

Ejercicios Introducción a R (2)

David Criado Ramón

9/11/2019

Introducción a R

1. Matrices

Crea los siguientes vectores

```
new_hope <- c(460.998007, 314.4)
empire_strikes <- c(290.475067, 247.9)
return_jedi = c(309.306177, 165.8)
```

Los datos se corresponden con las ventas en millones de la trilogía de la guerra de las galaxias. El primer número se corresponde a las ventas en US, y el segundo al resto de países. Recuerda que el decimal en los países anglosajones se expresa con un “.”.

- a) Construye la matriz `star_wars_matrix` con esos vectores.

```
star_wars_matrix <- matrix(c(new_hope, empire_strikes, return_jedi), 3, 2, byrow=T)
star_wars_matrix
```

```
##           [,1] [,2]
## [1,] 460.9980 314.4
## [2,] 290.4751 247.9
## [3,] 309.3062 165.8
```

- b) Añádeles nombres a las columnas y las filas de la matriz según las descripciones dadas anteriormente de los datos.

```
rownames(star_wars_matrix) <- c("New Hope", "Empire Strikes", "Return Jedi")
colnames(star_wars_matrix) <- c("Ventas USA", "Ventas resto países")
star_wars_matrix
```

```
##           Ventas USA Ventas resto países
## New Hope           460.9980           314.4
## Empire Strikes      290.4751           247.9
## Return Jedi         309.3062           165.8
```

- c) Calcula las ganancias mundiales de cada película y guárdalas en un vector que se llame `worldwide_vector`

```
worldwide_vector = star_wars_matrix[,1] + star_wars_matrix[,2]
worldwide_vector
```

```
##           New Hope Empire Strikes      Return Jedi
##           775.3980          538.3751          475.1062
```

- d) Añade este último vector como una columna nueva a la matriz `star_wars_matrix` y asigna el resultado a `all_wars_matrix`. Usa para ello la función `cbind()`.

```
all_wars_matrix <- cbind(star_wars_matrix, worldwide_vector)
all_wars_matrix
```

```
##           Ventas USA Ventas resto países worldwide_vector
## New Hope      460.9980           314.4       775.3980
## Empire Strikes 290.4751           247.9       538.3751
## Return Jedi    309.3062           165.8       475.1062
```

- e) Calcula las ganancias totales en USA y fuera de USA para las tres películas. Puedes usar para ello la función `colSums()`

```
colSums(star_wars_matrix)
```

```
##           Ventas USA Ventas resto países
##           1060.779           728.100
```

- f) Calcula la media de ganancias para todas las películas fuera de los estados unidos. Asigna a esa media la variable `non_us_all`

```
non_us_all <- mean(star_wars_matrix[,2])
non_us_all
```

```
## [1] 242.7
```

- g) Haz lo mismo pero sólo para las dos primeras películas. Asigna el resultado a la variable `non_us_some`

```
non_us_some <- mean(star_wars_matrix[1:2,2])
non_us_some
```

```
## [1] 281.15
```

- h) Calcula cuantos visitantes hubo para cada película en cada área geográfica. Ya tienes las ganancias totales en `star_wars_matrix`. Asume que el precio de las entradas es de cinco euros/dólares. (Nota: Las ganancias totales para cada película dividido por el precio del ticket te da el número de visitantes)

```
M <- star_wars_matrix * 100000 / 5
colnames(M) <- c("Entradas USA", "Entradas resto países")
M
```

```
##           Entradas USA Entradas resto países
## New Hope      9219960           6288000
## Empire Strikes 5809501           4958000
## Return Jedi    6186124           3316000
```

2. Subsetting matrices y arrays

Como hemos visto en teoría la sintaxis para acceder tanto a las matrices como a arrays bidimensionales es `array[rows, columns]`. Muchas funciones de R necesitan una matriz como dato de entrada. Si algo no funciona recuerda convertir el objeto a una matriz con la función `as.matrix(iris)`. Crea un array `i <- array(c(1:10),dim=c(5,2))`. ¿Que información te dan los siguientes comandos?

```
i <- array(c(1:10),dim=c(5,2))
i
```

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

```
dim(i)
```

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

dim(i) devuelve un vector con las dimensiones del array.

```
nrow(i)
```

```
## [1] 5
```

nrow(i) devuelve el número de filas del array, es igual que dim(i)[1].

```
ncol(i)
```

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

ncol(i) devuelve el número de columnas del array, es igual que dim(i)[2].

Descárgate el fichero array_datos.txt de PRADO (Datos/)

- impórtalo en tu workspace teniendo en cuenta que es un texto tabulado.

```
datos <- read.table("array_datos.txt")
datos
```

```
##   edad peso altura
## 1   20   65   174
## 2   22   70   180
## 3   19   68   170
```

- crea un documento con los mismos datos pero en formato csv en vez de tab separated.

```
write.csv2(datos, file = "array_datos.csv", row.names = F)
```