

Data Science amb Python

Estudiant: Cristiane de Souza da Silva

Lliurament tasca 4: Pràctica amb programació numèrica

Descripció

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

- 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 [1]: # Import the library numpy
import numpy as np

def summary_1d(array_id):
    if array_id.ndim != 1:
        print('Incorrect dimension')
    else:
        print('The average, standard deviation and median are, repectively:
              str(np.average(array_id)), str(np.std(array_id)), str(np.med:
```

```
In [2]: from numpy import random
x=random.randint(100, size=(12))
x
```

```
Out[2]: array([90, 80, 57, 23, 45, 12, 39, 78, 56, 45, 78, 13])
```

```
In [3]: summary_1d(x)
```

```
The average, standard deviation and median are, repectively:  51.333333333333
3336 25.567992142955266 50.5
```

```
In [ ]:
```

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

```
In [4]: def square_n_n(n):
        '''
        The function will create a square n x n.
        It's only needed to put the n , for example:
        a square 3 x 3, insert the number 3
        '''
        n = random.randint(100, size=(n, n))
        return n
```

```
In [5]: square_n_n(5)
```

```
Out[5]: array([[ 0, 25, 80, 19, 63],
               [92, 44, 26, 52, 46],
               [31, 12, 55, 24, 65],
               [ 2, 12, 73, 73, 77],
               [97, 80, 43, 84, 12]])
```

```
In [ ]:
```

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

```
In [6]: def total_2d(table_2d):
        if table_2d.ndim !=2:
            print('Incorrect dimension')
        else:

            total_row = table_2d.sum(axis=1)
            total_column = table_2d.sum(axis=0)

        return total_row, total_column
```

```
In [7]: y=square_n_n(2)
        y
```

```
Out[7]: array([[47,  6],
               [21, 33]])
```

```
In [8]: total_2d(y)
```

```
Out[8]: (array([53, 54]), array([68, 39]))
```

- Exercici 4 Implementa manualment una funció que calculi el coeficient de correlació. Informa't-en sobre els seus usos i interpretació.

```
In [9]: a=random.randint(100, size=(5))
        b=random.randint(100, size=(5))
```

```
In [11]: np.corrcoef(a, b)
```

```
Out[11]: array([[ 1.          , -0.43481177],  
               [-0.43481177,  1.          ]])
```

```
In [ ]:
```