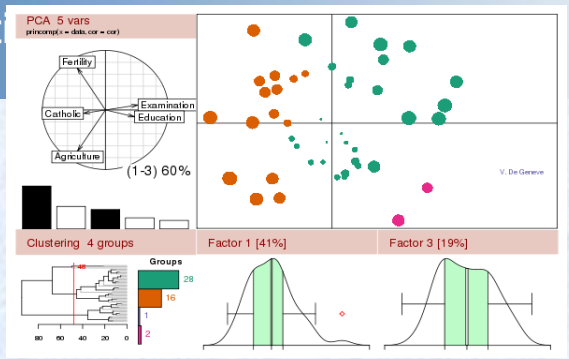
# ANÁLISIS DE DATO CON R


ENEI
Escuela Nacional de Estadística e Informática
INEI
INSTITUTO NACIONAL DE ESTADÍSTICA E INFORMÁTICA
contáctenos: enei@inei.gob.pe / 433-3127
Pasaje Hernán Velarde 285 Lima.
Entre la cuadra 01 y 02 de la Av. Arequipa.
Correo: enei@inei.gob.pe / campusvirtual@inei.gob.pe
Teléfonos: 433-3127 - 332-4650
Centro Andino de Formación y Capacitación en Estadística
Cursos Especializados en Estadística e Informática


# Arrays y Matrices

- Arrays
- Indexación de arrays
- La función array
- Facilidades con matrices
- Ejercicios propuestos

# Arrays

Un *array* es un conjunto de datos de k dimensiones. El caso más sencillo se da con k=2, lo que llamaremos matriz (*matrix*). Todos los elementos de un *array* han de ser del mismo tipo.

En R cualquier *array* ha de tener asociado un atributo llamado *dim* que indique los límites superiores de cada una de las dimensiones. Por definición el límite inferior es 1.

```
> a <- 1:42 # Creamos un vector de 42 posiciones
# Lo transformamos en un array añadiéndole el límite superior de cada dimensión. En este caso en un array de
# tres dimensiones de longitudes 3, 7 y 2. Nótese que las dimensiones que se "mueven" más rápido son las de
# más a la izquierda.
> dim(a) <- c(3,7,2)
> a
, , 1
     [,1] [,2] [,3] [,4] [,5] [,6] [,7]
[1,]    1    4    7   10   13   16   19
[2,]    2    5    8   11   14   17   20
[3,]    3    6    9   12   15   18   21

, , 2
     [,1] [,2] [,3] [,4] [,5] [,6] [,7]
[1,]   22   25   28   31   34   37   40
[2,]   23   26   29   32   35   38   41
[3,]   24   27   30   33   36   39   42
```

# Indexación de arrays (1)

Los *arrays* se indexan exactamente igual que los vectores. En este caso tenemos que establecer un vector de índices (sesión 2) para cada dimensión del *array*. Los diferentes vectores se pondrán separados por comas y toda la indexación también irá entre corchetes ([]).

Si sobre una determinada dimensión no queremos aplicar ningún tipo de selección simplemente dejaremos el espacio correspondiente a esa dimensión en blanco.

```
# Del array anterior seleccionamos la posición 1, 4, 2
> a[1,4,2]
 [1] 31

# Ahora fijamos las dos primeras dimensiones y dejamos libre la tercera
> a[2,4,]
 [1] 11 32

# a[,2,] es un array con vector de dimensiones c(3,2) y vector de datos que contiene los valores
# c(a[1,2,1], a[2,2,1], a[3,2,1], a[1,2,2], a[2,2,2], a[3,2,2])
> a[,2,]

     [,1] [,2]
[1,]    4   25
[2,]    5   26
[3,]    6   27
```

# Indexación de arrays (2)

Un *array* también se puede indexar con un único *array* de índices. Dicho *array* tendrá tantas columnas como dimensiones tenga el *array* de donde se quieren subseleccionar los elementos y un números de filas indeterminado, que equivaldrá al número de elementos que queramos seleccionar. En el caso de las matrices el proceso se hace más claro.

```
# Generamos una matriz 4 * 5
> x <- 1:20
> dim(x) <- c(4,5)
> x
     [,1] [,2] [,3] [,4] [,5]
[1,]    1    5    9   13   17
[2,]    2    6   10   14   18
[3,]    3    7   11   15   19
[4,]    4    8   12   16   20

# Ahora queremos seleccionar los elementos x[1,3], x[2,2] y x[3,1]. Para eso crearemos el array de índices,
# que en este caso tendrá dos columnas y tres filas.
> i <- c(1,2,3,3,2,1)   #Vector de valores
> dim(i) <- c(3,2)      #Vector de dimensiones
> i
     [,1] [,2]
[1,]    1    3
[2,]    2    2
[3,]    3    1
```

