

Proyecto 2

Rudik Roberto Rompich - Carlos Martínez

Ejercicios

A continuación se listan los comentarios de los códigos proporcionados.

Ejercicio 1

Se empieza declarando una lista de elementos.

```
peso <- c(19,14,15,17,20,23,30,19,25)
```

Estas dos líneas de código proporcionan una lista de valores booleanos. 1. Si los valores de la lista <20, entonces devuelve TRUE y si son >20 devuelve FALSE. 2. Si los valores son <20 o >30 entonces es TRUE, de lo contrario FALSE.

```
peso < 20
```

```
## [1] TRUE TRUE TRUE TRUE FALSE FALSE FALSE TRUE FALSE
```

```
peso < 20 | peso > 25
```

```
## [1] TRUE TRUE TRUE TRUE FALSE FALSE TRUE TRUE FALSE
```

Se crea una lista con los elementos menores a 20.

```
peso[peso < 20]
```

```
## [1] 19 14 15 17 19
```

Se crea una lista con los elementos menores a 20, siempre y cuando los elementos no sean 15.

```
peso[peso < 20 & peso != 15]
```

```
## [1] 19 14 17 19
```

Se crea una lista que contiene 3 Strings del String original insertado, por cada rep().

```
trat <- c(rep("A", 3), rep("B", 3), rep("C", 3))
trat
```

```
## [1] "A" "A" "A" "B" "B" "B" "C" "C" "C"
```

Se le asigna los primeros tres elementos de la lista a cada una de las "A"

```
peso[trat == "A"]
```

```
## [1] 19 14 15
```

A cada valor de peso se le asigna un valor de A o B.

```
peso[trat == "A" | trat == "B"]
```

```
## [1] 19 14 15 17 20 23
```

Con la función `split()` se separan 3 elementos de la lista `peso` con la lista `trat`. Lo que hace `$A` es seleccionar simplemente la lista con las variables `A`.

```
split(peso, trat)$A
```

```
## [1] 19 14 15
```

Se declara una variable `x` con un rango de 1 a 15. Luego con la función `length()` se calcula su longitud y posteriormente se imprime.

```
x <- 1:15  
length(x)
```

```
## [1] 15
```

```
x
```

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

Se crea una matriz de 3x4

```
y <- matrix(5, nrow = 3, ncol = 4)  
y
```

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

Con la función `dim()` se calcula las dimensiones de la matriz.

```
dim(y)
```

```
## [1] 3 4
```

Se verifica si la variable `x` es un vector. Da positivo.

```
is.vector(x)
```

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

Se verifica si la variable `y` es un vector. Da falso.

```
is.vector(y)
```

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

Se verifica si la variable `x` es un array. Da falso.

```
is.array(x)
```

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

Se crea una lista con rango de 1 a 5.

```
x1 <- 1:5  
x1
```

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

Se crea una lista con 5 elementos y se asigna el valor `patata` a la variable `x3`

```
x2 <- c(1, 2, 3, 4, 5)  
x2
```

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

```
x3 <- "patata"
```

El tipo de elemento. Entero.

```
typeof(x1)
```

```
## [1] "integer"
```

El tipo de elemento. Double.

```
typeof(x2)
```

```
## [1] "double"
```

El tipo de elemento. Character.

```
typeof(x3)
```

```
## [1] "character"
```

Es una generalidad para mostrar el “mode” de una variable. Numérico.

```
mode(x)
```

```
## [1] "numeric"
```

Es una generalidad para mostrar el “mode” de una variable. Numérico.

```
mode(y)
```

```
## [1] "numeric"
```

Es una generalidad para mostrar el “mode” de una variable. Character.

```
mode(x3)
```

```
## [1] "character"
```

```
z <- c(TRUE, FALSE)
```

Se determina el tipo de datos de la lista z. Que es tipo lógico.

```
typeof(z)
```

```
## [1] "logical"
```

Se determina el modo de datos de la lista z. Que es tipo lógico.

```
mode(z)
```

```
## [1] "logical"
```

Se determinan los atributos de la variable y, en este caso, sus dimensiones.

```
attributes(y)
```

```
## $dim
```

```
## [1] 3 4
```

Se declara una lista que tiene como elementos a un rango de 1 a 3 y un b con un elemento 5

```
w <- list(a = 1:3, b = 5)
```

Arroja la variable a que contiene a los elementos de b

```
w
```

```
## $a
## [1] 1 2 3
##
## $b
## [1] 5
```

Se crea un data frame de la variable y declarada anteriormente.

```
y <- as.data.frame(y)
y
```

```
##   V1 V2 V3 V4
## 1  5  5  5  5
## 2  5  5  5  5
## 3  5  5  5  5
```

