Introducción a R: Ejercicios Día 2

Juan Ramón Gómez Berzosa

17/10/2018

Ejercicio 1: Matrices

*** Ejecuta los siguientes comandos.**

```
matrix(data=5, nr=2, nc=2)
```

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

```
matrix(1:6, 2, 3)
```

```
##      [,1] [,2] [,3]
## [1,]    1    3    5
## [2,]    2    4    6
```

```
matrix(1:6, 2, 3, byrow=TRUE)
```

```
##      [,1] [,2] [,3]
## [1,]    1    2    3
## [2,]    4    5    6
```

*** Crea un vector z con los 30 primeros números y crea con el una matriz m con 3 filas y 10 columnas.**

```
z = c(1:30)

m = matrix(z,nrow= 3)
m
```

```
##      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10]
## [1,]    1    4    7   10   13   16   19   22   25    28
## [2,]    2    5    8   11   14   17   20   23   26    29
## [3,]    3    6    9   12   15   18   21   24   27    30
```

*** Escribe la tercera columna en un vector**

```
vector = m[,3]
vector
```

```
## [1] 7 8 9
```

*** Create in R the matrices**

```r
x = c(3,-1,21,1)
x = matrix(x,nrow = 2)
x
```

```
##      [,1] [,2]
## [1,]    3   21
## [2,]   -1    1
```

```r
y = c(1,0,4,1,0,-1)
y= matrix(y,nrow = 2)
y
```

```
##      [,1] [,2] [,3]
## [1,]    1    4    0
## [2,]    0    1   -1
```

*** Calcula los efectos de los siguientes comandos**

(a) x[1,]

(b) x[2,]

(c) x[,2]

(d) y[1,2]

(e) y[,2:3]

```r
x[1,]
```

```
## [1]  3 21
```

```r
x[2,]
```

```
## [1] -1  1
```

```r
x[,2]
```

```
## [1] 21  1
```

```r
y[1,2]
```

```
## [1] 4
```

```r
y[,2:3]
```

```
##      [,1] [,2]
## [1,]    4    0
## [2,]    1   -1
```

*** Transforma la matriz m que creaste en el ejercicio anterior en un array multidimensional. (Pista: averigua lo que puedas de la función dim().)**

```r
array = array(m,dim= dim(m))
array
```

```
##      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10]
## [1,]    1    4    7   10   13   16   19   22   25    28
## [2,]    2    5    8   11   14   17   20   23   26    29
## [3,]    3    6    9   12   15   18   21   24   27    30
```

*** Crea un array de 5 x 5 y rellénalo con valores del 1 al 25. Investiga la función array(). Llama al array x**

```
x = array(1:25,dim= c(5,5))
x
```

```
##      [,1] [,2] [,3] [,4] [,5]
## [1,]    1    6   11   16   21
## [2,]    2    7   12   17   22
## [3,]    3    8   13   18   23
## [4,]    4    9   14   19   24
## [5,]    5   10   15   20   25
```

*** Dadas las matrices m1 y m2 usa rbind() y cbind() para crear matrices nuevas utilizando estas funciones, llamalas M1 y M2. ¿En que se diferencian las matrices creadas?**

```
m1 <- matrix(1, nr = 2, nc = 2)
m2 <- matrix(2, nr = 2, nc = 2)

M1 = rbind(m1, c(69,13))
M2 = cbind(m2, c(14,22))
M1
```

```
##      [,1] [,2]
## [1,]    1    1
## [2,]    1    1
## [3,]   69   13
```

```
M2
```

```
##      [,1] [,2] [,3]
## [1,]    2    2   14
## [2,]    2    2   22
```

*** El operador para el producto de dos matrices es "% * %". Por ejemplo, considerando las dos matrices creadas en el ejercicio anterior utilízalo.**

```
M1%*%M2
```

```
##      [,1] [,2] [,3]
## [1,]    4    4   36
## [2,]    4    4   36
## [3,]  164  164 1252
```

*** Usa la matriz M1 del ejercicio anterior y aplica la función t(). ¿qué hace esa función?**

```
t(M1)
```

```
##      [,1] [,2] [,3]
## [1,]    1    1   69
## [2,]    1    1   13
```

Genera la matriz traspuesta.

*** Ejecuta los siguientes comandos basados en la función diag() sobre las matrices creadas anteriormente m1 y m2. ¿Qué tipo de acciones puedes ejecutar con ella?**

diag(m1)

diag(rbind(m1, m2) %*% cbind(m1, m2)) ###diag(m1) <- 10 ###diag(3) ###v <- c(10, 20, 30) ###diag(v) ###diag(2.1, nr = 3, nc = 5)

```
diag(m1)
```

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

```
diag(rbind(m1, m2) %*% cbind(m1, m2))
```

```
## [1] 2 2 8 8
```

```
diag(m1) <- 10
diag(3)
```

```
##      [,1] [,2] [,3]
## [1,]    1    0    0
## [2,]    0    1    0
## [3,]    0    0    1
```

```
v <- c(10, 20, 30)
diag(v)
```

```
##      [,1] [,2] [,3]
## [1,]   10    0    0
## [2,]    0   20    0
## [3,]    0    0   30
```

```
diag(2.1, nr = 3, nc = 5)
```

```
##      [,1] [,2] [,3] [,4] [,5]
## [1,]  2.1  0.0  0.0    0    0
## [2,]  0.0  2.1  0.0    0    0
## [3,]  0.0  0.0  2.1    0    0
```

Con la función diag() podemos extraer o reemplazar la diagonal de una matriz, además de poder construir una matriz diagonal.

