Strings: Extra Practice (Part 3)

You can download this .qmd file from here. Just hit the Download Raw File button.

```
library(tidyverse)
library(rvest)
library(httr)
```

On Your Own - Extra practice with strings and regular expressions

1. Describe the equivalents of ?, +, * in {m,n} form.

A ? means to repeat 0 or 1 times, so the equivalent would be {,1}, which means to repeat at most 1 time (and therefore also 0 times would work). A + means to repeat 1 or more times, so the equivalent would be {1,}. A * means to repeat 0 or more times, so the equivalent would be {0,}.

2. Describe, in words, what the expression "(.)(.)\2\1" will match, and provide a word or expression as an example.

This expression will detect any two characters that then are immediately followed by the same two characters in reverse order, in an abba pattern like "bottom".

3. Produce an R string which the regular expression represented by "\..\..\.." matches. In other words, find a string `y` below that produces a TRUE in `str_detect`.

```
test_string <- c("H.E.L.L.O. W.O.R.L.D")
str_detect(test_string, "\\..\\..\\..")
```

```
[1] TRUE
```

4. Solve with `str_subset()`, using the words from `stringr::words`:

- Find all words that start or end with x.

```r
str_subset(words, "^x|x$")
```

```
[1] "box" "sex" "six" "tax"
```

(there are no words that start with x).

- Find all words that start with a vowel and end with a consonant.

```r
str_subset(words, "^[aeiou].*[^aeiou]$")
```

```
  [1] "about"       "accept"      "account"     "across"      "act"        
  [6] "actual"      "add"         "address"     "admit"       "affect"     
 [11] "afford"      "after"       "afternoon"   "again"       "against"    
 [16] "agent"       "air"         "all"         "allow"       "almost"     
 [21] "along"       "already"     "alright"     "although"    "always"     
 [26] "amount"      "and"         "another"     "answer"      "any"        
 [31] "apart"       "apparent"    "appear"      "apply"       "appoint"    
 [36] "approach"    "arm"         "around"      "art"         "as"         
 [41] "ask"         "at"          "attend"      "authority"   "away"       
 [46] "awful"       "each"        "early"       "east"        "easy"       
 [51] "eat"         "economy"     "effect"      "egg"         "eight"      
 [56] "either"      "elect"       "electric"    "eleven"      "employ"     
 [61] "end"         "english"     "enjoy"       "enough"      "enter"      
 [66] "environment" "equal"       "especial"    "even"        "evening"    
 [71] "ever"        "every"       "exact"       "except"      "exist"      
 [76] "expect"      "explain"     "express"     "identify"    "if"         
 [81] "important"   "in"          "indeed"      "individual"  "industry"   
 [86] "inform"      "instead"     "interest"    "invest"      "it"         
 [91] "item"        "obvious"     "occasion"    "odd"         "of"         
 [96] "off"         "offer"       "often"       "okay"        "old"        
[101] "on"          "only"        "open"        "opportunity" "or"         
[106] "order"       "original"    "other"       "ought"       "out"        
[111] "over"        "own"         "under"       "understand"  "union"      
[116] "unit"        "university"  "unless"      "until"       "up"         
[121] "upon"        "usual"      
```

- Find all words that start and end with the same letter

```r
str_subset(words, "^(.).*\\1$")
```

```
 [1] "america"    "area"       "dad"        "dead"       "depend"
 [6] "educate"    "else"       "encourage"  "engine"     "europe"
[11] "evidence"   "example"    "excuse"     "exercise"   "expense"
[16] "experience" "eye"        "health"     "high"       "knock"
[21] "level"      "local"      "nation"     "non"        "rather"
[26] "refer"      "remember"   "serious"    "stairs"     "test"
[31] "tonight"    "transport"  "treat"      "trust"      "window"
[36] "yesterday"
```

