

Week-5: Code-along

Ariel Quek Xuan

2023-09-11

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)

```
# Function to print "Hello" message  
say_hello_to <- function(name) {print(paste0("Hello", name, "!"))}
```

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

```
# Testing different input names  
say_hello_to(" Kashif")
```

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

```
#Testing different input names  
say_hello_to(" Sach")
```

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

3. typeof primitive functions (Slide #16)

```
typeof(sum)
```

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

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

```
typeof(say_hello_to)
```

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

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

```
calc_sample_mean <- function(sample_size) {mean(rnorm(sample_size))}
```

6. Test your function (Slide #22)

```
calc_sample_mean(1000)
```

```
## [1] -0.01781874
```

```
calc_sample_mean(c(100,300,3000))
```

```
## [1] 0.1257216
```

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

```
library(tidyverse)
```

```
## Warning: package 'tidyverse' was built under R version 4.2.3
```

```
## Warning: package 'ggplot2' was built under R version 4.2.3
```

```
## Warning: package 'tibble' was built under R version 4.2.1
```

```
## Warning: package 'tidyr' was built under R version 4.2.2
```

```
## Warning: package 'readr' was built under R version 4.2.3
```

```
## Warning: package 'purrr' was built under R version 4.2.2
```

```
## Warning: package 'dplyr' was built under R version 4.2.2
```

```
## Warning: package 'stringr' was built under R version 4.2.2
```

```
## Warning: package 'forcats' was built under R version 4.2.3
```

```
## Warning: package 'lubridate' was built under R version 4.2.3
```

```
## — Attaching core tidyverse packages ————— tidyverse 2.0.0 —
## ✓ dplyr      1.1.0      ✓ readr      2.1.4
## ✓ forcats    1.0.0      ✓ stringr    1.5.0
## ✓ ggplot2    3.4.3      ✓ tibble     3.1.8
## ✓ lubridate  1.9.2      ✓ tidyr      1.3.0
## ✓ purrr      1.0.1
## — Conflicts ————— tidyverse_conflicts() —
## ✗ dplyr::filter() masks stats::filter()
## ✗ dplyr::lag()     masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to be
come errors
```

```
#create vector to test function
sample_tibble <- tibble(sample_sizes=c(100,300,3000))
sample_tibble %>%
  mutate(sample_means = calc_sample_mean(sample_sizes))
```

```
## # A tibble: 3 × 2
##   sample_sizes sample_means
##   <dbl>         <dbl>
## 1     100         0.350
## 2     300         0.350
## 3    3000         0.350
```

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] -0.2393486
```

9. Different input combinations (Slide #26)

```
calc_sample_mean(10,our_sd = 2)
```

```
## [1] 0.7948965
```

10. Different input combinations (Slide #27)

```
#Adding 2 to the input x  
add_two <- function(x) {x+2}  
add_two(4)
```

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

11. Some more examples (Slide #28)

```
add_two(-34)
```

```
## [1] -32
```

```
add_two(5.784)
```

```
## [1] 7.784
```

B. Scoping

12. Multiple assignment of z (Slide #36)

```
#Initialize Z  
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){  
  #reassigning Z  
  z <- 3  
  return (z+3)  
}  
foo()
```

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

13. Multiple assignment of z (Slide #37)

```
#Another reassignment of z  
foo(z=4)
```

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