Notas sobre análisis de datos en R

Curso de edX: 'Data analysis for life sciences' (07-2020)

Jhonny Lanzusi

Índice

Resumen

Notas del curso de edX, algo como un resumne de lo que voy aprendiendo

1. La doble linea a la derecha representa bloques de código. Los # marcan comentarios, y las lineas con ## marcan el resultado de ejecutar un comando

1 Cosas básicas de R

Lo básico: operaciones, aasignación, `for loops` y la función `which`.

1.1 Operaciones básicas y asignación

En R se puede operar como una simple calculadora, por ejemplo,[1]

```
(5+27)*2
## 64
```

Los operadores aritméticos de R que he usado estan en el cuadro 1, y los operadores lógicos —cuyo output es `false` o `true`— estan en el cuadro 2.

Como R es un lenguage de programación se pueden asignar variables, por ejemplo,

```
x <- 5+7
```

asigna el valor de 5+7 a la variable x. La asignación de puede usar con muchos tipos de elemntos en R.

1.2 Vectores y matrices

Cada vez que se da a R un número, este entiende en realidad un vector con una sola componente. Para construir vectores se usa la funcion `c()`, por ejemplo,

```
x <- c(2.23, 3.45, 1.87, 2.11, 7.33, 18.34)
```

asigna los valores entre parénteris a la variable x.

Los vectores en R pueden ser de dos tipos: atomic y lists. Los de tipo atomic son vectores en los que solo hay un tipo de datos, mientras que en las listas pueden haber tipos de datos distintos. Los vectores atomic son los mas sencilos, sus tipos se pueden ver en el cuadro 3

Cuadro 1: Operadores aritméticos

+	Suma
-	Resta
*	Multiplicación
\	División
^	Exponenciación.

Cuadro 2: Operadores Lógicos

`>, >=`	Mayor, mayor e igual.
`<, <=`	Menor, menor e igual.
`=!`	No igual
`a | b`	`True` si alguno de `a` o `b` es verdadero.
`a & b`	`True` si ambos, `a` y `b`, son verdaderos.
`isTRUE(x)`	Checkea si `x` es verdadero.

Cuadro 3: Tipos de vectores 'atomic'

numeric	Valores numéricos
logical	Valores del tipo TRUE y FALSE
character	Caracteres de texto

También existe la función **`vector()`** que permite crear un vector especificando su tipo (numeric, logical, etc) y despues el tamaño, por ejemplo

```
vec1 <- vector("numeric", 100)
```

crea un vector de tipo numérico con 100 entradas, las cuales aún no estan llenas.

1.3 Sequencias de números

Las dos maneras mas sencillas de crear sequencias de números son la funcione **`seq`** y el operador `:`, por ejemplo,[2]

```
1:20
## 1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18
   ↪ 19 20
pi:10
## 3.141593 4.141593 5.141593 6.141593 7.141593
   ↪ 8.141593 9.141593
15:1
## 15 14 13 12 11 10 9 8 7  6  5  4  3  2  1
```

2. El símbolo ↪ representa la continuación de la linea anterior

la función **`seq`** se comporta de manera similar,

```
seq(1,20)
## 1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18
   ↪ 19 20
```

pero tiene mas opciones que el operador `:`.

La función **`rep`** permite 'repetir'.

1.4 Bucles usando `for`

Se pueden hacer bucles para ejercutar el mismo código varias veces, por ejemplo el siguiente código suma los cuadrados de los primeros 25 números:

```
y <- 0

```

```
for (a in 1:25) {
  y[a] <- a^2
}
sum(y)
```

Muchos mas ejemplos de bucles se verán a lo largo de las notas.

2 Manejo de datos

Comenzamos leyendo un archivo con los datos y guardandolos en una variable,

```
miceData <- read.csv("femaleMiceWeights.csv")
head(miceData)
##   Diet Bodyweight
## 1 chow     21.51
## 2 chow     28.14
## 3 chow     24.04
## 4 chow     23.45
## 5 chow     23.68
## 6 chow     19.79
```

2.1 Introduccion a `dplyr`

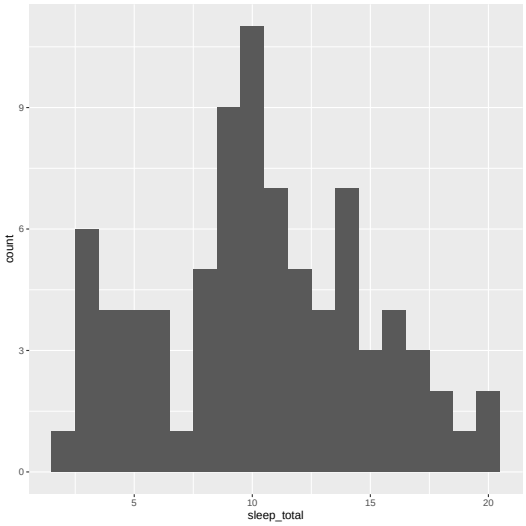
3 Visualizacion de datos

Usaremos herramientas básicas de visualizacion de datos partiendo de `mamsleep` (ver apartado 2.1). Para darnos una idea de como luce partes de los datos, veamos un histograma del total de sueño de los mamiferos.[3]

```
library(tidyverse)
plots <- list()
mamSleep_sleep <- mamSleep$sleep_total
( plots[[1]] <-
  ggplot(data = mamSleep, mapping = aes(x = sleep_
      ↪ total))+
  geom_histogram(binwidth = 1)
)
```

3. La notación `(plots[[i]] <-...)` puede ser ignorada con tranquilidad, es simplemente una lista que va guardando cada grafico.

Figura 1: Total de sueño, histograma.



A Manejo de archivos

En R se pueden manipular archivos usando funciones propias de R, sin depender de las herramientas específicas de cada sistema operativo. Por ejemplo, la función `getwd()` da el directorio actual.