** Ordena la matriz x <- matrix(1:100, ncol=10):

a. en orden descendente por su segunda columna y asigna el resultado a una nueva matrix x1. Pista: función order()

b. en orden descendente por su segunda fila y asigna el resultado a una nueva matrix x2

c. Ordena solo la primera columna de x de forma descendente

```
## a
x = matrix(1:100, ncol=10)

x1= x[order(x[,2], decreasing = TRUE),]
x1
```

```
##       [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10]
##  [1,]   10   20   30   40   50   60   70   80   90   100
##  [2,]    9   19   29   39   49   59   69   79   89    99
##  [3,]    8   18   28   38   48   58   68   78   88    98
##  [4,]    7   17   27   37   47   57   67   77   87    97
##  [5,]    6   16   26   36   46   56   66   76   86    96
##  [6,]    5   15   25   35   45   55   65   75   85    95
##  [7,]    4   14   24   34   44   54   64   74   84    94
##  [8,]    3   13   23   33   43   53   63   73   83    93
##  [9,]    2   12   22   32   42   52   62   72   82    92
## [10,]    1   11   21   31   41   51   61   71   81    91
```

```
## b
x2= x[,order(x[2,], decreasing = TRUE)]
x2
```

```
##       [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10]
##  [1,]   91   81   71   61   51   41   31   21   11     1
##  [2,]   92   82   72   62   52   42   32   22   12     2
##  [3,]   93   83   73   63   53   43   33   23   13     3
##  [4,]   94   84   74   64   54   44   34   24   14     4
##  [5,]   95   85   75   65   55   45   35   25   15     5
##  [6,]   96   86   76   66   56   46   36   26   16     6
##  [7,]   97   87   77   67   57   47   37   27   17     7
##  [8,]   98   88   78   68   58   48   38   28   18     8
##  [9,]   99   89   79   69   59   49   39   29   19     9
## [10,]  100   90   80   70   60   50   40   30   20    10
```

```
## c
x[,1] = x[order(x[,1],decreasing = TRUE),1]
x
```

```
##       [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10]
##  [1,]   10   11   21   31   41   51   61   71   81    91
##  [2,]    9   12   22   32   42   52   62   72   82    92
##  [3,]    8   13   23   33   43   53   63   73   83    93
##  [4,]    7   14   24   34   44   54   64   74   84    94
##  [5,]    6   15   25   35   45   55   65   75   85    95
##  [6,]    5   16   26   36   46   56   66   76   86    96
##  [7,]    4   17   27   37   47   57   67   77   87    97
##  [8,]    3   18   28   38   48   58   68   78   88    98
```

```
##  [9,]    2   19   29   39   49   59   69   79   89    99
## [10,]    1   20   30   40   50   60   70   80   90   100
```

*** Accede al dataset "women".**

Primero confirma que los datos están ordenados de forma creciente según la altura (height) y el peso (weight) sin mirar los datos

Crea una nueva variable "bmi". Este valor responde a la siguiente fórmula: BMI = (Weight in Pounds / (Height in inches) / (Height in inches)) x 703

Investiga las funciones is.unsorted(), sort() and order()

```
aux = women

aux = aux[order(aux$height),]

women == aux
```

```
##       height weight
##  [1,]   TRUE   TRUE
##  [2,]   TRUE   TRUE
##  [3,]   TRUE   TRUE
##  [4,]   TRUE   TRUE
##  [5,]   TRUE   TRUE
##  [6,]   TRUE   TRUE
##  [7,]   TRUE   TRUE
##  [8,]   TRUE   TRUE
##  [9,]   TRUE   TRUE
## [10,]   TRUE   TRUE
## [11,]   TRUE   TRUE
## [12,]   TRUE   TRUE
## [13,]   TRUE   TRUE
## [14,]   TRUE   TRUE
## [15,]   TRUE   TRUE
```

```
aux = aux[order(aux$weight),]

women == aux
```

```
##       height weight
##  [1,]   TRUE   TRUE
##  [2,]   TRUE   TRUE
##  [3,]   TRUE   TRUE
##  [4,]   TRUE   TRUE
##  [5,]   TRUE   TRUE
##  [6,]   TRUE   TRUE
##  [7,]   TRUE   TRUE
##  [8,]   TRUE   TRUE
##  [9,]   TRUE   TRUE
## [10,]   TRUE   TRUE
## [11,]   TRUE   TRUE
## [12,]   TRUE   TRUE
## [13,]   TRUE   TRUE
```

```
## [14,]   TRUE   TRUE
## [15,]   TRUE   TRUE
```

Como podemos ver los datos están ordenados por peso y altura de forma ascendente.

```
aux$bmi = (women$weight / women$height/ women$height ) * 703
aux
```

```
##    height weight      bmi
## 1      58    115 24.03240
## 2      59    117 23.62856
## 3      60    120 23.43333
## 4      61    123 23.23811
## 5      62    126 23.04318
## 6      63    129 22.84883
## 7      64    132 22.65527
## 8      65    135 22.46272
## 9      66    139 22.43274
## 10     67    142 22.23791
## 11     68    146 22.19680
## 12     69    150 22.14871
## 13     70    154 22.09429
## 14     71    159 22.17358
## 15     72    164 22.23997
```

*** 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)

```
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 numero corresponde a las ventas en US y el segundo al resto de países.**

a) Construye la matriz star_wars_matrix con esos vectores

b) Añádele nombres a las columnas y filas de la matriz seg??n las descripciones dadas anteriormente de los datos

c) Calcula las ganacias mundiales de cada película y guardalas en un vector que se llame worldwide_vector.

d) Añade éste 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().

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

f) Calcula la media de ganancias para todas las pel??culas fuera de los estados unidos. Asigna esa media la variable non_us_all.

g) Haz lo mismo pero solo par alas dos primeras películas . Asigna el resultado a la variable non_us_some.

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: el numero total de visitantes para cada pelicula dividido por el precio del ticket te da el numero de visitantes)

i) Calcula la media de visitantes en territorio USA y en territorio noUS.