5. What words in `stringr::words` have the highest number of vowels? What words have the highest proportion of vowels? (Hint: what is the denominator?) Figure this out using the tidyverse and piping, starting with `as_tibble(words) |>`.

```r
as_tibble(words) |>
  mutate(num_vowels = str_count(value, "[aeiou]")) |>
  arrange(desc(num_vowels))
```

```
# A tibble: 980 x 2
   value       num_vowels
   <chr>            <int>
 1 appropriate          5
 2 associate            5
 3 available            5
 4 colleague            5
 5 encourage            5
 6 experience           5
 7 individual           5
 8 television           5
 9 absolute             4
10 achieve              4
# i 970 more rows
```

```r
as_tibble(words) |>
  mutate(num_vowels = str_count(value, "[aeiou]"),
         prop_vowels = num_vowels/str_length(value)) |>
  arrange(desc(prop_vowels))
```

```
# A tibble: 980 x 3
   value  num_vowels prop_vowels
   <chr>       <int>       <dbl>
 1 a               1       1
 2 area             3       0.75 
 3 idea             3       0.75 
 4 age              2       0.667
 5 ago              2       0.667
 6 air              2       0.667
 7 die              2       0.667
 8 due              2       0.667
 9 eat              2       0.667
10 europe           4       0.667
# i 970 more rows
```

The words with the most values in words have 5 values; these words are as follows: appropriate, associate, available, colleague, encourage, experience, individual, and television. The word with the highest proportion of values is "a", which has a proportion of 1. The next two words have a proportion of 0.75, those words being "area" and "idea".

6. From the Harvard sentences data, use `str_extract` to produce a tibble with 3 columns: the sentence, the first word in the sentence, and the first word ending in "ed" (NA if there isn't one).

```r
as_tibble(sentences) |>
  mutate(first_word = str_extract(value, "\\b[^ ]+\\b"),
         first_ed = str_extract(value, "\\b[^ ]+ed\\b"))
```

```
# A tibble: 720 x 3
   value                                       first_word first_ed
   <chr>                                       <chr>      <chr>   
 1 The birch canoe slid on the smooth planks.  The        <NA>    
 2 Glue the sheet to the dark blue background. Glue       <NA>    
 3 It's easy to tell the depth of a well.      It's       <NA>    
 4 These days a chicken leg is a rare dish.    These      <NA>    
 5 Rice is often served in round bowls.        Rice       served  
 6 The juice of lemons makes fine punch.       The        <NA>    
 7 The box was thrown beside the parked truck. The        parked  
 8 The hogs were fed chopped corn and garbage. The        fed     
 9 Four hours of steady work faced us.         Four       faced   
10 A large size in stockings is hard to sell.  A          <NA>    
# i 710 more rows
```

