R for Psychology Research

Week 2 - Exercises

Marcus Lindskog

1. `if...else`, `for` - and `while` - loops, and Functions

1. With, `i <- 1`, write a `while` loop that prints the odd numbers from 1 through 7.
2. Using the following variables:

   `msg <- c("Hello")`; `i <- 1`

   Write a `while` loop that increments the variable, `i`, 6 times, and prints `msg` at every iteration.
3. Write a `for` loop that prints the first four numbers of this sequence: `x <- c(7, 4, 3, 8, 9, 25)`
4. Write a `for` loop that prints all the letters in `y <- c("q", "w", "e", "r", "z", "c")`.
5. Using `i <- 1`, write a `while` loop that prints the variable, `i`, and uses `break` to exit the loop if `i` equals 3.
6. Write a nested loop, where the outer `for` loop increments `a` 3 times, and the inner `for`m loop increments `b` 3 times. The nested loop should print the values of variables, `a` and `b`.
7. Write a `while` loop that prints the variable, `i`, that is incremented from 2 – 5, and uses the `next` statement, to skip the printing of the number 3.
8. Write a `for` loop that uses next to print all values except 3 in the following variable: `i <- 1:5`
9. Write a `for` loop to find the sum of the first 10 natural numbers (i.e., 1-10).
10. Write a `for` loop that prints the cube of the 20 first natural numbers in the console.
11. Write a `for` loop and a `while` loop that both print the first 10 natural numbers (i.e. 1-10) to the console.
12. Write a script that uses loops and `if...else` to print the first 10 even natural numbers to the console.
13. Write a script using `if...else` to check if a number `x` is negative or positive.
14. Write a script using `if...else` to check if a number `x` is negative, positive, or zero.
15. Write a script using `if...else` to check if a number `x` is divisible by 5, 11, or none of the two.
16. Write a script that recodes the vector `c(1,2,3,4,5,6,7,8,9,10)` such that values below 6 are 1 and above 6 are 0.
17. Write a script that recodes the vector `c(1,2,3,4,5,6,7,8,9,10)` such that odd values are 0 and even values are 1.
18. Write a function `both_na()` that takes two vectors of the same length and returns the number of positions that have NA in both vectors. Provide at least three examples if input that test your function.
19. Implement a fizzbuzz function. It should take a single number as input. If the number is divisible by three, it returns `fizz`. If the number is divisible by five, it returns `buzz` and if it is divisible by both three and five, it returns `fizzbuzz`
20. Write a function `length_checker` that takes to vectors `x` and `y` and checks if they are of equal length. The function should return `TRUE` if they are of equal length and `FALSE` if they are not.

2. Examination Exercises

The solution to the exercises in this section should be handed in as a part of the examination. Your solution should be contained in a single R-script that is emailed to `marcus.lindskog@psyk.uu.se`. Your code should be well commented and easy to follow. Answers to any questions below should be written as a comment in the R-script after the code that produces the answer.

1. Write a function `rand_below` that takes a threshold `theta` as the argument and uses a `while` loop to draw a random uniform number in the range $[0, 1]$ (`runif`) until it gets a number below the threshold. Count the number of iterations and return both the final random number and the number of iterations in a vector.

2. Write a function `to_power_of_n` that takes a numeric vector `x` and a number `n` as its arguments and returns a vector that has all the elements of `x` raised to the power of n (i.e. x^n).

3. Implement your own custom function `my_custom_function` that solves a statistical problem (e.g. calculates the skewness of a vector). It is okay to write a function to solve a problem for which there is already an available function in R. Your function should take a vector as its input. Further, your function should take **two** additional arguments that specify the details of the calculation. These arguments should be named and have default values. Finally, your function should contain a `for` loop, a `while` loop, or a `if...else` statement.