

“R para Ciencia de Datos”

Programar en R

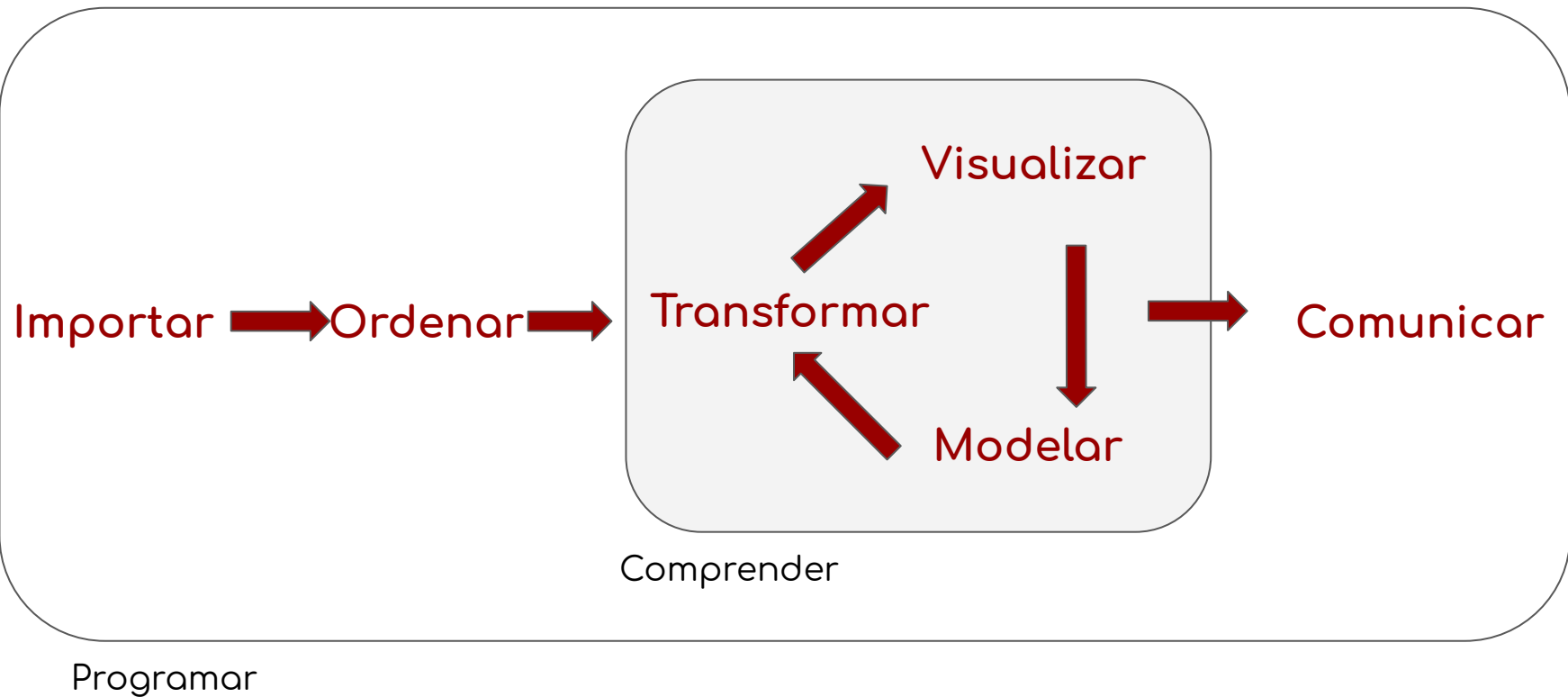


R-Ladies La Paz

MSc. Ing. Ruth Chirinos



“La programación es una destreza transversal en todo el trabajo en ciencia de datos: es necesario usar una computadora para hacer ciencia de datos”



Pipes

Package: magrittr



Ejemplo en R de Funciones:

- La pequeña Alicia corre por el bosque y a su paso va recogiendo frutillas y las guarda en su canasta
 - `alicia1 <- correr(foo_foo, a_traves = bosque)`
 - `alicia2 <- recoger(foo_foo_1, que = frutillas)`
 - `alicia3 <- guardar(foo_foo_2, en = canasta)`
 - `..`
 - `..`
 - `..`

Ejemplo usando Pipe:

- La pequeña Alicia corre por el bosque y a su paso va recogiendo frutillas y las guarda en su canasta

```
> alicia %>%
```