Es un resumen de los atributos del data frame.

```
attributes(y)

## $names
## [1] "V1" "V2" "V3" "V4"
##
## $class
## [1] "data.frame"
##
## $row.names
## [1] 1 2 3
```

Ejercicio 2

Se declara una lista de elementos. Primero a la x se le asigna una lista de 5 números y luego se cambia por una secuencia de 1 a 10. Para la variable se declara una lista de elementos de -5 a 3.

```
# Generación de secuencias
x <- c(1, 2, 3, 4, 5)
x <- 1 : 10
y <- -5 : 3
y <- -5 : 3
```

Una lista de (2,5)

```
1 : 4 + 1
```

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

Una lista de [1,5]

```
1 : (4 + 1)
```

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

Crea una secuencia de 2 a 18 en un intervalo de dos elementos.

```
x <- seq(from = 2, to = 18, by = 2)
```

Crea una secuencia de 2 a 18 que contiene 30 elementos.

```
x <- seq(from = 2, to = 18, length = 30)
```

Se crea una secuencia con el número de elementos de la variable anterior. Es decir, 30.

```
y <- seq(along = x)
```

Se crea una lista en un intervalo en función de una secuencia que va de 1 a 5, luego 7 a 10 y de -7 a 5 en un intervalo de 2 en 2.

```
z <- c(1 : 5, 7:10, seq(from=-7, to=5, by=2))
```

Se crea una lista de 1, de 5 elementos.

```
rep(1, 5) #repite el 1 cinco veces
```

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

Una lista de 1 a 3.

```
x <- 1:3
```

Se crea una lista duplicada del x anterior.

```
rep(x, 2)
```

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

Se crea una variable con una lista de numeros que contiene 3 cincos.

```
y <- rep(5, 3)
```

Una lista que repite 1, 2 y 3 varias veces, según el y indicado.

```
rep(x, y)
```

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

Se repiten los valores de 1 a 3, según el número de elementos del rep(5,3)

```
rep(1 : 3, rep(5, 3))
```

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

Según el x seleccionado, se repiten el número de elementos x, según la misma variable x.

```
rep(x, x)
```

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

Se repite x, hasta que hayan 8 elementos.

```
rep(x, length = 8)
```

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

De una lista de elementos de 1 a 3, cada elemento se repite 3 veces.

```
gl(3, 5) # es equivalente a rep(1:3, rep(5, 3))
```

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

```
## Levels: 1 2 3
```

De una lista de elementos de 1 a 4 se repiten hasta que se alcancen los 20 elementos.

```
gl(4, 1, length = 20) # gl genera factores
```

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

```
## Levels: 1 2 3 4
```

De una lista de 3 elementos, se repiten 4 veces cada uno.

```
gl(3, 4, label = c("Rubio", "Moreno", "Pelirrojo"))
```

```
## [1] Rubio      Rubio      Rubio      Rubio      Moreno     Moreno     Moreno
## [8] Moreno     Pelirrojo Pelirrojo Pelirrojo Pelirrojo Pelirrojo
## Levels: Rubio Moreno Pelirrojo
```

Ejercicio 3

Se crea una lista de elementos de 1 a 7.

```
#Selección de los elementos de un vector
x <- 1:7
```

Se selecciona el primer elemento.

```
x[1]
```

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

Se selecciona el tercer elemento.

```
x[3]
```

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

Se seleccionan los elementos 2 y 6 de la lista.

```
x[c(2,6)]
```

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

Se seleccionan los elementos mayores a 4.

```
x[x > 4]
```

```
## [1] 5 6 7
```

Es una operación booleana. Si cumple en el array retorna TRUE en el otro caso FALSE.

```
x > 4
```

```
## [1] FALSE FALSE FALSE FALSE TRUE TRUE TRUE
```

Se asignan los valores booleanos de $x > 4$ a una variable y.

```
y <- x > 4
```

x en función de la variable y.

```
x[y]
```

```
## [1] 5 6 7
```

Se excluyen los elementos 1 y 5.

```
x[-c(1, 5)]
```

```
## [1] 2 3 4 6 7
```

Lista de elementos que contiene los tres valores indicados.

```
y <- c(1, 2, 6)
```

x en función de y.

```
x[y]
```

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

Se les asignan nombres a las variables de una lista x.

```
names(x) <- c("a", "b", "c", "d", "e", "manzana") # Cuidado con las dobles las comillas (tecla del 2)
```

Se eleccionan los elementos requeridos en x.

```
x[c("a", "e", "manzana")]
```

```
##      a      e manzana
##      1      5      6
```

Ejercicio 4

Se declara una lista con los elementos siguientes:

```
#ordenación de vectores
x <- c(7, 4, 5, 9, 1)
```

