

DSP S03 T04 Programació_Numerica

May 31, 2021

1 Pràctica amb programació numèrica

Familiaritza't amb la Programació numèrica a través de la llibreria NumP

1.1 Nivell 1

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 dimensio, ha de mostrar un missatge d'error.

```
[76]: import numpy as np
```

Preparem un array de dimensió major que 1.

```
[77]: r=np.array([[1,2,3],[4,5,6],[11,11,11]],[[7,8,9],[11,12,13],[22,22,22]]])
print(r)
print(r.ndim)
```

```
[[[ 1  2  3]
 [ 4  5  6]
 [11 11 11]]
```

```
[[ 7  8  9]
 [11 12 13]
 [22 22 22]]]
```

3

Definim la funció...

```
[78]: def resum(a):
    if a.ndim!=1:
        print("L'array no és de dimensió 1!!")
    else:
        print(a)
        print("La longitud de l'array es "+str(len(a)))
        print("El número d'elements de l'array "+str(a.size))
        print("El typus dels elements és "+str(a.dtype))
        print("La shape de l'array es "+str(a.shape))
        print("La suma dels elements de l'array es "+str(a.sum()))
```

```

print("El valor màxim de l'array és "+str(a.max()))
print("El valor mínim de l'array és "+str(a.min()))
print("La mitja :"+str(a.mean()))
print("La mediana :"+str(np.median(a)))

```

```
[79]: resum(r)
```

L'array no és de dimensió 1!!

```
[80]: b=np.array([4,5,6,5,7,9,8,10,5,4,6,7,6,8,7,9,8])
```

```
[75]: resum(b)
b.sort()
print(b)
```

```

[ 4  5  6  5  7  9  8 10  5  4  6  7  6  8  7  9  8]
La longitud de l'array es 17
El número d'elements de l'array 17
El typus dels elements és int32
La shape de l'array es (17,)
La suma dels elements de l'array es 114
El valor màxim de l'array és 10
El valor mínim de l'array és 4
La mitja :6.705882352941177
La mediana :7.0
[ 4  4  5  5  5  6  6  6  7  7  7  8  8  8  9  9 10]

```

1.2 Exercici 2

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

```
[105]: from numpy import random
def quadrat(n):
    #n=int(input("Dimensió del quadrat NxN..."))
    x = random.randint(100,size=(n,n))
    return(x)
r=quadrat(3)
```

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.

```
[119]: def SumaColFil(n):
    r=quadrat(n)
    SumaCol=(r.sum(axis=0))
    SumaFil=(r.sum(axis=1))
    print(r)
    print("L'array amb la suma de les columnes és "+str(SumaCol))
```

```
print("L'array amb la suma de les files és "+str(SumaFil))
```

```
[120]: SumaColFil(2)
```

```
[[ 9 28]
 [85 78]]
L'array amb la suma de les columnes és [ 94 106]
L'array amb la suma de les files és [ 37 163]
```

1.4 Exercici 4

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

```
[150]: # El coeficient de correlacio entre dues variables ens indicara si hi ha una
        ↪ relacio directa entre les variables
Mates=np.array([2,3,4,4,5,6,6,7,7,8,10,10,6,7,5,7,9])
Fisic=np.array([1,3,2,4,4,4,6,4,6,7,9,10,8,6,4,8,7])

def Coeficient(Mates,Fisic):
    Prod=Mates*Fisic
    Mates2=Mates*Mates
    Fisic2=Fisic*Fisic
    #print(Prod)
    #print(Mates2)
    #print(Fisic2)
    #print("Media Mates :"+str(Mates.mean()))
    #print("Media Fisica:"+str(Fisic.mean()))
    #print("La covarianza es:")
    covarianza=(Prod.mean()-Mates.mean()*Fisic.mean())
    #print(covarianza)
    #print("La desviacion tipica de Mates es :"+str(math.sqrt(Mates2-pow(Mates.
    ↪ mean(),2))))
    #print("La desviacion tipica de Mates es :")
    dt1=(np.sqrt(Mates2.mean()-pow(Mates.mean(),2)))
    #print(dt1)
    #print("La desviacion tipica de Fisica es :")
    dt2=(np.sqrt(Fisic2.mean()-pow(Fisic.mean(),2)))
    #print(dt2)
    #print("El coeficiente de correlacion es:")
    coef=covarianza/(dt1*dt2)
    return(coef)

b=Coeficient(Mates,Fisic)
print(b)

a=np.corrcoef(Mates,Fisic)
print(a[0][1])
```

0.8780615581542278
0.8780615581542283