

String Manipulation and Regular Expressions Assignment

Tyler Kephart

2024-08-15

Required Libraries Load the stringr library

```
library(stringr)
```

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)
# string concat with () around area code and space between area code and number
str_c("(", area_codes, ") ", phone_nums)
```

```
## [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”

```
# create function
get_middle_character <- function(word) {
  # get string length
  word_length <- str_length(word)
  # check if length even or odd
  if (word_length %% 2 == 1) {
    # if odd, set middle character position
    middle <- (word_length + 1) / 2
  } else {
    # else even, set right middle character position
    middle <- word_length / 2 + 1
  }
  # return middle character
  str_sub(word, middle, middle)
}
# test with hamburger and hotdog
get_middle_character("hamburger")
```

```
## [1] "u"
```

```
get_middle_character("hotdog")
```

```
## [1] "d"
```

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!")

# define each sequence
double_quote <- "\"\""
single_quote <- "'"
backslash <- "\\\"
question_mark <- "\\?"
# combine sequences into a pattern
pattern <- str_c(double_quote, single_quote, backslash, question_mark)
# detect if pattern is in s2
str_detect(s2, pattern)
```

```
## [1] TRUE
```

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

```
# End with "ing" or "ise"
words[str_detect(words, "(ing|ise)$")]
```

```
## [1] "advertise" "bring"      "during"      "evening"      "exercise"      "king"
## [7] "meaning"   "morning"     "otherwise"    "practise"     "raise"         "realise"
## [13] "ring"      "rise"        "sing"        "surprise"     "thing"
```

```
# Do not follow the rule "i before e except after c"
words[str_detect(words, "(?<=c)ie|(?<!c)ei")]
```

```
## [1] "eight"      "either"      "science"     "society"     "weigh"
```

```
# Begin with at least two vowels and end with at least two consonants
words[str_detect(words, "^([aeiou]{2,}).*[bcdfghjklmnpqrstvwxyz]{2,}$")]
```

```
## [1] "authority" "each"       "early"      "east"       "easy"       "eight"
## [7] "ought"
```

```
# Contain a repeated pair of letters (e.g. "church" contains "ch" twice)
words[str_detect(words, "([[:alpha:]]{2}).*\\1")]
```

```
## [1] "appropriate" "church"      "condition"   "decide"      "environment"
## [6] "london"       "paragraph"   "particular"  "photograph"  "prepare"
## [11] "pressure"     "remember"    "represent"   "require"     "sense"
## [16] "therefore"    "understand"  "whether"
```

```
# Contain one letter other than e that is repeated in at least three places (e.g. "appropriate" contains three 'p's)
words[str_detect(words, "([^\e]).*\\1.*\\1")]
```

```
## [1] "appropriate" "available"    "business"    "discuss"     "environment"
## [6] "individual"   "paragraph"    "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.

```

# set up numbers
numbers <- c("one", "two", "three", "four", "five", "six", "seven", "eight", "nine", "ten", "eleven", "twelve")
# set up pattern
pattern <- str_c("\\b(", str_c(numbers, collapse = "|"), ")\\b\\s+(\\w+)")
# make sure matches is empty before filling up with number and word pairs
matches <- NULL
# loop thru each sentence
for (sentence in sentences) {
  # check if the pattern matches in the sentence
  if (str_detect(sentence, pattern)) {
    # if match, add number and word string to matches
    matches <- c(matches, str_match(sentence, pattern)[1])
  }
}
# look at the number and word pairs
matches

```

```

## [1] "seven books"      "two met"          "two factors"      "three lists"
## [5] "seven is"         "two when"         "ten inches"       "one war"
## [9] "one button"       "six minutes"      "ten years"        "two shares"
## [13] "two distinct"     "five cents"       "two pins"         "five robins"
## [17] "four kinds"       "three story"      "three inches"     "six comes"
## [21] "three batches"   "two leaves"

```

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.

```

# set "good" or "bad" pattern
pattern <- "\\b(good|bad)\\b"
# find sentence numbers that do contain the pattern
sentence_numbers <- which(str_detect(sentences, pattern))
# print the sentence numbers that have good or bad
print(sentence_numbers)

```

```

## [1] 139 187 195 259 360 389 460 508 612 656 668 692

```

```

# print the good or bad sentences
print(sentences[sentence_numbers])

```

```

## [1] "We frown when events take a bad turn."
## [2] "We admire and love a good cook."
## [3] "Sell your gift to a buyer at a good gain."
## [4] "These pills do less good than others."
## [5] "It takes a good trap to capture a bear."
## [6] "Much of the story makes good sense."
## [7] "The price is fair for a good antique clock."
## [8] "The water in this well is a source of good health."
## [9] "The team with the best timing looks good."
## [10] "To send it now in large amounts is bad."
## [11] "A good book informs of what we ought to know."
## [12] "It was a bad error on the part of the new judge."

```

```
# replace "bad" with "horrible" and "good" with "great"
sentences_replaced <- str_replace_all(sentences, "bad", "horrible")
sentences_replaced <- str_replace_all(sentences_replaced, "good", "great")
# print the sentences that had replacements
print(sentences_replaced[sentence_numbers])
```

```
## [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] "The team with the best timing looks great."
## [10] "To send it now in large amounts is horrible."
## [11] "A great book informs of what we ought to know."
## [12] "It was a horrible error on the part of the new judge."
```