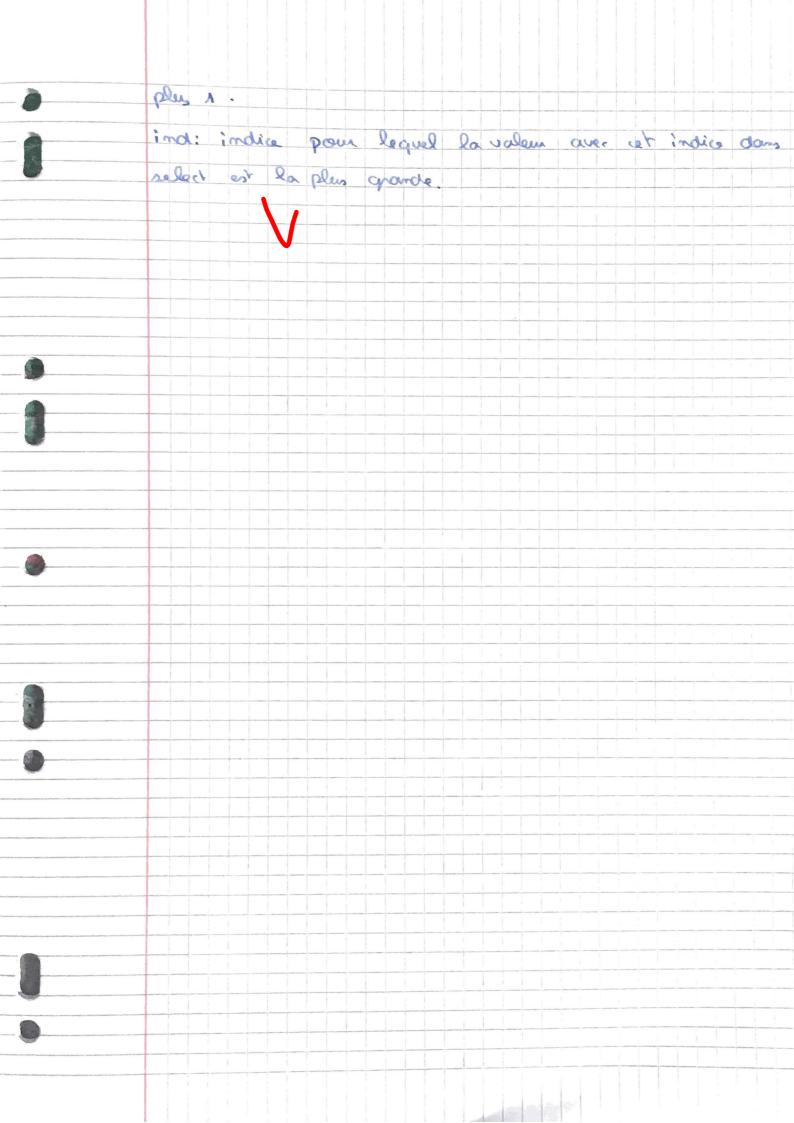


```
6) def reparation Par Groupe (dato, etat):
       N = len (dava.)
       0 = []
       X = C
       E = C 
        for R in range (N):
           if etar CR ] = 0 :
                O. append (dara [R.:])
           elif etat (2]=1:
             X. append (data [k:])
          else :
               E. append (data [ & : ])
        return [O, X E]
7) ARGS 1 = m, m, itm - m + j
ARG 52 = groupes (i), groupes (j), marker = mark [m*i+i]
ARGS3 = groupes (1)
                      Doit dépendre
TEST = i!=i
                          de k
```

