RESUMEN PRÁCTICA 2

En esta segunda práctica el trabajo también se enfoca, al igual que en la primera en introducir un poco el lenguaje de programación. Vamos a usar matrices, vectores y tablas.

En primer lugar vamos a ver cómo creamos un **vector**:

Le podemos poner el nombre que queramos (en este caso "v"). V será una "dirección" donde se guarde todo lo que vamos a poner a la derecha del igual.

> En caso de querer que el programa use/escriba ese vector, debemos poner únicamente el nombre que le hayamos dado.

Al lado del nombre pondremos un igual, y lo primero que escribiremos será una letra "c". Ese será el indicativo para el ordenador de que estamos creando un vector. Posteriormente, abriremos paréntesis y pondremos entre comas las componentes del vector.

```
v = c(10,-1.5,2/3,sin(pi),sqrt(3))
```

Podemos sumar vectores (v + u) y también multiplicarlos (u*v). En caso de querer realizar el producto escalar, tendremos que escribir (u%*%v)

En la práctica 2 también vamos a empezar a tratar con matrices.

Para crear una matriz tenemos dos métodos:
- Creando vectores y luego indicando que los juntamos para crear una matriz.
- Crear la matriz directamente con un vector y escribiendo el nº de filas y columnas (puede que sea más difícil colocar correctamente las filas y columnas)

- **1ª forma**:
  Vamos a crear varios vectores que incluyan datos:

```
v = c(10,-1.5,2/3,sin(pi),sqrt(3))
u= c(1/5,12,8/7,exp(2),3)
z= c(1,2,3,4,5,6)
```

Después, vamos a escribir el nombre de la matriz (MatA en nuestro caso), un igual y vamos a usar 2 comandos:

- **Rbind** → Va a colocar los vectores como filas de la matriz (Row)
- **Cbind** → Va a colocar los vectores como las columnas de la matriz (Column)

Entre paréntesis añadiremos los vectores que queremos usar (v,u,z)

```
MatA <- rbind (v,u,z)
MatB <- cbind (v,u,z)

MatA
MatB
```

```
> MatA
  [,1] [,2]      [,3]       [,4]     [,5] [,6]
v 10.0 -1.5 0.6666667 -0.6301012 1.732051 10.0
u  0.2 12.0 1.1428571  7.3890561 3.000000  0.2
z  1.0  2.0 3.0000000  4.0000000 5.000000  6.0
> MatB
              v         u z
[1,] 10.0000000  0.200000 1
[2,] -1.5000000 12.000000 2
[3,]  0.6666667  1.142857 3
[4,] -0.6301012  7.389056 4
[5,]  1.7320508  3.000000 5
[6,] 10.0000000  0.200000 6
```

❖ **2ª forma**

```
A<- matrix (c(1,2,3,4,5,6),nrow=3, ncol=2)
```

Vamos a obtener una matriz de 3x2. En este programa, la matriz se va rellanando con los valores del vector de forma vertical, es decir, rellena la primera columna, y después comienza a rellenar la segunda columna…

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

¿Qué pasa si no ponemos datos suficientes para rellenar la matriz completa? Pues vuelve a empezar a rellenar por el primer dato que hemos introducido (el 1)

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

```
> A
     [,1] [,2]
[1,]    1    4
[2,]    2    1
[3,]    3    2
```

Otra posibilidad que os ofrece este programa es calcular un determinante.

Solo tenemos que escribir: det(AA) y nos mostrará el resultado

```
> AA= matrix (c(1,2,3,4),nrow=2,ncol=2)
> det(AA)
[1] -2
```

También podemos calcular la matriz traspuesta (t(B))

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

Otro aspecto tratado en la práctica 2 es la resolución de un sistema de ecuaciones:

1º Construimos una matriz con los coeficientes (los datos que acompañan a las incógnitas)

```
#Consideramos el sistema de ecuaciones:

# 2x+3y-2z=18.5; -x+4y-z=12.3; x/4 +2/3y-8z=16.5

# Construir matriz de coeficientes (empleando rbind)
v= c (2,3,-2)
u= c(-1,4,-1)
z= c (1/4,2/3,-8)
COEF = rbind (v,u,z)
```

2º Construimos un vector con los términos independientes (lo que no lleva incógnitas)

```
#Construir vector términos independientes
a= c(18.5,12.3,16.5)
```

3º Usamos la función Solve (,) para resolver el sistema y lo almacenamos en un vector (z en nuestro caso):

```
#Resolver y almacenar el vector
z= solve(COEF,a)
z
```

```
> z= solve(COEF,a)
> z
[1]  2.597027  3.297621 -1.706541
```

Antes de proceder a explicar la última parte de la práctica tenemos que recordar que en un vector podemos introducir datos que no sean números (siempre entre comillas)

```
u=c(13,-14,"manzanas")
```

Por último, vamos a explicar como crear tablas.

Vamos a usar la función data.frame(,)

1º Creamos los vectores que necesitemos. Los vectores van a ser las columnas de la tabla.

2º Escribimos el data.frame (vector1, vector2, vector i, vector n) dentro de un vector para que sea más accesible después.

IMP → Los nombres de los vectores son los títulos de las columnas.

```
Nombre= c("Ana","María", "Juan", "Antonio")
Peso= c(57,85,93,102)
Estatura= c(158,190,99,100)

pp=data.frame(Nombre, Peso, Estatura)
pp
```

```
> pp
   Nombre Peso Estatura
1     Ana   57      158
2   María   85      190
3    Juan   93       99
4 Antonio  102      100
```

Esta segunda práctica fue también para dar los primeros pasos en este nuevo programa.