

VECTORES Y MATRICES

VECTORES

Un vector es un objeto que almacena un conjunto de valores en una sola fila. Su estructura es **variable=c ()**. Dentro del paréntesis se deben introducir los valores que le queremos asignar al vector, separados por comas. En caso de querer incluir nombres, estos deberán ir entre comillas.

```
Ejemplos: > vector1=c (1,3,8)
          > vector2=c (19,40,9,13)
          > vector1
          [1] 1 3 8
          > vector2
          [1] 19 40 9 13
          > z1=c (10, -3.5, "pepito")
```

En este caso, el vector z1 tiene 3 componentes, dos de las cuales son numéricas y una de ellas es de caracteres. Si queremos que se escriba en contenido del vector z1, se obtendrá:

```
> z1
[1] "10" "-3.5" "pepito"
```

En el caso de que se desee acceder únicamente a una de las componentes del vector, se debe introducir el nombre del vector y seguido un corchete con el número de la componente a la que queremos acceder.

Ejemplo: Si ahora queremos acceder a la tercera componente del vector z1, se introducirá:

```
> z1[3]
[1] "pepito"
```

Si se quisieran sumarse o restarse las componentes de dos vectores, todas sus componentes deben de ser numéricas. Si en alguno de los vectores hay una componente no numérica, dichos vectores no podrán ser utilizados para realizar operaciones matemáticas.

```
Ejemplos: > vector1=c (17/3, 4, exp(pi))
          > vector2=c(sin(4*pi/3), -36, 5^3)
          > vector1 + vector2
          [1] 14.80000 92.45000 88.66667 127.39000
          > vector2 - vector1
          [1] 11.20000 85.55000 -87.33333 -121.79000
          > vector1*vector2
          [1] 23.40000 307.05000 58.66667 348.85200
```

Esta última operación es el producto de dos vectores componente a componente. En el caso de querer realizar el producto escalar de dos vectores se introducirá:

```
> vector1%*%vector2
[1]
[1,] 2743.679
```

Una matriz es un conjunto bidimensional de números. En R, si queremos crear una matriz se puede realizar de distintas maneras, empleando distintos comandos.

1. CON MATRIX

Con este comando la matriz se crearía siguiendo la fórmula: **A=matrix (data, nrow, ncol)**. Siendo data los datos que queremos introducir a la matriz añadiéndolos en forma de vector (c= ()), nrow corresponde con el número de filas y ncol con el número de columnas.

Ejemplos: > A=matrix (c (10, 4, 8, 3, 6, 9), nrow=3, ncol=3)

```
> A
      [,1] [,2] [,3]
[1,]  10   3  10
[2,]   4   6   4
[3,]   8   9   8
```

Si deseamos acceder a una posición concreta, la introduciremos entre corchetes precedidos por el nombre que le hayamos asignado a la matriz.

Ejemplo: En una matriz queremos acceder a la posición (2,1):

```
> A=matrix (c (3, 4, 5, 6), nrow=2, ncol=2)
> A
      [,1] [,2]
[1,]   3   5
[2,]   4   6
> A [2,1]
[1] 4
```

2. CON CBIND

Con este comando la matriz se crearía siguiendo la fórmula: **A=cbind (a, b, c,...)**. Siendo a, b y c vectores, y estos vectores se dispondrán dentro de la matriz en forma de columnas. Hay que tener en cuenta que los vectores deben tener la misma dimensión.

Ejemplo: > a=c (2, 5, 7)

```
> b=c (4, 8, 1)
> c=c (3, 6, 9)
> A=cbind (a, b, c)
> A
      a  b  c
[1,]  2  4  3
[2,]  5  8  6
[3,]  7  1  9
```

3. CON RBIND

Con este comando la matriz se crearía siguiendo la fórmula: **A=rbind (a, b, c,...)**. Siendo a, b y c vectores, y estos vectores se dispondrán dentro de la matriz en forma de filas. Hay que tener en cuenta que los vectores deben tener la misma dimensión.

Ejemplo: > a=c (7, 8, 9)

```
> b=c (1, 3, 5)
> c=c (2, 4, 6)
> A=rbind (a, b, c)
> A
      [,1] [,2] [,3]
a       7   8   9
b       1   3   5
c       2   4   6
```

1. SUMA Y RESTA

Para sumar o restar dos matrices, deben ser del mismo orden, es decir, deben tener el mismo número de filas y de columnas.

Ejemplos: > A=matrix (c (3, 2, -1, 0, 7/6, 0.25), nrow=3, ncol=2)

> A

```
      [,1] [,2]
[1,]  3  0.000
[2,]  2  1.667
[3,] -1  0.25
```

> B=matrix (c (-3, 12, -10, 0.3, 3.5, 0.95), nrow=3, ncol=2)

> B

```
      [,1] [,2]
[1,] -3  0.30
[2,] 12  3.50
[3,] -10 0.95
```

> A + B

```
      [,1] [,2]
[1,]  0  0.300
[2,] 14  4.667
[3,] -11 1.200
```

> A - B

```
      [,1] [,2]
[1,]  6 -0.300
[2,] -10 -2.333
[3,]  9 -0.700
```

2. PRODUCTO DE MATRICES

Para que se puedan multiplicar dos matrices, deben tener una dimensión adecuada, es decir, el número de columnas de la primera matriz y el número de filas de la segunda debe coincidir. Para realizar el producto de las matrices se utiliza el símbolo %*%.

