

DS d'info

Question 1)

```
select idpatient from medical  
where etat = "hernie discale"
```

```
2) select nom, prenom from patient  
where id =  
( select idpatient from medical  
  where etat = "spondylolisthesis" )
```

```
3) select etat, count(*) df as nbpatients from
```

```
select etat, count(*) as nbpatients  
from medical  
group by etat
```

4) La bibliothèque Numpy permet un accès
pas plus facile aux données, ce qui est
utile pour les tableaux de grande taille

5) Dans le tableau, il y aura $N \times 6 = 600\,000$ valeurs

~~Dans le vecteur, il y aura $N \times 3 = 300\,000$ valeurs~~

Dans le vecteur, il y aura $N \times 1 = 100\,000$ valeurs

Le tableau est codé en 32 bits = 4 octets et
le vecteur en 8 bits = 1 octet

Donc la quantité de mémoire totale est

$$\text{de } 4 \times N \times 6 + 1 \times N = 2,5 \text{ Mo}$$

6) ~~def Separation-Par Group (data, etat):~~

~~S = [[], [], []]~~

~~for h in range(len(data)):~~

~~if etat[h] == 0:~~

~~def Separation-Par~~

def Separation-Par Group (data, etat):

S = [[], [], []]

for i in range(len(data)):

S[etat[i]] . append (data[i])

return S

7) D'après le script, le tableau est de dimension $n \times n$

De plus les boucles commencent à 0 donc

$$\text{ARGS } 1 = (n, n, (i+1) * (j+1))$$

Test = $i! \neq j!$ car les diagrammes de la diagonale sont différents des autres

8) Les diagrammes de la diagonale représentent les loi de répartition et par les diagrammes n -diagonaux représentent les liens entre les caractéristiques

```

10) def min_max(x):
    min, max = x[0], x[0]
    for k in range(len(x)):
        if x[k] < min:
            min = x[k]
        if x[k] > max:
            max = x[k]
    return min, max

```

```

11) def distance(y, data)
    tab = []
    for k in range(N):
        for i in range(m):
            a =
            a += sqrt((y[i] - data[k][i])**2)
            tab.append(a)
    return a

```

12) La partie 1 range les distances
par ordre croissant dans la liste T
avec leur élément

La
nb ~~partie 2~~ transforme une liste de
en une liste d'entiers ou pour

Questio = 15

```
def moyenne(x):  
    n = len(x)  
    somme = 0  
    for k in range(n):  
        somme = somme + x[k]  
    return somme / n
```

```
def variance(x):  
    s = 0  
    for k in x:  
        s = s + k**2  
    return s / len(x) - moyenne(x)**2
```

*tab = [[0,0,0]]

16/

def synthese (data, etat) :

*for k in range (len (data)):

for i in range (2)

if etat[k] == i:

tab[i][k] =

[maximize[k], variance[A]]

return tab