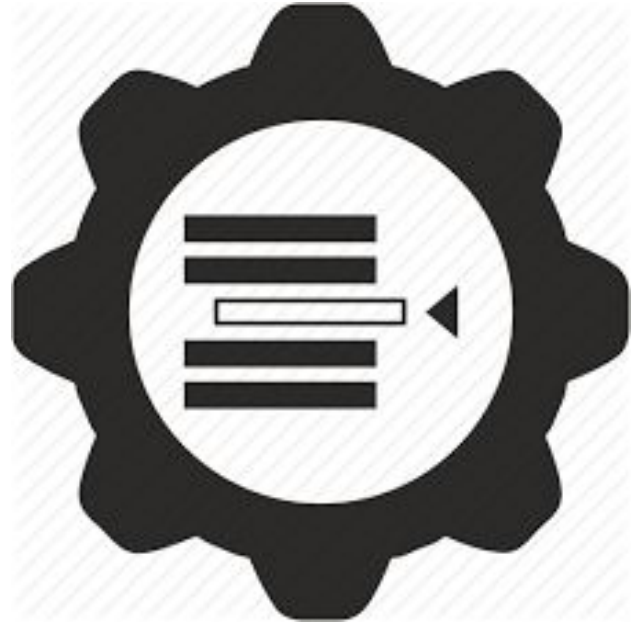
```
  correr(a_través = bosque) %>%
```

```
  recoger(que = frutillas) %>%
```

```
  guardar(en = canasta)
```

Funciones

Package: magrittr



Partes de una función

```
nombreFuncion <- function(arguments) {  
  
    #BODY OF THE FUNCTION  
  
}
```

Ejemplo...

Por qué usar funciones?

```
>df <- tibble::tibble(  
  a = rnorm(10),  
  b = rnorm(10),  
  c = rnorm(10),  
  d = rnorm(10)  
)
```

```
df$a <- (df$a - min(df$a, na.rm = TRUE)) /  
  (max(df$a, na.rm = TRUE) - min(df$a, na.rm = TRUE))
```

```
df$b <- (df$b - min(df$b, na.rm = TRUE)) /  
  (max(df$b, na.rm = TRUE) - min(df$a, na.rm = TRUE))
```

```
df$c <- (df$c - min(df$c, na.rm = TRUE)) /  
  (max(df$c, na.rm = TRUE) - min(df$c, na.rm = TRUE))
```

```
df$d <- (df$d - min(df$d, na.rm = TRUE)) /  
  (max(df$d, na.rm = TRUE) - min(df$d, na.rm = TRUE))
```

Errores

Duplicidad

Analiza tu función

- ¿Cuántos inputs tiene?

```
df$a <- (df$a - min(df$a, na.rm = TRUE)) /  
  (max(df$a, na.rm = TRUE) - min(df$a, na.rm = TRUE))
```

- Convertimos df\$a en variable

```
x <- df$a  
(x - min(x, na.rm = TRUE)) / (max(x, na.rm = TRUE) -  
min(x, na.rm = TRUE))
```

Analiza tu función (Continuación...)

- Usa la función range

```
rng <- range(x, na.rm = TRUE)
(x - rng[1]) / (rng[2] - rng[1])
```

- Finalmente nuestra función rescale01:

```
rescale01 <- function(x) {
  rng <- range(x, na.rm = TRUE, finite=TRUE)
  (x - rng[1]) / (rng[2] - rng[1])
}
```

```
rescale01(c(0, 5, 10))
#> [1] 0.0 0.5 1.0
```

Usando nuestra función

```
df <- tibble::tibble(  
  a = rnorm(10),  
  b = rnorm(10),  
  c = rnorm(10),  
  d = rnorm(10)  
)
```

```
rescale01(df$a)  
rescale01(df$b)  
rescale01(df$c)  
rescale01(df$d)
```

Partes de una función

The diagram illustrates the components of an R function definition. The function name 'rescale01' is highlighted with a red box and labeled 'Nombre'. The argument 'x' is highlighted with a red box and labeled 'Argumentos'. The function body, containing two lines of code, is enclosed in a red box and labeled 'Cuerpo'. The closing brace '}' is labeled 'Retornar una salida'.

```
rescale01 <- function(x) {  
  rng <- range(x, na.rm = TRUE, finite=TRUE)  
  (x - rng[1]) / (rng[2] - rng[1])  
}
```

