

S03T04

November 9, 2021

1 Nivell 1

1.1 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.

```
[ ]: import numpy as np
import numpy.random as random

[ ]: def summary( myarray ):
    if myarray.ndim == 1:
        print(f"Valor minim: {myarray.min()}")
        print(f"Valor maxim: {myarray.max()}")
        print(f"Mitja: {round(np.mean(myarray),2)}")
        print(f"Mitjana: {round(np.median(myarray),2)}")
        print(f"Variància: {round(np.var(myarray),2)}")
        print(f"Desviació standard: {round(myarray.std(), 2)}")
        print(f"Coeficient de correlacio: {round(np.corrcoef(myarray), 2)}")
    else:
        print(f"La dimensio del array es {myarray.ndim} i ha de ser 1")

myarray = random.randint(99, size=(9))
print("-----")
summary(myarray)
print("-----")
summary(myarray.reshape(3,3))
```

```
-----
Valor minim: 15
Valor maxim: 87
Mitja: 46.89
Mitjana: 45.0
Variància: 658.99
Desviació standard: 25.67
Coeficient de correlacio: 1.0
-----
La dimensio del array es 2 i ha de ser 1
```

1.2 Exercici 2

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

```
[ ]: def random_matrix2d(n):  
    return np.random.randint(0,100,size=(n,n))  
print(random_matrix2d(3))
```

```
[[23 75 49]  
 [77  5 23]  
 [ 5 76  8]]
```

1.3 Exercici 3

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

```
[ ]: def sumrowsumcol(mymatrix):  
    if mymatrix.ndim == 2:  
        return mymatrix.sum(axis=1),mymatrix.sum(axis=0)  
    else:  
        return None  
  
mymatrix=random_matrix2d(3)  
print(mymatrix)  
row,col=sumrowsumcol(mymatrix)  
print(f"suma filas: {row}\nsuma columnes: {col}")
```

```
[[75 70  6]  
 [31 80 84]  
 [16 12 81]]  
suma filas: [151 195 109]  
suma columnes: [122 162 171]
```

1.4 Exercici 4

Extret de [Wikipedia](#) Implementa manualment una funció que calculi el coeficient de correlació. Informa't-en sobre els seus usos i interpretació.

El coeficient de correlació de Pearson mesura la relació lineal entre dues variables quantitatives i és independent de l'escala de mesura de les variables. El coeficient de correlació entre dues variables aleatòries X i Y és el quocient de la seva covariància pel producte de les seves desviacions estàndard:

$$r = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2} \sqrt{\sum_{i=1}^n (y_i - \bar{y})^2}}$$

El valor de l'índex de correlació varia en l'interval [-1,+1]:

- Si $r = 1$, hi ha una correlació positiva perfecta. L'índex indica una dependència total entre les dues variables anomenada relació directa : quan una d'elles augmenta, l'altra també ho fa en proporció constant.

- Si $0 < r < 1$, hi ha una correlació positiva.
- Si $r = 0$, no existeix relació lineal. Però això no necessàriament implica que les variables són independents: poden existir encara relacions no lineals entre les dues variables.
- Si $-1 < r < 0$, hi ha una correlació negativa.
- Si $r = -1$, hi ha una correlació negativa perfecta. L'índex indica una dependència total entre les dues variables anomenada relació inversa : quan una d'elles augmenta, l'altra disminueix en proporció constant.

```
[ ]: def mycorrcoef(x, y):
    media_x = np.sum(x)/len(x)
    media_y = np.sum(y)/len(x)
    #coeficien correlacio
    r = (np.sum((x-media_x) * (y-media_y))) / (np.sqrt((np.sum((x -
    ↪ media_x)**2))) * np.sqrt((np.sum((y - media_y)**2))))
    return r

x = np.array([1, 2, 3, 4, 5, 6, 7, 8, 9])
y = np.array([1, 2, 3, 4, 5, 6, 7, 8, 9])

print(mycorrcoef(x,y))

x = np.array([-1, -2, -3, -4, -5, -6, -7, -8, -9])
y = np.array([1, 2, 3, 4, 5, 6, 7, 8, 9])

print(mycorrcoef(x,y))

x = np.array([-1, -2, -3, -4, -5, -6, -7, -8, -9])
y = np.array([-1, -2, -3, -4, -5, -6, -7, -8, -9])

print(mycorrcoef(x,y))

y = np.array([1, 2, 3, 4, 5, 26, 457, 788, 49])

print(mycorrcoef(x,y))
```

```
0.9999999999999999
-0.9999999999999999
0.9999999999999999
-0.5638336031319795
```

Los resultados del coeficiente de correlacion encajan con los previsto, 1 cuando son iguales los dos vectores, -1 cuando son inversos, con un vector con una correlacion del 50% aproximadamente tenemos 0.56 asi creo que funciona correctamente