

Programa de Doctorado

Programación básica



Manejo y visualización de datos en R

Paloma Ruiz Benito, Verónica Cruz Alonso,
Julen Astigarraga Urcelay

Enero - 2022

Contenido

Parte I.- Introducción a R

- Entornos de programación, introducción a R y RStudio.
- Tipos de variables y datos, operaciones aritméticas y lógicas, creación de vectores, matrices, listas y tablas. Selección de datos.

Día 1

Parte II.- Manejo de datos

- Flujo y funciones en la gestión de bases de datos.
- Recomendaciones para la estructura de las bases de datos y creación de código de programación.
- Introducción a la gestión de datos. Recomendaciones para su generación.
- Estructuras de programación: condicionales, bucles y funciones.

Parte III.- Visualización de datos

- Funciones básicas e introducción a ggplot para la visualización de datos.
- Generación de gráficos unidimensionales: histograma, dispersión, gráfico de cajas y bigotes, etc.
- Generación de gráficos bidimensionales: dispersión, boxplot, gráficos de barras, etc.
- Ejemplos y prácticas de visualización de gráficos en mapas.

Parte IV.- Trabajo reproducible

- Trabajo reproducible.
- Introducción a git y github.
- Rmarkdown.

Base R

Cheat Sheet

Getting Help

Accessing the help files

?mean

Get help of a particular function.

help.search('weighted mean')

Search the help files for a word or phrase.

help(package = 'dplyr')

Find help for a package.

More about an object

str(iris)

Get a summary of an object's structure.

class(iris)

Find the class an object belongs to.

Using Packages

install.packages('dplyr')

Download and install a package from CRAN.

library(dplyr)

Load the package into the session, making all its functions available to use.

dplyr::select

Use a particular function from a package.

data(iris)

Load a built-in dataset into the environment.

Working Directory

getwd()

Find the current working directory (where inputs are found and outputs are sent).

setwd('C://file/path')

Change the current working directory.

Use projects in RStudio to set the working directory to the folder you are working in.

Vectors

Creating Vectors

c(2, 4, 6)	2 4 6	Join elements into a vector
2:6	2 3 4 5 6	An integer sequence
seq(2, 3, by=0.5)	2.0 2.5 3.0	A complex sequence
rep(1:2, times=3)	1 2 1 2 1 2	Repeat a vector
rep(1:2, each=3)	1 1 1 2 2 2	Repeat elements of a vector

Vector Functions

sort(x)

Return x sorted.

table(x)

See counts of values.

rev(x)

Return x reversed.

unique(x)

See unique values.

Selecting Vector Elements

By Position

x[4] The fourth element.

x[-4] All but the fourth.

x[2:4] Elements two to four.

x[-(2:4)] All elements except two to four.

x[c(1, 5)] Elements one and five.

By Value

x[x == 10] Elements which are equal to 10.

x[x < 0] All elements less than zero.

x[x %in% c(1, 2, 5)] Elements in the set 1, 2, 5.

Named Vectors

x['apple'] Element with name 'apple'.

Programming

For Loop

```
for (variable in sequence){  
  Do something  
}
```

Example

```
for (i in 1:4){  
  j <- i + 10  
  print(j)  
}
```

While Loop

```
while (condition){  
  Do something  
}
```

Example

```
while (i < 5){  
  print(i)  
  i <- i + 1  
}
```

If Statements

```
if (condition){  
  Do something  
} else {  
  Do something different  
}
```

Example

```
if (i > 3){  
  print('Yes')  
} else {  
  print('No')  
}
```

Functions

```
function_name <- function(var){  
  Do something  
  return(new_variable)  
}
```

Example

```
square <- function(x){  
  squared <- x*x  
  return(squared)  
}
```

Reading and Writing Data

Also see the **readr** package.

Input	Output	Description
df <- read.table('file.txt')	write.table(df, 'file.txt')	Read and write a delimited text file.
df <- read.csv('file.csv')	write.csv(df, 'file.csv')	Read and write a comma separated value file. This is a special case of read.table/write.table.
load('file.Rdata')	save(df, file = 'file.Rdata')	Read and write an R data file, a file type special for R.

