

Stat 291 - Recitation 6

Orçun Oltulu

3 / 12 / 2021

1. Conditionals:

Exercise 1.1:

Write an if statement that prints absolute value of an integer. (without using abs() function)

```
x <- -5

if(x < 0){
  print(-x)
} else{
  print(x)
}
```

```
## [1] 5
```

```
# Alternatively,
ifelse(x < 0, -x, x)
```

```
## [1] 5
```

Exercise 1.2:

Write an if statement that returns “even” or “odd” based on the given value.

```
x <- 6

if(x%%2 == 1){
  print("odd")
} else{
  print("even")
}
```

```
## [1] "even"
```

```
# Alternatively
ifelse(x%%2 == 1, "odd", "even")
```

```
## [1] "even"
```

Exercise 1.3:

Write an if statement that returns the maximum value out of the elements of a numeric vector x of length 2.

```
x <- c(8,16)

if(x[2] > x[1]){
  cat("Max value is", x[2], "\n")
} else if (x[1] > x[2]){
  cat("Max value is", x[1], "\n")
} else{
  cat("They are equal", "\n")
}
```

```
## Max value is 16
```

Exercise 1.4:

Assume you have the following vector;

```
vec <- c(1,0,0,0,1,1,0)
```

Convert 0 and 1 to non-smoker and smoker.

```
vec <- c(1,0,0,0,1,1,0)
vec <- ifelse(vec == 0, "non-smoker", "smoker")
vec
```

```
## [1] "smoker"      "non-smoker" "non-smoker" "non-smoker" "smoker"
## [6] "smoker"      "non-smoker"
```

Exercise 1.5:

Consider the following continuous function:

$$f(x) = \begin{cases} 3x + 2 & x \leq 3 \\ 2x - 0.5x^2 & x > 3 \end{cases}$$

Write an if-else statement in R for the mathematical function given above, where x is 3.

```
x <- 3

if(x <= 3){
  y <- 3*x + 2
} else{
  y <- 2*x - 0.5 * x^2
}
print(y)
```

```
## [1] 11
```

```
# alternatively
```

```
x <- 3
y <- ifelse(x <= 3, 3*x + 2, 2*x - 0.5 * x^2)
y
```

```
## [1] 11
```

Exercise 1.6:

Write a code block which takes a character variable, controls with the password which is “mypassword123”. If it is correct prints “Correct”, if it is not, prints “False”. (Hint: to get an input from user use ‘readline’ command)

```
trial <- readline(prompt = "Enter your password: ")
```

```
## Enter your password:
```

```
true_password <- "mypassword123"
trial <- readline(prompt = "Enter your password: ")
```

```
## Enter your password:
```

```
if(true_password == trial){
  print("Correct")
} else{
  print("False")
}
```

```
## [1] "False"
```

```
# alternatively
```

```
ifelse(readline(prompt = "Enter your password: ") == true_password,
       "Correct", "False")
```

```
## Enter your password:
```

```
## [1] "False"
```

Exercise 1.7:

(a) Using if statements create a simple calculator which can calculate only addition, subtraction, multiplication and division for 2 values.

Use readline() function to get the math operation from user.

```
operation <- readline(prompt = "Specify the operation: ")
```

```
## Specify the operation:
```

```
a <- 5
```

```
b <- 2
```

```
operation <- readline(prompt = "Specify the operation: ")
```

```
## Specify the operation:
```

```
if(operation %in% c("+","addition")){  
  result <- a + b  
  cat("a + b = ",a, "+", b, "=", result)  
} else if(operation %in% c("-", "subtration")){  
  result <- a - b  
  cat("a - b = ",a, "-", b, "=", result)  
} else if(operation %in% c("*", "multiplication")){  
  result <- a * b  
  cat("a * b = ",a, "*", b, "=", result)  
} else if(operation %in% c("/", "division")){  
  result <- a / b  
  cat("a / b = ",a, "/", b, "=", result)  
} else {  
  print("Warning: There is no such operation !" )  
}
```

```
## [1] "Warning: There is no such operation !"
```

(b) Now, create the same calculator using Switch() function.

```
a <- 4
```

```
b <- 7
```

```
operation <- readline(prompt = "Specify the operation: ")
```

```
## Specify the operation:
```

```
result <- switch(operation,  
  "addition" = a + b,  
  "subtraction" = a - b,  
  "multiplication" = a * b,
```

```
        "division" = a / b)
result
```

```
## NULL
```

