



2ème Année Master Informatique, Semestre 1

Option : Systèmes informatiques Intelligents (SII)

Module : Data Mining

Rapport
Etude Exploratoire d'un dataset

Monôme :

- OUHOCINE Sarah

Professeur :

- M^{me} BABA ALI

SOMMAIRE

Introduction Générale.....	
1 – Etude du benchmark contenu dans le dataset HEART_Stat.txt.....	
1.1 Description des entrées et de chacun des attributs du dataset	
1.2 Manipulation du dataset.....	
1.3 Valeurs manquantes (Missing Value).....	
2 – Etude exploratoire des caractéristiques descriptives des données.....	
2.1 Statistiques	
2.1.1 Moyenne	
2.1.2 Médiane (Q2).....	
2.1.3 Mode	
2.2 Symétrie des données	
2.3 Boite a moustache (diagramme de boite)	
2.3.1 Min	
2.3.2 Q1.....	
2.3.3 Q3	
2.3.4 Max	
2.3.5 Déduction des valeurs aberrantes (Outliers).....	
2.3.5.1 valeurs aberrantes faibles	
2.3.5.2 valeurs aberrantes élevées	
2.3.6 Code et exécution des boites a moustaches	
2.4 Histogramme	
2.5 Diagramme de dispersion (Nuage de points)	
2.5.1 Regroupement	
2.5.2 Coefficient de Corrélation et corrélation	
2.5.3 valeurs aberrantes (Outliers).....	
2.5.4 Code du diagramme de dispersion	
2. 6 Existence des corrélations entre différents attributs.....	
2.7 QQPLOT (Droite de Henry).....	
3 – INTERFACE	
Conclusion Générale.....	

INTRODUCTION GENERALE

Nous sommes dans l'ère du big data où les données sont de plus en plus gigantesques.

Aujourd'hui, dû aux grandes tailles des bases de données actuelles, les données brutes sont généralement de faible qualité. Elles peuvent être **incomplètes** (valeurs manquantes), **bruitées** (valeurs erronées ou aberrantes) ou **incohérentes** (divergence entre attributs), par conséquent l'application des algorithmes de datamining sur de telles données complexifie l'apprentissage et nuit à la performance ainsi qu'à la fiabilité du modèle.

Pour cela, [un prétraitement des données](#) avant d'appliquer toute technique de DataMining est une étape importante, nécessaire et cruciale pour avoir une idée globale sur le comportement de ces données et améliorer leur qualité par la suite.

A partir de ce contexte-là, nous allons mettre en application le prétraitement des données concernant l'extraction des caractéristiques descriptives des données.

Pour ce faire, nous avons divisé notre travail en deux parties :

- 1- Etude du benchmark contenant les données brutes.
- 2- Etude exploratoire des caractéristiques descriptives des données.

1 – Etude du benchmark contenu dans le dataset HEART_Stat.txt

➤ 1.1 Description des entrées et de chacun des attributs du dataset :

Le dataset intitulé : « **HEART_Stat.txt** », c'est le dataset sur lequel nous voudrions effectuer une étude exploratoire. Il s'agit des données relatives à des maladies cardiaques observées chez une population de 270 personnes, extraites à partir des diagnostics de ces derniers, des images médicales, des analyses.... Chaque instance (**dossier d'un patient**) est décrite par 13 attributs (inputs), en plus de classe de l'instance (l'output) qui est en fait la variable à prédire :

- **absence de la maladie cardiaque** (1).

Ou - **présence de la maladie cardiaque** (2).

Ces attributs sont de type différents (Réel, Ordonné, Binaire, Nominale), et ils sont complets i.e. (le dataset ne contient pas des valeurs manquantes ou des champs vides).

Les attributs sont les suivants :

Numéro de l'attribut	Nom de l'attribut dans le dataset (en anglais)	Traduction et Description de l'attribut (en français)	Type de l'attribut
1	Age	L'Age du patient.	Réel
2	Sex	Le Sexe du patient, cet attribut peut prendre 2 valeurs possibles : 0 si c'est masculin ou 1 si c'est féminin ou vice-versa.	Binaire
3	Chest pain type	Le Type de douleur de poitrine ou « douleur thoracique » c'est une gêne , généralement à l'avant de la poitrine cet attribut peut prendre 4 valeurs possibles (1, 2, 3,4).	Nominale
4	Resting blood pressure	Tension artérielle au repos ou « tension artérielle » : c'est la pression du sang en circulation sur les parois des vaisseaux sanguins au repos, cet attribut varient entre 94 et 200.	Réel
5	Serum cholestoral	Sérum cholestoral ou « cholestérol », terme souvent associé aux maladies cardiaques : c'est un lipide qui constitue la matière grasse des êtres vivants, il se mesure en mg /dl, les valeurs de cet attribut varient entre 126 et 564.	Réel
6	Fasting blood sugar	Glycémie à jeun équivalent à dire le taux de glucose dans le sang il se mesure en mg /dl, cet attribut peut prendre 2 valeurs possibles : 0 si supérieur strictement à 120 mg/dl, 1 sinon. Ou vice-versa	Binaire
7	Resting electrocardiographic results	Résultats électrocardiographiques au repos , c'est le résultat du processus de production d'un électrocardiogramme (ECG/EKG), un enregistrement, un graphique de tension de l'activité électrique du cœur au repos, l'aide d'électrodes placées sur la peau. Cet attribut peut prendre 3 valeurs possibles (0, 1, 2).	Nominale
8	Maximum heart rate achieved	Fréquence cardiaque maximale atteinte : équivalent à dire la vitesse maximale du rythme cardiaque mesurée par le nombre de battements du cœur par minutes (bpm), les valeurs de cet attribut varient entre 71 et 202.	Réel
9	Exercise induced angina	Angine de poitrine induite par l'exercice ou la douleur thoracique ou la pression, généralement en raison de pas assez de flux sanguin vers le muscle cardiaque, cet attribut peut prendre 2 valeurs possible (0 ou 1)	Binaire
10	Oldpeak	ST dépression induite par l'exercice par rapport au repos : est la conclusion sur l'électrocardiogramme (au repos) dans laquelle la trace dans le segment ST est anormalement basse au-dessous de la ligne basse, les valeurs de cet attribut varient entre	Réel
11	The slope of the peak exercise ST segment	La pente du segment ST d'exercice de pointe	Ordonné
12	Number of major vessels	Nombre de vaisseaux sanguins principaux cad nombre de conduits qui transportent le sang dans l'organisme, ils sont colorés par la fluorescenc, les valeurs de cet attribut varient entre 0 et 3	Réel
13	Thal	Thalassémie ou « Troubles sanguins héréditaires », cet attribut peut prendre 3 valeurs possible (3 pour « Normal », 6 pour « défaut corrigé », et 7 pour « défaut réversible »).	Nominale