Conditions

a == b	Are equal	a > b	Greater than	a >= b	Greater than or equal to	is.na(a)	Is missing
a != b	Not equal	a < b	Less than	a <= b	Less than or equal to	is.null(a)	Is null

Programación básica en R

Entorno de trabajo en R studio

Usando paquetes y librerías

Pidiendo ayuda y viendo ejemplos

Directorio de trabajo

Creación vectores: asignación, tipos de variables, operaciones aritméticas básicas

Tipo de objetos: matriz, data frame, lista

Operaciones lógicas, seleccionar datos, cambiar entre tipos de objetos

1 – Programación básica en R

Usando paquetes y librerías

Using Packages

install.packages('dplyr')

Download and install a package from CRAN.

library(dplyr)

Load the package into the session, making all its functions available to use.

dplyr::select

Use a particular function from a package.

data(iris)

Load a built-in dataset into the environment.

1 – Programación básica en R

Pidiendo ayuda y viendo ejemplos

Getting Help

Accessing the help files

?mean

Get help of a particular function.

help.search('weighted mean')

Search the help files for a word or phrase.

help(package = 'dplyr')

Find help for a package.

More about an object

str(iris)

Get a summary of an object's structure.

class(iris)

Find the class an object belongs to.

1 – Programación básica en R

Pidiendo ayuda y viendo ejemplos



Si **necesitas ayuda** y **no encuentras el problema** puedes encontrarla en: stackoverflow (<http://stackoverflow.com>) y R-help mailing list

Asegurate:

- (1) Tener la última version de R y del paquete instalada.
- (2) Crea un ejemplo reproducible para que la comunidad pueda ayudarte (<http://stackoverflow.com/questions/5963269>).
- (3) Mira si alguien ha preguntado previamente sobre tu problema, es mucho más rápido que esperar una respuesta.

1 – Programación básica en R

Directorio de trabajo

Working Directory

getwd()

Find the current working directory (where inputs are found and outputs are sent).

setwd('C://file/path')

Change the current working directory.

Use projects in RStudio to set the working directory to the folder you are working in.

1 – Programación básica en R

Creación vectores: asignación, tipos de variables, operaciones aritméticas básicas

```
> 1
> 2+1
3
> A<-2
> B<-1
> A+B
3
> A<- "Hola"
> A
```



Los **vectores** son variables con **uno o más valores del mismo tipo** (numero entero o doble, carácter, etc).

En R una variable con una única posición es un vector de una posición en vez de escalar. Los vectores pueden estar vacíos (tener una longitud de 0) y esto puede ser útil cuando escribáis funciones

1 – Programación básica en R

Creación vectores: asignación, tipos de variables, operaciones aritméticas básicas

Número (entero, real)

Cadena de texto (“a character string”)

Factor (with factor levels)

Booleano (TRUE-FALSE)

```
> A <- 6
```

```
> A <- 66.6
```

```
> A <- "Hola Mundo"
```

Los **vectores** tienen tres características clave:

1) Tipo de variable: ¿Qué es?

`typeof(nombre)`

2) Longitud: ¿Cuántos elementos contiene?

`length(nombre)`

3) Atributos: metadatos adicionales

`attributes(nombre)`



¿Cuál es la diferencia entre un vector
y una lista?



¡¡En un vector todos los elementos son del mismo tipo, mientras que una lista puede estar compuesta por elementos de distinto tipo!!

1 – Programación básica en R

Creación vectores: asignación, tipos de variables, operaciones básicas

Operaciones aritméticas

Operador	Descripción
+	Suma
-	Resta
*	Multiplicación
/	División
^	Exponente

1 – Programación básica en R

Creación vectores: asignación, tipos de variables, operaciones aritméticas básicas



Vector

(colección ordenada de elementos del mismo tipo)

```
> A <- c(3, 5)
```

```
> A[1]
```

```
> seq(9, 1, -2)
```

```
[1] 9 7 5 3 1
```

```
> seq(9, 1, length=7)
```

```
[1] 9.000000 7.666667 6.333333 5.000000  
3.666667 2.333333 1.000000
```

```
> rep(1:3, 4)
```

```
[1] 1 2 3 1 2 3 1 2 3 1 2 3
```