Nombre

Argumentos

Cuerpo

Retornar una salida

R Functions

Operator	Description	Math
<code>abs(x)</code>	Takes the absolute value of x	
<code>log(x,base=y)</code>	Takes the logarithm of x with base y; if base is not specified, returns the natural logarithm	
<code>exp(x)</code>	Returns the exponential of x	
<code>sqrt(x)</code>	Returns the square root of x	
<code>factorial(x)</code>	Returns the factorial of x (x!)	
<code>mean(x)</code>	Mean of x	Statistics
<code>median(x)</code>	Median of x	
<code>var(x)</code>	Variance of x	
<code>sd(x)</code>	Standard deviation of x	
<code>scale(x)</code>	Standard scores (z-scores) of x	
<code>quantile(x)</code>	The quartiles of x	
<code>summary(x)</code>	Summary of x: mean, min, max etc..	

Condiciones

```
if (this) {  
    # haz aquello  
} else if (that) {  
    # haz otra cosa  
} else {  
    #  
}
```

Condiciones

```
rescale01 <- function(x, value) {  
  
  if(value == 1) {  
  
    rng <- range(x, na.rm = TRUE, finite=TRUE)  
    (x - rng[1]) / (rng[2] - rng[1])  
  
  } else {  
    return("Ha ingresado una config errónea")  
  }  
  
}
```


Vectores

```
library(tidyverse)
```



```
x <- c(1, 2, 3, 4, 5)
```

Vectores Básicos

- Vectores **atómicos**, de los cuales existen seis tipos: **lógico**, **entero**, **doble**, **caracter**, **complejo** y **sin procesar** (*raw*)
- Las **listas**, que a veces son denominadas como vectores recursivos debido a que pueden contener otras listas.

Vectores



Vectores (Continuación...)

Vector 1	Vector 2	Vector 3
Nombre	Puntuación	¿Es posterior a 2005?
Shrek	7.9	Falso
Shrek 2	7.2	Falso
<u>Shrek</u> tercero	6.1	Verdadero
<u>Shrek</u> : Felices por siempre...	6.3	Verdadero

Vectores (Continuación...)

```
# crear vector carácter con nombre de las películas  
nombre <- c("Shrek", "Shrek 2", "Shrek Tercero", "Shrek: Felices por  
siempre")
```

```
# crear vector numérico con puntuación de las películas  
puntuacion <- c(7.9, 7.2, 6.1, 6.3)
```

```
# crear vector lógico sobre si la película es posterior a 2015  
posterior_2005 <- c(FALSE, FALSE, TRUE, TRUE)
```

Vectores (Continuación...)

```
# sumar 2 a la puntuación  
puntuacion + 2
```

```
# dividir la puntuación entre 2  
puntuacion/2
```

```
# crea la puntuación de rafa  
puntuacion_de_rafa <- c(10, 9, 6, 7)
```

```
# calcular diferencia entre puntuaciones  
puntuacion_de_rafa - puntuacion
```

```
# seleccionar la primera y la última película  
nombre[c(1, 4)]
```

```
# seleccionar todas menos la segunda película  
nombre[-2]
```

Listas

```
# Cargando datos de videos anteriores
# informacion peliculas shrek.
nombre <- c("Shrek", "Shrek 2", "Shrek Tercero", "Shrek: Felices por siempre")
puntuacion <- c(7.9, 7.2, 6.1, 6.3)
posterior_2005 <- c(FALSE, FALSE, TRUE, TRUE)

# informacion estrenos de peliculas.
warner <- c(20, 20, 16, 17, 17, 22, 17, 18, 19)
disney <- c(11, 13, 11, 8, 12, 11, 12, 8, 10)
fox <- c(18, 15, 15, 15, 16, 17, 15, 13, 11)

# crear diferentes estructuras de datos en R
vector_titulos <- nombre
matriz_peliculas <- matrix(c(warner, disney, fox),
                           nrow = 9,
                           ncol = 3)
peliculas_df <- data.frame(nombre,
                           puntuacion,
                           posterior_2005)
```

