BESSON
Lucas

MPSI 1

DS d'info

Question 1)

```
select idpatient from medical
where etat = "hernie discal"
```

2)
```
select nom, prenom from patient
where id =
(select idpatient from medical
where etat = "spondylolisthesis")
```

Jointure ...

3) ~~select etat, count(*) as nbpatients from~~

```
select etat, count(*) as nbpatients
from medical
group by etat
```

Vu

4) La bibliothèque Numpy permet un accès bien plus facile aux données, ce qui est utile pour les tableaux de grande taille

5) Dans le tableau, il y aura $N \times 6 = 600\,000$ valeurs

~~Dans le vecteur, il y aura $N \times 3 = 300\,000$ valeurs~~

Dans le vecteur, il y aura $N \times 1 = 100\,000$ valeurs

Le tableau est codé en 32 bits = 4 octets et le vecteur en 8 bits = 1 octets

Donc la quantité de mémoire totale est de $4 \times N \times 6 + 1 \times N = 2,5$ Mo

6) def Separation_Par_Group (data, etat) :

```
    S = [[], [], []]
    for k in range (len(data)) :
        if etat[k] == 0 :
```

~~def Separation_Par~~

```
def SeparationPar Group (data, etat):
    S = [[], [], []]
    for i in range (len(data)):
        S[etat[i]].append(data(i))
    return S
```

7) D'après le script, le tableau est de dimension $n \times n$

De plus les boucles commencent à 0 donc

ARGS 1 = (n, n, (i+1) * (j+1))

Test = i != j car les diagrammes de la diagonale sont différents des autres

8) Les diagrammes de la diagonales représentent des loi de répartition et par les diagrammes n-diagonaux représentent les liens entre les caractéristiques

10) def min_max(X):
    min, max = X[0], X[0]
    for k in range(len(X)):
        if X[k] < min:
            min = X[k]
        if X[k] > max:
            max = X[k]
    return min, max

11) def distance(xj, data)
    tab = []
    for k in range(N):
        for i in range(m):
            a += sqrt((xj[i] - data[k][i])**2)
        tab.append(a)
    return a

12) La partie 1 range les distances par ordre croissant dans la liste T avec leur éléments

La partie 2 transforme une liste de nb en une liste d'entiers ou pers

Question 15

```
def moyenne(x) :
    n = len(x)
    somme = 0
    for k in range(n):
        somme = somme + x[k]
    return somme / n
```

```
def variance(x):
    s = 0
    for k in x :
        s = s + k**2
    return s / len(x) - moyenne(x)**2
```

16/ def synthese (data, etat) :

*tab = [[],[],[]]

*for k in range (len (data)) :

for i in range (2)

if etat [k] = i :

tab [i] [k] =

[moyenne [k], variance [k]]

return tab