Vectors

Creating Vectors

c(2, 4, 6)	2 4 6	Join elements into a vector
2:6	2 3 4 5 6	An integer sequence
seq(2, 3, by=0.5)	2.0 2.5 3.0	A complex sequence
rep(1:2, times=3)	1 2 1 2 1 2	Repeat a vector
rep(1:2, each=3)	1 1 1 2 2 2	Repeat elements of a vector

1 – Programación básica en R

Creación vectores: asignación, tipos de variables, operaciones aritméticas básicas

```
> rep(1:4, 2)
[1] 1 2 3 4 1 2 3 4
> rep(1:4, each = 2)
[1] 1 1 2 2 3 3 4 4
> rep(1:4, each = 2, times = 3)
[1] 1 1 2 2 3 3 4 4 1 1 2 2
[13] 3 3 4 4 1 1 2 2 3 3 4 4
> rep(1:4, 1:4)
[1] 1 2 2 3 3 3 4 4 4 4
>
rep(c("cat", "dog", "gerbil", "goldfish", "rat"), c(2, 3, 2,
1, 3))
[1] "cat" "cat" "dog" "dog" "dog" "gerbil"
[7] "gerbil" "goldfish" "rat" "rat" "rat"
```


1 – Programación básica en R

Creación vectores: asignación, tipos de variables, operaciones aritméticas básicas

```
y <- rnorm(1000, mean=3, sd=2)
```

1 – Programación básica en R

Creación vectores: asignación, tipos de variables, operaciones aritméticas básicas

Funciones matemáticas

Función	Descripción
log(x)	Logaritmo natural
exp(x)	Exponencial
max(x)	Máximo
min(x)	Minimo
round(x,n)	Redondear decimales
cor(x)	Correlación
sum(x)	Suma

1 – Programación básica en R

Creación vectores: asignación, tipos de variables, operaciones aritméticas básicas

Funciones matemáticas

Función	Descripción
mean(x)	Media
median(x)	Mediana
quantile(x)	cuantil
rank(x)	Ranking de elementos
var(x)	Varianza
sd(x)	Desviación estándar

1 – Programación básica en R

Creación vectores: asignación, tipos de variables, operaciones aritméticas básicas

Operadores lógicos and relacionales

Operador	Descripción
>	Mayor que
<	Menor que
>=	Menor o igual que
<=	Mayor o igual que
==	Exactamente igual a
!=	No igual a

Operator	Descripción
!	Logico NO
&	Logico Y
	Logico O
&&	Y con IF
	O con IF
%in%	Que contiene

1 – Programación básica en R

Creación vectores: asignación, tipos de variables, operaciones aritméticas básicas

Indexación con operadores lógicos

```
> vv <- c(4, 5, 6, 24, 5234, 46, 763452)
> vv[4]
[1] 24
> vv<20
[1] TRUE TRUE TRUE FALSE FALSE FALSE FALSE
> vv[vv<20]
[1] 4 5 6
> vv[1:3]
[1] 4 5 6
> cc<-c(4, 5, 6)
> vv %in% cc
> vv [vv %in% cc]
```

1 – Programación básica en R



Tipo de objetos básicos: `vector`, `matriz`, `data frame`, `lista`

- Almacenar información
 - **Vector**
 - **Matriz**
 - **Data frame**
 - **Lista**

¿Sabes las principales diferencias entre vector, lista, matriz y data.frame?



	Homogéneo	Heterogéneo
1-dimensión	Vector	Lista
2-dimensiones	Matriz	Data frame

La mejor manera de entender la estructura de un objeto en R es
`str()`

1 – Programación básica en R

Tipo de objetos: matriz, data frame, lista



Matriz

(una matriz es un vector con un atributo dimensión).

```
> MA <- matrix(3, 5)
```

```
> x<-1:10
```

```
> matrix(x, 5, 2)
```