```
#a
star_wars_matrix = cbind(cbind(new_hope,empire_strikes),return_jedi)
```

```
#b
rownames(star_wars_matrix) = c("ventas_US","ventas_resto")
View(star_wars_matrix)
```

```
#c
worldwide_vector = star_wars_matrix["ventas_US",] + star_wars_matrix["ventas_resto",]
worldwide_vector
```

```
##       new_hope empire_strikes    return_jedi
##       775.3980       538.3751       475.1062
```

```
#d
#El ejercicio pide hacerlo por columnas, pero como el resultado no tiene sentido lo har?? por filas
all_wars_matrix = rbind(star_wars_matrix,worldwide_vector)
all_wars_matrix
```

```
##                  new_hope empire_strikes return_jedi
## ventas_US         460.998       290.4751    309.3062
## ventas_resto      314.400       247.9000    165.8000
## worldwide_vector  775.398       538.3751    475.1062
```

```r
#e
#Al igual que en el ejercicio anterior, hemos construido la matriz de forma que las ventas
#est??n representadas en las filas y las pel??culas en las columnas, as?? que usaremos la funci??n rowS
#para que tenga sentido
rowSums(star_wars_matrix)
```

```
##    ventas_US ventas_resto
##     1060.779      728.100
```

```r
#f
non_us_all = mean(star_wars_matrix["ventas_resto",])
non_us_all
```

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

```r
#g
non_us_some = mean(star_wars_matrix["ventas_resto",1:2])
non_us_some
```

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

```r
#h
entrada = 5
visitantes = star_wars_matrix / 5
```

```r
#i

mean(visitantes["ventas_US",])
```

```
## [1] 70.71862
```

```r
mean(visitantes["ventas_resto",])
```

```
## [1] 48.54
```

Ejercicio 2: Subsetting matrices y arrays

*** Crea un array i <- array(c(1:10),dim=c(5,2)). ¿Que información te dan los siguientes comandos?**

dim(i);

nrow(i);

ncol(i)

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

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

```r
nrow(i)
```

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

```r
ncol(i)
```

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

La función dim nos da la dimensión del array en filas y columnas como si fuera una matriz, nrow nos da el número de filas y ncol el n??mero de columnas.

*** Crea un array de dimensiones 5 filas y dos columnas y rellénalo con valores del 1-5 y del 5 al 1.**

```
x = array(c(1:5,5:1),dim=c(5,2))
x
```

```
##      [,1] [,2]
## [1,]    1    5
## [2,]    2    4
## [3,]    3    3
## [4,]    4    2
## [5,]    5    1
```

*** ¿Qué hace el comando x[i] ??. Comprueba que tienes en x antes ?**

x[i] me da el valor que se almacena en la posición i del vector o array.

*** ¿y el comando x[i] <- 0?**

Asigna a la posición i de x el valor 0.

*** Descárgate el fichero array_datos.txt de PRADO (Datos/) e impórtalo en tu work space de R teniendo en cuenta que es un texto tabulado. Después crea un documento con los mismos datos pero en formato csv en vez de tab separated**

```
# el método read.delim tiene por defecto como separador el tabulador
array_datos = read.delim("array_datos.txt",header=TRUE)
array_datos
```

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

```
write.csv(array_datos, file = "array_datos.csv")
```

Ejercicio 3: Factores

*** Dado x = c(1, 2, 3, 3, 5, 3, 2, 4, NA), ¿cuáles son los levels de factor(x)?**

a) 1, 2, 3, 4, 5

b) NA

c) 1, 2, 3, 4, 5, NA

```
x = c(1, 2, 3, 3, 5, 3, 2, 4, NA)
factor(x,exclude = NULL )
```

```
## [1] 1    2    3    3    5    3    2    4    <NA>
## Levels: 1 2 3 4 5 <NA>
```

Los levels del factor son 1,2,3,4,5,NA la opción c, ya que factor si lo ejecutamos por defecto tiene la opción exclude = NA que exclude de la salida los valores NA, pero si ejecutamos el comando de forma que no los excluya, esta los incluye en los levels.

*** Dado x <- c(11, 22, 47, 47, 11, 47, 11) y la ejecución de la sentencia**

factor(x, levels=c(11, 22, 47), ordered=TRUE)¿cuál es el cuarto elemento de la salida? ###a. 11 ###b. 22 ###c. 47

```
x <- c(11, 22, 47, 47, 11, 47, 11)
factor(x, levels=c(11, 22, 47), ordered=TRUE)
```

```
## [1] 11 22 47 47 11 47 11
## Levels: 11 < 22 < 47
```

El cuarto elemento sería 47.

*** Para el factor z <- c("p", "a" , "g", "t", "b"), reemplaza el tercer elemento de z por "b".**

a.factor(z[3]) <- "b"

b.levels(z[3]) <- "b"

c.z[3] <- "b"

```
z <- c("p", "a" , "g", "t", "b")
z[3] <- "b"
z
```

```
## [1] "p" "a" "b" "t" "b"
```

El apartado c sería la opción correcta.

*** Dado z <- factor(c("p", "q", "p", "r", "q")) escribe una expresión de R que cambie el level "p" a "w"**

```
z <- factor(c("p", "q", "p", "r", "q"))
levels(z)[1] = "w"
z
```

```
## [1] w q w r q
## Levels: w q r
```

*** Usa el dataset "iris"" :**

***Escribe la expresión necesaria para convertir la variable "Sepal.Length" en un factor con cinco niveles (levels) . Pista(mira la función table() y la función cut().(**

```
levels1 = cut(iris$Sepal.Length, breaks = 5)
iris$Sepal.Length = levels1
levels1
```

