cryptography

Max Lang

10/27/2022

find_palindrome

This function returns TRUE if a given string is a palindrome, FALSE otherwise.

Input

- `input`: A `character(1)`.

Output

- Logical vector of length one

Code

```
is_palindrome <- function(input) {
  assertString(input)
  str <- strsplit(toupper(input), "")[[1]]
  str <- str[str %in% LETTERS]
  all(str == rev(str))
}
```

Worked example

```
is.a.palindrome <- "Was it a car or a cat I saw?"
not.a.palindrom <- "Hello, what is your name?"
is_palindrome(is.a.palindrome)
```

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

```
is_palindrome(not.a.palindrom)
```

```
## [1] FALSE
```

caesar_cipher

This function can encrypt and decrypt plain text using the caesar cyper.

Input

- plaintext: Text to encrypt / decrypt
- key: Letter used as the key.

- decrypt: Set true if you want to decrypt encrypted text

Output

Encrypted/decrypted text

Code

```
caesar_cipher <- function(plaintext, key, decrypt = FALSE) {
  assertString(plaintext, pattern = "^[A-Z ]*$")
  assertString(key, pattern = "^[A-Z ]$")
  assertLogical(decrypt)

  code <- c(" ", LETTERS)
  plaintext <- strsplit(plaintext, "")[[1]]
  decryptmultiplier <- if (decrypt) -1 else 1

  numbers <- (match(plaintext, code) - 1) + (match(key, code) - 1) * decryptmultiplier
  numbers <- numbers %% 27
  paste(code[numbers + 1], collapse = "")
}
```

Worked example (encrypt)

```
encrypted.text <- caesar_cipher("HELLO THIS IS A VERY SECRET MESSAGE",
                                key = "L",
                                decrypt = FALSE)

encrypted.text
```

```
## [1] "TQXX LETUDLUDLMLGQCJLDQOCQELYQDDMSQ"
```

Worked example (decrypt)

```
caesar_cipher(encrypted.text, key = "L", decrypt = TRUE)
```

```
## [1] "HELLO THIS IS A VERY SECRET MESSAGE"
```

Breaking the caesar cypher

We will use letter frequencies based on this Wiki article https://en.wikipedia.org/wiki/Letter_frequency.

```
letterfrequencies <- 1 / 100 * c(
  A = 6.756, B = 1.234, C = 2.302, D = 3.518, E = 10.508, F = 1.843, G = 1.667,
  H = 5.041, I = 5.763, J = 0.127, K = 0.639, L = 3.330, M = 1.990, N = 5.583,
  O = 6.210, P = 1.596, Q = 0.079, R = 4.953, S = 5.234, T = 7.492, U = 2.282,
  V = 0.809, W = 1.952, X = 0.124, Y = 1.633, Z = 0.061, ` ` = 17.272)
```

text_log_likelihood

This function estimates the log-likelihood that a given text is, in fact, a non-encrypted plain text, using the distribution of letters.

Input

- text: A `character(1)` string made up of upper case letters and space

Output

A scalar `numeric` giving the log likelihood of a given text.

Code

```
text_log_likelihood <- function(text) {
  # your code
  assertString(text, pattern = "^[A-Z ]+$")
  sum(log(letterfrequencies[strsplit(text, "")[[1]]]))
}
```

Worked Example

```
text_log_likelihood("HI WHAT IS UP")
```

```
## [1] -37.04098
```

estimate_key

This function estimates the most likely key for a given ciphertext. This is the key that generates a text that is most likely according to `text_log_likelihood`. The possible keys are the 26 letters as well as the space (`" "` – this one does not change the text).

Input

- `ciphertext`: A `character(1)` string made up of upper case letters and space

Output: a list with two entries:

- `key`: `character(1)` giving an upper case letter or space.
- `log.likelihood`: `numeric(1)` giving the log likelihood of the text when decrypting with this key.

Code

```
estimate_key <- function(ciphertext) {
  assertString(ciphertext, pattern = "^[A-Z ]+$")
  keys <- c(LETTERS, " ")
  result <- vapply(keys, function(k) {
    text_log_likelihood(caesar_cipher(ciphertext, k, TRUE))
  }, numeric(1))
  list(key = names(which.max(result)), log.likelihood = max(result))
}
```

Worked example

```
cipher <- caesar_cipher("HELLO THIS IS A SECRET MESSAGE TO CAESAR", key = "B", decrypt = FALSE)
estimate_key(cipher)
```

```
## $key
## [1] "B"
##
## $log.likelihood
## [1] -107.673
```

break_caesar

This function uses the previous function to break the caesar cipher.

Input:

- A character vector of length one containing an (caesar) encrypted text with unknown key.

Output:

Hopefully the correct message.

Code

```
break_caesar <- function(cipher) {
  assertCharacter(cipher, any.missing = FALSE, min.len = 1)
  infos <- do.call(rbind, lapply(cipher, function(x) as.data.frame(estimate_key(x))))
  needle <- which.max(infos$log.likelihood)
  caesar_cipher(cipher[[needle]], infos$key[[needle]], decrypt = TRUE)
}
```

Worked example

```
ciphertext
```

```
## [1] "BUQCLXYAUPLYIXPIXUPAUWYDCPLXYSXPXUPXQTPLYIXPXYBPQCTPIXUPHDATYUGHPLXYSXPXQTPQHHUBRAUTPVGDBPIXUPE
```

```
break_caesar(ciphertext) # The passage is from The Gallic Wars by Julius Caesar
```

```
## [1] "MEANWHILE WITH THE LEGION WHICH HE HAD WITH HIM AND THE SOLDIERS WHICH HAD ASSEMBLED FROM THE PI
```