Ejemplo: > A=matrix (c (3, 21, 2, 3, 4, 10), nrow=3, ncol=2)

> B=matrix (c (13, 1, pi, 3.5, 4, 100), nrow=2, ncol=3)

> A

```
      [,1] [,2]
[1,]  3  3
[2,] 21  4
[3,]  2 10
```

> B

```
      [,1] [,2] [,3]
[1,] 13  3.141593  4
[2,]  1  3.500000 100
```

> M=A%*%B

> M

```
      [,1] [,2] [,3]
[1,] 42  19.92478 312
[2,] 277 79.97345 484
[3,] 36  41.28319 1008
```

3. OTRAS OPERACIONES CON MATRICES

Además, se pueden realizar con matrices distintas operaciones aritméticas como las vistas anteriormente y nuevas (+, -, /, *, ** o ^). Pero siempre teniendo en cuenta como tienen que ser las dimensiones de las matrices para que dichas operaciones se puedan realizar sin problemas.

Otras operaciones con matrices			
t(A)	Traspuesta de la matriz A	svd(A)	Descomposición en valores singulares
det(A)	Determinante de la matriz A	qr(A)	Descomposición QR
solve(A,b)	Solución del sistema de ecuaciones	eigen(A)	Valores y vectores propios
Ax=b	Sistema	diag(a)	Matriz diagonal (b es un vector)
solve(A)	Inversa de la matriz A	Diag(A)	Matriz diagonal (A es una matriz)

4. ALGUNAS FUNCIONES SOBRE MATRICES

Hay algunas funciones que pueden darnos información de las matrices y de los vectores. Estas funciones son:

- **Dim (A)**= Indica las dimensiones de la matriz creada.
- **Length (A)**= Indica el número total de los elementos de la matriz.
- **A [número, número]**= Te permite acceder a elementos dentro de la matriz.

Estas mismas funciones son aplicables a los vectores.

Ejemplos: > A=matrix (c (3, 4, 5, 6), nrow=2, ncol=2)

```
> A
     [,1] [,2]
[1,]   3   5
[2,]   4   6
> dim (A)
[1] 2  2
> length (A)
[1] 4
> A [1,1]
[1] 3
```

TABLAS

Para crear una tabla se suele utilizar la instrucción **data.frame (vector1, vector2, vector3...)**. Este comando es una estructura de datos que generaliza matrices y permite introducir datos de distintos tipos (números, letras, palabras...). Eso sí, los datos de cada vector o matriz creados deben de ser del mismo tipo, no se deben mezclar. Además, para crear el data.frame los vectores o matrices que se quieren introducir en la tabla deben tener la misma longitud. Siempre se debe tener en cuenta que, en caso de querer introducir palabras o letras, estas deberán ir entre comillas.

En cuanto al uso de tildes en las palabras, se pueden utilizar, sin embargo, en algunos sistemas operativos no aparecen o dan error o aparece en otro formato la palabra, por lo que también se debe de tener cuidado en su uso.

Ejemplo: > Grados=c ("Biotecnología", "Medicina", "Matemáticas", "Bioquímica")

```
> Notacorte=c (13.168, 13.476, 12.825, 12.944)
> Alumnos=c (80, 120, 30, 100)
> Tabla=data.frame (Grados, Notacorte, Alumnos)
> Tabla
```

Grados	Notacorte	Alumnos
Biotecnología	13.168	80
Medicina	13.476	120
Matemáticas	12.825	30
Bioquímica	12.944	100

GRÁFICAS

Para hacer una representación gráfica en R, se deben tener dos vectores con datos almacenados para asignarle a uno de ellos los valores correspondientes a las abscisas y al otro el de las ordenadas. Para realizar el gráfico se debe emplear el comando **plot (X, Y, xlab= '...', ylab='...', main='...', col='...', pch=n, type='...')**.

Donde está escrita la X, se debe introducir el vector que lleve los valores que queremos almacenar en el eje de abscisas.

En la Y, introduciremos el vector que lleve los valores que queremos almacenar en el eje de ordenadas.

En xlab e ylab se deben introducir entre comillas los nombres que le queremos asignar a los ejes.

En main se debe introducir el título que le queremos asignar a la gráfica.

Los subcomandos col, pch, y type, son optativos para introducir color en la línea del gráfico, distintas figuras en los puntos de la función y el tipo de gráfico, respectivamente.

