UNIVERSIDAD NACIONAL DE LOJA

PERIODO ACADÉMICO: OCTUBRE 2019 MARZO 202(
PRAC
TICA # 4
ASIGNATURA:
SIMULACIÓN
RESULTADO DE APRENDIZAJE DE LA PRÁCTICA: Entiende las estructuras básicas de programación en R
TIEMPO PLANIFICADO: 3 HORAS
NUMERO DE ESTUDIANTES: Sexto ciclo (Paralelo A)

1. **TEMA: Programación básica en R**

2. **OBJETIVOS:**

- Comprende las estructuras básicas if, ifelse, for.
- Comprende el uso de vectorización.
- Usa los conocimientos aprendidos en teoría para su posterior aplicación práctica.

3. **RECURSOS NECESARIOS:**

- R-studio.
- Computador de Laboratorios

4. **INSTRUCCIONES:**

- Prohibido consumo de alimentos
- Prohibido equipo de diversión, celulares etc.
- Prohibido jugar
- Prohibido mover o intercambiar los equipos de los bancos de trabajo
- Prohibido sacar los equipos del laboratorio sin autorización.
- Ubicar los equipos y accesorios en el lugar dispuesto por el responsable del laboratorio, luego de terminar las prácticas.
- Uso adecuado de equipos

5. **ACTIVIDADES POR DESARROLLAR:**

1. What will this conditional expression return?

```
x <- c(1, 2, -3, 4)

     if(all(x>0)){ print("All Postives")
     } else{
     print("Not all positives")
     }
```

```
[1] "Not all positives"
```

2. Which of the following expressions is always FALSE when at least one entry of a logical vector xis TRUE?
   A. all(x)
   B. any(x)
   C. any(!x)
   D. all(!x)

3. The function nchar tells you how many characters long a character vector is. Write a line of code that assigns to the object new_names the state abbreviation when the state name islonger than 8 characters.

```
library(dslabs)
data("murders")
ifelse(nchar(murders$state)>8,  murders$abb, murders$state)
```

```
 [1] "Alabama"  "Alaska"   "Arizona"  "Arkansas" "CA"       "Colorado"
 [7] "CT"       "Delaware" "DC"       "Florida"  "Georgia"  "Hawaii"
[13] "Idaho"    "Illinois" "Indiana"  "Iowa"     "Kansas"   "Kentucky"
[19] "LA"       "Maine"    "Maryland" "MA"       "Michigan" "MN"
[25] "MS"       "Missouri" "Montana"  "Nebraska" "Nevada"   "NH"
[31] "NJ"       "NM"       "New York" "NC"       "ND"       "Ohio"
[37] "Oklahoma" "Oregon"   "PA"       "RI"       "SC"       "SD"
[43] "TN"       "Texas"    "Utah"     "Vermont"  "Virginia" "WA"
[49] "WV"       "WI"       "Wyoming"
```

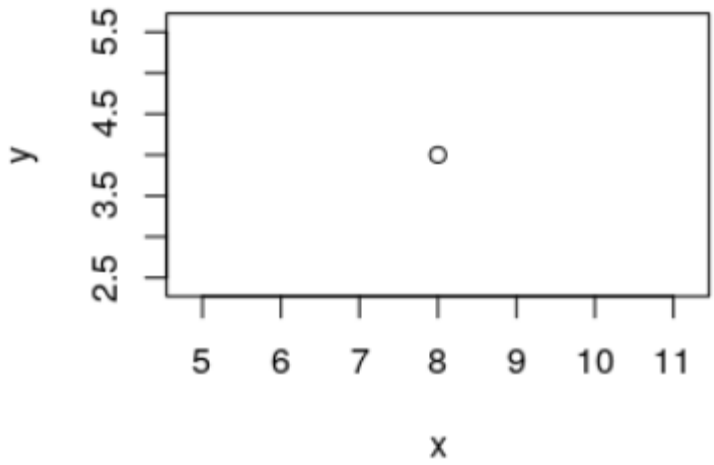
4. Create a function sum_n that for any given value, say n, computes the sum of the integers from 1 to n (inclusive). Use the function to determine the sum of integers from 1 to 5,000.

```
suma_n <- function(x){
s <- sum(x)
s
}
n <-5000
suma_n(1:n)
```

```
[1] 12502500
```