Crea una lista de elementos distintos al anterior.

```
order(x)
```

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

Ordenada los elementos de menor a mayor.

```
sort(x)
```

```
## [1] 1 4 5 7 9
```

Desordenada los elementos.

```
rev(x)
```

```
## [1] 1 9 5 4 7
```

Desordenada los elementos aleatoriamente.

```
rank(x)
```

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

En función de la lista order(x) se selecciona los elementos en x.

```
x[order(x)]
```

```
## [1] 1 4 5 7 9
```

Se crea un lista y con varios elementos.

```
y <- c(1, 5, 5, 5, 7, 7, 9, 9, 9, 9)
```

Se vuelve a imprimir una lista desordenada de los elementos de x.

```
rank(x)
```

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

Aquí el uso de rank() es evidente, ya que se les asigna una misma variable a cada mismo valor.

```
rank(y)
```

```
## [1] 1.0 3.0 3.0 3.0 5.5 5.5 8.5 8.5 8.5 8.5
```

El valor mínimo de x.

```
min(x)
```

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

El uso de which.min no está claro, ya que esta función sirve para devolver el índice de valores booleanos y en este caso no se tiene una condición.

```
which.min(x)
```

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

El which sirve para encontrar el índice en donde x== al valor minimo de x.

```
which(x == min(x))
```

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

Se vuelve a crear otra lista y.

```
y <- c(1, 1, 2, 2, 3)
```

Se asignan nuevos valores en orden de los valores previos de y.

```
order(y)
```

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

Ejercicio 5

Es una forma de crear una lista que agrupe una de las letras indicadas con un número de la secuencia 3 a 5, y el separador no es ninguno.

```
#Vectore de caracteres  
paste1 <- paste(c("I", "J", "M"), 3:5, sep = "")
```

Es una forma de crear una lista que agrupe una de las letras indicadas con un número de la secuencia 3 a 5, y el separador es un punto.

```
paste2 <- paste(c("I", "J", "M"), 3:5, sep = ".")
```

Es una forma de crear una lista que agrupe una de las palabras indicadas separadas por un espacio.

```
unir1 <- paste(c("el", "sol", "brilla"), collapse = " ")
```

Se separan las letras indicadas por un guion.

```
unir2 <- paste(c("el", "sol", "brilla"), collapse = "-")
```

Se crean 3 strings separadas en la lista.

```
unir3 <- paste(c("el", "sol", "brilla"))
```

Se crea una variable letras con las primeras 9 letras del abecedario.

```
letras <- LETTERS[1:9]
```

Una lista de números del 7 al 15.

```
numeros <- 7:15
```

Se juntas las letras con los números de la variable números.


```
unir4 <- paste(letras, numeros, sep = "")
```

Se le pide a substr() que se seleccionen los elementos en la posición 2 a 4.

```
substr("abcdef", 2, 4)
```

```
## [1] "bcd"
```

Se hace una lista de letras pegadas, según el rango indicado.

```
x <- paste(LETTERS[1:7], collapse = "")
```

Se seleccionan los elementos 3 a 7 de la variable x. Entonces, se les asignan los valores xyz en la posición 3.

```
substr(x, 3, 7) <- c("xyz")
```

```
x
```

```
## [1] "ABxyzFG"
```

Se crea una lista con 3 valores.

```
x <- c(60, 90, 903)
```

Se le asigna a y el factor x y luego se imprimen X y Y.

```
y <- factor(x)
```

```
x
```

```
## [1] 60 90 903
```

```
y
```

```
## [1] 60 90 903
```

```
## Levels: 60 90 903
```

Solo sirve para hacer una conversión entre character a numérico.

```
as.numeric(as.character(y))
```

```
## [1] 60 90 903
```

Se le asigna el factor de los 3 elementos a una variable.

```
factor1 <- factor(c("alto", "bajo", "medio"))
```

Se le asigna el factor de los 3 elementos a una variable, basándose en una variable levels con otro orden.

```
factor2 <- factor(c("alto", "bajo", "medio"), levels = c("bajo", "medio", "alto")) # cambia el orden de
```

Ejercicio 6

Se declara una variable a de 1 a 24

```
#arrays y matrices
```

```
a <- 1:24
```

dim() se utiliza para visualizar las dimensiones de a y el valor al que está igualado es una lista. a se divide en 3 listas.

```
dim(a) <- c(3,4,2)
```

Se dividen en una lista de 2 según la lista (4,5) los valores de 7

```
arr1 <- array(7, dim = c(4,5))
```

Se crea una matriz de 5*4 que agrupa los valores indicados.

```
mat1 <- matrix(10:29, nrow = 5)
mat1
```