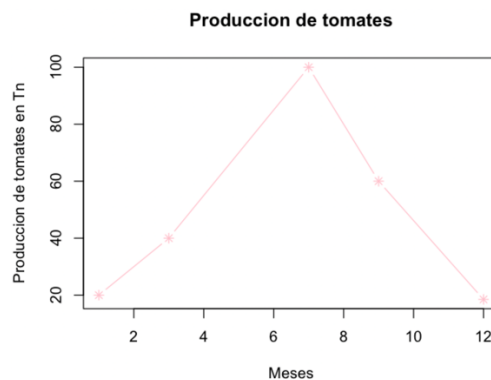
Ejemplo: Tenemos los datos de la producción de tomates en toneladas durante diferentes meses del año, y se quiere representar los puntos en un gráfico bidimensional. Se haría de esta manera:

```
> meses=c (1, 3, 7,9, 12)
```

```
> produccion=c (20, 40, 100, 60, 18.5)
```

```
> plot (meses, produccion, xlab="Meses", ylab="Producción de tomates en Tn", main="Producción tomates", col="pink",  
pch=8, type="b")
```

Y el resultado se mostraría así:



Con el comando **type="..."** tenemos la opción de elegir el tipo de gráfica que se quiere obtener, y para ello se deben utilizar distintas letras:

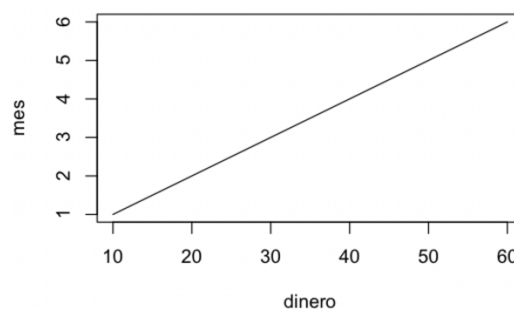
- Utilizaremos la letra **l** en caso de querer unir los puntos con una línea sin querer que estos mismos aparezcan.

Ejemplo:

```
> dinero=c (10, 20, 30, 40, 50, 60)
```

```
> mes=c (1, 2, 3, 4, 5, 6)
```

```
> plot(dinero, mes, type='l')
```



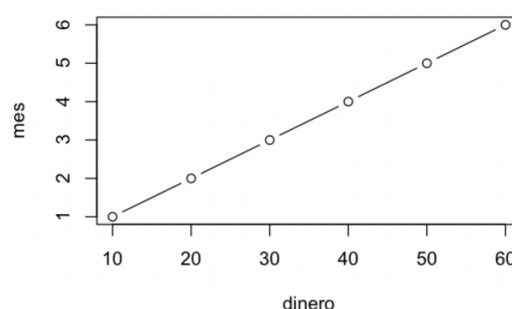
- Utilizaremos la letra **b** en caso de querer unir una línea con puntos, incluyendo los puntos.

Ejemplo:

```
> dinero=c (10, 20, 30, 40, 50, 60)
```

```
> mes=c (1, 2, 3, 4, 5, 6)
```

```
> plot(dinero, mes, type='b')
```

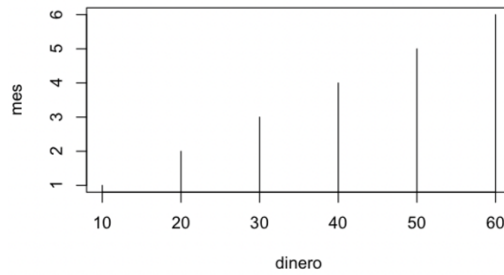


- Utilizaremos la letra **h** en caso de querer que el gráfico tenga forma de histograma.

Ejemplo: > dinero=c(10, 20, 30, 40, 50, 60)

> mes=c(1, 2, 3, 4, 5, 6)

> plot(dinero, mes, type='h')



Para que el gráfico aparezca en un color determinado, se utilizará el comando **col="..."**, dentro de las comillas se deberá introducir el color que deseamos que aparezca en inglés.

Asimismo, se puede añadir una leyenda al gráfico con el comando **legend ()**, en el que se puede especificar con los siguientes comandos:

- Se puede utilizar **x=n, y=n**, siendo n las coordenadas donde se encuentra la leyenda. Además, al comando **x="..."**, se le puede añadir algunos de los siguientes datos para colocar directamente la leyenda: "bottomright", "bottom", "bottomleft", "left", "topleft", "top", "topright", "right", "center".

- Se puede utilizar **legend=n**, siendo n las etiquetas de los datos que queremos describir con la leyenda.

- Se puede utilizar **fill="color"**, en el que se introduce en inglés, los colores que acompañan a las etiquetas. Se debe tener en cuenta que estos colores deben coincidir con los que hemos usado en el gráfico.

- Se puede utilizar **title="..."**, para poner un título.

Ejemplo: > dinero=c(10, 20, 30, 40, 50, 60)

> mes=c(1, 2, 3, 4, 5, 6)

> plot (mes, dinero, xlab="Meses", ylab="Dinero", col="pink", type="b")

> legend(x="topright", legend=dinero, fill="pink", title="Meses representados")

