

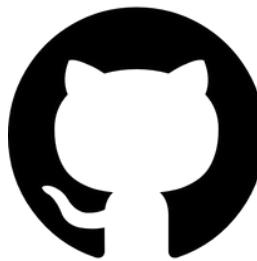
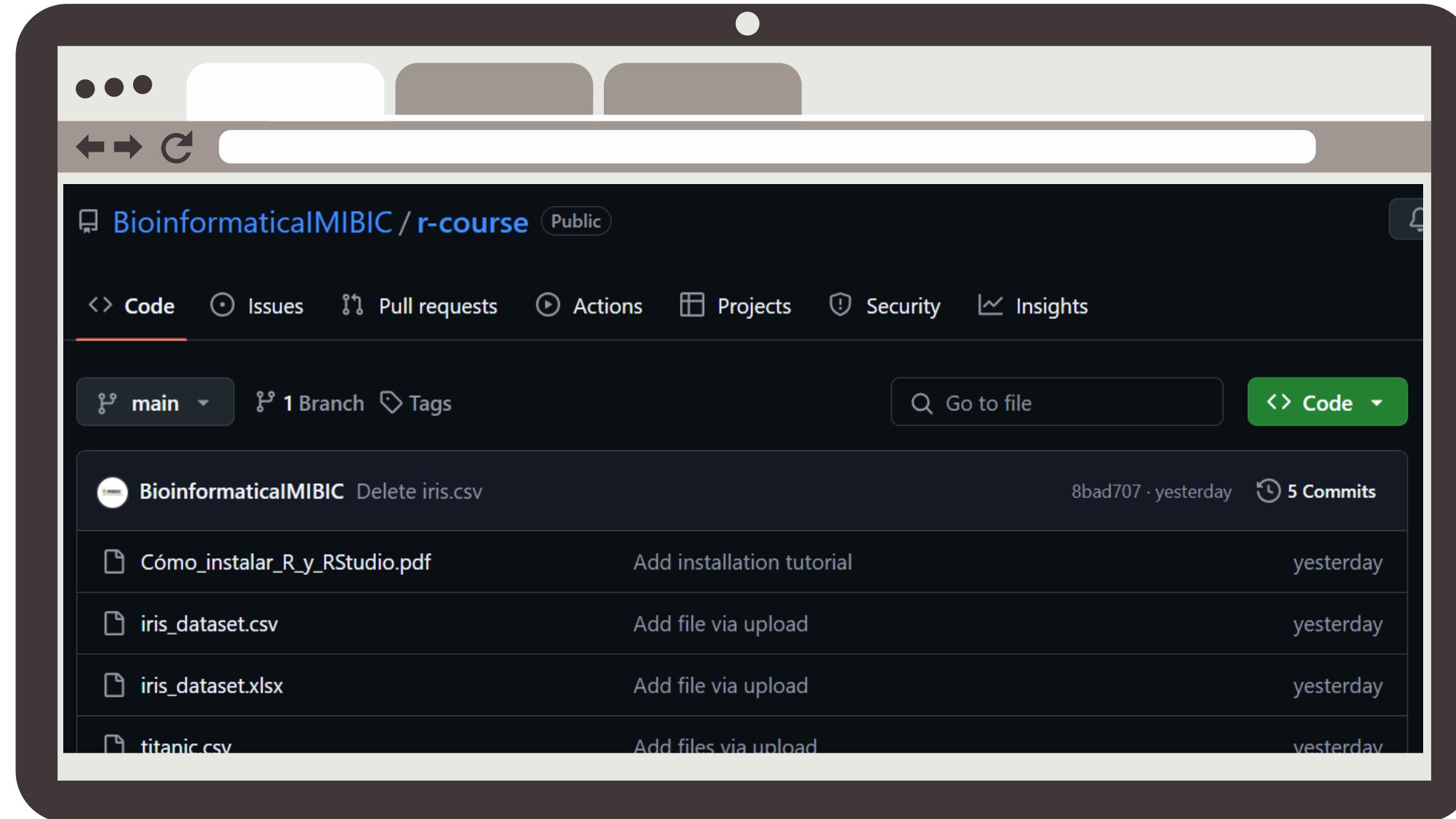
<!--Estudio IMIBIC-->