# Indexación arrays (3)

```
# Vemos los elementos seleccionados
> x[i]
[1] 9 6 3

# Ponemos esos elementos a cero
> x[i] <- 0
> x
     [,1] [,2] [,3] [,4] [,5]
[1,]    1    5    0   13   17
[2,]    2    0   10   14   18
[3,]    0    7   11   15   19
[4,]    4    8   12   16   20
```

## La función *array*

Además de con un vector de datos y un atributo dim, los *arrays* también pueden ser construídos con la función *array*, a la cual se le ha de pasar como argumentos el vector de valores y el vector de dimensiones. Esta función actúa igual que el método "manual", pero acepta vectores de dimensiones que no encajen exactamente con el tamaño del vector de datos (reciclaje).

```
> Z1 <- array(x,dim=c(4,5)) # Creamos el mismo array que antes

> Z2 <- array(x,dim=c(5,5)) # Los valores del vector de datos se reciclan

> Z3 <- array(x,dim=c(4,4)) # Hay valores del vector de datos que no se incluyen en el nuevo array
```

# Facilidades con matrices (1)

Las matrices son una particularización de los *arrays*, donde el número de dimensiones es 2. Debido a su gran uso, R dispone de funciones específicas para matrices.

| Función | Operador |
|---|---|
| número filas | `nrow` |
| número columnas | `ncol` |
| producto matricial | `%*%` |
| transposición | `t` |
| diagonal | `diag` |

Al igual que la función *array*, la función *matrix* nos permite crear matrices.

```
# Creamos una matriz de dimensiones 5 * 4 con la función matrix.
# Si queremos que la llene por filas le añadiremos el parámetro
# byrow=TRUE.
> a <- matrix(1:20,5,4)
> a
     [,1] [,2] [,3] [,4]
[1,]    1    6   11   16
[2,]    2    7   12   17
[3,]    3    8   13   18
[4,]    4    9   14   19
[5,]    5   10   15   20

# Miramos las dimensiones. También lo podemos hacer con la función dim.
> nrow(a)
[1] 5
> ncol(a)
[1] 4
```

# Facilidades con matrices (2)

R permite utilizar los operadores matemáticos clásicos (+, - , *, /,...) para operaciones con matrices de mismas dimensiones. En este caso se realiza la operación para cada par de elementos

```
# Creamos dos matrices de iguales dimensiones
> a <- matrix(1:20,5,4)
> b <- matrix(21:40,5,4,byrow=TRUE)

# Ahora podemos hacer operaciones con las dos matrices
> a + b #Suma posición a posición.
     [,1] [,2] [,3] [,4]
[1,]   22   28   34   40
[2,]   27   33   39   45
[3,]   32   38   44   50
[4,]   37   43   49   55
[5,]   42   48   54   60
```

Las matrices también se pueden construir a base de la unión de vectores individuales, ya sea apilándolos por filas o por columnas. Para eso tenemos las funciones *rbind* y *cbind*. Para transformar un array en vector utilizaremos la función *as.vector*

```
> rbind(1:5,11:15,21:25) #Apilamos tres vectores por filas. El resultado es un objeto de tipo matrix.
     [,1] [,2] [,3] [,4] [,5]
[1,]    1    2    3    4    5
[2,]   11   12   13   14   15
[3,]   21   22   23   24   25
```

# Ejercicios propuestos

1. Generar una matriz de dimensiones 3 * 5 llena de ceros de 2 maneras diferentes.

2. Utilizando la función *array*, crear un *array* tridimensional 4*4*4, llenándolo con los números del 1 al 64. Hacer una subselección de los índices 2 y 4 de la primera dimensión y 1 y 3 de la tercera. Mirar las dimensiones del nuevo array. Convertir el resultado de esta subselección en un vector, y después construir con ese mismo vector una matriz 4 * 4, llenándola por filas.

3. Utilizando la matriz 4 * 4 del problema anterior, invertir el orden de sus columnas y transponerla. A todos superiores a 30 de esta nueva matriz cambiarles el signo. Hacer el producto matricial de la matriz original por la que acabamos de obtener.

4. Sin utilizar *cbind* ni *rbind*, crear la siguiente matriz 3 * 10 (no hay que escribir el vector (1,2,3) 10 veces):

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

¿Cómo podemos obtener la matriz siguiente?

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

Hacer la multiplicación posición a posición de las dos matrices obtenidas.

# Recuerda siempre nuestro correo para cualquier consulta

**campusvirtual@inei.gob.pe**

# Comunicación constante con la Escuela del INEI

**Correo de la Escuela del INEI**
**enei@inei.gob.pe**

**Área de Educación Virtual**
**(campusvirtual@inei.gob.pe)**