Week-5: Code-along

Deng Chunxi

2023-09-13

II. Code to edit and execute using the Code-along.Rmd file

A. Writing a function

1. Write a function to print a “Hello” message (Slide #14)

```
# Enter code here
name <- "Chunxi"
say_hello_to <- function(name) {
print(paste0("Hello ", name, "!"))
}
```

2. Function call with different input names (Slide #15)

```
# Enter code here
say_hello_to('Chunxi')
```

```
## [1] "Hello Chunxi!"
```

3. typeof primitive functions (Slide #16)

```
# Enter code here
typeof(`+`)
```

```
## [1] "builtin"
```

4. typeof user-defined functions (Slide #17)

```
# Enter code here
typeof(say_hello_to)
```

```
## [1] "closure"
```

5. Function to calculate mean of a sample (Slide #19)

```
# Enter code here
calc_sample_mean <- function(sample_size) {
mean(rnorm(sample_size))
}
```

6. Test your function (Slide #22)

```
# With one input
calc_sample_mean(1000)
```

```
## [1] -0.03825892
```

```
# With vector input
calc_sample_mean(c(100, 300, 3000))
```

```
## [1] -0.5870522
```

7. Customizing the function to suit input (Slide #23)

```
# Enter code here
```

8. Setting defaults (Slide #25)

```
# First define the function
calc_sample_mean <- function(sample_size,
our_mean=0,
our_sd=1) {
sample <- rnorm(sample_size,
mean = our_mean,
sd = our_sd)
mean(sample)
}
# Call the function
calc_sample_mean(sample_size = 10)
```

```
## [1] 0.3500892
```

9. Different input combinations (Slide #26)

```
# Enter code here
calc_sample_mean(10, our_sd = 2)
```

```
## [1] -0.02311989
```

```
calc_sample_mean(10, our_mean = 6)
```

```
## [1] 5.928345
```

10. Different input combinations (Slide #27)

```
# set error=TRUE to see the error message in the output
# Enter code here
calc_sample_mean(our_mean = 5)
```

```
## Error in rnorm(sample_size, mean = our_mean, sd = our_sd): argument "sample_size"
is missing, with no default
```

11. Some more examples (Slide #28)

```
# Enter code here
add_two <- function(x) {
x+2
}
add_two(4)
```

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

B. Scoping

12. Multiple assignment of z (Slide #36)

```
# Enter code here
z <- 1
sprintf("The value assigned to z outside the function is %d",z)
```

```
## [1] "The value assigned to z outside the function is 1"
```

```
foo <- function(z = 2) {
z <- 3
return(z+3)
}
foo()
```

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

13. Multiple assignment of z (Slide #37)

```
# Enter code here
z <- 1
foo <- function(z = 2) {
z <- 3
return(z+3)
}
foo(z = 4)
```

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

```
sprintf("The final value of z after reassigning it to a different value inside the fu
nction is %d",z)
```

```
## [1] "The final value of z after reassigning it to a different value inside the fun
ction is 1"
```