R para Todos: El poder para explotar tus Datos {

<Por="Adrián Santiago Ortiz"/>

}

Material disponible

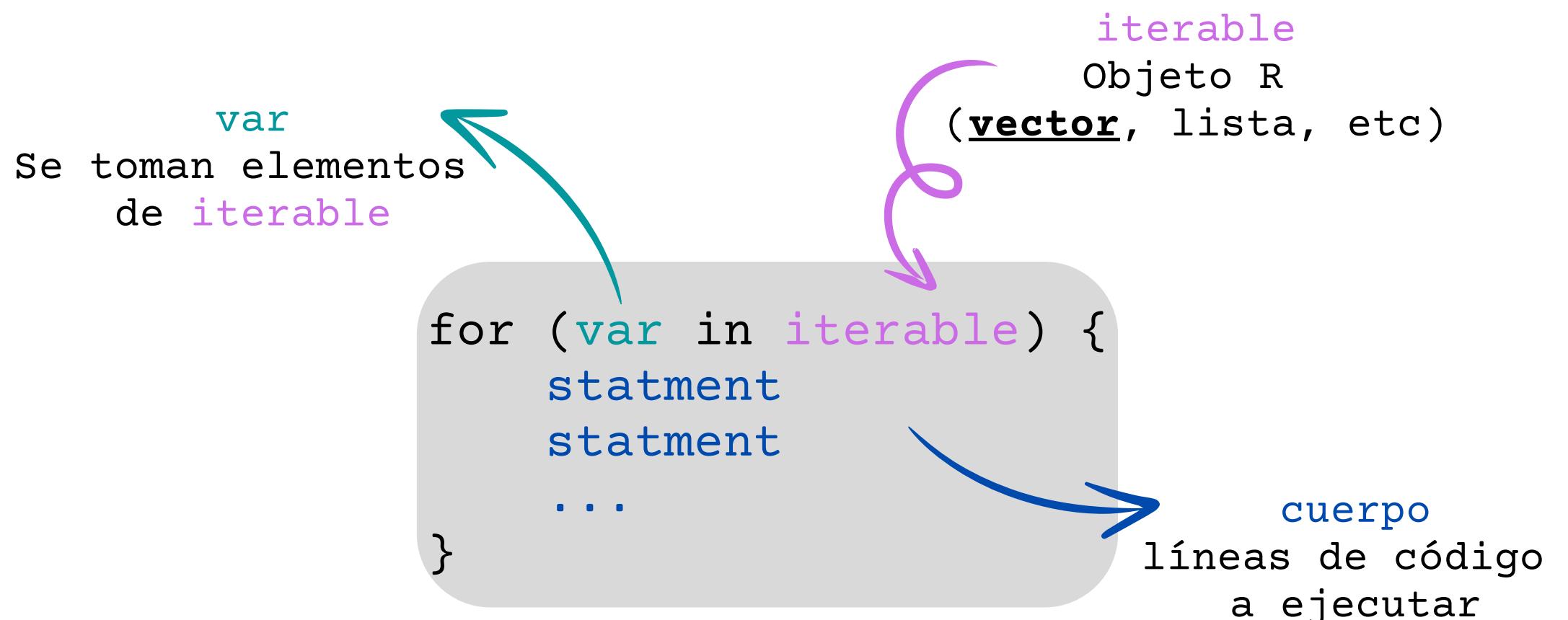


<https://github.com/BioinformaticalMIBIC/r-course>

Repaso de la sesión 2



Bucle FOR: repite un bloque de código un número específico de veces



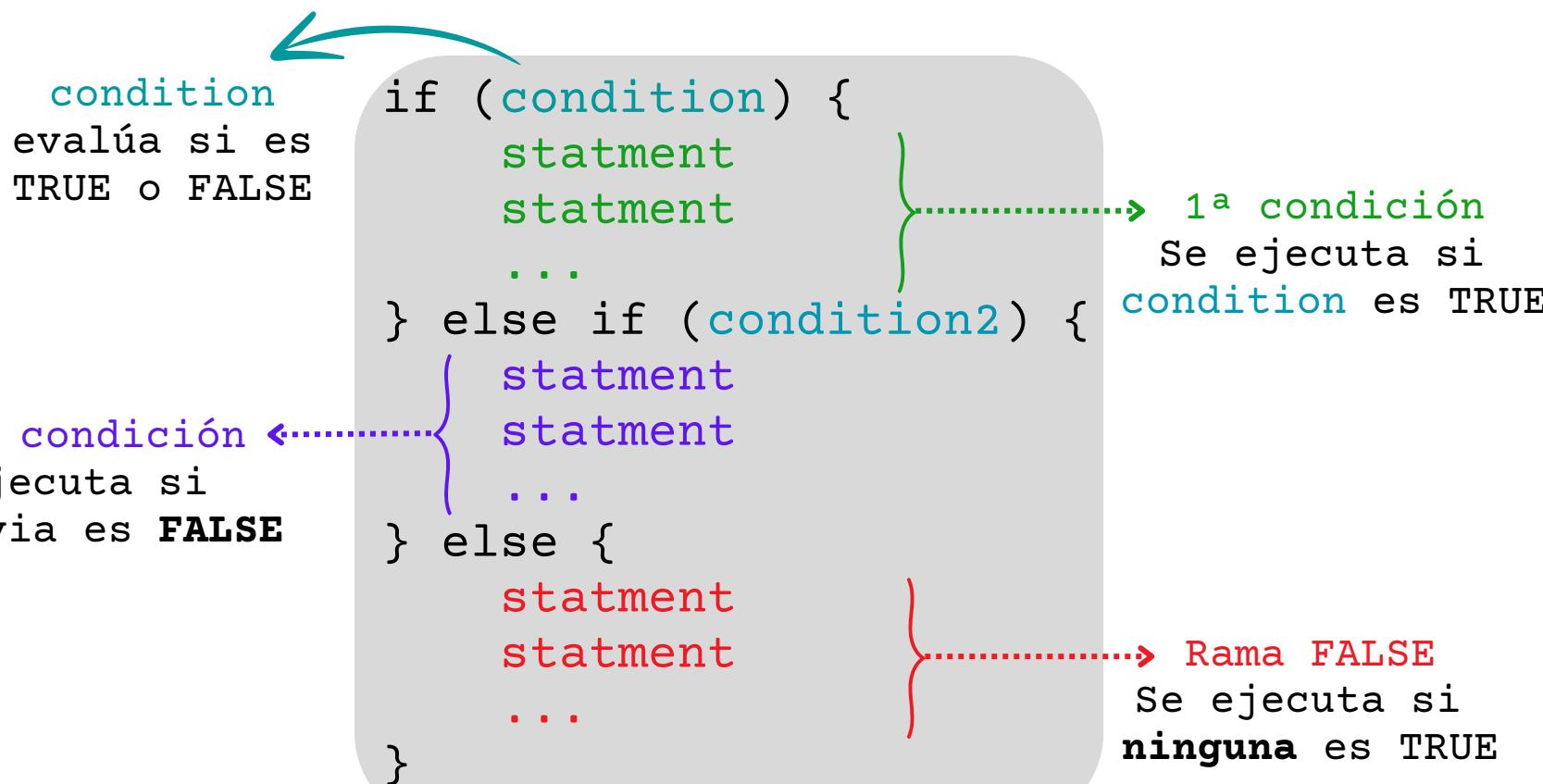
```
for (i in 1:10) {  
  if (i != 5) {  
    print (i)  
  }  
}
```

```
[1] 1  
[1] 2  
[1] 3  
[1] 4  
[1] 6  
[1] 7  
[1] 8  
[1] 9  
[1] 10
```

Repaso de la sesión 2



if/else if/else: ejecuta un bloque de código solo si se cumple una condición específica



```
horas_sueño <- 6  
desayuno <- TRUE  
  
if (horas_sueño >= 8 & desayuno == TRUE) {  
    mensaje <- "¡A comerte el día!"  
} else if (horas_sueño >= 8 & desayuno == FALSE) {  
    mensaje <- "No está mal pero igual necesitas un café"  
} else if (horas_sueño < 8 & desayuno == TRUE) {  
    mensaje <- "Un poco de sueño, pero al menos desayunaste"  
} else {  
    mensaje <- "¡Sobrevivir al lunes será tu misión!"  
}  
  
# Mostrar el mensaje  
print(mensaje)
```

Optimización de bucles

Funciones *apply*

apply

lapply

sapply

tapply

rapply

mapply

Por fila (1)

apply(m, 1, sum)

objeto: m

función: sumar

1	3	5
2	4	6

$$\text{sum}(1, 3, 5) = 9$$

$$\text{sum}(2, 4, 6) = 12$$

[9, 12]

Contenidos

01 Introducción a R y Rstudio

02 Tipos de datos

03 Operadores en R

04 Tipos de objetos y operaciones

05 Estructuras de control y flujo de ejecución

06 Creación de funciones **Sesión 3**

07 Visualización de resultados en gráficos

Contenidos

01

02

03

04

05

06 Creación de funciones

07

• Funciones, ¿ventajas de su uso?



