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.

? {0,1} + {1,} * {0,}

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

The expression will flip any two characters around, so for example, it would transform "we" into "ew".

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`.

```
str_detect(".w.w.w", "\\..\\..\\..")
```

```
[1] TRUE
```

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

- Find all words that start or end with x.
- Find all words that start with a vowel and end with a consonant.
- Find all words that start and end with the same letter

```
str_subset(stringr::words, "(^x|x$)")
```

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

```
str_subset(stringr::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"
```

```
str_subset(stringr::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) |>`.

```
vowel_tbl <- as_tibble(words) |>
  mutate(
    vowels = str_count(value, "[aeiou]"),
    prop_vowels = vowels / str_length(value)
  )

vowel_tbl |>
  arrange(desc(vowels))
```

```
# A tibble: 980 x 3
   value       vowels prop_vowels
   <chr>        <int>       <dbl>
 1 appropriate      5       0.455
 2 associate        5       0.556
 3 available        5       0.556
 4 colleague        5       0.556
 5 encourage        5       0.556
 6 experience       5       0.5
 7 individual       5       0.5
 8 television       5       0.5
 9 absolute         4       0.5
10 achieve          4       0.571
# i 970 more rows
```

```
vowel_tbl |>
  arrange(desc(prop_vowels))
```

```
# A tibble: 980 x 3
   value  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
```

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).

```
sentence_tbl <- as_tibble(sentences) |>
  mutate(first_word = str_extract(sentences, "^\\b[^ ]+\\b")) |>
  mutate(ed_end = str_extract(sentences, "\\b[^ ]*ed\\b"))
```

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

```
str_extract(sentences, "\\b[^ ]*'\\b")
```

```
  [1] NA          NA          "It'"       NA          NA          NA
  [7] NA          NA          NA          NA          NA          NA
 [13] NA          NA          NA          NA          NA          "man'"
 [19] NA          NA          NA          NA          NA          NA
 [25] NA          NA          NA          NA          NA          NA
 [31] NA          NA          NA          NA          NA          NA
 [37] NA          NA          NA          NA          NA          NA
 [43] NA          NA          NA          NA          NA          NA
 [49] NA          NA          NA          NA          NA          NA
 [55] NA          NA          NA          NA          NA          NA
 [61] NA          NA          NA          NA          NA          NA
 [67] NA          NA          NA          NA          NA          NA
 [73] NA          NA          NA          NA          NA          NA
 [79] NA          NA          NA          NA          NA          NA
 [85] NA          NA          NA          NA          NA          NA
 [91] NA          NA          NA          NA          NA          NA
 [97] NA          NA          NA          NA          NA          NA
[103] NA          "don'"      NA          NA          NA          NA
[109] NA          NA          NA          NA          NA          NA
[115] NA          NA          NA          NA          NA          NA
[121] NA          NA          NA          NA          NA          NA
```

```
[127] NA          NA          NA          NA          NA          NA
[133] NA          NA          NA          NA          NA          NA
[139] NA          NA          NA          NA          NA          NA
[145] NA          NA          NA          NA          NA          NA
[151] NA          NA          NA          NA          NA          NA
[157] NA          NA          NA          NA          NA          NA
[163] NA          NA          NA          NA          NA          NA
[169] NA          NA          NA          NA          NA          NA
[175] NA          NA          NA          NA          NA          "store'"
[181] NA          NA          NA          NA          NA          NA
[187] NA          NA          NA          NA          NA          NA
[193] NA          NA          NA          NA          NA          NA
[199] NA          NA          NA          NA          NA          NA
[205] NA          NA          NA          NA          NA          NA
[211] NA          NA          NA          NA          NA          NA
[217] NA          NA          NA          NA          NA          NA
[223] NA          NA          NA          NA          NA          NA
[229] NA          NA          NA          NA          NA          NA
[235] NA          NA          NA          NA          NA          NA
[241] NA          NA          NA          NA          NA          NA
[247] NA          NA          NA          NA          NA          NA
[253] NA          NA          NA          NA          NA          NA
[259] NA          NA          NA          NA          NA          NA
[265] NA          NA          NA          NA          NA          NA
[271] NA          NA          NA          NA          NA          NA
[277] NA          NA          NA          NA          NA          NA
[283] NA          NA          NA          NA          NA          NA
[289] NA          NA          NA          NA          NA          NA
[295] NA          NA          NA          NA          NA          NA
[301] NA          NA          "workman'"  NA          NA          NA
[307] NA          "Let'"      NA          NA          NA          NA
[313] NA          NA          NA          NA          NA          NA
[319] NA          NA          NA          NA          NA          NA
[325] NA          "sun'"      NA          NA          NA          NA
[331] NA          NA          NA          NA          NA          NA
[337] NA          NA          NA          NA          NA          NA
[343] NA          NA          NA          NA          NA          "child'"
[349] NA          NA          NA          NA          NA          NA
[355] NA          NA          "king'"     NA          NA          NA
[361] NA          NA          NA          NA          NA          NA
[367] NA          NA          NA          NA          NA          NA
[373] NA          NA          NA          NA          NA          NA
[379] NA          NA          NA          NA          NA          NA
```

