Challenge-5

Nicole Lim

2023-09-10

Questions

Question-1: Local Variable Shadowing Create an R function that defines a global variable called `x` with a value of 5. Inside the function, declare a local variable also named `x` with a value of 10. Print the value of `x` both inside and outside the function to demonstrate shadowing.

Solutions:

```
# Enter code here
define_x <- function(x) {
    x <<- 5
    x <- 10
    print(x)
}

define_x(x)
```

```
## [1] 10
```

```
print(x)
```

```
## [1] 5
```

Question-2: Modify Global Variable Create an R function that takes an argument and adds it to a global variable called `total`. Call the function multiple times with different arguments to accumulate the values in `total`.

Solutions:

```
# Enter code here
totaller <- function(initial = 0, amount_to_add, total) {

    total <- (initial + amount_to_add)
    total <<- total
    print(total)
    # initial <<- total
}

totaller(amount_to_add = 2)
```

```
## [1] 2
```

```
totaller(initial = total, amount_to_add = 4)
```

```
## [1] 6
```

```
totaller(initial = total, amount_to_add = 6)
```

```
## [1] 12
```

```
print(total)
```

```
## [1] 12
```

Question-3: Global and Local Interaction Write an R program that includes a global variable `total` with an initial value of 100. Create a function that takes an argument, adds it to `total`, and returns the updated `total`. Demonstrate how this function interacts with the global variable.

Solutions:

```
# Enter code here
totaller_2 <- function(total, number_to_add_2) {

    total <<- 100
    total <- (total + number_to_add_2)
    print(total)
}

totaller_2(total = total, number_to_add_2 = 27)
```

```
## [1] 127
```

```
# This argument interacts with the global variable by
# drawing from its global value in the execution of the
# function.
```

Question-4: Nested Functions Define a function `outer_function` that declares a local variable `x` with a value of 5. Inside `outer_function`, define another function `inner_function` that prints the value of `x`. Call both functions to show how the inner function accesses the variable from the outer function's scope.

Solutions:

```
# Enter code here
outer_function <- function(x) {

    x <- 5
    inner_function <- function(x) print(x)
    inner_function(x)
}

outer_function(x)
```

```
## [1] 5
```

Question-5: Meme Generator Function Create a function that takes a text input and generates a humorous meme with the text overlaid on an image of your choice. You can use the `magick` package for image manipulation. You can find more details about the commands offered by the package, with some examples of annotating images here: https://cran.r-project.org/web/packages/magick/vignettes/intro.html

Solutions:

```
# Enter code here
library(magick)
```

```
## Linking to ImageMagick 6.9.12.93
## Enabled features: cairo, fontconfig, freetype, heic, lcms, pango, raw, rsvg, webp
## Disabled features: fftw, ghostscript, x11
```

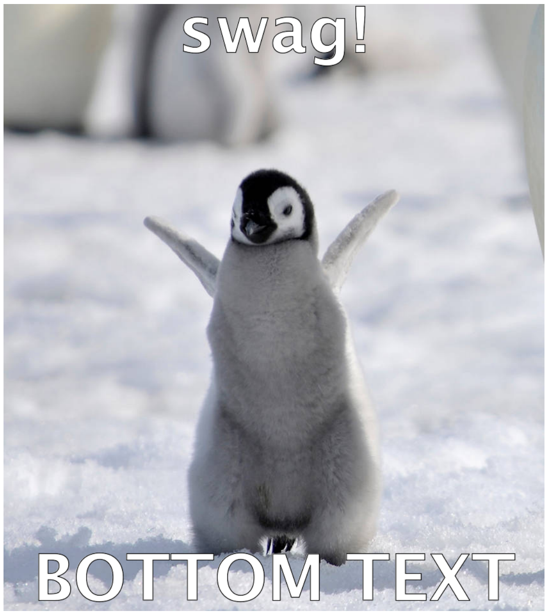
```
memeify <- function(meme_uppertext, meme_lowertext) {

    meme_image <- image_read("jingxiang2.jpeg")
    meme1 <- image_annotate(meme_image, meme_uppertext, size = 100,
        gravity = "north", color = "white", strokecolor = "black")
    meme2 <- image_annotate(meme1, meme_lowertext, size = 100,
        gravity = "south", color = "white", strokecolor = "black")
    print(meme2)

}

memeify("swag!", "BOTTOM TEXT")
```

```
##   format width height colorspace matte filesize density
## 1   JPEG   844    947       sRGB  TRUE        0 300x300
```



Question-6: Text Analysis Game Develop a text analysis game in which the user inputs a sentence, and the R function provides statistics like the number of words, characters, and average word length. Reward the user with a “communication skill level” based on their input.

Solutions:

```
# Enter code here
suppressMessages(library("tidyverse"))
library(tidyverse)
library(magick)
sentencealyser <- function(sentence) {

    stats_tibble <- tibble(number_of_words = lengths(strsplit(sentence,
        " ")), characters = nchar(sentence), average_word_length = characters/number_of_words)

    # print(stats_tibble)

    number_of_words <- stats_tibble$number_of_words
    characters <- stats_tibble$characters
    average_word_length <- stats_tibble$average_word_length

    wordiness = case_when(number_of_words > 20 ~ 3, number_of_words >
        10 ~ 2, number_of_words > 5 ~ 1, number_of_words < 5 ~
        0)
    characteriness = case_when(characters > 60 ~ 3, characters >
        40 ~ 2, characters > 20 ~ 1, characters < 20 ~ 0)
    lengthiness = case_when(average_word_length > 8 ~ 3, average_word_length >
        6 ~ 2, average_word_length > 4 ~ 1, average_word_length <
        4 ~ 0)

    # print(characteriness) print(wordiness)
    # print(lengthiness)

    eloquence = (characteriness + lengthiness + wordiness)

    # print(eloquence)

    comm_skill_level = case_when(eloquence > 7 ~ 3, eloquence >
        4 ~ 2, eloquence > 1 ~ 1, eloquence < 1 ~ 0, )

    replace_na(comm_skill_level, 0)

    congratulations <- paste0("Congratulations! Your communication skill level score is ",
        comm_skill_level, " out of 3!")
    medal <- image_read("medal.jpeg")
    medal_engraved <- image_annotate(medal, paste0("Skill Level: ",
        comm_skill_level), size = 30, gravity = "center", color = "black",
        location = "+0+50")

    print(congratulations)
    print(medal_engraved)

}

sentencealyser("It is a truth universally acknowledged, that a single man in possession of a good fortu
```

```
## [1] "Congratulations! Your communication skill level score is 2 out of 3!"
## # A tibble: 1 x 7
##   format width height colorspace matte filesize density
##   <chr>  <int>  <int> <chr>      <lgl>    <int> <chr>
## 1 JPEG     370    466 sRGB       TRUE         0 300x300
```