

BARON TANYA MPS12 DS info

1) select ~~the~~ 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) from MEDICAL group by etat

4) La bibliothèque Numpy effectue plus rapidement les calculs matriciels elle est plus efficace que les listes python

5) On a $100\,000 \times 6 = 600\,000$

d'où $600\,000 \times 32 = 19\,200\,000$ bits

or 1 octet = 8 bits et $1\text{ Mo} = 1 \times 10^6$ octets

d'où stockage nécessaire est de 2,4 Mo

Puis pour le vecteur: $1 \times 100\,000 = 100\,000$ bits
soit 0,1 Mo

Donc au total: 2,5 Mo. nécessaire

6) def repartitionParGroupe(data, etat):
 groupes = [[] for _ in range(3)]
 for i in range(len(data)):
 groupes[etat[i]].append(data[i])
 return groupes

7)

8) Hors diagonales: cela permet de repérer plus rapidement le nombre de patients atteints des mêmes maladies.

Diagonales: —

9) On a

$$x_{\text{norm},j} = \frac{x_j - \min(x)}{\max(x) - \min(x)}$$

10) `import numpy as np`
`def min_max(X):`
 `X = array(X)`
 `m = np.min(X)`
 `n = np.max(X)`
 `print(m)`
 `print(n)`

11) `def distance(z, data):`

12) La partie 1 trie la liste T
 La partie 2

15) `def moyenne(x):`
 `n = len(x)`
 `s = 0`
 `for i in range(n):`
 `s = s + x[i]`
 `return s/n`

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

20) L'échelle logarithmique est souvent utilisée car elle permet de mieux représenter les données, d'où l'utilisation du logarithme