

Info DS9

1. `SELECT idpatient FROM MEDICAL
WHERE etat = 'hernie discale';`
2. `SELECT nom, prenom FROM PATIENT
JOIN MEDICAL ON PATIENT.id = MEDICAL.idpatient
WHERE etat = 'spondylolisthésis';`
3. `SELECT etat, COUNT(idpatient) AS nbpatients
FROM MEDICAL
GROUP BY etat;`

5. Dans le tableau il y a $6 \times 100\,000 = n \times N$
 $= 600\,000$ réels codés sur 32 bits = 4 octets
Pour stocker il faut une mémoire de
 $600\,000 \times 4 = 2\,400\,000$ octets c'est-à-dire
2,4 Mo.

Pour le vecteur il y a N valeurs codées sur
8 bits = 1 octet. Donc il faut une mémoire
de $100\,000$ octets = 0,1 Mo pour stocker le
vecteur.

Donc en tout cela fait 2,5 Mo.

6. `def separationParGroupe(data, etat):`

`L = []`

`M = []`

`P = []`

`for i in range(0, N+1):`

`if etat[i] == 0:`

```

L.append(data[i,:])
elif etat[i] == 1 :
M.append(data[i,:])
else
P.append(data[i,:])

```

```

return [L, M, P].

```

7. ARG S1 = (n, n, i x j)
 ARG S2 = (groupe[i], groupe[j], marker = mark(R))
 ARG S3 = (data[i])
 TEST = i != j

9.
$$x_{normj} = \frac{x_j - \min(X)}{\max(X) - \min(X)}$$

On retrouve bien 0 si $x_j = \min(X)$
 et 1 si $x_j = \max(X)$.

10. def min-max(X):
 min = X[0]
 max = X[0]
 for i in range(1, len(X)+1):
 if X[i] < min:
 min = X[i]
 elif X[i] > max:
 max = X[i]
 return min, max


```

11. def distance(z, data):
    d = 0
    Q = []
    for k in range(N+1):
        for i in range(len(z)):
            d += (z[i] - data[k, i])**2
        Q.append(sqrt(d))
    return Q.

```

12. La partie 1 crée une liste T de listes comprenant 2 éléments.
 Cette liste est après triée.

T représente une liste.
 dist est aussi une liste.
 select aussi une liste
 et ind est un entier

```

15 def moyenne(x):
    S = 0
    for i in range(0, len(x)):
        S = S + x[i]
    return S / len(x)

```

```

def variance(x):
    S = 0
    n = len(x)
    for i in range(0, len(x)):
        S = S + (x[i] - moyenne(x))**2
    Return S / n

```