Creación de funciones: sintaxis

```
name <- function (args) {
```

```
}
```

```
variable <- name(args)
```

args

lista de
valores a usar

Creación de funciones: sintaxis

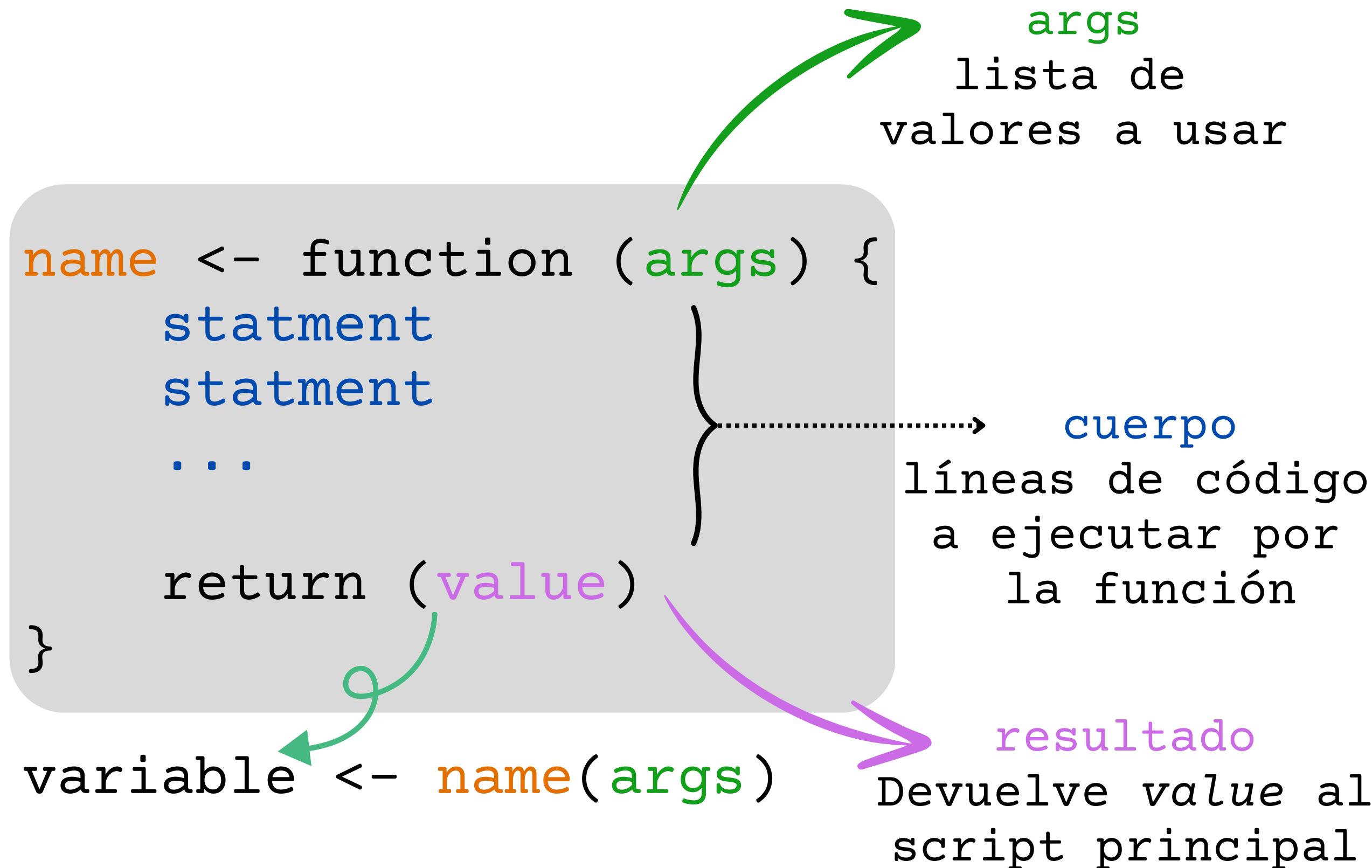
```
name <- function (args) {  
  statement  
  statement  
  ...  
}
```

args
lista de
valores a usar

cuerpo
líneas de código
a ejecutar por
la función

```
variable <- name(args)
```

