

Challenge-5

Elise Wong

2023-09-11

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:

```
# Define x as a global variable with a value of '5',
x <- 5

# Create a function,
demonstrate_shadowing <- function(x) {
  x <- 10
  return(x)
}

# Print the value of `x` inside the function,
demonstrate_shadowing()
```

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

```
# Print the value of `x` outside the function,
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:

```
# Create 'total' as a global variable,
total <- 0

# Create a function to accumulate values,
add_argument <- function(argument) {
  total <- total + argument
}

# To call the function multiple times with different arguments,
add_argument(argument = 2)
add_argument(argument = 5)
add_argument(argument = 8)
add_argument(argument = 25)

# To print the accumulated total,
print(total)
```

```
## [1] 40
```

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:

```
# Create global variable,
total <- 100

# Create function,
updated_total <- function(argument) {
  total <- total + argument
  return(total)
}

# To demonstrate how this function interacts with the global variable,
updated_total(15) ## <- within the function, the total increases by argument = 15
```

```
## [1] 115
```

```
total ## <- however, the global variable remains unaffected.
```

```
## [1] 100
```

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:

```
# Define the functions,
outer_function <- function() {
  x <- 5
  inner_function <- function() {
    x
  }

# Call both functions,
  inner_function()
}

outer_function()
```

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

We call the inner function within the outer function to demonstrate that the inner function can access variables from its parent function's scope. Upon calling the outer function outside the nested functions, it demonstrates the relation between outer ('parent') and inner function - the inner function accesses 'x' from the outer function's scope.

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> (<https://cran.r-project.org/web/packages/magick/vignettes/intro.html>)

Solutions:

```

# I put eval=FALSE as the document would not knit to HTML or PDF without a valid 'i
image_path'.
library(magick)

generate_meme <- function(text_top, text_bottom, image_path, output_path) {

# For the image,
  uploaded_image <- image_read(image_path)
  updated_image <- image_charcoal(uploaded_image)

# For text,
  half_image <- image_annotate(updated_image, text_top, font = 'Comic Sans', size =
60, gravity = "north", color = "red")
  final_image <- image_annotate(half_image, text_bottom, font = 'Comic Sans', size
= 60, gravity = "south", color = "red")

# To generate the meme,
  image_write(final_image, path = output_path)
}

print(paste("Welcome to the 'charcoal' meme creator!"))
image_path <- readline("Insert the link to your image here: ")
text_top <- readline("What do you want the top line of your meme to say? Write here
: ")
text_bottom <- readline("What do you want the bottom line of your meme to say? Writ
e here: ")
output_path <- readline("Name your file here: ")
final_output_path <- paste0(output_path, ".png")

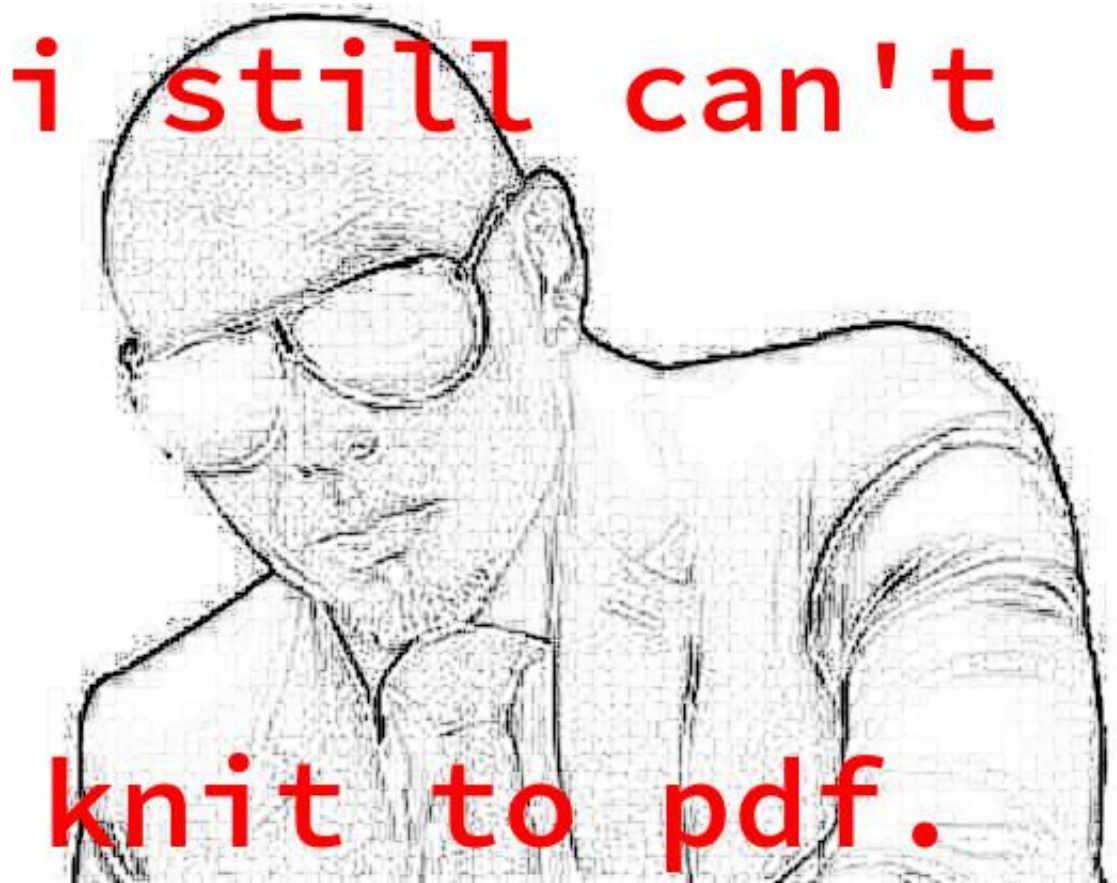
generate_meme(text_top, text_bottom, image_path, final_output_path)
print(paste0("Your meme has been created and saved as ", final_output_path, "."))

```

```

# This is evidence of a successful attempt.
library(knitr)
knitr::include_graphics("https://drive.google.com/uc?export=download&id=12ioYzeP8zU
tfFEyeotZ9gz8EWhIDXGM1")

```



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
text_analysis_game <- function(x) {
  word_number <- length(strsplit(x, " ")[[1]])
  character_number <- nchar(x)
  average_length <- character_number / word_number

  communication_skill_level <- ifelse(average_length <= 5, "Bronze",
                                     ifelse(average_length <= 8, "Silver", "Gold"))

  print(paste0("Thank you for playing the word game, ", user_name, "!"))
  print(paste0("Your total number of words is ", word_number, "."))
  print(paste0("Your total number of characters is ", character_number, "."))
  print(paste0("The average word length of your sentence is ", average_length, "."))
}

print(paste0("Your overall communication skill level is ", communication_skill_level, "."))
}
```

```
user_name <- readline("Please enter your name: ")
```

```
## Please enter your name:
```

```
user_input <- readline("Please enter a sentence: ")
```

```
## Please enter a sentence:
```

```
text_analysis_game(user_input)
```

```
## [1] "Thank you for playing the word game, !"  
## [1] "Your total number of words is 0."  
## [1] "Your total number of characters is 0."  
## [1] "The average word length of your sentence is NaN."  
## [1] "Your overall communication skill level is NA."
```

<br