7. Find and output all contractions (words with apostrophes) in the Harvard sentences, assuming no sentence has multiple contractions.

```r
str_extract(sentences, "\\b[^ ]+\\'[^ ]\\b") # will output lots of NA's, can also do:
```

```
  [1] NA           NA           "It's"       NA           NA          
  [6] NA           NA           NA           NA           NA          
 [11] NA           NA           NA           NA           NA          
 [16] NA           NA           "man's"      NA           NA          
 [21] NA           NA           NA           NA           NA          
 [26] NA           NA           NA           NA           NA          
 [31] NA           NA           NA           NA           NA          
 [36] NA           NA           NA           NA           NA          
 [41] NA           NA           NA           NA           NA          
 [46] NA           NA           NA           NA           NA          
 [51] NA           NA           NA           NA           NA          
 [56] NA           NA           NA           NA           NA          
 [61] NA           NA           NA           NA           NA          
 [66] NA           NA           NA           NA           NA          
 [71] NA           NA           NA           NA           NA          
 [76] NA           NA           NA           NA           NA          
 [81] NA           NA           NA           NA           NA          
 [86] NA           NA           NA           NA           NA          
 [91] NA           NA           NA           NA           NA          
 [96] NA           NA           NA           NA           NA          
[101] NA           NA           NA           "don't"      NA          
[106] NA           NA           NA           NA           NA          
[111] NA           NA           NA           NA           NA          
[116] NA           NA           NA           NA           NA          
[121] NA           NA           NA           NA           NA          
[126] NA           NA           NA           NA           NA          
[131] NA           NA           NA           NA           NA          
[136] NA           NA           NA           NA           NA          
[141] NA           NA           NA           NA           NA          
[146] NA           NA           NA           NA           NA          
[151] NA           NA           NA           NA           NA          
[156] NA           NA           NA           NA           NA          
[161] NA           NA           NA           NA           NA          
[166] NA           NA           NA           NA           NA          
[171] NA           NA           NA           NA           NA          
[176] NA           NA           NA           NA           "store's"   
[181] NA           NA           NA           NA           NA          
[186] NA           NA           NA           NA           NA          
[191] NA           NA           NA           NA           NA          
[196] NA           NA           NA           NA           NA          
```

```
[201] NA            NA            NA            NA            NA
[206] NA            NA            NA            NA            NA
[211] NA            NA            NA            NA            NA
[216] NA            NA            NA            NA            NA
[221] NA            NA            NA            NA            NA
[226] NA            NA            NA            NA            NA
[231] NA            NA            NA            NA            NA
[236] NA            NA            NA            NA            NA
[241] NA            NA            NA            NA            NA
[246] NA            NA            NA            NA            NA
[251] NA            NA            NA            NA            NA
[256] NA            NA            NA            NA            NA
[261] NA            NA            NA            NA            NA
[266] NA            NA            NA            NA            NA
[271] NA            NA            NA            NA            NA
[276] NA            NA            NA            NA            NA
[281] NA            NA            NA            NA            NA
[286] NA            NA            NA            NA            NA
[291] NA            NA            NA            NA            NA
[296] NA            NA            NA            NA            NA
[301] NA            NA            "workman's"   NA            NA
[306] NA            NA            "Let's"       NA            NA
[311] NA            NA            NA            NA            NA
[316] NA            NA            NA            NA            NA
[321] NA            NA            NA            NA            NA
[326] "sun's"       NA            NA            NA            NA
[331] NA            NA            NA            NA            NA
[336] NA            NA            NA            NA            NA
[341] NA            NA            NA            NA            NA
[346] NA            NA            "child's"     NA            NA
[351] NA            NA            NA            NA            NA
[356] NA            "king's"      NA            NA            NA
[361] NA            NA            NA            NA            NA
[366] NA            NA            NA            NA            NA
[371] NA            NA            NA            NA            NA
[376] NA            NA            NA            NA            NA
[381] NA            NA            NA            NA            NA
[386] NA            NA            NA            NA            NA
[391] NA            NA            "It's"        NA            NA
[396] NA            NA            NA            NA            NA
[401] NA            NA            NA            NA            NA
[406] NA            NA            NA            NA            NA
[411] NA            NA            NA            NA            NA
```

```
[416] NA            NA            NA            NA            NA
[421] NA            NA            NA            NA            NA
[426] NA            NA            NA            NA            NA
[431] NA            NA            NA            NA            NA
[436] NA            NA            NA            NA            NA
[441] NA            NA            NA            NA            NA
[446] NA            NA            NA            NA            NA
[451] NA            NA            NA            NA            NA
[456] NA            NA            NA            NA            NA
[461] NA            NA            NA            NA            NA
[466] NA            NA            NA            NA            NA
[471] NA            NA            NA            NA            NA
[476] NA            "don't"       NA            NA            NA
[481] NA            NA            NA            NA            NA
[486] NA            NA            NA            NA            NA
[491] NA            NA            NA            NA            NA
[496] NA            NA            NA            NA            NA
[501] NA            NA            NA            NA            NA
[506] NA            NA            NA            NA            NA
[511] NA            NA            "queen's"     NA            NA
[516] NA            NA            NA            NA            NA
[521] NA            NA            NA            NA            NA
[526] NA            NA            NA            NA            NA
[531] NA            NA            NA            NA            NA
[536] "don't"       NA            NA            NA            NA
[541] NA            NA            NA            NA            NA
[546] NA            NA            NA            NA            NA
[551] NA            NA            NA            NA            NA
[556] NA            NA            NA            NA            NA
[561] NA            NA            NA            NA            NA
[566] NA            NA            NA            NA            NA
[571] NA            NA            NA            NA            NA
[576] NA            NA            NA            NA            NA
[581] NA            NA            NA            NA            NA
[586] NA            NA            NA            NA            NA
[591] NA            NA            NA            NA            NA
[596] NA            NA            NA            NA            NA
[601] NA            NA            NA            NA            NA
[606] NA            NA            NA            NA            NA
[611] NA            NA            NA            NA            NA
[616] NA            NA            NA            NA            NA
[621] NA            NA            NA            "don't"       NA
[626] NA            NA            NA            NA            NA
```

```
[631] NA           NA           NA           NA           NA          
[636] "don't"      NA           NA           NA           NA          
[641] NA           NA           NA           NA           NA          
[646] NA           NA           NA           NA           NA          
[651] NA           NA           NA           NA           NA          
[656] NA           NA           NA           NA           NA          
[661] NA           NA           NA           NA           NA          
[666] NA           NA           NA           NA           NA          
[671] NA           NA           NA           NA           NA          
[676] NA           NA           "don't"      NA           "pirate's"  
[681] NA           NA           NA           NA           NA          
[686] NA           NA           NA           NA           NA          
[691] NA           NA           NA           NA           NA          
[696] NA           NA           NA           NA           NA          
[701] NA           NA           NA           NA           NA          
[706] NA           NA           NA           NA           NA          
[711] NA           NA           "neighbor's" NA           NA          
[716] NA           NA           NA           NA           NA          
```

```r
as_tibble(sentences) |>
  mutate(contractions = str_extract(value, "\\b[^ ]+\\'[^ ]\\b")) |>
  select(contractions) |>
  filter(!is.na(contractions)) #tibble, no NAs
```

```
# A tibble: 18 x 1
   contractions
   <chr>       
 1 It's        
 2 man's       
 3 don't       
 4 store's     
 5 workman's   
 6 Let's       
 7 sun's       
 8 child's     
 9 king's      
10 It's        
11 don't       
12 queen's     
13 don't       
14 don't       
15 don't       
```

```
16 don't
17 pirate's
18 neighbor's
```