```
##   [1] (5.02,5.74] (4.3,5.02]  (4.3,5.02]  (4.3,5.02]  (4.3,5.02]
##   [6] (5.02,5.74] (4.3,5.02]  (4.3,5.02]  (4.3,5.02]  (4.3,5.02]
##  [11] (5.02,5.74] (4.3,5.02]  (4.3,5.02]  (4.3,5.02]  (5.74,6.46]
##  [16] (5.02,5.74] (5.02,5.74] (5.02,5.74] (5.02,5.74] (5.02,5.74]
##  [21] (5.02,5.74] (5.02,5.74] (4.3,5.02]  (5.02,5.74] (4.3,5.02]
##  [26] (4.3,5.02]  (4.3,5.02]  (5.02,5.74] (5.02,5.74] (4.3,5.02]
##  [31] (4.3,5.02]  (5.02,5.74] (5.02,5.74] (5.02,5.74] (4.3,5.02]
##  [36] (4.3,5.02]  (5.02,5.74] (4.3,5.02]  (4.3,5.02]  (5.02,5.74]
##  [41] (4.3,5.02]  (4.3,5.02]  (4.3,5.02]  (4.3,5.02]  (5.02,5.74]
##  [46] (4.3,5.02]  (5.02,5.74] (4.3,5.02]  (5.02,5.74] (4.3,5.02]
##  [51] (6.46,7.18] (5.74,6.46] (6.46,7.18] (5.02,5.74] (6.46,7.18]
##  [56] (5.02,5.74] (5.74,6.46] (4.3,5.02]  (6.46,7.18] (5.02,5.74]
##  [61] (4.3,5.02]  (5.74,6.46] (5.74,6.46] (5.74,6.46] (5.02,5.74]
##  [66] (6.46,7.18] (5.02,5.74] (5.74,6.46] (5.74,6.46] (5.02,5.74]
##  [71] (5.74,6.46] (5.74,6.46] (5.74,6.46] (5.74,6.46] (5.74,6.46]
##  [76] (6.46,7.18] (6.46,7.18] (6.46,7.18] (5.74,6.46] (5.02,5.74]
##  [81] (5.02,5.74] (5.02,5.74] (5.74,6.46] (5.74,6.46] (5.02,5.74]
##  [86] (5.74,6.46] (6.46,7.18] (5.74,6.46] (5.02,5.74] (5.02,5.74]
##  [91] (5.02,5.74] (5.74,6.46] (5.74,6.46] (4.3,5.02]  (5.02,5.74]
##  [96] (5.02,5.74] (5.02,5.74] (5.74,6.46] (5.02,5.74] (5.02,5.74]
## [101] (5.74,6.46] (5.74,6.46] (6.46,7.18] (5.74,6.46] (6.46,7.18]
## [106] (7.18,7.9]  (4.3,5.02]  (7.18,7.9]  (6.46,7.18] (7.18,7.9]
## [111] (6.46,7.18] (5.74,6.46] (6.46,7.18] (5.02,5.74] (5.74,6.46]
## [116] (5.74,6.46] (6.46,7.18] (7.18,7.9]  (7.18,7.9]  (5.74,6.46]
## [121] (6.46,7.18] (5.02,5.74] (7.18,7.9]  (5.74,6.46] (6.46,7.18]
## [126] (7.18,7.9]  (5.74,6.46] (5.74,6.46] (5.74,6.46] (7.18,7.9]
## [131] (7.18,7.9]  (7.18,7.9]  (5.74,6.46] (5.74,6.46] (5.74,6.46]
## [136] (7.18,7.9]  (5.74,6.46] (5.74,6.46] (5.74,6.46] (6.46,7.18]
## [141] (6.46,7.18] (6.46,7.18] (5.74,6.46] (6.46,7.18] (6.46,7.18]
## [146] (6.46,7.18] (5.74,6.46] (6.46,7.18] (5.74,6.46] (5.74,6.46]
## Levels: (4.3,5.02] (5.02,5.74] (5.74,6.46] (6.46,7.18] (7.18,7.9]
```

Escribe la expresión necesaria para generar una tabla de frecuencias con dos filas y tres columnas . Las filas deben referirse a si la variable "Sepal.length" es menor que 5 y las columnas a las diferentes expecies.

```
# El vector c(4,4.9,8) hace referencia a los intervalos para las divisiones
# serían del (4, 4.9] y del (4.9,8]
rm(iris)
tabla<-cut(iris$Sepal.Length, breaks =c(4,4.9,8), labels=(c("TRUE", "FALSE")) )
factor(tabla)
```

```
##   [1] FALSE TRUE  TRUE  TRUE  FALSE FALSE TRUE  FALSE TRUE  TRUE  FALSE
##  [12] TRUE  TRUE  TRUE  FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [23] TRUE  FALSE TRUE  FALSE FALSE FALSE FALSE TRUE  TRUE  FALSE FALSE
##  [34] FALSE TRUE  FALSE FALSE TRUE  TRUE  FALSE FALSE TRUE  TRUE  FALSE
##  [45] FALSE TRUE  FALSE TRUE  FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [56] FALSE FALSE TRUE  FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [67] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [78] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [89] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [100] FALSE FALSE FALSE FALSE FALSE FALSE FALSE TRUE  FALSE FALSE FALSE
## [111] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [122] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [133] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [144] FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## Levels: TRUE FALSE
```

```
table(tabla, iris$Species)
```

```
##
## tabla   setosa versicolor virginica
##   TRUE      20          1         1
##   FALSE     30         49        49
```

*** El factor responses se define como:**

responses <- factor(c("Agree", "Agree", "Strongly Agree", "Disagree", "Agree")), sin embargo nos damos cuenta que tiene un nuevo nivel, "Strongly Disagree", que no estaba presente cuando se creó. Añade el nuevo nivel al factor y conviértelo en un factor ordenado de la siguiente forma:

Levels: Strongly Agree < Agree < Disagree < Strongly Disagree

```
responses <- factor(c("Agree", "Agree", "Strongly Agree", "Disagree", "Agree"))
responses = factor(responses, levels =  c("Agree", "Strongly Agree", "Disagree", "Strongly Disagree"), o
responses
```

```
## [1] Agree          Agree          Strongly Agree Disagree
## [5] Agree
## Levels: Agree < Strongly Agree < Disagree < Strongly Disagree
```

*** Dado el factor:**

x <- factor(c("high", "low", "medium", "high", "high", "low", "medium"))

escribe la expresión en R que permita dar valores numéricos únicos para los distintos niveles (levels) de x según el siguiente esquema:

level high => value 1

level low => value 2

level medium => value 3

Pista: investiga la función unique() y los parámetros de data.frame()