```
[385] NA          NA          NA          NA          NA          NA
[391] NA          NA          "It'"       NA          NA          NA
[397] NA          NA          NA          NA          NA          NA
[403] NA          NA          NA          NA          NA          NA
[409] NA          NA          NA          NA          NA          NA
[415] NA          NA          NA          NA          NA          NA
[421] NA          NA          NA          NA          NA          NA
[427] NA          NA          NA          NA          NA          NA
[433] NA          NA          NA          NA          NA          NA
[439] NA          NA          NA          NA          NA          NA
[445] NA          NA          NA          NA          NA          NA
[451] NA          NA          NA          NA          NA          NA
[457] NA          NA          NA          NA          NA          NA
[463] NA          NA          NA          NA          NA          NA
[469] NA          NA          NA          NA          NA          NA
[475] NA          NA          "don'"      NA          NA          NA
[481] NA          NA          NA          NA          NA          NA
[487] NA          NA          NA          NA          NA          NA
[493] NA          NA          NA          NA          NA          NA
[499] NA          NA          NA          NA          NA          NA
[505] NA          NA          NA          NA          NA          NA
[511] NA          NA          "queen'"    NA          NA          NA
[517] NA          NA          NA          NA          NA          NA
[523] NA          NA          NA          NA          NA          NA
[529] NA          NA          NA          NA          NA          NA
[535] NA          "don'"      NA          NA          NA          NA
[541] NA          NA          NA          NA          NA          NA
[547] NA          NA          NA          NA          NA          NA
[553] NA          NA          NA          NA          NA          NA
[559] NA          NA          NA          NA          NA          NA
[565] NA          NA          NA          NA          NA          NA
[571] NA          NA          NA          NA          NA          NA
[577] NA          NA          NA          NA          NA          NA
[583] NA          NA          NA          NA          NA          NA
[589] NA          NA          NA          NA          NA          NA
[595] NA          NA          NA          NA          NA          NA
[601] NA          NA          NA          NA          NA          NA
[607] NA          NA          NA          NA          NA          NA
[613] NA          NA          NA          NA          NA          NA
[619] NA          NA          NA          NA          NA          "don'"
[625] NA          NA          NA          NA          NA          NA
[631] NA          NA          NA          NA          NA          "don'"
[637] NA          NA          NA          NA          NA          NA
```

```
[643] NA          NA          NA          NA          NA          NA
[649] NA          NA          NA          NA          NA          NA
[655] NA          NA          NA          NA          NA          NA
[661] NA          NA          NA          NA          NA          NA
[667] NA          NA          NA          NA          NA          NA
[673] NA          NA          NA          NA          NA          "don'"
[679] NA          "pirate'"   NA          NA          NA          NA
[685] NA          NA          NA          NA          NA          NA
[691] NA          NA          NA          NA          NA          NA
[697] NA          NA          NA          NA          NA          NA
[703] NA          NA          NA          NA          NA          NA
[709] NA          NA          NA          NA          "neighbor'" NA
[715] NA          NA          NA          NA          NA          NA
```

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

```
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
```

This code changes the first and last letters (first line) and evaluates if it is in the words dictionary(lines 2 and 3), and then prints them (line 4).

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.

```
# 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)
```

Trim helps us replace some of the whitespace in the reviews (before and after) so that it is easier to do analysis and manipulation on.

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.

```
lower_reviews <- str_to_lower(top_reviews)

emotion_cry_reviews <- str_subset(lower_reviews, "emotion|cry")

emotion_cry_reviews
```

```
[1] "a wonderful return to form for pixar, who again deliver the emotional and creative goods
[2] "at worst it suggests that the brains trust at pixar, after 22 years of peerless output
[3] "funny, irreverent and eye-popping. it will also make you want to cry at least once but p
```

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

```
top_reviews <- str_replace_all(top_reviews, "\\bPixar\\b", "Pixar Animation Studios")
```

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.

```
str_count(user_reviews, "\\b(I|i)\\b")
```

```
 [1] 3 0 0 2 0 2 0 0 2 1 0 4 0 0 1 0 3 0 0 0
```

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".

```
str_replace_all(user_reviews, "(\\d),(\\d)", "\\1\\2")
```

```
 [1] "I loved the plot twist at the end, and the designs of some of the characters are peak.
 [2] "Captivatingly beautiful. Will tug at your heartstrings."
 [3] "Such A Beautiful Film, This Story Is Disney/Pixar At Their Finest! The Visuals And Musi
 [4] "I find myself as a grown man watching this every once and while when I smoke a joint an
 [5] "This was a good and funny musical that everyone can in joy."
 [6] "Definitely the best movie of 2017 and what makes it so are the beautiful tearjerking en
 [7] "Great movie excited about the second movie"
 [8] "flawless film. just stunning."
 [9] "All Time Classic. Just masterfully done across the board. The music is really distinct
[10] "I've seen this movie a hundred times, and every time it gets to me still. It's amazingl
[11] "The film's visuals and narrative take you on a journey through culture, family, life, a
[12] "How can I not love this movie? Coco is my favorite movie in Pixar. I have watched this
[13] "What's not to love?  This tender story of the little boy who journeys to the Land of th
[14] "Another enjoyable movie from Pixar.  Heartwarming and great for the entire family."
```

```
[15] "It's one of the best movies pixar has made in these late years. No doubt about it.\nThe
[16] "Very colorful much like Encanto and had a fun way of dealing with death and remembering
[17] "Mild spoilers: \n\nI've heard people make fun of the fact that there is a twist-villain
[18] "So good, visually stunning."
[19] "Excellent coulorful movie explaining the Mexican culture of the Day of The Dead"
[20] "A very good and well made pixar film. The musics are fire and the plot twist was undpre
```

```
str_replace_all(top_reviews, "(\\d),(\\d)", "\\1\\2")
```

```
 [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
```

```
critic_commas <- str_count(top_reviews, ",")
user_commas <- str_count(user_reviews, ",")

mean(critic_commas)
```

```
[1] 1.35
```

```
mean(user_commas)
```

```
[1] 2
```

Users have a higher average number of commas used.