# Challenge-5: Solutions

Narayani Vedam

2023-09-15

**Welcome!** Hope you have watched the lecture videos and followed the instructions in code-along. Go through the steps described below, *carefully*. It is totally fine to get stuck - **ASK FOR HELP**; reach out to your friends, TAs, or the discussion forum on Canvas.

Here is what you have to do,

1. **Pair** with a neighbor and work
2. **Download** the `Challenge-5.pdf` and `Challenge-5.Rmd` from Canvas
3. **Move** the downloaded files to the folder, "Week-5" that you created previously
4. **Set** it as the working directory
5. **Edit** content in `Challenge-5.Rmd` wherever indicated following instructions in `Challenge-5.pdf`
6. **Remember** to set `eval=TRUE` in the code chunk to generate the output
7. **Ensure** that `echo=TRUE` so that the code is rendered in the final document
8. **Code output** may not be required in all cases, use your discretion
9. **Inform** the tutor/instructor upon completion
10. **Submit** the document on Canvas after they approve
11. **Attendance** will be marked only after submission
12. Once again, **do not hesitate** to reach out to the tutors/instructor, if you are stuck

# 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
x <- 5

demo <- function() {
  x <- 10
  print(x) # Prints 10 (local variable)
}

demo()
```

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

```
print(x) # Prints 5 (global variable)
```

```
## [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:** Notice the use of `<<-` inside the function to affect the value of the global variable.

```
# Enter code here
total <- 0

add_to_total <- function(value) {
  total <<- total + value
}

# Example usage
add_to_total(5)
add_to_total(7) # Here, the value of total is 5
print(total) # Prints 12 because the we had changed value of total from 0 to 5 and then added 7
```

```
## [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:** Notice the use of `<<-` inside the function to affect the value of the global variable

```
# Enter code here
total <- 100

add_to_total <- function(value) {
  total <<- total + value
  return(total)
}

new_total <- add_to_total(20)
print(new_total) # Prints 120
```

```
## [1] 120
```

### 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 <- 5

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

  inner_function()
}

outer_function()
```

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

```
outer_function()
```

```
## [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:** The outline of the code is below. They have the freedom to use any image etc.

```
# Enter code here
library(magick)

generate_meme <- function(text, image_path) {
  meme <- image_read(image_path)
  meme <- image_annotate(meme, text, gravity = "south", size = 50)
  image_write(meme, path = "output_meme.png")
}

# Example usage:
generate_meme("When your code finally works", "meme_image.jpg")
```

### 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(input_sentence) {
  cat("Welcome to the Text Analysis Game!\n")
  sentence <- input_sentence

  words <- unlist(strsplit(sentence, " "))
  num_words <- length(words)
  num_characters <- nchar(sentence)
  avg_word_length <- num_characters / num_words

  cat("Statistics:\n")
  cat("Number of Words:", num_words, "\n")
  cat("Number of Characters:", num_characters, "\n")
  cat("Average Word Length:", round(avg_word_length, 2), "\n")

  skill_level <- ifelse(avg_word_length >= 5, "Advanced", "Beginner")
  cat("Your Communication Skill Level:", skill_level, "\n")
}

text_analysis_game("Hi there!")
```

```
## Welcome to the Text Analysis Game!
## Statistics:
## Number of Words: 2
## Number of Characters: 9
## Average Word Length: 4.5
## Your Communication Skill Level: Beginner
```