VIALY
Carla

MPSI 2

DS d'informatique
Cycle 3-4

Q1: select idpatient from MEDICAL where etat = "hernie discale"

Etrange et pas franchement une jointure

Q2: select nom, prenom from PATIENT where id = (select idpatient from MEDICAL where etat = "spondylolisthésis").

Q3: select etat, count(distinct idpatient) from MEDICAL group by etat

Q4: Par Numpy, les tableaux de grandes tailles sont plus efficacement gérés que par les listes Python.

Q5: on a donc un tableau à 100 000 lignes et 6 colonnes. Chaque case contient un réel codé sur 32 bits, donc 4 octets.
Ainsi il faudra $4 \times 100\,000 \times 6 = 2\,400\,000$ octets
soit 2,4 Mo

Q6:
```
def separationParGroupe(data, etat):
    groupes = [[] for k in range(3)]
    for i in range(len(data)):
        groupe[etat[i]].append(data[i])
    return groupes.
```

Q7 : ARGS1 = (n,0 , (i+1) , (j+1))

ARGS2 = (groupe[i], groupe[j], marker = mark[k])

TEST = i != j.

ARGS3 = data[i].

Doit dépendre de k

Q8 : Les diagrammes de la diagonale permettent de visualiser la répartition des nombre de patient en fonction de l'évolution d'un attribut.

Les diagrammes hors de la diagonale mettent en lien les attributs entre eux et introduisent également l'état du patient.

Q9 : $x_{norm_j} = \frac{x_j - min(X)}{max(X) - min(X)}$

Q10 :

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

Une seule boucle suffit.

Qu'en est-il de la complexité ?

Q11:
```
def distance(z, data):
    d=0
    for i in range(len(z)):
        d = d+ (z[i]- data[i]) **2
    return sqrt(d)
```

Q12: Partie 1 : création d'un tableau dont les colonnes sont les distances de z avec la colonne i ainsi que le numéro i

Partie 2 :

Partie 1 : création et tri de la liste T

Partie 2 :

. . .

Q13 :