```
x <- factor(c("high", "low", "medium", "high", "high", "low", "medium"))
x
```

```
## [1] high   low    medium high   high   low    medium
## Levels: high low medium
```

```
data.frame(levels = unique(x), value = as.numeric(unique(x)))
```

```
##   levels value
## 1   high     1
## 2    low     2
## 3 medium     3
```

4. Acceso y selección de secciones de un data frames

*** Vamos a trabajar con un ejemplo que viene por defecto en la instalación de R USArrests. Este data frame contiene la información para cada estado Americano de las tasas de criminales (por 100.000 habitantes). Los datos de las columnas se refieren a Asesinatos, violaciones y porcentaje de la población que vive en áreas urbanas. Los datos son de 1973. Contesta a las siguientes preguntas sobre los datos:**

```
#Las dimensiones del dataframe
dim(USArrests)
```

```
## [1] 50  4
```

```
#La longitud del dataframe (filas o columnas)
length(USArrests)
```

```
## [1] 4
```

```
#Numero  de columnas
ncol(USArrests)
```

```
## [1] 4
```

```
#Cómo calcularlas el número de filas?
nrow(USArrests)
```

```
## [1] 50
```

```r
#Obtén el nombre de las filas y las columnas para este data frame
row.names.data.frame(USArrests)
```

```
##  [1] "Alabama"        "Alaska"         "Arizona"        "Arkansas"
##  [5] "California"     "Colorado"       "Connecticut"    "Delaware"
##  [9] "Florida"        "Georgia"        "Hawaii"         "Idaho"
## [13] "Illinois"       "Indiana"        "Iowa"           "Kansas"
## [17] "Kentucky"       "Louisiana"      "Maine"          "Maryland"
## [21] "Massachusetts"  "Michigan"       "Minnesota"      "Mississippi"
## [25] "Missouri"       "Montana"        "Nebraska"       "Nevada"
## [29] "New Hampshire"  "New Jersey"     "New Mexico"     "New York"
## [33] "North Carolina" "North Dakota"   "Ohio"           "Oklahoma"
## [37] "Oregon"         "Pennsylvania"   "Rhode Island"   "South Carolina"
## [41] "South Dakota"   "Tennessee"      "Texas"          "Utah"
## [45] "Vermont"        "Virginia"       "Washington"     "West Virginia"
## [49] "Wisconsin"      "Wyoming"
```

```r
colnames(USArrests)
```

```
## [1] "Murder"   "Assault"  "UrbanPop" "Rape"
```

```r
#Échale un vistazo a los datos, por ejemplo a las seis primeras filas
USArrests[1:6,]
```

```
##            Murder Assault UrbanPop Rape
## Alabama      13.2     236       58 21.2
## Alaska       10.0     263       48 44.5
## Arizona       8.1     294       80 31.0
## Arkansas      8.8     190       50 19.5
## California    9.0     276       91 40.6
## Colorado      7.9     204       78 38.7
```

```r
#Ordena de forma decreciente las filas de nuestro data frame según el porcentaje de población
#en el área urbana. Para ello investiga la función order () y sus parámetros.

USArrests[order(USArrests$UrbanPop,decreasing = TRUE),]
```

```
##               Murder Assault UrbanPop Rape
## California       9.0     276       91 40.6
## New Jersey       7.4     159       89 18.8
## Rhode Island     3.4     174       87  8.3
## New York        11.1     254       86 26.1
## Massachusetts    4.4     149       85 16.3
## Hawaii           5.3      46       83 20.2
## Illinois        10.4     249       83 24.0
## Nevada          12.2     252       81 46.0
## Arizona          8.1     294       80 31.0
## Florida         15.4     335       80 31.9
## Texas           12.7     201       80 25.5
## Utah             3.2     120       80 22.9
## Colorado         7.9     204       78 38.7
## Connecticut      3.3     110       77 11.1
## Ohio             7.3     120       75 21.4
## Michigan        12.1     255       74 35.1
## Washington       4.0     145       73 26.2
## Delaware         5.9     238       72 15.8
```

```
## Pennsylvania      6.3     106       72 14.9
## Missouri          9.0     178       70 28.2
## New Mexico       11.4     285       70 32.1
## Oklahoma          6.6     151       68 20.0
## Maryland         11.3     300       67 27.8
## Oregon            4.9     159       67 29.3
## Kansas            6.0     115       66 18.0
## Louisiana        15.4     249       66 22.2
## Minnesota         2.7      72       66 14.9
## Wisconsin         2.6      53       66 10.8
## Indiana           7.2     113       65 21.0
## Virginia          8.5     156       63 20.7
## Nebraska          4.3     102       62 16.5
## Georgia          17.4     211       60 25.8
## Wyoming           6.8     161       60 15.6
## Tennessee        13.2     188       59 26.9
## Alabama          13.2     236       58 21.2
## Iowa              2.2      56       57 11.3
## New Hampshire     2.1      57       56  9.5
## Idaho             2.6     120       54 14.2
## Montana           6.0     109       53 16.4
## Kentucky          9.7     109       52 16.3
## Maine             2.1      83       51  7.8
## Arkansas          8.8     190       50 19.5
## Alaska           10.0     263       48 44.5
## South Carolina   14.4     279       48 22.5
## North Carolina   13.0     337       45 16.1
## South Dakota      3.8      86       45 12.8
## Mississippi      16.1     259       44 17.1
## North Dakota      0.8      45       44  7.3
## West Virginia     5.7      81       39  9.3
## Vermont           2.2      48       32 11.2
```

```
#¿Podrías añadir un segundo criterio de orden?, ¿cómo?

