

## PRACTICA 2:

El **objetivo** es realizar operaciones básicas con lo visto en la práctica anterior y ver otros objetos de interés como los vectores y matrices, así como realizar operaciones con ellos

### **Introducción:**

<https://youtu.be/D39db7hnpTs>

### **Cálculo Con Vectores:**

Primeramente tendremos que definir dos vectores, para ello utilizaremos lo aprendido en la practica 1:

```
> v=c(2, sin(3), 100)
> v
[1] 2.00000 0.14112 100.00000
> y=c(14, cos(4), 32)
> y
[1] 14.0000000 -0.6536436 32.0000000
```

Una vez que hemos definido dos vectores (como minimo), empezaremos con las operaciones básicas:

- **Suma:**

```
> y+v
[1] 16.0000000 -0.5125236 132.0000000
```

- **Resta:**

```
> y-v
[1] 12.0000000 -0.7947636 -68.0000000
```

- **Producto Escalar:** usaremos %\*%

```
> y%*%v
      [,1]
[1,] 3227.908
```

### **Calculo Con Matrices:**

Primeramente tendremos que definir dos matrices, para ello utilizaremos lo aprendido en la práctica 1:

```
> A=matrix(c(-6, 4, 5, 7, 10, sqrt(pi)), nrow = 3, ncol = 2)
> A
```

```

      [,1] [,2]
[1,] -6 7.000000
[2,]  4 10.000000
[3,]  5 1.772454
> B=matrix(c(9, 10, exp(3), sin(1), 4, -10), nrow = 3, ncol = 2)
> B
      [,1] [,2]
[1,] 9.00000 0.841471
[2,] 10.00000 4.000000
[3,] 20.08554 -10.000000

```

Una vez que hemos definidas dos matrices (como minimo), empezaremos con las operaciones básicas:

- **Suma:**

```

> A+B
      [,1] [,2]
[1,] 3.00000 7.841471
[2,] 14.00000 14.000000
[3,] 25.08554 -8.227546

```

- **Resta:**

```

> A-B
      [,1] [,2]
[1,] -15.00000 6.158529
[2,] -6.00000 6.000000
[3,] -15.08554 11.772454

```

- **Producto:**

```

> C=A*B
> C
      [,1] [,2]
[1,] -54.0000 5.890297
[2,] 40.0000 40.000000
[3,] 100.4277 -17.724539

```

Otras **operaciones** de interes:

- **Matriz Transpuesta:** para obtener la matriz traspuesta, usaremos el comando “**t ()**”:

```

> t(A)
      [,1] [,2] [,3]
[1,] -6  4 5.000000
[2,]  7 10 1.772454

```

- **Determinante de una Matriz:** para obtener el determinante de la matriz , usaremos el comando “**det ()**”:

```
> C=matrix(c(2, 25, sin(2), 45), nrow = 4, ncol = 4)
```

```
> C
```

```
      [,1] [,2] [,3]
[1,] 2.0000000 2.0000000 2.0000000
[2,] 25.0000000 25.0000000 25.0000000
[3,] 0.9092974 0.9092974 0.9092974
[4,] 45.0000000 45.0000000 45.0000000
```

```
      [,4]
[1,] 2.0000000
[2,] 25.0000000
[3,] 0.9092974
[4,] 45.0000000
```

```
> det(C)
```

```
[1] 0
```

- **Matriz Inversa:** para obtener una matriz inversa, primeramente tendremos que definir la matriz identidad:

```
> I=diag(1,nrow=4) # Matriz diagonal de dimensión 4
```

```
> I
```

```
      [,1] [,2] [,3] [,4]
[1,] 1 0 0 0
[2,] 0 1 0 0
[3,] 0 0 1 0
[4,] 0 0 0 1
```

Una vez creada la matriz identidad, usaremos el comando “**solve ()**” (introduciremos I y la matriz que deseamos calcular su inversa) para hallar la matriz inversa:

```
> solve(C,I)
```

```
      [,1] [,2] [,3]
[1,] -0.7307692 0.2307692 0.6538462
[2,] 0.5000000 0.0000000 -0.5000000
[3,] 0.1538462 -0.1538462 0.2307692
```

- **Sistema de Ecuaciones:**

**Ejemplo:** se quiere resolver un sistema de ecuaciones se debe utilizar el siguiente planteamiento sea  $2X+3Y=4$  Y  $4x+2y=9$

- **Para las variables x:**

```
> x<-matrix(c(2,4,3,2), 2,2)
```

```
> x
```

```
      [,1] [,2]
[1,] 2 3
```

```
[2,] 4 2
```

- **Para las variables y:**

```
> y<-matrix(c(4,9),2,1)
> y
     [,1]
[1,]  4
[2,]  9
```

- **Lo resolvemos:**

```
> r<-solve(t(x)%*%x)%*%t(x)%*%y
> r
     [,1]
[1,] 2.375
[2,] -0.250
```

### ***Uso de Data.Frame:***

Se utiliza para almacenar datos u objetos en forma de hoja de datos. Cada fila, corresponderá a una observación o valor de un objeto, mientras que en cada columna corresponde a un vector que contiene los datos de la variable.

**Ejemplo:** Crea una tabla, donde se represente el peso, el nombre y la estatura de cuatro personas al alzar.

- **Primero:** definimos los vectores “edad”, “peso” y “estatura”

```
> nombre=c("Julia","Pepe", "Roberto", "Escolastica")
> peso=c(57,85,93,102)
> estatura=c(158,190,99,100)
```

- **Segundo:** crearemos el cuadro de datos con el comando “**data.frame ()**”

```
> pp=data.frame(nombre,peso,estatura)
> pp
  nombre peso estatura
1  Julia  57    158
2  Pepe  85    190
3 Roberto  93     99
4 Escolastica 102   100
```