hw 6

Jacqueline Nguyen

2024-06-01

question 1:

```
words <- function(w) {
    w[nchar(w) > 0]
}

keep_words <- function(words) {
  words[nchar(words) > 0]
}

is_special_ending <- function(ending) {
  is_es <- all(ending == c("e", "s"))
  is_ed <- all(ending == c("e", "d"))
  is_e_not_le <- ending[2] == "e" & ending[1] != "l"
  is_es | is_ed | is_e_not_le
}

is_vowel <- function(letter) {
  letter %in% c("a", "e", "i", "o", "u", "y")
}

count_syllables <- function(word) {
  word_letters <- unlist(strsplit(word, split = ""))
  if (length(word_letters) <= 3) {
    1
  } else {
    word_letters <- rm_special_endings(word_letters)
    word_vowels <- is_vowel(word_letters)
    sum(word_vowels) - sum(diff(which(word_vowels)) == 1)
    }
}

rm_special_endings <- function(word_letters) {
  word_tail <- tail(word_letters, n = 2)
  if (is_special_ending(word_tail)) {
    if (word_tail[2] == "e") {
      word_letters[-length(word_letters)]
} else {
  head(word_letters, n = -2)
}
} else {
  word_letters
  }
}

reading_ease <- function(text) {
  #counting sentences:
  sentences <- tolower(strsplit(text, split = "[.!?:;]")[[1]])
  sentences <- gsub(pattern = "[[:punct:]]", replacement = "", sentences)
  total_sentences <- length(sentences)

  #counting words
  words <- unlist(strsplit(sentences, split = " "))
  words <- lapply(words, keep_words)
```

```
  total_words <- sum(unlist(lapply(words, length)))

  #counting syllables
  syllables <- lapply(words, count_syllables)
  total_syllables <-  sum(unlist(lapply(syllables, length)))

  RE <- 206.835 - (1.015 * (total_words/total_sentences)) - (84.6*(total_syllables/total
_words))

  RE
}
waffles_vec <-
c("We need to remember what's important in life: friends, waffles, work.",
"Or waffles, friends, work.",
"Doesn't matter, but work is third."
)

reading_ease(waffles_vec)
```

```
## [1] 108.9616
```

question 2:

```
my_nchar <- function(x) {
   chars <- strsplit(x, split = "")
   chars
   vapply(chars, length, numeric(1))
}
```

question 3:

```
my_strrep<- function(vec,times){
    if(length(times)>length(vec)){
        newVec<-rep(NA,length(times))
        i <- 1
        for(idx in 1:length(times)){
            j <- times[idx]
            string <- vec[i]
            term <- string
            l <- 1
            while(l < j){
                string <- paste(string,term,sep="")
                l <- l + 1
            }
            newVec[idx] <- string
            i <- i + 1
            if (i > length(vec)){
                i <- 1;
            }
        }
        newVec
    }else{
        newVec<-rep(NA,length(vec))
        i <- 1
        for(idx in 1:length(vec)){
            j <- times[i]
            string <- vec[idx]
            term <- string
            l <- 1
            while(l < j){
                string <- paste(string,term,sep="")
                l <- l+1
            }
            newVec[idx] <- string
            i <- i+1
            if (i > length(times)){
                i <- 1;
            }
        }
        newVec
    }
}

x <- c("a", "b", "c")
times <- c(2, 3, 5, 1)
result <- my_strrep(x, times)
result
```

```
## [1] "aa"    "bbb"   "ccccc" "a"
```

question 4:

```
#`%is_in%` <- function(x,table) {
#  apply(x, any(x==table), logical(length(x)))
#}
```

question 5:

```
# my_gregexpr <- function(pattern, text) {
#  if()
#}
```