USArrests[order(USArrests$UrbanPop,USArrests$Rape,decreasing = TRUE),]
```

```
##                Murder Assault UrbanPop Rape
## California        9.0     276       91 40.6
## New Jersey        7.4     159       89 18.8
## Rhode Island      3.4     174       87  8.3
## New York         11.1     254       86 26.1
## Massachusetts     4.4     149       85 16.3
## Illinois         10.4     249       83 24.0
## Hawaii            5.3      46       83 20.2
## Nevada           12.2     252       81 46.0
## Florida          15.4     335       80 31.9
## Arizona           8.1     294       80 31.0
## Texas            12.7     201       80 25.5
## Utah              3.2     120       80 22.9
## Colorado          7.9     204       78 38.7
## Connecticut       3.3     110       77 11.1
## Ohio              7.3     120       75 21.4
## Michigan         12.1     255       74 35.1
## Washington        4.0     145       73 26.2
```

```
## Delaware          5.9     238       72 15.8
## Pennsylvania      6.3     106       72 14.9
## New Mexico       11.4     285       70 32.1
## Missouri          9.0     178       70 28.2
## Oklahoma          6.6     151       68 20.0
## Oregon            4.9     159       67 29.3
## Maryland         11.3     300       67 27.8
## Louisiana        15.4     249       66 22.2
## Kansas            6.0     115       66 18.0
## Minnesota         2.7      72       66 14.9
## Wisconsin         2.6      53       66 10.8
## Indiana           7.2     113       65 21.0
## Virginia          8.5     156       63 20.7
## Nebraska          4.3     102       62 16.5
## Georgia          17.4     211       60 25.8
## Wyoming           6.8     161       60 15.6
## Tennessee        13.2     188       59 26.9
## Alabama          13.2     236       58 21.2
## Iowa              2.2      56       57 11.3
## New Hampshire     2.1      57       56  9.5
## Idaho             2.6     120       54 14.2
## Montana           6.0     109       53 16.4
## Kentucky          9.7     109       52 16.3
## Maine             2.1      83       51  7.8
## Arkansas          8.8     190       50 19.5
## Alaska           10.0     263       48 44.5
## South Carolina   14.4     279       48 22.5
## North Carolina   13.0     337       45 16.1
## South Dakota      3.8      86       45 12.8
## Mississippi      16.1     259       44 17.1
## North Dakota      0.8      45       44  7.3
## West Virginia     5.7      81       39  9.3
## Vermont           2.2      48       32 11.2
```

```
#Muestra por pantalla la columna con los datos de asesinato

USArrests$Murder
```

```
##  [1] 13.2 10.0  8.1  8.8  9.0  7.9  3.3  5.9 15.4 17.4  5.3  2.6 10.4  7.2
## [15]  2.2  6.0  9.7 15.4  2.1 11.3  4.4 12.1  2.7 16.1  9.0  6.0  4.3 12.2
## [29]  2.1  7.4 11.4 11.1 13.0  0.8  7.3  6.6  4.9  6.3  3.4 14.4  3.8 13.2
## [43] 12.7  3.2  2.2  8.5  4.0  5.7  2.6  6.8
```

```
#Muestra las tasas de asesinato para el segundo, tercer y cuarto estado

USArrests[2:4,"Murder"]
```

```
## [1] 10.0  8.1  8.8
```

```
#Muestra las primeras cinco filas de todas las columnas

USArrests[1:5,]
```

```
##            Murder Assault UrbanPop Rape
## Alabama      13.2     236       58 21.2
## Alaska       10.0     263       48 44.5
## Arizona       8.1     294       80 31.0
```

```
## Arkansas      8.8     190       50 19.5
## California    9.0     276       91 40.6
```

```
#Muestra todas las filas para las dos primeras columnas

USArrests[,1:2]
```

```
##                Murder Assault
## Alabama          13.2     236
## Alaska           10.0     263
## Arizona           8.1     294
## Arkansas          8.8     190
## California        9.0     276
## Colorado          7.9     204
## Connecticut       3.3     110
## Delaware          5.9     238
## Florida          15.4     335
## Georgia          17.4     211
## Hawaii            5.3      46
## Idaho             2.6     120
## Illinois         10.4     249
## Indiana           7.2     113
## Iowa              2.2      56
## Kansas            6.0     115
## Kentucky          9.7     109
## Louisiana        15.4     249
## Maine             2.1      83
## Maryland         11.3     300
## Massachusetts     4.4     149
## Michigan         12.1     255
## Minnesota         2.7      72
## Mississippi      16.1     259
## Missouri          9.0     178
## Montana           6.0     109
## Nebraska          4.3     102
## Nevada           12.2     252
## New Hampshire     2.1      57
## New Jersey        7.4     159
## New Mexico       11.4     285
## New York         11.1     254
## North Carolina   13.0     337
## North Dakota      0.8      45
## Ohio              7.3     120
## Oklahoma          6.6     151
## Oregon            4.9     159
## Pennsylvania      6.3     106
## Rhode Island      3.4     174
## South Carolina   14.4     279
## South Dakota      3.8      86
## Tennessee        13.2     188
## Texas            12.7     201
## Utah              3.2     120
## Vermont           2.2      48
## Virginia          8.5     156
## Washington        4.0     145
```

```
## West Virginia     5.7      81
## Wisconsin         2.6      53
## Wyoming           6.8     161
```

```
#Muestra todas las filas de las columnas 1 y 3

USArrests[,c(1,3)]
```

```
##                Murder UrbanPop
## Alabama          13.2       58
## Alaska           10.0       48
## Arizona           8.1       80
## Arkansas          8.8       50
## California        9.0       91
## Colorado          7.9       78
## Connecticut       3.3       77
## Delaware          5.9       72
## Florida          15.4       80
## Georgia          17.4       60
## Hawaii            5.3       83
## Idaho             2.6       54
## Illinois         10.4       83
## Indiana           7.2       65
## Iowa              2.2       57
## Kansas            6.0       66
## Kentucky          9.7       52
## Louisiana        15.4       66
## Maine             2.1       51
## Maryland         11.3       67
## Massachusetts     4.4       85
## Michigan         12.1       74
## Minnesota         2.7       66
## Mississippi      16.1       44
## Missouri          9.0       70
## Montana           6.0       53
## Nebraska          4.3       62
## Nevada           12.2       81
## New Hampshire     2.1       56
## New Jersey        7.4       89
## New Mexico       11.4       70
## New York         11.1       86
## North Carolina   13.0       45
## North Dakota      0.8       44
## Ohio              7.3       75
## Oklahoma          6.6       68
## Oregon            4.9       67
## Pennsylvania      6.3       72
## Rhode Island      3.4       87
## South Carolina   14.4       48
## South Dakota      3.8       45
## Tennessee        13.2       59
## Texas            12.7       80
## Utah              3.2       80
## Vermont           2.2       32
## Virginia          8.5       63
```