8. *Carefully* explain what the code below does, both line by line and in general terms.

```r
temp <- str_replace_all(words, "^([A-Za-z])(.*)([a-z])$", "\\3\\2\\1")
as_tibble(words) |>
  semi_join(as_tibble(temp)) |>
  print(n = Inf)
```

```
Joining with `by = join_by(value)`
```

```
# A tibble: 45 x 1
   value
   <chr>
 1 a
 2 america
 3 area
 4 dad
 5 dead
 6 deal
 7 dear
 8 depend
 9 dog
10 educate
11 else
12 encourage
13 engine
14 europe
15 evidence
16 example
17 excuse
18 exercise
19 expense
20 experience
21 eye
22 god
23 health
24 high
25 knock
26 lead
```

```
27 level
28 local
29 nation
30 no
31 non
32 on
33 rather
34 read
35 refer
36 remember
37 serious
38 stairs
39 test
40 tonight
41 transport
42 treat
43 trust
44 window
45 yesterday
```

The first line creates a temporary tibble called temp that switches the first and last letter of every word (so “woman” becomes “nomaw”, for example). The second line turns the list of normal words into a tibble, then the third line takes the two tables and returns only the values from the original words table that are also found in the temporary table with the letters switched, and the last line prints out all the results. Overall, this code produces a tibble of words from the words data that are still a word if you switch the first and last letters, whether this is because those are the same letter (like in “dad”), because the switch makes a new word (like how “deal” becomes “lead”), or because the word only has one letter (as in “a”).

Coco and Rotten Tomatoes

We will check out the Rotten Tomatoes page for the 2017 movie Coco, scrape information from that page (we’ll get into web scraping in a few weeks!), clean it up into a usable format, and answer some questions using strings and regular expressions.

```r
# used to work
# coco <- read_html("https://www.rottentomatoes.com/m/coco_2017")

# robotstxt::paths_allowed("https://www.rottentomatoes.com/m/coco_2017")

library(polite)
```

```
Warning: package 'polite' was built under R version 4.4.3
```

```
coco <- "https://www.rottentomatoes.com/m/coco_2017" |>
  bow() |>
  scrape()

top_reviews <-
  "https://www.rottentomatoes.com/m/coco_2017/reviews?type=top_critics" |>
  bow() |>
  scrape()
top_reviews <- html_nodes(top_reviews, ".review-text")
top_reviews <- html_text(top_reviews)

user_reviews <-
  "https://www.rottentomatoes.com/m/coco_2017/reviews?type=user" |>
  bow() |>
  scrape()
user_reviews <- html_nodes(user_reviews, ".js-review-text")
user_reviews <- html_text(user_reviews)
```

9. `top_reviews` is a character vector containing the 20 most recent critic reviews (along with some other junk) for Coco, while `user_reviews` is a character vector with the 10 most recent user reviews.

a) Explain how the code below helps clean up both `user_reviews` and `top_reviews` before we start using them.

```
user_reviews <- str_trim(user_reviews)
top_reviews <- str_trim(top_reviews)
```

This code will get rid of excessive whitespace (tabs and such) at the start and end of a string. Before running this code, there is a large blank space at the start of each review, which this code gets rid of.

