

DS Informatique :

Q₁: `SELECT idpatient from MEDICAL where etat = "hernie discale"`

Q₂: `SELECT nom, prénom from PATIENT join MEDICAL on MEDICAL.idpatient = PATIENT.id where MEDICAL.etat = 'spondylolisthésis'`

Q₃: `Select etat, Count(etat) from MEDICAL group by etat.`

Q₄: intérêt d'utilisation de la bibliothèque Numpy :

Quand les tableaux sont de grandes tailles, cette bibliothèque optimise l'accès à l'intégralité des données proposées dans le tableau.

Q₅: Quantité de Mémoire :

$N = 100\,000$ lignes et $n = 6$ (colonnes)

Pour le tableau data on a : $N \times n = 600\,000$ données.

Pour le tableau etat on a : $N \times n = 100\,000$ données.

(avec $n = 1$ car on a 1 seule)

data est codé sur 32 bits et états sur 8 bits donc la quantité de mémoire nécessaire est :

$$600\,000 \times 4 + 100\,000 \times 1 = 2\,500\,000 \text{ octets}$$

soit 2.5 Mo

Q₆: `def separationParGroupe (data, etat) :`

`x = [(), (), ()]`

`y = len(data)`

`for i in range (y) :`

`x [etat[i]], append (data [i])`

`return x`

Q7: —

Q8: les diagrammes de la diagonale : permettent de voir le nombre de patients en fonction d'un attribut et donc si ces valeurs sont centrées autour d'une moyenne.

Quant aux diagrammes hors diagonales, ça représente un attribut en fonction d'un autre différent.

Q9: Expression de $\alpha_{norm,j}$:

$$\alpha_{norm,j} = \frac{\alpha_j - \min(X)}{-\min(X) + \max(X)}$$

Q10: def min_max(X):

i, j = X[0], X[0]

for n in X:

if n < i:

n = i

if n > j:

n = j

return i, j

Q11: def distance(z, data):

d = 0

for i in range(len(z)):

z[i]

Q15: def moyenne (X):
 b = 0
 for n in X:
 b += n
 return b / len(X)

def variance (X):
 b = 0
 m = moyenne (X)
 for a in X:
 b += a ** 2
 return b / len(X) - m ** 2

Q16: def synthese (data, etat):

Q17: def gaussienne (a, moy, v):

Q18: def prediction

Q18: def probabilerGroupe (z, data, etat):