5. Create a function altman_plot that takes two arguments, x and y, and plots the difference against the sum.

```
altmat_plot <- function(x,  y){
 plot(x+y,  x-y)
  }
 x <- 8
 y <- 4
 plot(x,y)
```



6. After running the code below, what is the value of x?

```
x<- 3
 my_func <- function(y){
x <- 5
y+5

}
x
```

x=3 fuera de la función; x= 5 dentro de la función

7. Write a function compute_s_n that for any give n computes the sum s n = 12 + 22 + 32 +42+. . . +n2. Report the value of the sum when n=10 .

```
compute_s_n <- function(n){
    numero <- 1:n
    sum(numero)
  }
  n=10
  compute_s_n(n)
```

[1] 55

8. Define an empty numerical vector `s_n` of size 25 using `s_n <- vector("numeric", 25)` and store in the results of S_1,S_2,...S_{25} using a for-loop

```
s_n <- vector("numeric", 25)
  n <- 25
  for(i in 1:n){s_n[i] <- sum(i)
  }
  s_n
```

```
 [1]  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25
```

9. Repeat exercise 8, but this time use sapply.

```
s_n <- vector("numeric", 25)
compute_s_n <- function(n){
sum(n)
}
n=1:25
s_n <- sapply(n, compute_s_n)
s_n
```

```
 [1]  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25
```

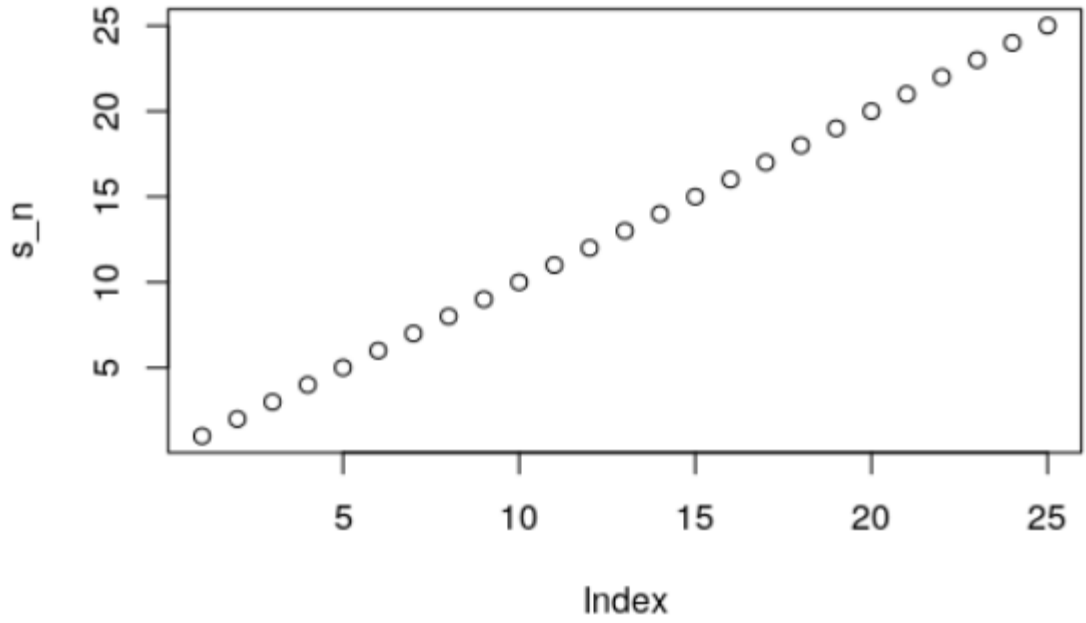
10. Repeat exercise 8, but this time use map_dbl.

```
install.packages("tidyverse")
install.packages("purrr")
library(purrr)
s_n <- vector("numeric", 25)
compute_s_n <- function(n){
        sum(n)
}
n <- 1:25
s_n <- map_dbl(n, compute_s_n)
s_n
```

```
 [1]  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25
```