b) Print out the critic reviews where the reviewer mentions "emotion" or "cry". Think about various forms ("cried", "emotional", etc.) You may want to turn reviews to all lower case before searching for matches.

```
as_tibble(top_reviews) |>
  mutate(lower = str_to_lower(value),
         emo_cry = str_detect(lower, "emotion|cry|cried")) |>
  filter(emo_cry) |>
  select(value)
```

```
# A tibble: 3 x 1
  value
  <chr>
1 A wonderful return to form for Pixar, who again deliver the emotional and cre~
2 At worst it suggests that the brains trust at Pixar, after 22 years of peerle~
3 Funny, irreverent and eye-popping. It will also make you want to cry at least~
```

c) In critic reviews, replace all instances where "Pixar" is used with its full name: "Pixar Animation Studios".

```r
str_replace(top_reviews, "Pixar", "Pixar Animation Studios")
```

```
 [1] "A fine addition to the Pixar Animation Studios legacy… a very sweet film about family,
 [2] "An unexpectedly brilliant and dynamic story about lineage, connection, and self-discove
 [3] "In a country with an ever increasing Hispanic and Mexican population, a film like Coco
 [4] "I don't think there's any question that Coco is really great."
 [5] "Several times I found myself sobbing without knowing exactly why only to realize why th
 [6] "A wonderful return to form for Pixar Animation Studios, who again deliver the emotional
 [7] "The film has a galloping rhythm, and the animation is scrupulous and ravishing, from it
 [8] "On paper, the mythology scans as complicated and dark, but in the capable hands of Acad
 [9] "Pixar Animation Studios's latest project is a glittering return to non-franchise form a
[10] "Its victorious denouement offers everyone a different way to think about what it means
[11] "At worst it suggests that the brains trust at Pixar Animation Studios, after 22 years o
[12] "Funny, irreverent and eye-popping. It will also make you want to cry at least once but
[13] "This is a charming and very memorable film."
[14] "Despite the fact that it's so well told and really beautifully directed, it doesn't hav
[15] "... Coco is another triumph for Pixar Animation Studios..."
[16] "Funny and heart-tugging with some knockout tunes, the movie glows with warmth. And how
[17] "Not top-tier Pixar Animation Studios. But decent enough."
[18] "Pixar Animation Studios has raised the animation bar again, with its most musical - and
[19] "While the animation is Pixar Animation Studios perfect, I don't think the story grips t
[20] "Every plot point and thematic implication slots into place, but the pleasures of Coco a
```

d) Find out how many times each user uses "I" in their review. Remember that it could be used as upper or lower case, at the beginning, middle, or end of a sentence, etc.

```r
as_tibble(user_reviews) |>
  mutate(i_count = str_count(value, "[iI]"))
```

```
# A tibble: 20 x 2
   value                                                                   i_count
   <chr>                                                                     <int>
```

```
 1 "\n                      Coco is more than just an animated film - i~      64
 2 "\n                      LOVE THIS! When my favorite movies is such ~       7
 3 "\n                      Esse foi um dos melhores desenhos que eu já~       2
 4 "\n                      Film commovente ma non un capolavoro \n    ~       1
 5 "\n                      Amazing movie. Really makes you think what ~      15
 6 "\n                      Coco is Pixar's most human film. The movie ~       7
 7 "\n                      This movie EASILY has the best Disney plot ~      15
 8 "\n                      A very emotional story. A kid character tha~      10
 9 "\n                      I can’t stop crying plz help\n             ~       2
10 "\n                      Cried more with this than with Heidi. Absol~      13
11 "\n                      This animation has everything. \nA great st~       9
12 "\n                      such wonderful heartworming film \n        ~       2
13 "\n                      Coco is a nice movie about family and remem~      17
14 "\n                      What I liked most about Coco was the music ~      18
15 "\n                      When I see the absolute cinema meme I think~      19
16 "\n                      This is meaning to anyone who actually feel~      14
17 "\n                      It was so emotional and it had some laughs,~       4
18 "\n                      Pixar at its finest since Toy Story 3\n    ~       4
19 "\n                      20th Century Studios had the Book of Life, ~      67
20 "\n                      One of my favourite Pixar movies ever. Made~      13
```

e) Do critics or users have more complex reviews, as measured by average number of commas used? Be sure your code weeds out commas used in numbers, such as “12,345”.

```r
mean(str_count(user_reviews, ", "))
```

```
[1] 1.45
```

```r
mean(str_count(top_reviews, ", "))
```

```
[1] 1.35
```

The average number of commas in a user review is 1.45, while the average number in a critic review is 1.35. Thus, user reviews are slightly more complex on average.