

DS d'informatique

Q1: SELECT idpatient from MEDICAL WHERE  
etat = "hernie discale"

Q2: SELECT nom, prenom from PATIENT WHERE

id = (SELECT idpatient FROM MEDICAL WHERE  
etat = "spondylolisthésis")

Joëlle

Q3:

~~etat~~

~~(\*) as~~

SELECT etat, count(\*) as patients par etat FROM MEDICAL  
GROUP BY etat. ✓

Q4: Dans le cas d'un tableau de grande taille, numpy  
facilite l'accès aux données et le rend plus rapide  
d'accès. ✓

Q5:

du tableau data

Il y a 6 attributs donc le nombre de données total est:

$$N \times 6 = 600\ 000$$

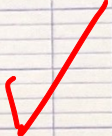
Le 1<sup>er</sup> tableau étant codé en 32 bits = 4 octets et le 2<sup>e</sup> en octets,  
le nombre total d'octets nécessaire est:

$$N \times 6 \times 4 + N \times 1 \times 1 = 100\ 000 \times (6 \times 4 + 1) \\ = 2,5\ Mo$$



Q6

```
def separation par groupe (data, etat):  
    etat 0 = []  
    etat 1 = []  
    etat 2 = []  
    lendata = len (data)  
    for k in range (lendata):  
        if etat [k] == 0:  
            etat 0.append (data [k])  
        elif etat [k] == 1:  
            etat 1.append (data [k])  
        elif etat [k] == 2:  
            etat 2.append (data [k])  
    return etat 0, etat 1, etat 2
```



Q7.

Le tableau est de dimensions n x n (n lignes, n colonnes) et, en corrigeant le défaut à 0 de la fonction range()

ARGS 1 = (n, n, (~~i+1~~ \* (j+1)))

Les diagrammes de la diagonale sont différents des autres,  
le TEST permet ainsi de choisir les diagrammes  
hors diagonale : TEST  $i \neq j$

Q10

```
def min_max (X)  
    min, max = X[0], X[0]  
    for x in X:  
        if x > max:  
            max = x  
        if x < min:  
            min = x  
    return min, max
```



Q12

- la partie 1 crée une liste contenant toutes les distances entre le  $n$ -uplet  $z$  et chaque  $n$ -uplet  $x$ .  
Elle  $H_i$  ensuite cette liste.

- la partie 2 transforme une liste de nb 0 en une liste contenant un 1 pour les coefficients dont les indices sont égaux aux indices  $i$  des  $K$  premiers couples de la liste  $T$  envoyée par la partie 1.  
il ne faut pas se contenter de lire l'algo.

Q15:

```
def moyenne (X):  
    S = 0  
    for x in X:  
        S += x  
    return S/len(X)
```

```
def variance (X):  
    a = 0  
    n = len(X)  
    for i in range (n):  
        a = a + X[i]**2  
    return a/n - moyenne(X)**2
```