Creación de funciones: sintaxis



Ejercicio. Obtén la tabla de multiplicar completa del 4.

```
tbl_mult <- function(n) {  
}  
tbl_mult(4)
```

Return no necesario

```
> tbl_mult(4)  
[1] "4 x 0 = 0"  
[1] "4 x 1 = 4"  
[1] "4 x 2 = 8"  
[1] "4 x 3 = 12"  
[1] "4 x 4 = 16"  
[1] "4 x 5 = 20"  
[1] "4 x 6 = 24"  
[1] "4 x 7 = 28"  
[1] "4 x 8 = 32"  
[1] "4 x 9 = 36"  
[1] "4 x 10 = 40"
```

Ejercicio. Obtén la tabla de multiplicar completa del 4.

```
tbl_mult <- function(n) {  
  for (i in 0:10) {  
    res <- n * i  
    output <- paste(n, "x", i, "=", res)  
    print (output)  
  }  
}  
  
tbl_mult(4)
```

Return no necesario

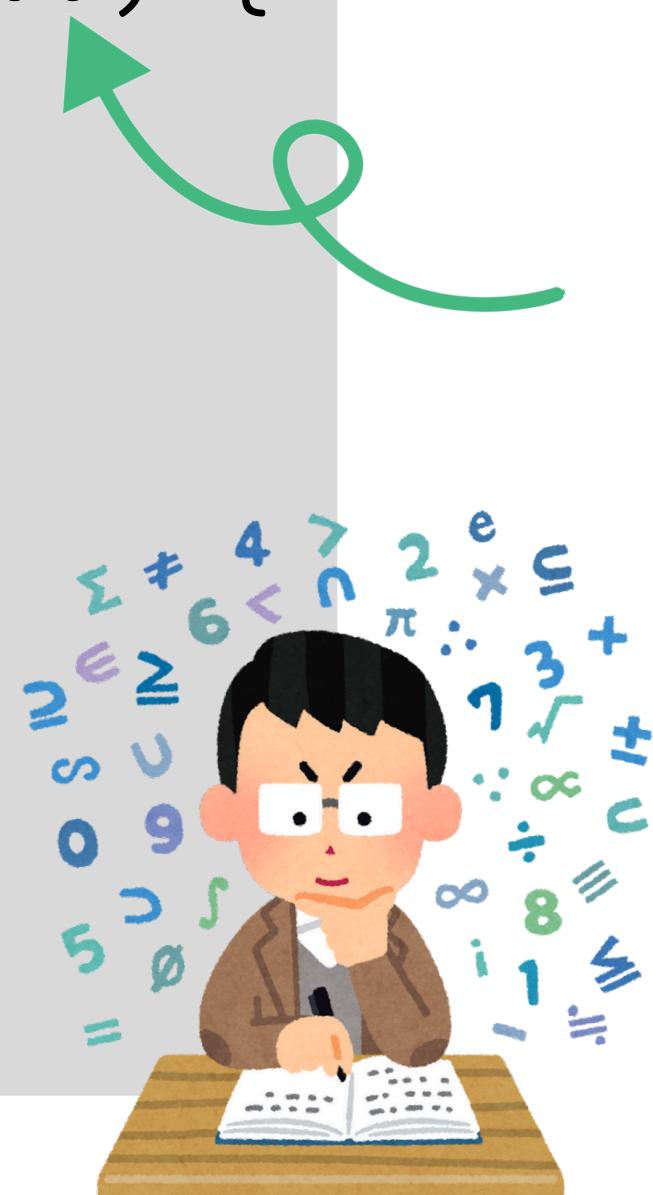


```
> tbl_mult(4)  
[1] "4 x 0 = 0"  
[1] "4 x 1 = 4"  
[1] "4 x 2 = 8"  
[1] "4 x 3 = 12"  
[1] "4 x 4 = 16"  
[1] "4 x 5 = 20"  
[1] "4 x 6 = 24"  
[1] "4 x 7 = 28"  
[1] "4 x 8 = 32"  
[1] "4 x 9 = 36"  
[1] "4 x 10 = 40"
```

• Funciones: número variable de parámetros

```
suma_cuadrados <- function(...) {  
  valores <- list(...)  
  res <- 0  
  
  for (valor in valores) {  
    res <- res + valor ^ 2  
  }  
  
  return (res)  
}
```

```
suma_cuadrados(-4, 2, 0, 3)  
[1] 29
```



