Assignment 4

Reagan Anderson

2024-10-23

Exercise 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. Make sure that your test set of strings has several examples that match as well as several that do not. Show at least two examples that return TRUE and two examples that return FALSE. *If you copy the Rmarkdown code for these exercises directly from my source pages, make sure to remove the* `eval=FALSE` *from the R-chunk headers.*

Here is an example of what a solution might look like.

q) This regular expression matches:

Any string that contains the lower-case letter "a".

```
strings <- c('Adel', 'Mathematics', 'able', 'cheese')
data.frame( string = strings ) %>%
  mutate( result = str_detect(string, 'a') )
```

```
##        string result
## 1        Adel  FALSE
## 2 Mathematics   TRUE
## 3        able   TRUE
## 4      cheese  FALSE
```

Please complete the questions below.

a) This regular expression matches:

Any string containing the letters "ab" together, in that order.

```
strings <- c( 'able', 'either', 'establishment', 'kindness' )
data.frame( string = strings ) %>%
  mutate( result = str_detect(string, 'ab') )
```

```
##          string result
## 1          able   TRUE
## 2        either  FALSE
## 3 establishment   TRUE
## 4      kindness  FALSE
```

b) This regular expression matches:

Any string that contains either letters "a" or "b"

```
strings <- c( 'been', 'apple', 'curteous', 'pi' )
data.frame( string = strings ) %>%
  mutate( result = str_detect(string, '[ab]') )
```

```
##     string result
## 1     been   TRUE
## 2    apple   TRUE
## 3 curteous  FALSE
## 4       pi  FALSE
```

c) This regular expression matches:

Any string containing "a" or "b" in the beginning of the string

```
strings <- c( 'each', 'ample', 'kindness', 'beacon' )
data.frame( string = strings ) %>%
  mutate( result = str_detect(string, '^[ab]') )
```

```
##     string result
## 1     each  FALSE
## 2    ample   TRUE
## 3 kindness  FALSE
## 4   beacon   TRUE
```

d) This regular expression matches:

Any string containing any digit, any white space with only 'a' or 'A', letters followed by the digit and space.

```
strings <- c( '1 Anonymous', '56 ample', 'purple', 'lighthouse' )
data.frame( string = strings ) %>%
  mutate( result = str_detect(string, '\\d+\\s[aA]') )
```

```
##        string result
## 1 1 Anonymous   TRUE
## 2    56 ample   TRUE
## 3      purple  FALSE
## 4  lighthouse  FALSE
```

e) This regular expression matches:

Any strings containing, any digit followed by any space with repetitions and then either letter 'A' or 'a'.

```
strings <- c( 'false', '5 5 and', 'money', '9 5 Apples' )
data.frame( string = strings ) %>%
  mutate( result = str_detect(string, '\\d+\\s*[aA]') )
```

```
##       string result
## 1      false  FALSE
## 2    5 5 and   TRUE
## 3      money  FALSE
## 4 9 5 Apples   TRUE
```

f) This regular expression matches:

Any string containing any character repeating or not

```
strings <- c( 'banana', '911', 'string', 'indecisive' )
data.frame( string = strings ) %>%
  mutate( result = str_detect(string, '.*') )
```

```
##       string result
## 1     banana   TRUE
## 2        911   TRUE
## 3     string   TRUE
## 4 indecisive   TRUE
```

g) This regular expression matches:

Any string containing any alphanumeric character repeating twice at the beginning of the string

```
strings <- c('nonobar', 'barbar', 'aabar', '55bar' )
data.frame( string = strings ) %>%
  mutate( result = str_detect(string, '^\\w{2}bar') )
```

```
##    string result
## 1 nonobar  FALSE
## 2  barbar  FALSE
## 3   aabar   TRUE
## 4   55bar   TRUE
```

h) This regular expression matches:

Ant string containing 'foo.bar' or a string with two digits at the beginning followed by bar.

```
strings <- c('food', '55barbar', 'foo.bar', 'winning')
data.frame( string = strings ) %>%
  mutate( result = str_detect(string, '(foo\\.bar)|(^\\w{2}bar)') )
```

```
##     string result
## 1     food  FALSE
## 2 55barbar   TRUE
## 3  foo.bar   TRUE
## 4  winning  FALSE
```

Exercise 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 <- str_replace_all(file.names, pattern='\\.', replacement = '_')
file.names <- str_split_fixed(file.names, pattern = '_', n=6)
file.names
```

```
##      [,1]   [,2] [,3]  [,4]       [,5]     [,6]
## [1,] "S123" "P2" "C10" "20120621" "213422" "jpg"
## [2,] "S10"  "P1" "C1"  "20120622" "050148" "jpg"
## [3,] "S187" "P2" "C2"  "20120702" "023501" "jpg"
```

```
site <- file.names[,1]

plot <- file.names[,2]

camera <- file.names[,3]

date_split <- str_split_fixed(file.names[,4], pattern = '', n=8)

year <- paste(date_split[1,1:4], sep='', collapse ='')
year <- c(year, paste(date_split[2,1:4], sep='', collapse =''))
year <- c(year, paste(date_split[3,1:4], sep='', collapse =''))

month <- paste(date_split[1,5:6], sep='', collapse='')
month <- c(month, paste(date_split[2,5:6], sep='', collapse=''))
month <- c(month, paste(date_split[3,5:6], sep='', collapse=''))

day <- paste(date_split[1,7:8], sep='', collapse='')
day <- c(day, paste(date_split[2,7:8], sep='', collapse=''))
day <- c(day, paste(date_split[3,7:8], sep='', collapse=''))

time_split <- str_split_fixed(file.names[,5], pattern='', n=6)

hour <- paste(time_split[1,1:2], sep='', collapse='')
hour <- c(hour, paste(time_split[2,1:2], sep='', collapse=''))
hour <- c(hour, paste(time_split[3,1:2], sep='', collapse=''))

minute <- paste(time_split[1,3:4], sep='', collapse='')
minute <- c(minute, paste(time_split[2,3:4], sep='', collapse=''))
minute <- c(minute, paste(time_split[3,3:4], sep='', collapse=''))

second <- paste(time_split[1,5:6], sep='', collapse='')
second <- c(second, paste(time_split[2,5:6], sep='', collapse=''))
second <- c(second, paste(time_split[3,5:6], sep='', collapse=''))
```

```
camera_df <- data.frame(
  Site = site,
  Plot = plot,
  Camera = camera,
  Year = year,
  Month = month,
  Day = day,
  Hour = hour,
  Minute = minute,
  Second = second
)
camera_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
```

Exercise 3:

The full text from Lincoln's Gettysburg Address is given below. It has been provided in a form that includes lots of different types of white space. Your goal is to calculate the mean word length of Lincoln's Gettysburg Address! *Note: you may consider 'battle-field' as one word with 11 letters or as two words 'battle' and 'field'. The first option a bit more difficult and technical!.*

```
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_split <- str_replace_all(Gettysburg, pattern='-', replacement='')
gettysburg_split <- str_replace_all(gettysburg_split, pattern = '\n', replacement='')
gettysburg_split <- str_split(gettysburg_split, pattern=' ')
```

```r
mean( str_length(gettysburg_split[[1]]) )
```

```
## [1] 4.278986
```