string, '(foo\\.bar)|(^\\w{2}bar)') )
```

```
##    string result
## 1 foo.bar   TRUE
## 2   abbar   TRUE
## 3   12bar   TRUE
## 4 foo bar  FALSE
## 5  foobar  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
```

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

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+"),
    DateTime = str_extract(file.names, "\\d{8}_\\d{6}"),
    Year = substr(DateTime, 1, 4),
    Month = substr(DateTime, 5, 6),
    Day = substr(DateTime, 7, 8),
    Hour = substr(DateTime, 10, 11),
    Minute = substr(DateTime, 12, 13),
    Second = substr(DateTime, 14, 15)
  ) %>%
  select(-file.names, -DateTime)
```

```
##   Site Plot Camera Year Month Day Hour Minute Second
## 1  123    2     10 2012    06  21   21     34     22
## 2   10    1      1 2012    06  22   05     01     48
## 3  187    2      2 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_remove_all(Gettysburg, ",")
Gettysburg <- str_replace_all(Gettysburg, "\n"," ")
Gettysburg <- str_remove_all(Gettysburg, "-")
Gettysburg <- str_remove_all(Gettysburg, "\\.")

words <- str_split(Gettysburg, " ")
words <- data.frame(reduce(words, rbind))
colnames(words) <- "word"
words <- words %>%
  filter(nchar(word)>0)
mean(nchar(words$word))
```

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

```
mean
```

```
## function (x, ...)
## UseMethod("mean")
## <bytecode: 0x146de25c8>
## <environment: namespace:base>
```