Listas (Continuación...)

```
# crear lista en R
lista_curso <- list(vector_titulos,
                    matriz_peliculas)

# mostrar lista
lista_curso

# Seleccionar vector de la lista
lista_curso[['vector']] # recomiendo usar este

# Seleccionar el tercer elemento del vector de la lista
lista_curso[['vector']][3]

# Seleccionar fila 5 y columna 3 de la matriz de la lista
lista_curso[['matriz']][5,3]

# agregar dataframe a lista
lista_curso[['dataframe']] <- peliculas_df

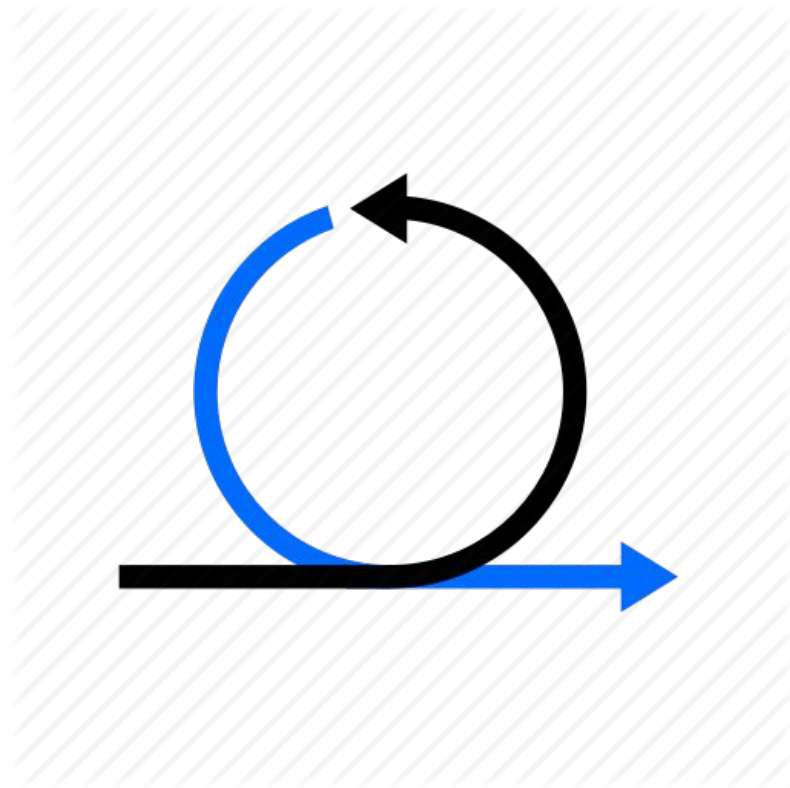
# eliminar un elemento de lista
lista_curso[['vector']] <- NULL

# revisar que no está el vector
lista_curso
```


Iteration

```
library(tidyverse)
```

```
library(datos)
```



Iteraciones

```
#Iterar del 1 al 4
for (i in 1:4) {
  print("Hemoglobina")
}
```

