

ATIVIDADE PRÁTICA

MAP2112 – Introdução a Lógica de Programação e Modelagem Computacional

Prof. Dr. Luis Carlos de Castro Santos

1º semestre 2019

Os exercícios abaixo são selecionados de:

<https://www.r-exercises.com/start-here-to-learn-r/>

O site contém mais de 4000 exercícios e serve como mais um auxílio ao aprendizado. Alguns comandos podem não ter sido apresentados durante as aulas, aproveite a oportunidade para conhecê-los use o help do R Studio. O site também contém as respostas aos exercícios, mas procure não utilizá-las e tente encontrar a sua solução mesmo que por tentativa e erro. Lembre-se que pode haver mais de uma forma de se obter o mesmo resultado e todas são válidas.

Para cada questão proponha uma solução e execute o código em R para verificar sua resposta.

Loops and Conditional Execution

Conditional Execution

Exercise 1

Create an R script that returns the absolute value of an integer vector x of length one.

Exercise 2

Create an R script that calculates the square root of a given integer vector x of length one, if the value contained in x is negative it should return NA.

Exercise 3

Create an R script that returns the maximum value out of the elements of a numeric vector x of length 2.

Exercise 4

Create an R script that returns TRUE if the elements of a vector x , with length 3, are strictly increasing.

Exercise 5

Create an R script that returns the max value of a vector x with length 3. Don't use the aid of an auxiliary variable.

Exercise 6

Create an R script that returns the amount of values that are larger than the mean of a vector. You are allowed to use `mean()`.

Exercise 7

Create an R script that, given a numeric vector x with length 3, will print the elements by order from high to low.

If else sequences Exercises

Procure seção do help sobre o comando `ifelse` para entender seu funcionamento antes de realizar os exercícios dessa seção.

Exercise 1

What is the output of:

```
ifelse(sqrt(9)<2, sqrt(9), 0)
```

Exercise 2

What is the output of:

```
ifelse(sqrt(100)>9, sqrt(100), 0)
```

Exercise 3

What is the output y of:

```
x=12
if(is.numeric(x)) y=x*2
```

Exercise 4

What is the output x, y of:

```
z=-1
if(z<0){x=abs(z);y=z*3}
```

Exercise 5

What is the output y of:

```
z=6
if(z<0) y=z*3 else y=z*5
```

Exercise 6

What is the output z of:

```
x=15
y=3
if(is.numeric(x))
if(is.numeric(y) & y!=0)
z=x/y
```

Exercise 7

What is the output of:

```
x=letters[20]
if(is.numeric(x)) print('is numeric') else
if(is.character(x)) print('is character')
```

Exercise 8

What is the output of:

```
x=90
ifelse(x<100,ifelse(x/2==trunc(x/2),x/2,0),
ifelse(x/100==trunc(x/100),x/100,-1))
x=120
ifelse(x<100,ifelse(x/2==trunc(x/2),x/2,0),
ifelse(x/100==trunc(x/100),x/100,-1))

x=200
ifelse(x<100,ifelse(x/2==trunc(x/2),x/2,0),
ifelse(x/100==trunc(x/100),x/100,-1))
```

Exercise 9

What is the output n of:

```
z='i'
if(z %in% letters)
if(z=='a') n=1 else
if(z=='e') n=2 else
if(z=='i') n=3 else
if(z=='o') n=4 else n=5
```

Exercise 10

What is the output n of:

```
z='u'
if(z %in% letters)
```

```
if (z=='a') n=1 else
if (z=='e') n=2 else
if (z=='i') n=3 else
if (z=='o') n=4 else n=5
```

Scripting Loops In R

Exercise 1

The repeat{} loop processes a block of code until the condition specified by the break statement, (that is mandatory within the repeat{} loop), is met.

The structure of a repeat{} loop is:

```
repeat {
  commands
  if(condition) {
    break
  }
}
```

For the first exercise, write a repeat{} loop that prints all the even numbers from 2 – 10, via incrementing the variable, “i <- 0”.