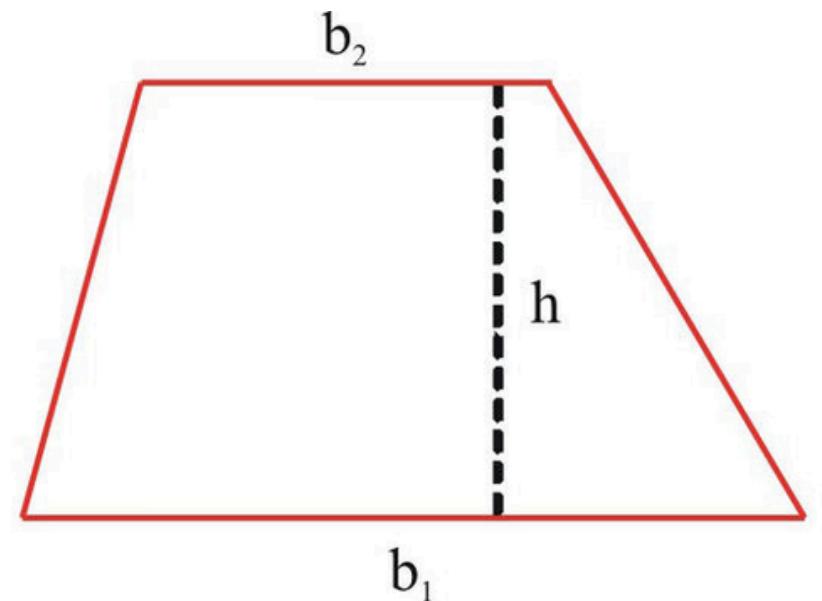
Ejercicios

Calcula el área de un trapezoide cuya base mayor (b_1) es igual a 6, su base menor (b_2) es igual a 4 y su altura es 5.

```
calc_area <- function () {
```

```
}
```

```
res
```



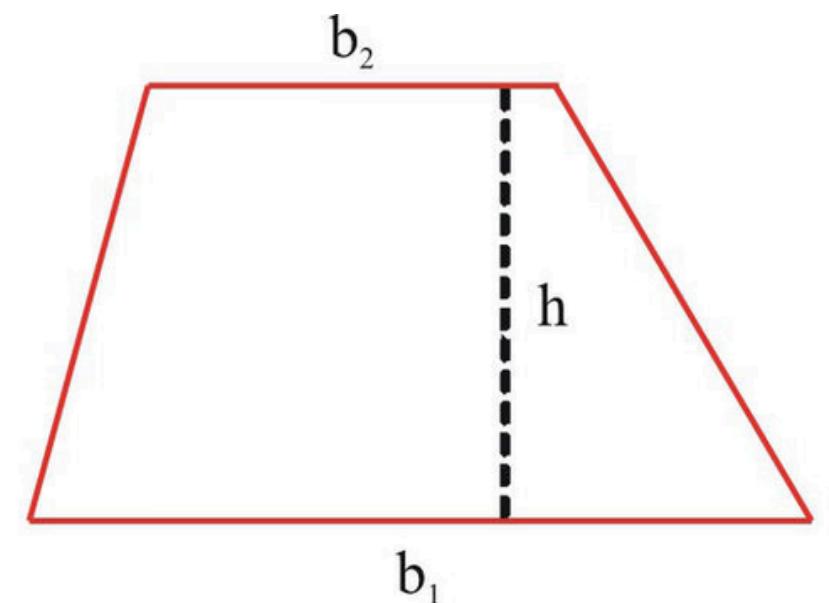
$$A = \frac{(b_1 + b_2)h}{2}$$

Ejercicios

Calcula el área de un trapezoide cuya base mayor (b_1) es igual a 6, su base menor (b_2) es igual a 4 y su altura es 5.

```
calc_area <- function (b1, b2, h) {  
  A <- (b1 + b2) * h / 2  
  return(A)  
}  
  
res <- calc_area(6, 4, 5)
```

> res



$$A = \frac{(b_1 + b_2)h}{2}$$

Ejercicios

Calcula el área de un trapezoide cuya base mayor (b_1) es igual a 6, su base menor (b_2) es igual a 4 y su altura es 5.

