String Manipulation and Regular Expressions Assignment

Assignment Instructions Complete all questions below. After completing the assignment, knit your document, and download both your .Rmd and knitted output. Upload your files for peer review.

For each response, include comments detailing your response and what each line does. Ensure you test your functions with sufficient test cases to identify and correct any potential bugs.

Required Libraries Load the stringr library

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.1     v tibble    3.2.1
## v lubridate 1.9.4     v tidyr     1.3.1
## v purrr     1.0.4
## -- 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 errors
```

Question 1. Use str_c to put (before the area codes followed by) and a space followed by the phone number.

```
### Answer should be of the form "(703) 5551212" "(863) 1234567" "(404) 7891234" "(202) 4799747"
area_codes <- c(703, 863, 404, 202)
phone_nums <- c(5551212, 1234567, 7891234, 4799747)
area_codes <- str_c("(", area_codes, ")") # format area codes (xxx)
comb <- str_c(area_codes, phone_nums, sep = " ") # combine area codes and phone numbers w/ " "
print(comb)
```

```
## [1] "(703) 5551212" "(863) 1234567" "(404) 7891234" "(202) 4799747"
```

Question 2. Create a function that receives a single word as an input. Use str_length() and str_sub() to extract the middle character from the string. What will you do if the string has an even number of characters? Test your function on the strings “hamburger” and “hotdog”

```
mid_char <- function (s) {
    slen <- str_length(s) # get length
    middle_index <- slen / 2 # find middle
    if(slen %% 2 == 0) { # if even length
        # subset string to extract two middle characters
        middle_char <- str_sub(s, middle_index, middle_index + 1)
```

```
    } else { # if odd length
        middle_index <- ceiling(middle_index) # round index for middle char
        middle_char <- str_sub(s, middle_index, middle_index) # subset str for middle char
    }
    return(middle_char)
}
```

Question 3. How would you match the sequence "'? Note this is a double quote, single quote, backslash and question mark. Build it up one piece at a time. Use it to identify that sequence contained in s2 below.

```
s <- "\"'\\?"
s2 <- str_c("some stuff",s,"more!")

str_view(s2, "(\"'\\\\\\?)")
```

```
## [1] | some stuff<"'\?>more!
```

Question 4. Using the words provided in stringr::words, create regular expressions that find all words that:

```
# a.  End with "ing" or "ise"
words %>% str_view(".+i(ng|se)$")
```

```
##  [15] | <advertise>
## [113] | <bring>
## [251] | <during>
## [280] | <evening>
## [288] | <exercise>
## [448] | <king>
## [512] | <meaning>
## [533] | <morning>
## [588] | <otherwise>
## [637] | <practise>
## [674] | <raise>
## [681] | <realise>
## [709] | <ring>
## [710] | <rise>
## [765] | <sing>
## [834] | <surprise>
## [860] | <thing>
```

```
# b.  Do not follow the rule "i before e except after c"
words %>% str_view(".*[^c].*ei|.*[c].*ie")
```

```
##   [7] | <achie>ve
## [158] | <clie>nt
## [684] | <recei>ve
## [726] | <scie>nce
## [781] | <socie>ty
## [939] | <wei>gh
```

```
# c.  Begin with at least two vowels and end with at least two consonants
words %>% str_view("^[aeiou]{2,}.*[^aeiou]{2,}")
```

```
##  [61] | <authority>
## [252] | <each>
## [253] | <early>
## [254] | <east>
## [255] | <easy>
## [261] | <eight>
## [262] | <eith>er
## [589] | <ought>
```

```
# d.  Contain a repeated pair of letters (e.g. "church" contains "ch" twice)
words %>% str_view(".*(..).*\\1.*")
```

```
##  [48] | <appropriate>
## [152] | <church>
## [181] | <condition>
## [217] | <decide>
## [275] | <environment>
## [487] | <london>
## [598] | <paragraph>
## [603] | <particular>
## [617] | <photograph>
## [638] | <prepare>
## [641] | <pressure>
## [696] | <remember>
## [698] | <represent>
## [699] | <require>
## [739] | <sense>
## [858] | <therefore>
## [903] | <understand>
## [946] | <whether>
```

```
# e.  Contain one letter other than e that is repeated in at least three places (e.g. "appropriate" con
words %>% str_view(".*([^e])(.*\\1.*){2,}")
```

```
##  [48] | <appropriate>
##  [62] | <available>
## [119] | <business>
## [233] | <discuss>
## [275] | <environment>
## [423] | <individual>
## [598] | <paragraph>
## [877] | <tomorrow>
```

Question 5. Using the sentences provided in stringr::sentences, find all words that come after a "number" like "one", "two", ... "twelve". Pull out both the number and the word.

```r
# create number vector
nums <- c("one","two","three","four","five","six","seven","eight","nine","ten","eleven","twelve","[0-9]+
# create regex for numbers
numpat <- str_c("\\b(", str_c(nums, collapse = "|"), ")\\b")
# create tibble from sentences
sen1 <- tibble(sentence = sentences)
res <- sen1 %>%
    # make number and word columns
    mutate(
        number = str_extract(sentence,numpat), # extract number
        word = str_extract(sentence, str_c(numpat, "\\s(\\w+)")) %>% str_extract("\\w+") # extract word
    ) %>%
    filter(!is.na(number)) %>%  filter(!is.na(word)) # filter NAs
print(res)
```

```
## # A tibble: 22 x 3
##    sentence                                    number word
##    <chr>                                       <chr>  <chr>
##  1 The rope will bind the seven books at once. seven  seven
##  2 The two met while playing on the sand.      two    two
##  3 There are more than two factors here.       two    two
##  4 Type out three lists of orders.             three  three
##  5 Two plus seven is less than ten.            seven  seven
##  6 Drop the two when you add the figures.      two    two
##  7 There the flood mark is ten inches.         ten    ten
##  8 We are sure that one war is enough.         one    one
##  9 His shirt was clean but one button was gone. one   one
## 10 The fight will end in just six minutes.     six    six
## # i 12 more rows
```

Question 6. Using the sentences provided in stringr::sentences, view all sentences that contain the word "good" or the word "bad". Get the sentence numbers where those words occur. Use str_replace_all() to replace the word "bad" with the word "horrible" and the word "good" with the word "great". Look at the sentence numbers you found before to verify the words were replaced correctly.

```r
# Find index of line for each sentence with good or bad
str_indices <- sentences %>%
    str_detect("\\s(good|bad)\\s") %>%
    which()
# replace all goods wth greats and all bads with horribles
new_sentences <- sentences %>%
    str_replace_all("good", "great") %>%
    str_replace_all("bad", "horrible")
# check your work
new_sentences[str_indices]
```

```
##  [1] "We frown when events take a horrible turn."
##  [2] "We admire and love a great cook."
##  [3] "Sell your gift to a buyer at a great gain."
##  [4] "These pills do less great than others."
##  [5] "It takes a great trap to capture a bear."
##  [6] "Much of the story makes great sense."
```

```
##  [7] "The price is fair for a great antique clock."
##  [8] "The water in this well is a source of great health."
##  [9] "A great book informs of what we ought to know."
## [10] "It was a horrible error on the part of the new judge."
```