

# Lliurament tasca 4 - Pràctica amb programació numèrica

April 19, 2021

## 1 Lliurament tasca 4: Pràctica amb programació numèrica

### 1.1 Descripció

Familiaritza't amb la Programació Numèrica a través de la llibreria NumPy.

### 1.2 Nivell 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.

```
[1]: import numpy as np
import statistics as stat

def programa(array):
    array = np.array(array)
    if array.ndim == 1:
        print(f'Resum dels estadístics per l'array {array.ndim} dimensió:')
        print(f'Mitjana: {round(array.mean(),2)}')
        print(f'Màxim: {array.max()}')
        print(f'Moda: {stat.mode(array)}')
        print(f'Desviació típica: {round(array.std(),2)}')
    else:
        print(f'Error! Només podem saber el resum estadístic pels arrays d'una')
        print(f'dimensió. El teu array té {array.ndim} dimensions.')
# Exemple 1: Array 1 dimensió
programa([5,10,20,5,4,7,8,9,0,10,20,45])
print("")

# Exemple 2: Array 2 dimensions
programa([[1,2,3],[5,6,7]])
```

```
Resum dels estadístics per l'array 1 dimensió:
Mitjana: 11.92
Màxim: 45
Moda: 5
Desviació típica: 11.48
```

Error! Només podem saber el resum estadístic pels arrays d'una dimensió. El teu array té 2 dimensions.

### 1.3 Exercici 2

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

```
[2]: def matriu_quadrat():
    matriu = np.random.randint(0,100,(10,10))
    return matriu

print(matriu_quadrat())
```

```
[[71 35 78 14 50  3 14 66 38 77]
 [84 85 88 84 64 81 52 99 97 33]
 [11 94 21 23 31 62 74 98 78 16]
 [85 46 98 76 72 49 83 97 44  5]
 [67 82 11  8 98 10 34  6 11 58]
 [13 65 50 37 80 38 29 12 11 88]
 [92 65 25 82 38 52 79 73  3  1]
 [87 86 90 62 35 85  6 89 67 33]
 [30 78 75 88 11 67 74 63  5 47]
 [62 11 47 84 53 64 98 46 67 46]]
```

### 1.4 Exercici 3

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

```
[5]: def dos_dim(array2dim):
    matriu = np.array(array2dim)
    print(f'La matriu generada és:\n {matriu}\n')
    for i in matriu:
        print(f'Suma fila {i} és: {i.sum()}')
    for i in matriu.T:
        print(f'Suma columna {i} és: {i.sum()}')

dos_dim([[3,2,1],[5,2,1]])
```

La matriu generada és:

```
[[3 2 1]
 [5 2 1]]
```

```
Suma fila [3 2 1] és: 6
Suma fila [5 2 1] és: 8
Suma columna [3 5] és: 8
Suma columna [2 2] és: 4
Suma columna [1 1] és: 2
```

## 1.5 Exercici 4

Implementa manualment una funció que calculi el coeficient de correlació. Informa't-en sobre els seus usos i interpretació.

```
[4]: import numpy as np

x = np.random.randn(10)

y = np.random.randn(10)

corr = np.mean(np.multiply((x-np.mean(x)),(y-np.mean(y))))/(np.std(y)*np.std(x))
print(f'Coeficient de correlació entre x i y: {round(corr,2)}')
```

Coeficient de correlació entre x i y: -0.3

## 2 Coeficient de correlació

El coeficient de correlació és una mesura de regressió que quantifica la dependència lineal entre 2 variables.

**Condicions** Per tal de que el coeficient de correlació sigui vàlid, les condicions que s'han de cumplir són les següents:

- \* La relació a estudiar de tipus lineal.
- \* Les dues variables han de ser numèriques.
- \* Normalitat: Les dues variables han de distribuir-se de forma normal.
- \* Homocedasticitat: La variança d'Y ha de ser constant al llarg de la variable d'X.

**Valors del coeficient** Els valors del coeficient es mouen entre -1 y 1.

- \* Correlació perfecta positiva = 1
- \* Correlació perfecta negativa = -1
- \* No hi ha correlació = 0

**Interpretació** Quan la correlació es **positiva**, quan una variable augmenta l'altra també augmenta..

Quan la correlació es **negativa**, quan una variable augmenta, l'altra disminueix.

Quan la correlació es **0 (o al voltant de 0)**, no existeix relació lineal.