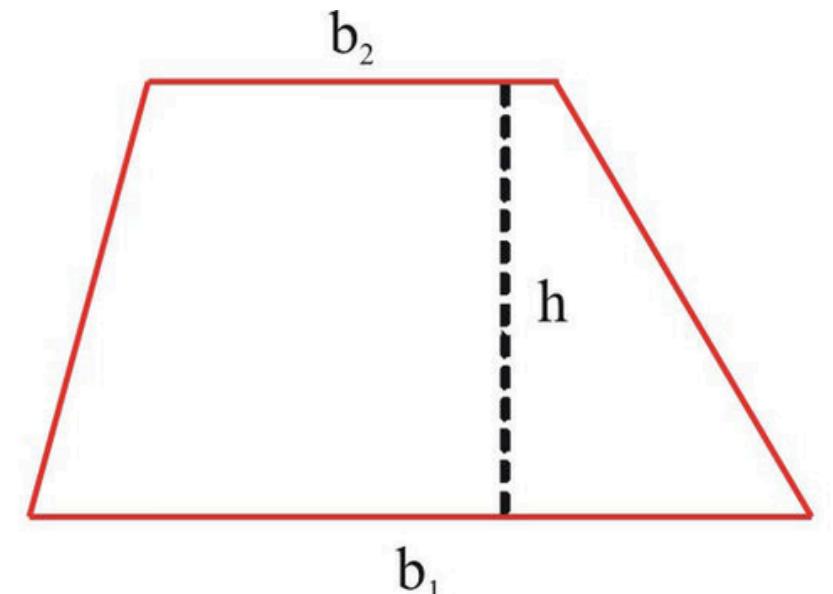
```
calc_area <- function (b1, b2, h) {  
  A <- (b1 + b2) * h / 2  
  return(A)  
}  
  
res <- calc_area(6, 4, 5)
```

> res

25

```
res <- calc_area(6, 5, 4)  
> res
```

?



$$A = \frac{(b_1 + b_2)h}{2}$$

Ejercicios

Calcula el área de un trapezoide cuya base mayor (b_1) es igual a 6, su base menor (b_2) es igual a 4 y su altura es 5.

```
calc_area <- function (b1, b2, h) {  
  A <- (b1 + b2) * h / 2  
  return(A)  
}  
  
res <- calc_area(6, 4, 5)
```

> res

25

! El orden **SÍ** importa

```
res <- calc_area(6, 5, 4)
```

> res X

22

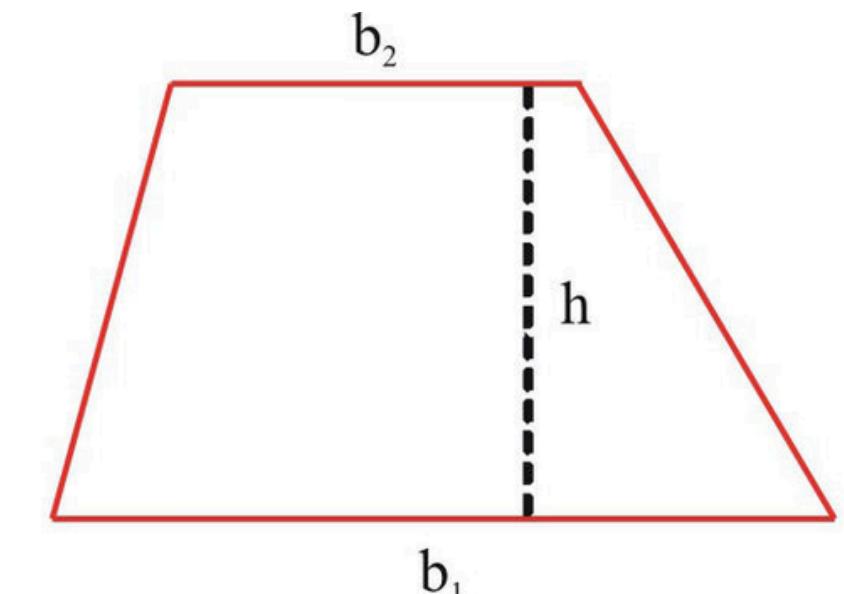
```
res <- calc_area(b1 = 6, h = 5, b2 = 4)
```