2. Loops:

Exercise 2.1:

Using for loop print integers from 1 to 9.

```
for(i in 1:9){
  print(i)
}
```

```
## [1] 1
## [1] 2
## [1] 3
## [1] 4
## [1] 5
## [1] 6
## [1] 7
## [1] 8
## [1] 9
```

Exercise 2.2:

Using for loop create the following sequence;

```
for(i in 1:9){
  cat(1:i, "\n")
}
```

```
## 1
## 1 2
## 1 2 3
## 1 2 3 4
## 1 2 3 4 5
## 1 2 3 4 5 6
## 1 2 3 4 5 6 7
## 1 2 3 4 5 6 7 8
## 1 2 3 4 5 6 7 8 9
```

Exercise 2.3:

Calculate factorial for a given 'x' using a loop and show the result for even steps. (eg. $2! = 2$, $4! = 24$, $6! = 720$...)

```
x <- 11
result <- 1

for(i in 1:x){
  result <- result * i
  if(i %% 2 == 0){
    cat(i,"!", " = ", result,"\n", sep = "")
  }
}
```

```
## 2! = 2
## 4! = 24
## 6! = 720
## 8! = 40320
## 10! = 3628800
```

Exercise 2.4:

Create an 5x5 upper triangular matrix of 1s. (Hint: Firstly, create a zero matrix then fill 1's with for loops)

```
A <- matrix(0,ncol = 5, nrow = 5)
A
```

```
##      [,1] [,2] [,3] [,4] [,5]
## [1,]    0    0    0    0    0
## [2,]    0    0    0    0    0
## [3,]    0    0    0    0    0
## [4,]    0    0    0    0    0
## [5,]    0    0    0    0    0
```

```
for(i in 1:5){
  for(j in i:5){
    A[i,j] = 1
  }
}
```

```
##      [,1] [,2] [,3] [,4] [,5]
## [1,]    1    1    1    1    1
## [2,]    0    1    1    1    1
## [3,]    0    0    1    1    1
## [4,]    0    0    0    1    1
```

```
## [5,]    0    0    0    0    1
# Alternatively,
a <- matrix(0, nrow = 5, ncol = 5)

for (i in 1:5){
  column = c(rep(1,i), rep(0,(5-i)))
  a[,i] = column
}
```

```
# Alternatively,

m <- matrix(0,5,5)
m
```

```
##      [,1] [,2] [,3] [,4] [,5]
## [1,]    0    0    0    0    0
## [2,]    0    0    0    0    0
## [3,]    0    0    0    0    0
## [4,]    0    0    0    0    0
## [5,]    0    0    0    0    0
```

```
for(i in 1:5){

  m[1:i,i] <- 1

}
```

Exercise 2.5:

Generate a random sample of size 20 between 1 to 100 with the following code block;

```
set.seed(291)
x <- sample(1:100, size = 20)
```

- Then, count the number of even numbers in vector y using for loops.

```
count <- 0

for(i in 1:length(x)){
  if(x[i] %% 2 == 0)
    count = count + 1
}

count
```

```
## [1] 7
```

- Now, find how many even numbers in your vector without using any loop.

```
# alternatively  
sum(ifelse(x %% 2 == 0, 1, 0))
```

```
## [1] 7
```

```
# alternatively  
sum(x %% 2 == 0)
```

```
## [1] 7
```

Exercise 2.6:

Create the following 5x5 matrix using for loops. (Hint: Create a zero matrix, use paste function)

```
R <- C <- 5  
  
mat <- matrix(0, nrow = R, ncol = C)  
  
for(i in 1:R){  
  for(j in 1:C){  
    mat[i,j] <- paste("X",i,j,sep="")  
  }  
}
```

```
##      [,1] [,2] [,3] [,4] [,5]  
## [1,] "X11" "X12" "X13" "X14" "X15"  
## [2,] "X21" "X22" "X23" "X24" "X25"  
## [3,] "X31" "X32" "X33" "X34" "X35"  
## [4,] "X41" "X42" "X43" "X44" "X45"  
## [5,] "X51" "X52" "X53" "X54" "X55"
```