

Clase matrices y arrays

Práctica 3

Matrices: array de 2 dimensiones

Las matrices pueden ser descritas como vectores bidimensionales. Al igual que un vector, únicamente pueden contener datos de un sólo tipo, pero tienen dos dimensiones (filas y columnas).

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

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

```
dim(matriz)
```

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

En este caso matriz tiene cuatro filas y una columna. Tener en cuenta que la dimensión 1 son las filas y la dimensión 2 las columnas.

Matrices: array de 2 dimensiones

La función `matrix` es una de las funciones que se usa para crear matrices, por default completa por columnas pero puedo especificarle que sea por filas (`byrow=TRUE`).

```
matriz_2<- matrix(c(1,2,3,4), ncol = 2, nrow = 2, byrow = TRUE)
matriz_2
```

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

Matrices - Funciones útiles

- Generar: `matrix`, `array`, `diag`
- Unir o concatenar: `rbind` (uno filas), `cbind` (uno columnas)
- Propiedades: `dim`, `max`, `min`, `length` (cantidad total de elementos)
- Operaciones: `det`, `solve`, `mean`, `colMeans`, `rowMeans`, `sd`

Matrices - Indexación

La principal manera para acceder a los datos es usar `matriz[filas, columnas]` a esta forma se la denomina INDEXACIÓN MÚLTIPLE.

$$A = \begin{pmatrix} 1 & 5 & 8 \\ 4 & 8 & 7 \\ 9 & 1 & 3 \end{pmatrix} \quad A = \begin{bmatrix} [1,1] & [1,2] & [1,3] \\ [2,1] & [2,2] & [2,3] \\ [3,1] & [3,2] & [3,3] \end{bmatrix}$$

Ejercicio: crear la matriz A y acceder al elemento que se encuentra en la columna 2 - fila 3.

Matrices - Indexación

Ejercicio: crear la matriz A y acceder al elemento que se encuentra en la columna 2 - fila 3.

```
# Creo la matriz M indicandole que complete por filas  
A <- matrix(data = c(1, 5, 8, 4, 8, 7, 9, 1, 3),  
            ncol = 3, nrow = 3, byrow = T)
```

```
A #muestro la matriz A por pantalla
```

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

```
# Accedo al elemento que se encuentra en la fila 3 - columna 2  
A[3,2]
```

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

Matrices indexación

También puedo acceder al elemento de una matriz indicando el número de posición del elemento. A esto se denomina indexación lineal.

$$A = \begin{pmatrix} 1 & 5 & 8 \\ 4 & 8 & 7 \\ 9 & 1 & 3 \end{pmatrix} \quad A = \begin{bmatrix} [1] & [4] & [7] \\ [2] & [5] & [8] \\ [3] & [6] & [9] \end{bmatrix}$$

Matrices indexacion (which)

Como en el caso de los vectores, podemos utilizar la funcion `which` para seleccionar ciertas posiciones (y luego elementos) que cumplen alguna condicion de una matriz.

En este caso, si queremos que la funcion `which` nos proporcione la fila y columna de la matriz tenemos que agregar el argumento `arr.ind=TRUE`

Matrices indexación (which)

Ejemplo: la matriz velocidad nos proporciona la intensidad de la velocidad del viento (m/s) en una provincia

```
velocidad<- matrix(c(10, NA, 15, 1, 5, NA, 20, NA, 50, NA, 12, 16), ncol =4 )  
velocidad
```

```
##      [,1] [,2] [,3] [,4]  
## [1,]  10   1  20  NA  
## [2,]  NA   5  NA  12  
## [3,]  15  NA  50  16
```

Queremos saber en que posiciones hay datos faltantes:

```
which(is.na(velocidad))
```

```
## [1]  2  6  8 10
```

```
which(is.na(velocidad), arr.ind = T)
```

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

Matrices ejercicio

Nos proporcionaron datos mensuales de precipitación de diciembre, enero y febrero de tres estaciones meteorológicas de Argentina:

Estación	Pp diciembre	PP enero	PP febrero
Mendoza	28 mm	40 mm	43 mm
Jujuy	130 mm	153.2 mm	152.9 mm
Buenos Aires	118.9 mm	135.4 mm	127.2 mm

- Construir una matriz (`matriz_pp_verano`) de 3 columnas donde cada columna representa cada estación meteorológica
- Para recordar que es lo que está guardado en la matriz, ponerle nombres a las filas y a las columnas
- Calcular la precipitación media de verano para las tres estaciones y guardar estos valores en un vector
- Agregar a la matriz `matriz_pp_verano` una columna con la precipitación medias de verano

Matrices ejercicio

- Agregar la informacion de otra estacion a nuestra matriz:

Estación	Pp diciembre	PP enero	PP febrero	Valor medio verano
Neuquen	11.3 mm	12.5 mm	11.9 mm	27.8 mm

- Calcular la pp media de enero de todas las estaciones. Para eso utilizar la funcion `mean()` y ademas indexación múltiple[,].