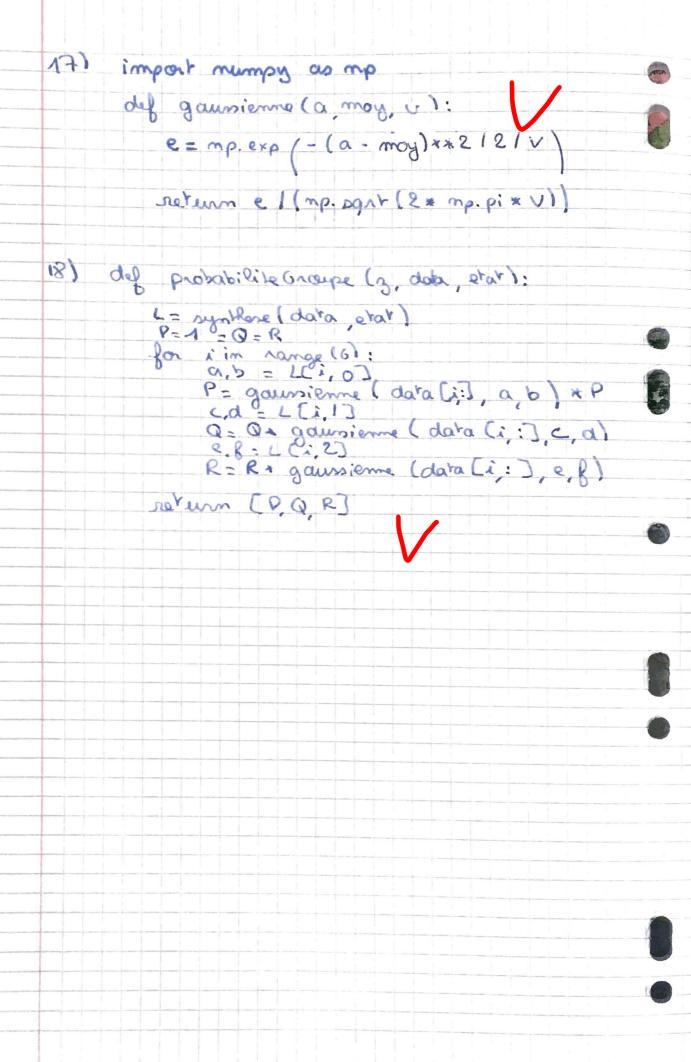
8) Les diagrammes diagrammes permettent d'avoir un récapitulatif du nombre de patients en fonction des valeurs d'un attributs. Bela permet de voir où re situe la majorité des personnes. Les diagrammes mon diagonaire permettent de regarder le lien entre deux attributo. Cela peut permetta d'exprimen la course d'un attribut. g) $x_{monmin} = \frac{x_{0} - min(x)}{mox(x) - min(x)}$ 101 def min marex): return min (x), man(x) Sans les fonctions min et max: def min_man(X): m= len(x) Colx = xcol for i'm range (1, m): if X[i] > max: [i] X = xom if X[i] < min: Complexité? min = XCi] return min max 11) del distance (z, data): min, max = min_max (data [:, 3]) do =

121 Partie 1. On vier une liste Toure la distance entre le muplet à classer et le conner et sa valeur d'étair. Puis on la trè par ordre croissant des distances. Partie ?: On créé une matrice ligne nulle de taille no Ombinajoule 1 et la valeur de état pour tous les enties entre oet k-1. Partie 3: on Jarde la plus grando valeur d'indice pour lequel sa valeur correspondante dans select pour l'indice est le plus grande. T: liste avec la distance entre le replet à danser et calle avec le muplet conner ains, que sa valeur d'état. dist: les distances enclidiennes ente le n-uplet à et chaque nuplet de data select: matrice ligne qui contient la voleur de état TCI ICI)



```
15) del moyenno (x):
          for & in range (lan(x)):
              S=S+x[R]
          return Silenix!
del variance (2):
       12 mayenne (x)
      for & in range (len(x)):
           5=5+2(R)-4xx2
      return SI len(x)
16) def synthese (data, etat):
      for Rin range (6):
for i'm range (len (data)).
if etat (i): 0:
                 Xo append (dara Ci)
            if star C: ] = 1:
                 X1. append (data Ci)
             elif:
                 X2. append (dara(i))
        A= [moyenne (xo), variance (xo)]
        B= [mayenne(X1), variance(X1)]
        C= [moyenne (x2), variance (x2)]
         XO= X1 = X2 = []
        Lappend [A,B,C]
```

return L



191 del prediction (3 data, etal): PO, P1, P2= probabilik Grange (z, data, etat) if P1 > Po : if P2>P1: return 2 else: return 1 ig P2>Po: S murar else: neturn O 80) Les valeurs de probabilité mouvées sont très petites. L'ordinateur peut avoir du mal à les différencier. En prenant le logarithme décimal on se ramère à des voleurs que peut calcula l'ordinateur sans evens. Par exemple, PO = 1,1×10-15 Rog PO == -15 P1=7,4x1016 log P1 = -16