

Sprint 2 : S02 T04: Pràctica amb programació numèrica

Nivell 1

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 [32]:

```
import numpy as np
arr = np.array([56, 10.4, 54, 3, 6.6,12,87,45])
#arr = np.array([[1, 2, 3], [4, 5, 6], [7, 8, 9], [10, 11, 12]])

def estadistica(arr):
    dimension_shape = np.shape(arr)
    print("Array.shape: ", dimension_shape)
    dimension = np.ndim(arr)
    print ("Dimensión: " , dimension)
    if dimension > 1:
        print("La dimensión del array es errónea")
    else:
        media_arr = np.mean(arr)
        print("La Media es: ", media_arr)
        mediana_arr= np.median(arr)
        print ("La Mediana es :", mediana_arr)
        desv_std_arr = np.std(arr)
        print ("La Desviación Estándar es: ", desv_std_arr)
        primer_cuantil = np.percentile(arr, 25)
        tercer_cuantil = np.percentile(arr,75)
        print("Primer y Tercer cuantil:" , primer_cuantil,",", tercer_cuantil)
        print("Rango Intercuantil: ", tercer_cuantil - primer_cuantil)

estadistica(arr)
```

```
Array.shape: (8,)
Dimensión: 1
La Media es: 34.25
La Mediana es : 28.5
La Desviación Estándar es: 28.64921464892188
Primer y Tercer cuantil: 9.45 , 54.5
Rango Intercuantil: 45.05
```

Exercici 2

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

In [131]...

```
import random

# generar números enteros aleatorios entre 0 y 100 : radom.randint (0,100)
## Genera numero flotante aleatorio entre 0 y 100: random.uniform(0,100))
```

```

numero = 0

def cuadradoAleatorio(control):
    lista_numeros = []
    ## Creamos una lista con 10 números aleatorios entre 0 y 100
    for i in range(control):
        lista_numeros.append(random.randint(0,100))

    print("Los números aleatorios generados son:" , lista_numeros)
    print ("El cuadrado de los número aleatorios es: ", np.power(lista_numeros,2))

cuadradoAleatorio(10)

```

Los números aleatorios generados son: [39, 69, 72, 69, 89, 73, 85, 28, 36, 44]
 El cuadrado de los número aleatorios es: [1521 4761 5184 4761 7921 5329 7225 784 1296 1936]

Exercici 3

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

In [132...

```

import numpy as np
datos = np.array([[1,3], [4, 6]])

def suma(arr):
    suma_columna = np.sum(arr, axis=0)
    suma_fila = np.sum(arr, axis=1)
    print("La suma de las filas es: ", suma_fila, ". La suma de las columnas es: ",

suma(datos)

```

La suma de las filas es: [4 10] . La suma de las columnas es: [5 9]

Exercici 4

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

In [140...

```

import numpy as np

# Datos simulados de dos variables a y b
a = np.array([12,9,6,7,2,5,4,0,1,8])
print(a)
b = np.array([3,5,1,9,5,3,7,2,10,5])
print(b)
rango= a.size
print("Elementos de la muestra: ", rango)

def coeficienteCorr(a, b):
    media_a = np.mean(a)
    media_b = np.mean(b)

    i = 0
    numerador = 0
    denominador = 0
    denominador_a = 0
    denominador_b = 0

```

```
for i in range(rango):
    numerador = numerador + ((a[i]-media_a)*(b[i]-media_b))
    denominador_a = denominador_a + ((a[i] - media_a)**2)
    denominador_b = denominador_b + ((b[i] - media_b)**2)
denominador = (denominador_a * denominador_b)**(1/2)
coeficiente_corr = numerador /denominador
print ("El coeficiente de correlación es: ", coeficiente_corr)

coeficienteCorr(a,b)
```

[12 9 6 7 2 5 4 0 1 8]

[3 5 1 9 5 3 7 2 10 5]

Elementos de la muestra: 10

El coeficiente de correlación es: -0.16987094710168293