

PEYRON Romane MPSi1

- Q1. `SELECT idpatient FROM MEDICAL WHERE etat = hernie discale;`
- Q2. `SELECT nom, prenom FROM PATIENT WHERE idpatient = (SELECT idpatient FROM MEDICAL WHERE etat = spondylolisthésis);`
- Q3. `SELECT etat, COUNT(*) FROM MEDICAL WHERE etat = normal OR etat = spondylolisthésis OR etat = hernie discale;`
- Q4. Cette bibliothèque contient de nombreux outils mathématiques pour manipuler les tableaux numpy, de plus elle permet d'optimiser la mémoire mise en jeu.
- Q5. Quantité de mémoire = $32 \times 6 \times N$ bits
$$= \frac{32 \times 6 \times N}{8000000} \text{ Mo}$$
$$= 2,4 \text{ Mo}$$
- Q6. `def separationParGroupe(data, etat):` import numpy as np
 `etat_0 = np.array()`
 `etat_1 = np.array()`
 `etat_2 = np.array()`
 for i in range(N):
 if etat[i] == 0:
 `etat_0.append(data[i, j] for j in range(6))`
 else if etat[i] == 1:
 `etat_1.append(data[i, j] for j in range(6))`
 else if etat[i] == 2:
 `etat_2.append(data[i, j] for j in range(6))`
 return [etat_0, etat_1, etat_2]

Q8: Les diagrammes en diagonale présentent le nb de patients qui, dans un certain état ont tel ou tel caractéristique ou un attribut: par exemple on repère l'incidence de l'arthrose la plus répandue chez les patients dans l'état i (0, 1 ou 2). Chaque diagonale correspond à un attribut différent.

Les diagrammes en le côté permettent de repérer les différences de profil par les états 0, 1 ou 2; si on voit un gros nuage de ronds, bien distinct d'un nuage de 'x' (hernie discale), on peut imaginer que la hernie discale influe (ou est influencée) sur resp. par) ces paramètres.

Q7. TEST correspond à $i \neq j$

Q9.
$$\alpha_{\text{norm},j} = \frac{|x_j - \min(X)|}{|\min(X) - \max(X)|}$$

Q10. Complexité linéaire: on balaye le vecteur X :

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


Q11: def distance(γ , data):

Somme = 0

$L = \text{len}(\text{data}[1,:])$

Liste = []

for i in range(N):

for j in range(L):

Somme += abs($\gamma[j] - \text{data}[i,j]$)

Liste.append(Somme)

return Liste

Q12: Partie 1: T contient ^{dans des sous-listes de 2 éléments} toutes les distances à γ , et l'identifiant du patient (numéro de la ligne dans data, et len(data) ici). kn(T) permet juste de trier la liste T (sinon par ordre croissant de distance à γ).

Partie 2: sélectionner les ^{identifiants de} K plus proches voisins de γ (les K premières listes de 2 éléments de la liste triée T) en remplaçant γ comme le plus proche de lui-même.

leur numéro de ligne dans le tableau data

select contient les "identifiants" des K patients les plus proches de γ , triés des plus proches aux moins proches.