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

Se crea una matrix de 5*5 y se le dice que se ordenen por fila.

```
mat2 <- matrix(10:29, nrow = 5, byrow = TRUE)
mat2
```

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

Se crea una lista de 10 a 29.

```
mat3 <- 10:29
```

Se crea un array con los elementos de mat3 en un array de 5*4

```
dim(mat3) <- c(5, 4)
mat3
```

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

El elemento (1,1) de la lista 1 del array a

```
a[1,1,1]
```

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

El elemento (1,1) de la lista 2 del array a

```
a[1,1,2]
```

```
## [1] 13
```

El elemento (3,4) de la lista 2 del array a

```
a[3,4,2]
```

```
## [1] 24
```

Se selecciona la fila 2 de los dos arrays del array a

```
a[2, , ] # es un array de dimensi?n c(4,2)
```

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

```
## [2,]    5   17
## [3,]    8   20
## [4,]   11   23
```

Se selecciona la columna 3 de los dos arrays del array a.

```
a
```

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

```
a[,3, ] # es un array de dimensi?n c(3,2)
```

```
##      [,1] [,2]
## [1,]    7   19
## [2,]    8   20
## [3,]    9   21
```

Se selecciona la primera fila de mat3

```
mat3
```

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

```
mat3[1, ]
```

```
## [1] 10 15 20 25
```

Se selecciona la 2 columna de mat3

```
mat3[, 2]
```

```
## [1] 15 16 17 18 19
```

Se selecciona un elemtno específico en función de los valores ingresados en las dos listas.

```
mat3[c(1, 3), c(2, 4)]
```

```
##      [,1] [,2]
## [1,]   15   25
## [2,]   17   27
```

Se asigna una matriz con 4 elementos y dos filas.

```
mat4 <- matrix(c(1, 3, 2, 4), nrow = 2)
```

Se imprime

```
mat4

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

Se evalúan elementos de mat4 en mat3 y da como resultado dos elementos.

```
mat3[mat4] # coordenadas matricialmente
```

```
## [1] 15 27
```

Se crea una matrix con 4 elementos y parámetros para columnas y filas.

```
m1 <- matrix(1: 4, nr = 2, nc = 2)
m1
```

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

Se crea una matrix con 4 elementos y parámetros para columnas y filas.

```
m2 <- matrix(6: 9, nr = 2, nc = 2)
m2
```

```
##      [,1] [,2]
## [1,]    6    8
## [2,]    7    9
```

rbind() juntó las matrices m1 y m2 en una sola, por filas.

```
rbind(m1, m2) # rbind une matrices por filas
```

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

cbind() juntó las matrices m1 y m2 en una sola, por columnas.

```
cbind(m1, m2) # cbind une matrices por columnas
```

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

diag() mostró la diagonal principal de la matriz.

```
diag(m1) # diagonal principal de la matriz
```

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

Ejercicio 7

Se crea una lista con una secuencia de 1 a 4.

```
#Data frames y listas
x <- 1:4
```

Se crea una variable.

```
n <- 10
```

Se crea una lista con dos elementos.

```
M <- c(10, 35)
```

Se crea otra variable con una lista de 2 a 5.

```
y <- 2:5
```

Se imprimen

```
x
```

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

```
n
```

```
## [1] 10
```

```
M
```

```
## [1] 10 35
```

```
y
```

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

Se crea un dataframe con los datos de x y n

```
data.frame(x, n)
```

```
##   x  n
```

```
## 1 1 10
```

```
## 2 2 10
```

```
## 3 3 10
```

```
## 4 4 10
```

Se crea un dataframe con los datos de x y M

```
data.frame(x, M)
```

```
##   x  M
```

```
## 1 1 10
```

```
## 2 2 35
```

```
## 3 3 10
```

```
## 4 4 35
```

Se crea un dataframe con los datos de x y y

```
data.frame(x, y)
```

```
##   x y
```

```
## 1 1 2
```

```
## 2 2 3
```

```
## 3 3 4
```

```
## 4 4 5
```

Se crean dos listas con los elementos de x y y. Independientemente.

```
l1 <- list(x, y)
```

```
l1
```

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

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

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

Se crea una lista independiente asignando un índice: A y B.

```
l2 <- list(A = x, B = y)  
l2
```

```
## $A  
## [1] 1 2 3 4  
##  
## $B  
## [1] 2 3 4 5
```

Se intenta visualizar los nombres de las variables de la lista l1. Retorna null.

```
names(l1)
```

```
## NULL
```

Se intenta visualizar los nombres de las variables de la lista l2. Retorna dos variables.

```
names(l2)
```

```
## [1] "A" "B"
```