

# S03 T04: Pràctica amb programació numèrica

Exercici 1: Crea una funció que donat un Array d'una dimensió, et faci un resum estadístic bàsic de les dades. Si detecta que l'array té més d'una dimensió, ha de mostrar un missatge d'error.

```
In [48]: import numpy as np

def estadistica(m):
    if m.ndim > 1:
        print('Error. la matriu ha de ser 1D')
    else:
        print('La mitjana és: ' + str(m.mean()))
        print('La mediana és: ' + str(np.median(m)))
        print('El coeficient de correlació és: ' + str(np.corrcoef(m)))
        print('La desviació estàndard és: ' + str(np.std(m)))
```

```
In [49]: # provem la funció en un array 1D:

m = np.array([2, 3, 4, 3, 2])
estadistica(m)

La mitjana és: 2.8
La mediana és: 3.0
El coeficient de correlació és: 1.0
La desviació estàndard és: 0.7483314773547882
```

```
In [50]: # provem la funció en un array > 1D:

a = np.array([[2, 3, 4], [3, 2, 3]])
estadistica(a)
```

Error. la matriu ha de ser 1D

Exercici 2: Crea una funció que et generi un quadrat NxN de nombres aleatoris entre el 0 i el 100.

```
In [198... def quadrat(N):
    N = int(input('Introdueix un nombre enter N per generar una matriu NxN: '))
    return np.random.randint(0, 101, size=(N, N))
```

```
In [193... # provem la funció:

quadrat(N)

Introdueix un nombre enter N per generar una matriu NxN: 4

Out[193... array([[51, 27, 86, 43],
       [66, 43, 62, 89],
       [ 3,  4, 75, 39],
       [15, 53, 49, 87]])
```

```
In [283... import random
res = []
N = int(input('Tria un valor de N per crear una matriu NxN: '))
for x in range (0,N):
    line = []
    for y in range (0,N):
        line.append(int(random.uniform(0,101)))
    res.append(line)

print(res)
```

Tria un valor de N per crear una matriu NxN: 3  
[[1, 78, 37], [16, 42, 42], [95, 12, 87]]

Exercici 3: Crea una funció que donada una taula de dues dimensions, et calculi els totals per fila i els totals per columna.

```
In [171... def sumatori(m):
    #suma files
    files = m.sum(axis=1)
    print('el sumatori de les files és: ' + str(files))

    #suma columnes
    columnes = m.sum(axis=0)
    print('El sumatori de les columnes és: ' + str(columnes))
```

```
In [284... # provem la funció amb un exemple:

a = np.array([
    [1, 2, 0],
    [3, 4, 5]])

sumatori(a)

el sumatori de les files és: [ 3 12]
El sumatori de les columnes és: [4 6 5]
```

Exercici 4: Implementa manualment una funció que calculi el coeficient de correlació. Informa-te'n sobre els seus usos i interpretació.

```
In [311... def coeficient_correlacio (X, Y):

    # primer calculem la mitjana dels conjunts de dades:
    mean_X = sum(X) / len(X)
    mean_Y = sum(Y) / len(Y)

    # COVARIANÇA(X,Y):
    XY = []
    for i in range (0, len(X)):
        XY.append(X[i]*Y[i])
    mean_XY = sum(XY) / len(XY)

    cov_XY = mean_XY - (mean_X * mean_Y)
    print('La covariança entre les variables és: ' + str(cov_XY))

    if cov_XY > 0:
        print('Dependència directa (positiva) entre variables.')
    if cov_XY < 0:
        print('Dependència inversa o negativa entre variables.')
    if cov_XY == 0:
        print('no existeix relació lineal entre ambdues variables.')

    # DESVIACIÓ TÍPICA(SD):
    X_sqr = 0
    for i in X:
        X_sqr = X_sqr + (i**2)
    SD_X = ((X_sqr / len(X)) - (mean_X**2))**0.5

    Y_sqr = 0
    for i in Y:
        Y_sqr = Y_sqr + (i**2)
    SD_Y = ((Y_sqr / len(Y)) - (mean_Y**2))**0.5

    # COEFICIENT CORRELACIÓ (R):
    # R = covar(x,y) / (desv. típ. X * desv. típ. Y)

    R = cov_XY / (SD_X * SD_Y)
    print('El coeficient de correlació és: ' + str(R))

    if R == 1:
        print('Correlació positiva perfecta: dependència total entre les dues variables. Si una augmenta, l\'altra augmenta.')
    if 0 < R < 1:
        print('Correlació positiva. A valors alts d\'una variable li corresponen valors alts de l\'altra, i igualment amb els valors baixos.')
    if R == 0:
        print('No existeix relació lineal entre variables.')
    if -1 < R < 0:
        print('Correlació negativa. A valors alts d\'una de les variables li corresponen valors baixos de l\'altra, i igualment amb els valors alts.')
    if R == -1:
        print('Correlació negativa/inversa perfecta: dependència total entre les dues variables. Quan una augmenta, l\'altra disminueix.')
```

```
In [312... # provem la funció per veure la correlació de 12 estudiants entre les notes de mates i de física:

mates = [2, 3, 4, 4, 5, 6, 6, 7, 7, 8, 10, 10]
fisica = [1, 3, 2, 4, 4, 4, 4, 6, 4, 6, 7, 9, 10]

coeficient_correlacio (mates, fisica)
```

La covariança entre les variables és: 5.9166666666666664  
Dependència directa (positiva) entre variables.  
El coeficient de correlació és: 0.9355071411331451  
Correlació positiva. A valors alts d'una variable li corresponen valors alts de l'altra, i igualment amb els valors baixos.