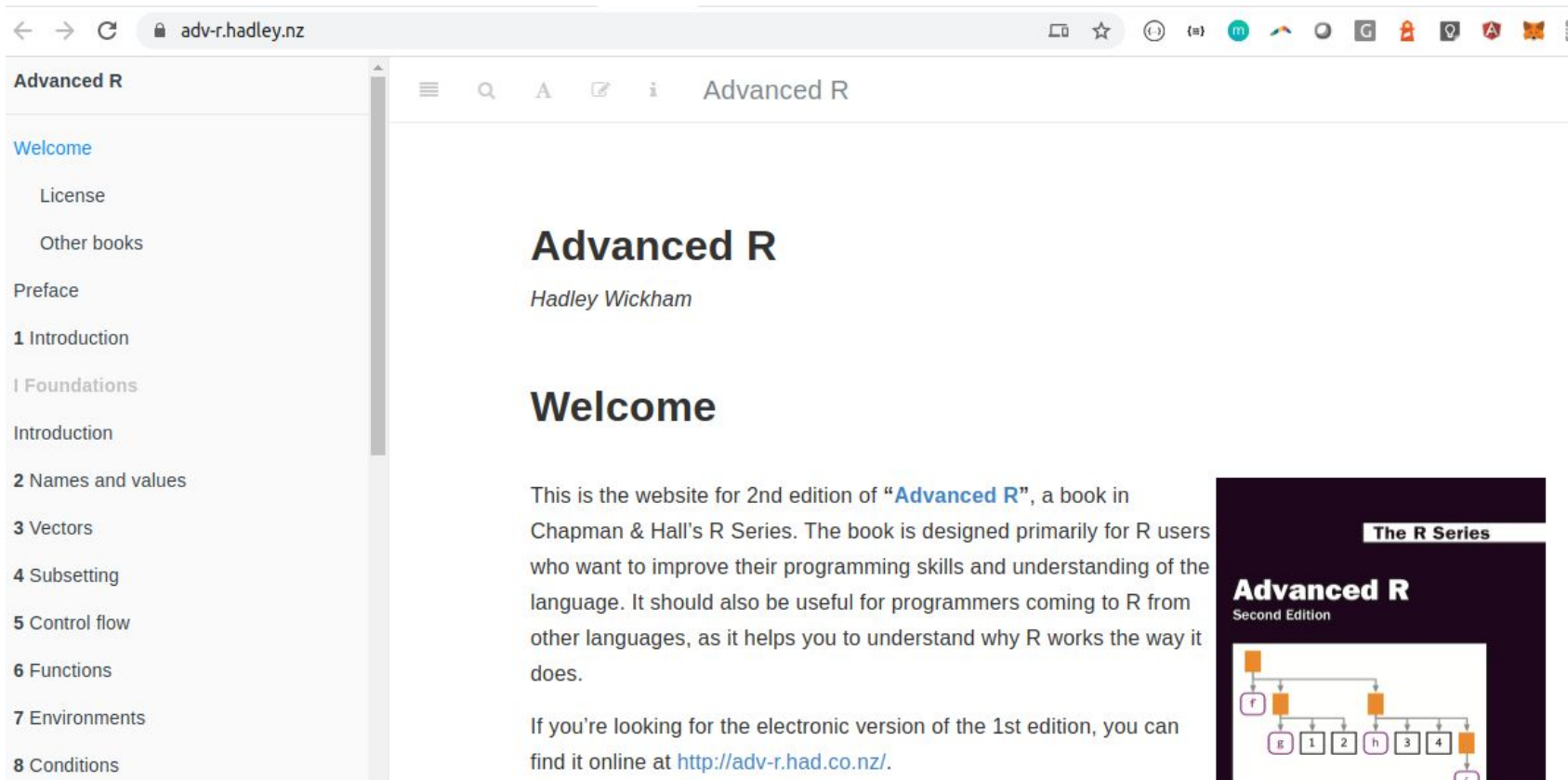
```
#Iterar del 1 al 5
for (i in 1:5) {
  print("El valor i:")
  print(i)
}
```

```
#Iterar un vector
nombre_mascotas <- c("cucho", "bobi", "morti", "sultan", "fido", "perlita")

for (i in 1:length(nombre_mascotas)) {
  print(paste("Mi perro se llamaba:", nombre_mascotas[i]))
}
```

Recursos

<https://adv-r.hadley.nz/>



The screenshot shows a web browser displaying the 'Advanced R' website. The browser's address bar shows 'adv-r.hadley.nz'. The website has a light gray header with the title 'Advanced R' and a navigation menu on the left. The main content area has a large heading 'Advanced R' by 'Hadley Wickham', followed by a 'Welcome' section. The 'Welcome' text describes the book as the 2nd edition of 'Advanced R' in the Chapman & Hall's R Series, designed for R users wanting to improve their programming skills. It also mentions the 1st edition is available at <http://adv-r.had.co.nz/>. The left sidebar contains a table of contents with links to 'Welcome', 'License', 'Other books', 'Preface', '1 Introduction', 'I Foundations', 'Introduction', '2 Names and values', '3 Vectors', '4 Subsetting', '5 Control flow', '6 Functions', '7 Environments', and '8 Conditions'.

Advanced R

Welcome

License

Other books

Preface

1 Introduction

I Foundations

Introduction

2 Names and values

3 Vectors

4 Subsetting

5 Control flow

6 Functions

7 Environments

8 Conditions

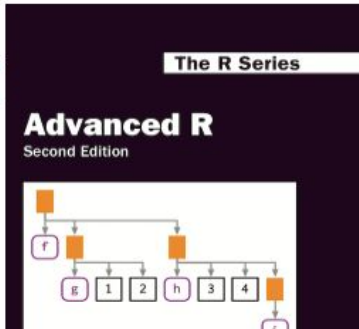
Advanced R

Hadley Wickham

Welcome

This is the website for 2nd edition of “**Advanced R**”, a book in Chapman & Hall’s R Series. The book is designed primarily for R users who want to improve their programming skills and understanding of the language. It should also be useful for programmers coming to R from other languages, as it helps you to understand why R works the way it does.

If you’re looking for the electronic version of the 1st edition, you can find it online at <http://adv-r.had.co.nz/>.



The book cover for 'Advanced R' Second Edition is shown. It features a dark purple background with the title 'Advanced R' in large white letters, and 'Second Edition' below it. The 'The R Series' logo is in the top right. A diagram of the R language structure is visible at the bottom, showing a hierarchy of components like 'f', 'g', '1', '2', 'h', '3', '4', and 'i'.

Thank you!