

ISLR Capítulo 2. Laboratorio de R

Javier Santibáñez

22 de agosto de 2015

Introducción

Este documento corresponde al laboratorio de R del capítulo 2 del libro *Introduction to Statistical Learning with applications in R*. Básicamente se reproduce el código en el texto y se agregan los comentarios pertinentes.

Antes de iniciar se fija el directorio de trabajo para conservar los archivos correspondientes.

```
setwd('C:/Users/Javier/OneDrive/Documentos/Ejercicios resueltos/ISLR/Capítulo 2')
```

Comandos básicos

Para crear un vector podemos utilizar la función `c`. El siguiente comando crea un vector llamado `x` con valores 1, 3, 2 y 5. Si queremos ver el valor del vector simplemente escribimos su nombre.

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

```
## [1] 1 3 2 5
```

También podemos guardar valores en objetos si escribimos `=` en lugar de `<-`.

```
x = c(1, 6, 2)
x
```

```
## [1] 1 6 2
```

```
y = c(1, 4, 3)
```

Para sumar dos vectores (elemento a elemento) utilizamos el comando `+`. Esta operación se puede realizar sólo si los vectores tienen la misma longitud (o bien si la longitud de uno es múltiplo de la longitud del otro). Para verificar que esto se cumpla podemos utilizar la función `length()`.

```
length(x)
```

```
## [1] 3
```

```
length(y)
```

```
## [1] 3
```

```
x + y
```

```
## [1] 2 10 5
```

La función `ls()` imprime una lista de los objetos que hemos creado durante la sesión actual (variables, funciones o conjuntos de datos). Por ejemplo, en esta sesión únicamente hemos declarado las variables `x` y `y`.

```
ls()
```

```
## [1] "x" "y"
```

La función `rm()` nos permite eliminar objetos de nuestro espacio de trabajo. Por ejemplo, podemos borrar los objetos que hemos creado en esta sesión. Escribiremos nuevamente `ls()` para verificar que los objetos han sido eliminados.

```
rm(x, y)
ls()
```

```
## character(0)
```

Podemos utilizar la combinación `rm(list = ls())` para eliminar todos los objetos en nuestro espacio de trabajo.

La función `matrix()` nos permite crear matrices en R. Se puede consultar la ayuda sobre esta función con los comandos `?matrix` o `help("matrix")`. Esta función tiene tres argumentos principales: las entradas de la matriz, el número de columnas y el número de filas. Consideremos el siguiente ejemplo:

```
x <- matrix(data = c(1, 2, 3, 4), nrow = 2, ncol = 2)
x
```

```
##      [,1] [,2]
## [1,]    1    3
## [2,]    2    4
```

Como en cualquier función de R, podemos omitir los nombres de los argumentos, sólo debemos recordar declararlos en el orden correcto.

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

R rellena por defecto la matrix por columnas. Si queremos cambiar esto agregamos la opción `byrow = TRUE`.

```
matrix(c(1, 2, 3, 4), 2, 2, byrow = TRUE)
```

```
##      [,1] [,2]
## [1,]    1    2
## [2,]    3    4
```

Podemos utilizar la función `sqrt()` para obtener la raíz cuadrada elemento a elemento, es decir, la matriz (o vector) cuyas entradas son las raíces cuadradas de las entradas en el argumento.

```
sqrt(x)
```

```
##      [,1]      [,2]
## [1,] 1.000000 1.732051
## [2,] 1.414214 2.000000
```

El operador `^` se aplica elemento a elemento y nos permite obtener cualquier potencia del argumento. Por ejemplo, `x^2` nos devuelve la matriz cuyos entradas son los cuadrados de las entradas de la matriz `x`.

```
x^2
```

```
##      [,1] [,2]  
## [1,]    1    9  
## [2,]    4   16
```

La función `rnorm()` nos permite crear una serie de números (pseudo)aleatorios de una distribución normal. Los argumentos principales son: `n` = número de observaciones, `mean` = media, `sd` = desviación estándar. Los valores por defecto son `mean = 0` y `sd = 1`.

El siguiente código genera una muestra aleatoria de tamaño 50 de una distribución normal estándar y los guarda en `x`, después crea una variable `y` correlacionada con `x`. Para verificar la correlación entre las dos variables se utiliza la función `cor()`. Para más información sobre esta última función se pueden usar los comandos `?cor` o `help("cor")`.

```
x <- rnorm(50)  
y <- x + rnorm(50, mean = 50, sd = 0.1)  
cor(x, y)
```

```
## [1] 0.995628
```

Existen otras funciones en R que nos permiten generar muestras aleatorias de distintas distribuciones. Para más información se puede usar el comando `?distributions`.

Cuando se requiere reproducir resultados que dependen de cantidades aleatorias es aconsejable fijar una semilla para generar los números aleatorios, esto nos permitirá generar siempre el mismo conjunto de números. La función es `set.seed()` y toma como argumento cualquier entero arbitrario (excepto 0, ya que esto nos fija como semilla una cantidad que depende del reloj del sistema, que prácticamente vuelve a ser no reproducible).

Gráficos