Arrays en R

```
# Por ejemplo quiero quedarme con todos los datos del dia 2,  
dia_2 <- datos[,2] #en este caso le estoy diciendo quiero  
                   # todas las filas, todas las columnas,  
                   # pero en la tercera dimension le digo que quiero el dia 2.
```

```
dim(dia_2)
```

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

```
dia_2
```

```
##      [,1] [,2] [,3] [,4] [,5] [,6]  
## [1,]  31  36  41  46  51  56  
## [2,]  32  37  42  47  52  57  
## [3,]  33  38  43  48  53  58  
## [4,]  34  39  44  49  54  59  
## [5,]  35  40  45  50  55  60
```

Arrays - apply

apply se utiliza para aplicar una función a cada elemento de un array. Los argumentos de esta función son: el array al que le quiero aplicar la función, las dimensiones que quedan fijas y la función que aplico.

```
datos<-array(data = c(1:120), dim = c(5,6,4))  
dim(datos)
```

```
## [1] 5 6 4
```

```
# Queremos hacer un promedio en los dias  
promedio_dias <- apply(datos, c(1,2), mean )  
  
dim(promedio_dias)
```

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

```
promedio_dias
```

```
##      [,1] [,2] [,3] [,4] [,5] [,6]  
## [1,]  46  51  56  61  66  71  
## [2,]  47  52  57  62  67  72  
## [3,]  48  53  58  63  68  73  
## [4,]  49  54  59  64  69  74  
## [5,]  50  55  60  65  70  75
```

Array - apply

A veces es conveniente “reacomodar” el array antes de usar la función apply. Veamos el caso del ejemplo anterior con el array datos

```
datos<-array(data = c(1:120), dim = c(5,6,4))
datos
```

```
## , , 1
##
##      [,1] [,2] [,3] [,4] [,5] [,6]
## [1,]    1    6   11   16   21   26
## [2,]    2    7   12   17   22   27
## [3,]    3    8   13   18   23   28
## [4,]    4    9   14   19   24   29
## [5,]    5   10   15   20   25   30
##
## , , 2
##
##      [,1] [,2] [,3] [,4] [,5] [,6]
## [1,]   31   36   41   46   51   56
## [2,]   32   37   42   47   52   57
## [3,]   33   38   43   48   53   58
## [4,]   34   39   44   49   54   59
## [5,]   35   40   45   50   55   60
##
## , , 3
##
##      [,1] [,2] [,3] [,4] [,5] [,6]
## [1,]   61   66   71   76   81   86
## [2,]   62   67   72   77   82   87
```

Array - apply

```
## Ahora en vez de tener 5 longitudes (filas), 6 latitudes (columnas) y 4 dias,  
# reacomodemos para que sean 30 filas (lon*lat) y 4 columnas (dias)  
datos_2<- matrix(data= datos, ncol = 4, nrow = 30)  
  
## Si ahora usamos la funcion rowMeans obtendriamos el promedio en los dias para cada lon-lat,  
## y el resultados seria similar al que obtuvimos en promedio_dias  
  
promedio_filas_datos2<- rowMeans(datos_2)  
promedio_filas_datos2  
  
## [1] 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70  
## [26] 71 72 73 74 75
```

Array - ejercicio

A partir del archivo “datos_viento.RData” que contiene datos diarios de la componente u del viento en diferentes niveles de la atmósfera para el mes de diciembre de 2017 sobre Sudamérica:

(Cargar el archivo con la funcion `load(“ruta_Archivo/nombre_Archivo.RData”)`)

- Obtener el promedio de la temperatura de diciembre para cada nivel y punto de retícula (AYUDA: el resultado debería ser una matriz para cada nivel).
- Seleccione el nivel de 850hPa y guardar en una matriz el promedio de temperatura sobre el dominio.
- Para el promedio de la temperatura de diciembre del nivel de 200 hPa, sumar todos los valores del dominio y guardarlos en un vector

Ayuda: en el archivo “datos_dimensiones_viento.RData” se encuentran las dimensiones del objeto que se encuentra en el archivo “datos_viento.RData”