> res

25



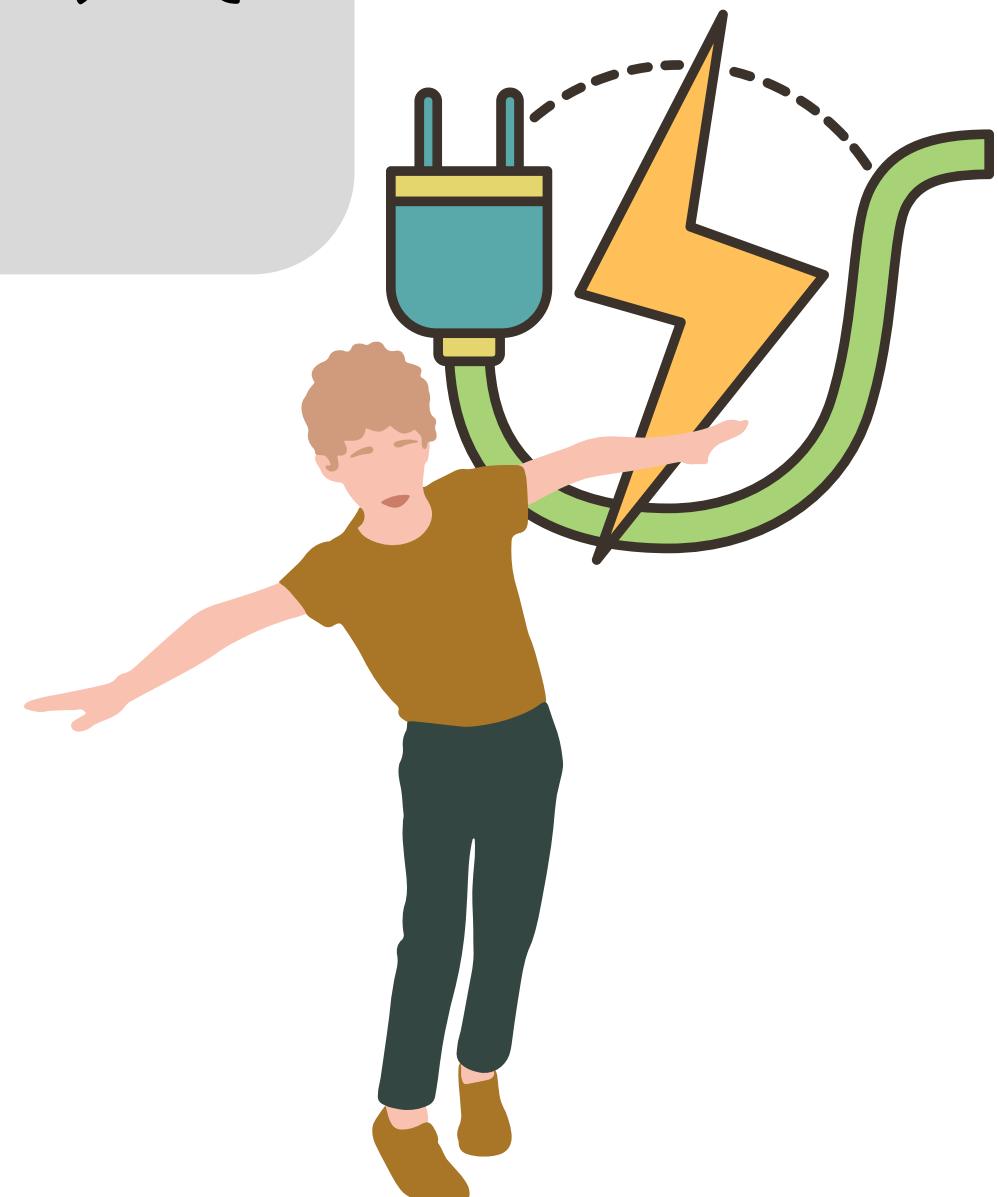
$$A = \frac{(b_1 + b_2)h}{2}$$



Ejercicio. Tasa metabólica basal

```
TMB <- function(sexo, altura, peso, edad) {  
}  
  
resultado_TMB <- TMB( , , )
```

Hombres	$TMB \text{ (kcal)} = (10 \times \text{peso en kg}) + (6,25 \times \text{altura en cm}) - (5 \times \text{edad en años}) + 5$
Mujeres	$TMB \text{ (kcal)} = (10 \times \text{peso en kg}) + (6,25 \times \text{altura en cm}) - (5 \times \text{edad en años}) - 161$



Ecuaciones de Harris-Benedict
revisadas por Mifflin y
St Jeor en 1990

<!--Estudio IMIBIC-->

Enhorabuena {

<"Has superado la tercera
sesión"/>

}

UCAIB BIOINFORMÁTICA Y
BIOESTADÍSTICA
(IMIBIC)