11. Plot Sn versus n. Use points defined by n=1,…,25.

```
s_n <- vector("numeric", 25)
compute_s_n <- function(n){
        sum(n)
}
n=1:25
s_n <- sapply(n, compute_s_n)
s_n
plot(n)
plot(s_n)
```



12. Confirm that the formula for this sum is Sn=n(n+1)(2n+1)/6.

```
s_n <- vector("numeric", 25)
compute_s_n <- function(n){
        sum(n)
}
n=1:25
compute_s_n(n)
s_n <- sapply(n, compute_s_n)
s_n
sn <-n*(n+1)*(2*n+1)/6
identical(s_n, sn )
```

```
[1] FALSE
```

6. INVESTIGACIÓN COMPLEMENTARIA (a elaborar por el estudiante)

Investigar acerca de las funciones lapply, tapply, mapply, vapply, y replicate

- **Laply**: Se diferencia con apply en que opera con listas. Recibe una lista y devuelve una lista

```
A <- matrix(1:9,nrow = 3, ncol = 3)
B <- matrix(11:19,nrow = 3, ncol = 3)
C <- matrix(21:29,nrow = 3, ncol = 3)
mi_lista <- list(A,B,C)
lapply(mi_lista,"[",1,1)
[[1]]
[1] 1

[[2]]
[1] 11

[[3]]
[1] 21
```

- **Tapply**: Realiza una operación (parámetro 3) respecto a un vector (parámetro 1) agrupada por los factores que se indiquen como argumento (parámetro 2).

```
x <- 1:20
y <- factor(rep(letters[1:5], each = 4))
x
 [1]  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20
y
 [1] a a a a b b b b c c c c d d d d e e e e
Levels: a b c d e

tapply(x, y, sum)
 a  b  c  d  e
10 26 42 58 74
```

- **Mapply**: Realiza operaciones entre matrices y devuelve una lista o vector, a continuación se muestran algunas de las operaciones que admite: Suma el primer elemento de cada vector, después el segundo y así sucesivamente:

```
mapply(sum, 1:5, 1:5, 1:5)
[1]  3  6  9 12 15
```

Repite cada elemento del primer vector el número de veces que indique el segundo vector:

```
mapply(rep, 1:4, 4:1)
[[1]]
 [1] 1 1 1 1
[[2]]
 [1] 2 2 2
[[3]]
 [1] 3 3
[[4]]
 [1] 4
```

- **Vapply**: Devuelve un vector con la longitud que tiene cada una de las listas introducidas como parámetro. Los bucles de la familia apply.

```
(x <- list(A = 1, B = 1:3, C = 1:7))
```

```
## $A
## [1] 1
##
## $B
## [1] 1 2 3
##
## $C
## [1] 1 2 3 4 5 6 7
```

```
vapply(x, FUN = length, FUN.VALUE = 0L)
```

```
## A B C
## 1 3 7
```

7. DISCUSIÓN (a elaborar por el estudiante)

- los ejercicios planteados permiten asimilar el conocimiento de la utilización de R Estudios, dentro del cual se aprende a utilizar funciones, vectores, operaciones, bucles, condiciones etc.

8. CONCLUSIONES (a elaborar por el estudiante)

- R Studios dispone de muchas funciones que ahorra la elaboración de códigos tediosos y largos

9. RECOMENDACIONES (elaborar por el estudiante)

- Se recomienda practicar con diferentes ejercicios para asimilar lo conceptos nuevos que se va aprendiendo en R Studios

10. BIBLIOGRAFÍA:

- A Brief History R: Past and Future History, Ross Ihaka, Statistics Department, The University of Auckland, Auckland, New Zealand, available from the CRAN website
- What's new in R? What's new in R?

Firma del Presidente de Curso de Sexto A

Ing. Marlon Santiago Viñan Ludeña

Mg. Sc DOCENTE CIS