Exercise 2

Using the following variables:

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

Write a repeat{} loop that breaks off the incrementation of, “i”, after 5 loops, and prints “msg” at every increment.

Exercise 3

while() loop will repeat a group of commands until the condition ceases to apply. The structure of a while() loop is:

```
while(condition) {
  commands
}
```

With, `i <- 1`, write a `while()` loop that prints the odd numbers from 1 through 7.

Exercise 4

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.

Exercise 5

The `for()` loop repeats commands until the specified length of the condition is met. The structure of a `for()` loop is:

```
for(condition) { commands }
```

For example:

```
for(i in 1:4) {
```

```
  print("variable"[i])
```

```
}
```

```
for(i in seq("variable")) {
```

```
  print(i)
```

```
}
```

```
for(i in seq_along("variable")) {
```

```
  print("variable"[i])
```

```
}
```

```
for(letter in "variable") {
```

```
  print(letter)
```

```
}
```

For this exercise, write a `for()` loop that prints the first four numbers of this sequence: `x <- c(7, 4, 3, 8, 9, 25)`

Exercise 6

For the next exercise, write a `for()` loop that prints all the letters in `y <- c("q", "w", "e", "r", "z", "c")`.

Exercise 7

The `break` statement is used within loops to exit from the loop. If the `break` statement is within a nested loop, the inner loop is exited, and the outer loop is resumed.

Using `i <- 1`, write a `while()` loop that prints the variable, “i”, (that is incremented from 1 – 5), and uses `break` to exit the loop if “i” equals 3.

Exercise 8

Write a nested loop, where the outer `for()` loop increments “a” 3 times, and the inner `for()` loop increments “b” 3 times. The `break` statement exits the inner `for()` loop after 2 incrementations. The nested loop prints the values of variables, “a” and “b”.

Exercise 9

The `next` statement is used within loops in order to skip the current evaluation, and instead proceed to the next evaluation.

Therefore, 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.

Exercise 10

Finally, write a `for()` loop that uses `next` to print all values except “3” in the following variable: `i <- 1:5`

Loops in R

Exercise 1

Write a `for` loop that iterates over the numbers 1 to 7 and prints the cube of each number using `print()`.

Exercise 2

Write a `for` loop that iterates over the column names of the inbuilt iris dataset and print each together with the number of characters in the column name in parenthesis. Example output: Sepal.Length (12). Use the following functions `print()`, `paste0()` and `nchar()`.

Exercise 3

Write a `while` loop that prints out standard random normal numbers (use `rnorm()`) but stops (breaks) if you get a number bigger than 1.

Exercise 4

Using `next` adapt the loop from last exercise so that doesn't print negative numbers.

Exercise 5

Using a `for` loop simulate the flip a coin twenty times, keeping track of the individual outcomes (1 = heads, 0 = tails) in a vector that you preallocate.

Exercise 6

Use a nested `for` loop (a `for` loop inside a `for` loop) that produces the following matrix, preallocate the matrix with NA values.

0	1	2	3	4
1	0	1	2	3
2	1	0	1	2
3	2	1	0	1
4	3	2	1	0

Exercise 7

Use a while loop to investigate the number of terms required before the product

$1 \cdot 2 \cdot 3 \cdot 4 \cdot \dots$

reaches above 10 million.

Exercise 8

Use a while loop to simulate one stock price path starting at 100 and random normally distributed percentage jumps with mean 0 and standard deviation of 0.01 each period. How long does it take to reach above 150 or below 50?

Exercise 9

Implement a simple version of Guess the number game using a while loop. The user should guess a number between 1 and 10, you can use `scan()` to get user input. The loop should break if the user guesses 5.

Exercise 10

Implement a multiplication game. A while loop that gives the user two random numbers from 2 to 12 and asks the user to multiply them. Only exit the loop after five correct answers. Try using `as.integer(readline())` instead of `scan()` this time.

FUNCTIONS

Functions Exercises 1

Exercise 1

Create a function that will return the sum of 2 integers.

