Data Handling: Programming with R

Aurélien Sallin

2023-09-28

Exercise 1: Matrices and Matrix Indexing

Objective: Familiarize yourself with creating matrices and matrix indexing in R.

Task 1.1: Create a Matrix

Create a 3x3 matrix named my_matrix with the following values:

1 2 4

9.9 5 6

7 1/3 9

Task 1.2: Indexing

Using indexing, perform the following tasks:

- Extract the second row of my_matrix
- Extract the value in the third row and first column of my_matrix
- Replace the middle value of fun_matrix with 10.

Task 1.3: Sum it up!

With a for-loop, loop through each column of my_matrix and compute the sum as well as the mean of each column.

Advanced: repeat the task with apply

Exercise 2: For-loop and While-loop

Objective: Practice using for-loop and while-loop to iterate through sequences and perform operations.

Task 2.1: Looping Through Colors

Given a vector of colors: colors <- c(“red”, “green”, “blue”), write a for-loop to print each color on a new line.

```
colors <- c("red", "green", "blue")
```

Task 2.2: Check colors

Given the vector of colors defined above, run a for-loop that runs through the vector and checks whether the color corresponds to your favorite color. The loop returns TRUE if it corresponds to your favorite color, and FALSE otherwise. Hint: define your favorite color in a variable.

Task 2.3: Countdown Timer

Write a while-loop that counts down from 5 to 1, printing each number with the message “Counting down: number!”. After reaching 1, print “Blast off!”.

Task 2.3: Yoloop

Write a for-loop that iterates over numbers 1 to 15. If a number is divisible by 3, print “Yo”. If it’s divisible by 5, print “No-Yo”. If it’s divisible by both 3 and 5, print “YOLO”. Otherwise, print the number. **Hint**: use the `%%` modulo and the integer division `%/%` presented in the course to solve this exercise.

Exercise 3: Functions

Objective: Develop skills in creating and using functions in R.

Task 3.1: Square and Cube

Create a function named square_cube that takes a number as input and returns a list containing its square and cube.

Task 3.1: Data Frame Processing

Write a function named process_matrix that takes a numeric matrix as input and returns a matrix with each value squared.

```
my_matrix <- cbind(c(9,9,9), c(2,1,4), c(8,4,2))
my_matrix2 <- cbind(c(81, 81, 81), c(4,1,16), c(64,16,4))

# Test (remove the comment to use this code)
# process_matrix(my_matrix) == my_matrix2
```