```
## Washington        4.0       73
## West Virginia     5.7       39
## Wisconsin         2.6       66
## Wyoming           6.8       60
```

```
#Muestra solo las primeras cinco filas de las columnas 1 y 2

USArrests[1:5,1:2]
```

```
##            Murder Assault
## Alabama      13.2     236
## Alaska       10.0     263
## Arizona       8.1     294
## Arkansas      8.8     190
## California    9.0     276
```

```
#Extrae las filas para el ??ndice Murder
USArrests[,"Murder"]
```

```
##  [1] 13.2 10.0  8.1  8.8  9.0  7.9  3.3  5.9 15.4 17.4  5.3  2.6 10.4  7.2
## [15]  2.2  6.0  9.7 15.4  2.1 11.3  4.4 12.1  2.7 16.1  9.0  6.0  4.3 12.2
## [29]  2.1  7.4 11.4 11.1 13.0  0.8  7.3  6.6  4.9  6.3  3.4 14.4  3.8 13.2
## [43] 12.7  3.2  2.2  8.5  4.0  5.7  2.6  6.8
```

```
#¿Que estado tiene la menor tasa de asesinatos? ¿qué línea contiene esa información?, obtén esa informa
datos = USArrests[order(USArrests$Murder,decreasing = FALSE),]
datos
```

```
##                Murder Assault UrbanPop Rape
## North Dakota      0.8      45       44  7.3
## Maine             2.1      83       51  7.8
## New Hampshire     2.1      57       56  9.5
## Iowa              2.2      56       57 11.3
## Vermont           2.2      48       32 11.2
## Idaho             2.6     120       54 14.2
## Wisconsin         2.6      53       66 10.8
## Minnesota         2.7      72       66 14.9
## Utah              3.2     120       80 22.9
## Connecticut       3.3     110       77 11.1
## Rhode Island      3.4     174       87  8.3
## South Dakota      3.8      86       45 12.8
## Washington        4.0     145       73 26.2
## Nebraska          4.3     102       62 16.5
## Massachusetts     4.4     149       85 16.3
## Oregon            4.9     159       67 29.3
## Hawaii            5.3      46       83 20.2
## West Virginia     5.7      81       39  9.3
## Delaware          5.9     238       72 15.8
## Kansas            6.0     115       66 18.0
## Montana           6.0     109       53 16.4
## Pennsylvania      6.3     106       72 14.9
## Oklahoma          6.6     151       68 20.0
## Wyoming           6.8     161       60 15.6
## Indiana           7.2     113       65 21.0
## Ohio              7.3     120       75 21.4
## New Jersey        7.4     159       89 18.8
## Colorado          7.9     204       78 38.7
```

```
## Arizona           8.1     294       80 31.0
## Virginia          8.5     156       63 20.7
## Arkansas          8.8     190       50 19.5
## California        9.0     276       91 40.6
## Missouri          9.0     178       70 28.2
## Kentucky          9.7     109       52 16.3
## Alaska           10.0     263       48 44.5
## Illinois         10.4     249       83 24.0
## New York         11.1     254       86 26.1
## Maryland         11.3     300       67 27.8
## New Mexico       11.4     285       70 32.1
## Michigan         12.1     255       74 35.1
## Nevada           12.2     252       81 46.0
## Texas            12.7     201       80 25.5
## North Carolina   13.0     337       45 16.1
## Alabama          13.2     236       58 21.2
## Tennessee        13.2     188       59 26.9
## South Carolina   14.4     279       48 22.5
## Florida          15.4     335       80 31.9
## Louisiana        15.4     249       66 22.2
## Mississippi      16.1     259       44 17.1
## Georgia          17.4     211       60 25.8
```

```
USArrests[which(USArrests == datos[1,"Murder"]),]
```

```
##              Murder Assault UrbanPop Rape
## North Dakota    0.8      45       44  7.3
```

El estado del Norte de Dakota es el que menor tasa de asesinatos tiene y está en la posición 34 del dataset USArrests.

```
# ¿Que estados tienen una tasa inferior al 4%?, obtén esa información.
USArrests[which(USArrests$Murder < 4),]
```

```
##               Murder Assault UrbanPop Rape
## Connecticut      3.3     110       77 11.1
## Idaho            2.6     120       54 14.2
## Iowa             2.2      56       57 11.3
## Maine            2.1      83       51  7.8
## Minnesota        2.7      72       66 14.9
## New Hampshire    2.1      57       56  9.5
## North Dakota     0.8      45       44  7.3
## Rhode Island     3.4     174       87  8.3
## South Dakota     3.8      86       45 12.8
## Utah             3.2     120       80 22.9
## Vermont          2.2      48       32 11.2
## Wisconsin        2.6      53       66 10.8
```

```
# Que estados estan en el cuartil superior (75) en lo que a poblacion en zonas urbanas se refiere
quantile(USArrests$UrbanPop)
```

```
##    0%   25%   50%   75%  100%
## 32.00 54.50 66.00 77.75 91.00
```

```
USArrests[which(USArrests$UrbanPop >= quantile(USArrests$UrbanPop)[4]) ,]
```