	[, 1]	[, 2]
[1,]	1	6
[2,]	2	7
[3,]	3	8
[4,]	4	9
[5,]	5	10

1 – Programación básica en R

Tipo de objetos: matriz, data frame, lista

Matriz

```
> X <- c(45, 87, 22, 1, 66, 47, 89, 23, 71)
> X <- matrix(x, 3, 3)
> colnames(X) <- c('peso', 'numero de hijos', 'dinero')
> rownames(X) <- c('Javier', 'Paco', 'Amancio')
> dimnames(X) <-
list(c('Julio', 'Pablo', 'Pedro'), c('X', 'Y', 'Z'))
```

	peso	numero de hijos	dinero
Mariano	45	1	89
Paco	87	66	23
Amancio	22	47	71

1 – Programación básica en R

Tipo de objetos: matriz, data frame, lista



Lista

(conjunto de objetos que pueden ser de distinta naturaleza)

```
> Lista<-  
list(c(56,76,84,91),c(1.70,1.71,1.87,1.90),c("loco",  
"estudiante","votante"))  
> Lista  
[[1]]  
[1] 56 76 84 91  
[[2]]  
[1] 1.70 1.71 1.87 1.90  
[[3]]  
[1] "loco"          "estudiante" "votante"  
> Lista[[1]]  
[1] 56 76 84 91  
> Lista[[1]][1]  
[1] 56
```

1 – Programación básica en R

Tipo de objetos: matriz, data frame, lista



Data.frame

(las columnas pueden ser de distinto tipo de datos y tener nombre, todas tienen la misma longitud)

```
> a <- c(10,20,30,40)
> b <- c("libro", "bolígrafo", "manual", "estuche")
> c <- c(TRUE,FALSE,TRUE,FALSE)
> d <- c(2.5, 8, 10, 7)
> df <- data.frame(a,b,c,d)
> names(df) <- c('ID', 'items', 'store', 'price')
> df
```

	ID	items	store	price
1	10	libro	TRUE	2.5
2	20	bolígrafo	FALSE	8.0
3	30	manual	TRUE	10.0
4	40	estuche	FALSE	7.0

1 – Programación básica en R

Operaciones lógicas, seleccionar datos,
cambiar entre tipos de objetos

```
vector1 <- c(5,10,23,44,65)
vector2 <- c(T,F,T)
vector3 <- c("red","blue","yellow","green")
```

```
#hacer un data.:
mydataframe<- data.frame (vector1, vector2,
vector1)
```

```
mydataframe [1,2]
mydataframe [c(2,3), 1]
mydataframe [-c(2,5), 1:3]
mydataframe [2, ]
mydataframe [,2]
mydataframe [vector1 > 3,2]
```

1 – Programación básica en R

Operaciones lógicas, seleccionar datos, cambiar entre tipos de objetos

```
Mymatrix <- matrix(c(1:25), nrow = 5, ncol = 5)
```

```
class(mymatrix)
```

```
colnames(mymatrix) <- c(letters [1:5])
```

```
names(mymatrix)
```

```
rownames(mymatrix)
```

```
dim(mymatrix)
```

```
nrow(mymatrix)
```

```
ncol(mymatrix)
```

```
mymatrix[1,4]
```

```
Nombres <- c("Temp", "Prec", "alt")
```

```
m_dataframe <- as.data.frame (mymatrix)
```

```
names(m_dataframe) <- nombres
```

```
class(m_dataframe)
```

1 – Programación básica en R



Operaciones lógicas, seleccionar datos,
cambiar entre tipos de objetos

Cambios entre tipos de objetos

	A vector más largo	A matriz	A data frame
de vector	<code>c(x, y)</code>	<code>cbind(x, y)</code> <code>rbind(x, y)</code>	<code>data.frame(x, y)</code>
de matriz	<code>as.vector(mym atrix)</code>		<code>as.data.frame (mymatrix)</code>
de data frame		<code>as.matrix(myf rame)</code>	

1 – Programación básica en R

Operaciones lógicas, seleccionar datos, cambiar entre tipos de objetos

```
> data.frame(ID = c(10,20,30,40),  
  items = c("libro", "bolígrafo", "manual", "estuche"),  
  store = c(TRUE,FALSE,TRUE,FALSE),  
  price = c(2.5, 8, 10, 7))  
> cantidad<- c(10, 35, 40, 5)  
> df$cantidad <- cantidad  
> df$cantidad <- NA  
> expensive <- subset(df, price > 5)  
> expensive$store <- subset(df, price > 5 & store == T)  
> sel<-c("libro","manual")  
> kk<-df[df$items %in% sel,]
```


use **R**!