Challenge-5

Heng Javier

13/09/2023

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:

```r
# Enter code here
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.3     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.0
## v ggplot2   3.4.3     v tibble    3.2.1
## v lubridate 1.9.2     v tidyr     1.3.0
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```r
library(dplyr)
x <-5
double <- function(x){
  x*2
}
print(x)
```

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

```r
print(double(x))
```

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

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:

```r
# Enter code here
data <- 5
add_to <- function(total){
  sum(data)
}
add_to(total)
```

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

```r
data <- c(2,4,1,7)
add_to <- function(total){
  sum(data)
}
add_to(total)
```

```
## [1] 14
```

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:

```r
# Enter code here
total <- 100
add<-30

new_total <- function(total,add){
  return(total+add)
}
total <- new_total(total,add)
total
```

```
## [1] 130
```

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:

```r
# Enter code here


outer_function <- function(){
  x<-5
  inner_function <- function(){
  sprintf("x is %d",x)
  }
  inner_function()
}


outer_function()
```

```
## [1] "x is 5"
```

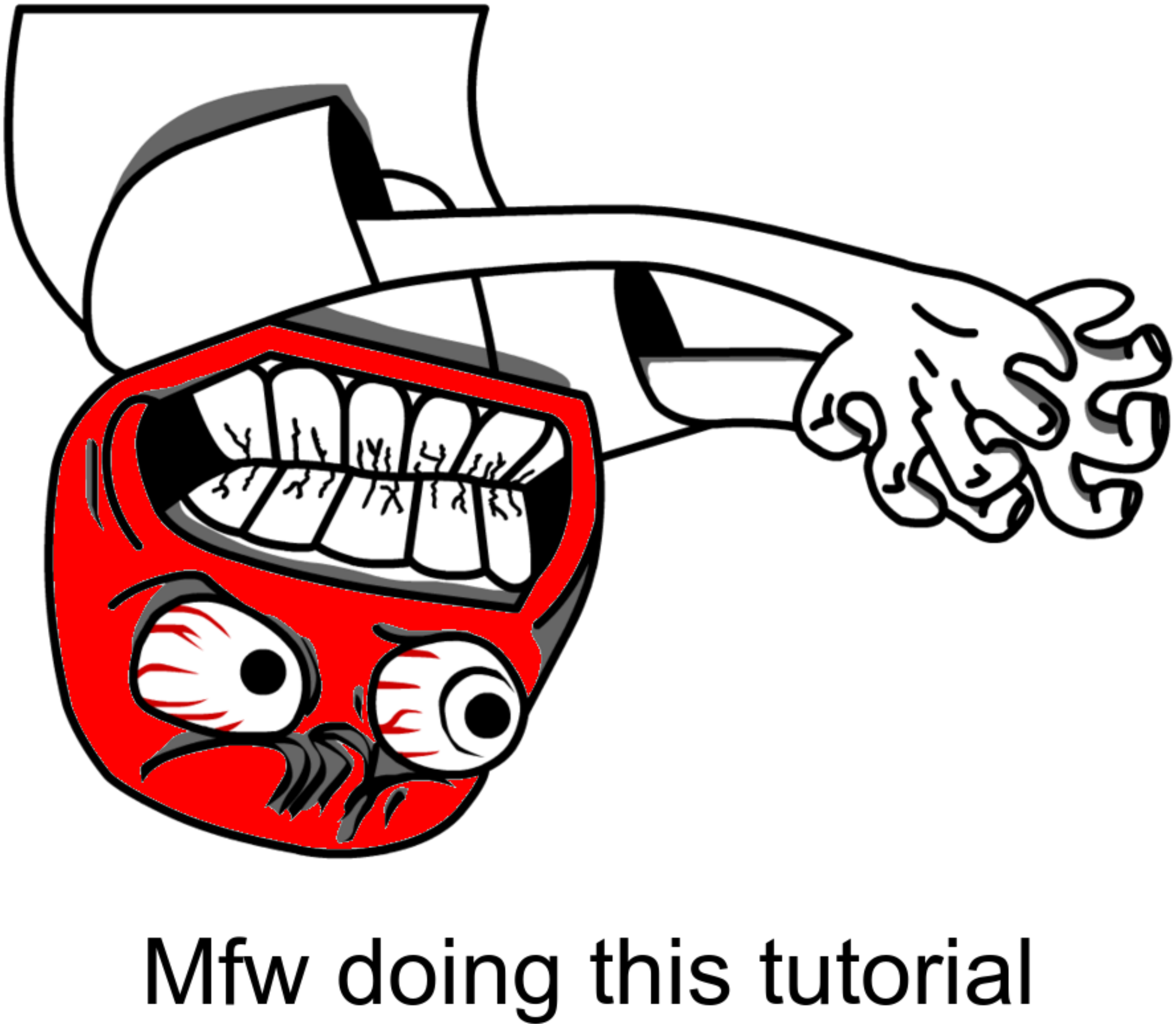
Question-5: Meme Generator Function Create a function that takes a text input and generates a humorous meme with the text overlaid on an image of your choice. You can use the `magick` package for image manipulation. You can find more details about the commands offered by the package, with some examples 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, freetype, fftw, ghostscript, heic, lcms, pango, raw, rsvg, webp
## Disabled features: fontconfig, x11
```

```
angry_man <- image_read("https://freepngimg.com/download/internet_meme/12-2-angry-meme-transparent-png.
Meme <- function(angry_man){
  angry_man<-image_flip(image_fill(angry_man, "red", point = "+100+200", fuzz = 20))
  image_annotate(angry_man, "Mfw doing this tutorial", size = 70, gravity = "south", color = "black")
}
Meme(angry_man)
```



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:

```r
# Enter code here
library(stringi)
sentence<- "I love R Studio!"
Game<-function(sentence){
  print("Total characters used:")
  print(nchar(sentence)-stri_count_words(sentence)+1)
  print("Number of words:")
  print(stri_count_words(sentence))
  print("Average word length:")
  Mean_characters<-(nchar(sentence)-stri_count_words(sentence)+1)/stri_count_words(sentence)
  print(Mean_characters)
  if(Mean_characters>5){
    print("You are a great communicator!")
  } else if(Mean_characters>1){
    print("You can do better than that!")
  } else ("You are probably a toddler")
}
Game(sentence)
```

```
## [1] "Total characters used:"
## [1] 13
## [1] "Number of words:"
## [1] 4
## [1] "Average word length:"
## [1] 3.25
## [1] "You can do better than that!"
```

```
sentence<- "Lorem ipsum dolor sit amet. Et rerum galisum quo ducimus accusamus qui pariatur mollitia. Ec

Qui nobis quibusdam sit exercitationem dolore ut rerum Quis eos blanditiis dolores! Ea maxime voluptatem

Et architecto autem qui molestias odit qui itaque consequatur. Qui officia voluptatum vel dignissimos do

Game(sentence)
```

```
## [1] "Total characters used:"
## [1] 550
## [1] "Number of words:"
## [1] 95
## [1] "Average word length:"
## [1] 5.789474
## [1] "You are a great communicator!"
```

```
sentence<- "u r a b f p o s"

Game(sentence)
```

```
## [1] "Total characters used:"
## [1] 8
## [1] "Number of words:"
## [1] 8
## [1] "Average word length:"
## [1] 1

## [1] "You are probably a toddler"
```