Exercise 2

Create a function what will return TRUE if a given integer is inside a vector.

Exercise 3

Create a function that given a data frame will print by screen the name of the column and the class of data it contains (e.g. Variable1 is Numeric).

Exercise 4

Create the function unique, which given a vector will return a new vector with the elements of the first vector with duplicated elements removed.

Exercise 5

Create a function that given a vector and an integer will return how many times the integer appears inside the vector.

Exercise 6

Create a function that given a vector will print by screen the mean and the standard deviation, it will optionally also print the median.

Exercise 7

Create a function that given an integer will calculate how many divisors it has (other than 1 and itself). Make the divisors appear by screen.

Exercise 8

Create a function that given a data frame, and a number or character will return the data frame with the character or number changed to NA.

Functions Exercises 2

Exercise 1

Create a function that given a data frame and a vector, will add a the vector (if the vector length match with the rows number of the data frame) as a new variable to the data frame.

Exercise 2

Consider a data frame df:

```
Id=c(1:10)
```

```
Age=c(14,12,15,10,23,21,41,56,78,12)
```

```
Sex=c('F','M','M','F','M','F','M','M','F','M')
```

```
Code=letters[1:10]
```

```
df=data.frame(Id,Age,Sex,Code)
```

Create a function that, given a data frame and two indexes, exchanges two values of the Code variable with each other.

For example, if the index is 1 and 3, you assign:

```
df[1,'Code']=df[3,'Code']
```

```
df[3,'Code']=df[1,'Code']
```

Exercise 3

Consider two variables x,y and a data frame df:

x,y integer

```
A=c(1:10)
```

```
B=seq(100,10,-10)
```

```
H=seq(-200,-50,along.with=B)
```

```
df=data.frame(A,B,H)
```

Create a function that given a data frame df calculate a new variable 'SUM_x_y'(If x=2 and y=3, then the new variable will be 'SUM_2_3',

if x=4 and y=10, then the new variable will be 'SUM_4_10'),such that for each row 'i' is equal to:

```
sum(x*df[1:i,1])+sum(y*df[1:i,2])
```

Exercise 4

Create a function that given a numeric vector, sort this in ascending order and duplicate it by two.

Exercise 5

Create a function that given a vector alpha numeric, keep only the numbers and apply the function created on exercise 4.

For example, if the input is a vector w="a" "v" "7" "4" "q" , the function will return w=8 14.

Exercise 6

Create a function that given a string

```
ST='NAME: Maria /COUNTRY:uruguay /EMAIL: mariaUY@gmail.com'
```

return a matrix

[1] [2]

[1,] "NAME" " Maria "

[2,] "COUNTRY" "uruguay "

[3,] "EMAIL" " mariaUY@gmail.com"

Exercise 7

Consider a vector:

```
ST=c('NAME:Maria /COUNTRY:uruguay /EMAIL:mariaUY@gmail.com','NAME:Paul/COUNTRY:UK  
/EMAIL:PaulUK@gmail.com',
```

```
'NAME:Jhon /COUNTRY:USA /EMAIL:JhonUSA@gmail.com','NAME:Carlos /COUNTRY:Spain  
/EMAIL:CarlosSP@gmail.com')
```

Create a function that given a vector string ST return a matrix:

[1] [2] [3] [4] [5]

[1,] "NAME" "Maria " "Paul" "Jhon " "Carlos "

[2,] "COUNTRY" "uruguay " "UK " "USA " "Spain "

[3,] "EMAIL" "mariaUY@gmail.com" "PaulUK@gmail.com" "JhonUSA@gmail.com" "CarlosSP@gmail.com"

Exercise 8

Create a function that given a numeric vector X returns the digits 0 to 9 that are not in X. If X=0 2 4 8

the function return 1 3 5 6 7 9

Exercise 9

Create a function that given two strings (one word each), check if one is an anagram of another.

Exercise 10

Create a function that given one word, return the position of word's letters on letters vector.

For example, if the word is 'abc', the function will return 1 2 3.