```
##               Murder Assault UrbanPop Rape
```

```
## Arizona          8.1     294       80 31.0
## California       9.0     276       91 40.6
## Colorado         7.9     204       78 38.7
## Florida         15.4     335       80 31.9
## Hawaii           5.3      46       83 20.2
## Illinois        10.4     249       83 24.0
## Massachusetts    4.4     149       85 16.3
## Nevada          12.2     252       81 46.0
## New Jersey       7.4     159       89 18.8
## New York        11.1     254       86 26.1
## Rhode Island     3.4     174       87  8.3
## Texas           12.7     201       80 25.5
## Utah             3.2     120       80 22.9
```

*** Vamos a trabajar con otro dataframe. Descarga el fichero student.txt de la plataforma PRADO, almacena la informaci??n en una variable llamada "students". Ten en cuenta que los datos son tab-delimited y tienen un texto para cada columna. Comprueba que R ha leído correctamente el fichero imprimiendo el objeto en la pantalla**

```
students = read.delim(file = "student.txt")
students
```

```
##    height shoesize gender population
## 1     181       44   male     kuopio
## 2     160       38 female     kuopio
## 3     174       42 female     kuopio
## 4     170       43   male     kuopio
## 5     172       43   male     kuopio
## 6     165       39 female     kuopio
## 7     161       38 female     kuopio
## 8     167       38 female    tampere
## 9     164       39 female    tampere
## 10    166       38 female    tampere
## 11    162       37 female    tampere
## 12    158       36 female    tampere
## 13    175       42   male    tampere
## 14    181       44   male    tampere
## 15    180       43   male    tampere
## 16    177       43   male    tampere
## 17    173       41   male    tampere
```

Imprime solo los nombres de la columnas

```
colnames(students)
```

```
## [1] "height"     "shoesize"   "gender"     "population"
```

Llama a la columna height solo

```
students$height
```

```
##  [1] 181 160 174 170 172 165 161 167 164 166 162 158 175 181 180 177 173
```

??Cuantas observaciones hay en cada grupo?. Utiliza la función table(). Este commando se puede utilizar para crear tablas cruzadas (cross-tabulations)

```
table(students$gender)
```

```
##
## female   male
##      9      8
```

```
table(students$gender, students$height)
```

```
##
##          158 160 161 162 164 165 166 167 170 172 173 174 175 177 180 181
##   female   1   1   1   1   1   1   1   1   0   0   0   1   0   0   0   0
##   male     0   0   0   0   0   0   0   0   1   1   1   0   1   1   1   2
```

Crea nuevas variables a partir de los datos que tenemos. Vamos a crear una variable nueva "sym" que contenga M si el genero es masculino y F si el genero es femenino. Busca en la ayuda información sobre la función ifelse(). Crea una segunda variable "colours" cuyo valor será "Blue"" si el estudiante es de kuopio y "Red" si es de otro sitio.

```
students$gender
```

```
##  [1] male   female female male   male   female female female female female
## [11] female female male   male   male   male   male
## Levels: female male
```

```
sym = ifelse(students$gender == "male", "M", "F")
sym
```

```
##  [1] "M" "F" "F" "M" "M" "F" "F" "F" "F" "F" "F" "F" "M" "M" "M" "M" "M"
```

```
colours = ifelse(students$population == "kuopio", "Blue", "Red")
colours
```

```
##  [1] "Blue" "Blue" "Blue" "Blue" "Blue" "Blue" "Blue" "Red"  "Red"  "Red"
## [11] "Red"  "Red"  "Red"  "Red"  "Red"  "Red"  "Red"
```

Con los datos anteriores de height y shoesize y las nuevas variables crea un nuevo data.frame que se llame students.new

```
students.new = data.frame(students$height, students$shoesize, sym, colours)
students.new
```

```
##    students.height students.shoesize sym colours
## 1              181                44   M    Blue
## 2              160                38   F    Blue
## 3              174                42   F    Blue
## 4              170                43   M    Blue
## 5              172                43   M    Blue
## 6              165                39   F    Blue
## 7              161                38   F    Blue
```

```
## 8              167                38   F     Red
## 9              164                39   F     Red
## 10             166                38   F     Red
## 11             162                37   F     Red
## 12             158                36   F     Red
## 13             175                42   M     Red
## 14             181                44   M     Red
## 15             180                43   M     Red
## 16             177                43   M     Red
## 17             173                41   M     Red
```

Comprueba que la clase de student.new es un dataframe

```
class(students.new)
```

```
## [1] "data.frame"
```

Crea dos subsets a partir del dataset student. Dividelo dependiendo del sexo. Para ello primero comprueba que estudiantes son hombres (male). Pista: busca información sobre la función which.

```
which(students$gender == "male")
```

```
## [1]  1  4  5 13 14 15 16 17
```

Basándote en esa selección dada por which() toma solo esas filas del dataset student para generar el subset stundent.male

-Basándote en esa selección dada por which() toma solo esas filas del dataset student para generar el subset stundent.male

- Repite el procedimiento para seleccionar las estudiantes mujeres (females)

```
students.male = students[which(students$gender == "male"),]
students.male
```

```
##    height shoesize gender population
## 1     181       44   male     kuopio
## 4     170       43   male     kuopio
## 5     172       43   male     kuopio
## 13    175       42   male    tampere
## 14    181       44   male    tampere
## 15    180       43   male    tampere
## 16    177       43   male    tampere
## 17    173       41   male    tampere
```

```
students.female = students[which(students$gender == "female"),]
students.female
```

```
##    height shoesize gender population
## 2     160       38 female     kuopio
## 3     174       42 female     kuopio
```

```
## 6     165       39 female     kuopio
## 7     161       38 female     kuopio
## 8     167       38 female    tampere
## 9     164       39 female    tampere
## 10    166       38 female    tampere
## 11    162       37 female    tampere
## 12    158       36 female    tampere
```

Utiliza la function write.table() para guarder el contenido de student.new en un archivo.

```
write.table(students.new, file = "student_new.txt", sep = "\t")
```