DS informatique

1) Select idpatient from MEDICAL where etat = 'hernie discale'

2) Select nom, prenom from PATIENT join MEDICAL on
MEDICAL.idpatient = PATIENT.id where MEDICAL.etat = 'spondylodiscite'

3) Select etat, count(etat) from MEDICAL group by etat

4) La bibliothèque numpy permet d'accéder rapidement à toutes les données du tableau, plus rapidement qu'avec de listes.

5) Le nombre de données du tableau, nommé, Avec N = 100 000,
data
N-6 et celui avec les états N

Le nombre d'octets est de N × 6 × 4 + N = 25 . N

= 2 500 000 o = 2,5 Mo

6) def separationParGroupe(data, etat):
a = [[], [], []]
n = len(data)
for k in range(n):
a[etat[k]].append(data[k])
return a

8) Les diagrammes de la diagonale permettent de voir si les valeurs sont réparties autour d'une valeur moyenne:
Les diagrammes hors diagonale permettent de voir s'il y a une corrélation avec un autre facteur.

9) L'expression de $x_{norm\,j}$ est

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

10)
```
def min_max(X):
    a, b = X[0], X[0]
    for i in X:
        if i < a:
            a = i
        if i > b:
            b = i
    return a, b
```

11)
```
def distance(z, data):
    d = 0
    for i in range(len(z)):
        d += (z[i] - data[i]) ** 2
    return sqrt(d)
```

14) Le taux de réussite ne change pas quand le nombre de voisin augmente
L'algorithme est moyennement efficace avec un taux de réussite
d'environ 72%.

15) def moyenne (x):

    a = 0

    for i in x :

        a += i

    return a / len(x)

def variance (x):

    a = 0

    m = moyenne (x)

    for i in x :

        a += i ** 2

    return a / len(X) - m ** 2

16) def synthese (data , etat) :

    X

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

    X

18) def probabiliteGroupe (z , data , etat) :

    X