

Sprint03_Tasca01

October 7, 2022

Sprint 3

Tasca 1: 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.
- Exercici 2: Crea una funció que et generi un quadrat NxN de nombres aleatoris entre el 0 i el 100.
- Exercici 3: Crea una funció que donada una taula de dues dimensions (NxM), et calculi els totals per fila i els totals per columna.
- Exercici 4: Implementa manualment una funció que calculi el coeficient de correlació. Informa-te'n sobre els seus usos i interpretació.

```
[1]: import numpy as np
      from scipy import stats
      import random
```

0.0.1 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.

```
[2]: def array_summary(array):
      if array.ndim > 1:
          print('Error, array de més d\' una dimensió')
      else:
          mean = np.mean(array)
          median = np.median(array)
          mode = stats.mode(array)[0][0]
          std = np.std(array)
          var = np.var(array)
          skew = stats.skew(array)

          text = 'Resum estadístic del array:\nMitjana: ' + str(mean) + '\n'
          ↪ '\nMediana: ' + str(median) + '\nModa: ' + str(mode) + '\nDesviació típica: '
          ↪ str(std) + '\nVariança: ' + str(var) + '\nAsimetria: ' + str(skew)
```

```
print(text)
```

```
[3]: array = np.array([2, 4, 5, 6, 2, 1, 9])
     array_summary(array)
```

Resum estadístic del array:
Mitjana: 4.142857142857143
Mediana: 4.0
Moda:2
Desviació típica:2.5872528966106905
Variança: 6.6938775510204085
Asimetria: 0.5848157341219765

0.0.2 Exercici 2

Crea una funció que et generi un quadrat $N \times N$ de nombres aleatoris entre el 0 i el 100.

```
[4]: def random_array(num):
     arr = np.zeros((num, num))
     print(arr)
     for i in range(0, num):
         row = []
         for j in range(0, num):
             row.append(random.randint(0, 100))
             print(row)
         np.append(arr, np.array(row))
     print(arr)
```

```
[5]: def random_array(num):
     arr = []
     for i in range(0, num):
         row = []
         for j in range(0, num):
             row.append(random.randint(0, 100))
         arr.append(row)
     arr = np.array(arr)
     return arr
```

```
[6]: print(random_array(15))
```

```
[[ 64   0  36  66  99  80  65  62  49  15  30  66  77  27  31]
 [ 24  21  42  54  40   5  13  67  98   7   0  76  57  40  82]
 [   1  14  56  64  81  99  90  15  66  70  80  68  27  37  78]
 [ 88  37  98  70  84  36   9  42  25  35  66  66  90  34  30]
 [ 26  34  83   5  79  99   1  40  85  53  78  23  52  53  59]
 [ 26  74  58  18  75  51  51  97  11   6  41  60  62  40  66]
 [ 33   0  62  72  63  34  32  36  40  46  61  24  94  60  17]
 [ 94  80  34  14  21  77  10  84  93   7  95 100  94  81  94]
```

```
[ 99  19  21  38  20  75  64  79  59  98  34  83  48  84  42]
[ 13  49  54   2  20   5  57  30  13  58  42   5  32   3  58]
[ 92  20  45  90  51  73 100  94  82  70   8  21  29  73  49]
[   3  79  88   6  64   8  57  57  92  38  41  96  73  89  49]
[ 59  94  27  10   6  77  10  60   2  95  84  91  98   5  47]
[ 43  78  87  22  38  45  87  14  11  35  42  90  24  26  79]
[   2  23  55  53  71  23  13  93   6  35  60  95  37  88  51]]
```

```
[7]: random.randint(0, 100)
```

```
[7]: 17
```

0.0.3 Exercici 3

Crea una funció que donada una taula de dues dimensions ($N \times M$), et calculi els totals per fila i els totals per columna.

```
[8]: def random_array_2(rows, cols):
    arr = []
    for i in range(0, rows):
        row = []
        for j in range(0, cols):
            row.append(random.randint(0, 100))
        arr.append(row)
    arr = np.array(arr)
    return arr

def sums(arr):
    col_sum = np.sum(arr, axis = 0)
    row_sum = np.sum(arr, axis = 1)

    print('Suma de les columnes')
    print(col_sum)
    print('Suma de les files')
    print(row_sum)
```

```
[9]: sums(random_array_2(2, 3))
```

```
Suma de les columnes
[ 97 104  80]
Suma de les files
[129 152]
```

0.0.4 Exercici 4

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

```
[12]: # Format funció

def manual_corr(arr):

    #Calculem la mitjana de cada columna:

    x_mean = 0
    y_mean = 0

    for i in range (0, len(arr[0])):
        x_mean += arr[0][i]

    x_mean = x_mean / len(arr[0])

    for i in range (0, len(arr[0])):
        y_mean += arr[1][i]

    y_mean = y_mean / len(arr[1])

    # Calculem la covariança. Primer els operadors del numerador:

    oper_x = 0
    oper_y = 0
    cov_xy = 0

    for i in range (0, len(arr[0])):
        cov_xy += (arr[0][i] - x_mean) * (arr[1][i] - y_mean) / len(arr[0])

    #Calculem la desviació estandar

    num_x = 0
    num_y = 0

    for i in range (0, len(arr[0])):
        num_x += (arr[0][i] - x_mean)**2

    for i in range (0, len(arr[1])):
        num_y += (arr[1][i] - y_mean)**2

    desv_x = np.sqrt(num_x/len(arr[0]))

    desv_y = np.sqrt(num_y/len(arr[1]))

    # Calculem la correlació

    corr = cov_xy / (desv_x * desv_y)
```

```
print(corr)
return corr
```

```
[13]: # Proves càlcul manual correlació

arr_1 = random_array_2(2, 10)

arr_2 = np.array([[7, 14, 32, 52, 85, 91, 105, 121], [1, 2, 3, 3, 5, 5, 6, 6]])
arr_3 = np.array([[7, 14, 32, 52, 85, 91, 105, 121], [5, 5, 4, 3, 3, 2, 1, 0]])

print('Test random. Les variables, al prendre valors aleatoris, no mostraran_
↳ correlació')
print('Correlació manual')
manual_corr(arr_1)
print('Correlació amb la funció de Numpy')
print(np.corrcoef(arr_1[0], arr_1[1]))

print('Test correlació positiva. Les dades de les variables estan_
↳ correlacionades de tal forma que al creixer una, també ho fa l\'altra')
manual_corr(arr_2)
print(np.corrcoef(arr_2[0], arr_2[1]))

print('Test correlació negativa. Les dades de les variables estan_
↳ correlacionades de tal forma que al creixer una, l\'altra decreix')
manual_corr(arr_3)
print(np.corrcoef(arr_3[0], arr_3[1]))
```

```
Test random. Les variables, al prendre valors aleatoris, no mostraran correlació
Correlació manual
-0.3033088837789841
Correlació amb la funció de Numpy
[[ 1.          -0.30330888]
 [-0.30330888  1.          ]]
Test correlació positiva. Les dades de les variables estan correlacionades de
tal forma que al creixer una, també ho fa l'altra
0.9813888179844993
[[1.          0.98138882]
 [0.98138882  1.          ]]
Test correlació negativa. Les dades de les variables estan correlacionades de
tal forma que al creixer una, l'altra decreix
-0.9652006490640259
[[ 1.          -0.96520065]
 [-0.96520065  1.          ]]
```