➤ 1.2 Manipulation du dataset :

Nous allons utiliser un script python  pour extraire à partir du dataset « **HEART_Stat.txt** » juste les informations nécessaires i.e. (la partie @data) :

```
@data
70,1,4,130,322,0,2,109,0,2.4,2,3,3,present
67,0,3,115,564,0,2,160,0,1.6,2,0,7,absent
57,1,2,124,261,0,0,141,0,0.3,1,0,7,present
64,1,4,128,263,0,0,105,1,0.2,2,1,7,absent
74,0,2,120,269,0,2,121,1,0.2,1,1,3,absent
65,1,4,120,177,0,0,140,0,0.4,1,0,7,absent
56,1,3,130,256,1,2,142,1,0.6,2,1,6,present
59,1,4,110,239,0,2,142,1,1.2,2,1,7,present
60,1,4,140,293,0,2,170,0,1.2,2,2,7,present
63,0,4,150,407,0,2,154,0,4,2,3,7,present
59,1,4,135,234,0,0,161,0,0.5,2,0,7,absent
53,1,4,142,226,0,2,111,1,0,1,0,7,absent
44,1,3,140,235,0,2,180,0,0,1,0,3,absent
61,1,1,134,234,0,0,145,0,2.6,2,2,3,present
57,0,4,128,303,0,2,159,0,0,1,1,3,absent
71,0,4,112,149,0,0,125,0,1.6,2,0,3,absent
46,1,4,140,311,0,0,120,1,1.8,2,2,7,present
53,1,4,140,203,1,2,155,1,3.1,3,0,7,present
64,1,1,110,211,0,2,144,1,1.8,2,0,3,absent
40,1,1,140,199,0,0,178,1,1.4,1,0,7,absent
67,1,4,120,229,0,2,129,1,2.6,2,2,7,present
48,1,2,130,245,0,2,180,0,0.2,2,0,3,absent
43,1,4,115,303,0,0,181,0,1.2,2,0,3,absent
```

Pour ce faire, il faut d'abord **ouvrir** le fichier source « **HEART_Stat.txt** », puis le **lire** en ignorant son contenu jusqu'à arriver à la ligne qui suit "@data" qui nous intéresse, ensuite à partir de cette ligne là on commence à **écrire** (copier) les données dans un autre fichier texte (le fichier destination) disons « **HEART_Stat_Data.txt** » et enfin on **affiche** le contenu de ce dernier et on ferme les deux fichiers ouverts.

Ce qui nous donne les premières lignes de notre script :

- **OUVERTURE :**

```
15 #Ouverture du fichier source HEART_Stat.txt
16 HEART_Stat=open("HEART_Stat.txt","r")
```

- **LECTURE / ECRITURE :**

```
18 #Lecture du fichier source HEART_Stat.txt
19 HEART_Stat=HEART_Stat.read()
20 for ligne in HEART_Stat :
21     if (ligne.startswith ("%")) or (ligne.startswith ("@")) :
22         {#ne rien faire; passer a la ligne suivante
23         }
24     else:
25         #Ecriture des données dans Le fichier destination HEART_Stat_Data.txt
26         HEART_Stat_Data = open("HEART_Stat_Data.txt","w")
27         HEART_Stat_Data.write(ligne)
```

- AFFICHAGE :

```
29 #Affichage du fichier destination
30 HEART_Stat_Data = open("HEART_Stat_Data.txt", "r").read()
31 print(HEART_Stat_Data)
```

- FERMETURE :

```
33 #Fermeture des deux fichier source et destination
34 HEART_Stat.close()
35 HEART_Stat_Data.close()
```

- RECUPERATION DES DONNEES :

Désormais, nous allons travailler avec le fichier « **HEART_Stat_Data.txt** » contenant les données brutes en exploitant les bibliothèques **csv**, **numpy**, **pandas** afin de pouvoir facilement récupérer :

- Les lignes (les dossiers des patients).

```
63 def ReadEntree(numRow):
64     # "this allow to print a row "
65     with open('HEART_Stat_Data.txt') as csv_file:
66         csv_reader = csv.reader(csv_file, delimiter=',')
67         line_count = 0
68         #a=[]
69         for row in csv_reader:
70             if(line_count == numRow):
71                 z = len(row) - 1
72                 b=[]
73                 for i in range(0,z):
74                     b.insert(i,float(row[i]))
75                     i+=1
76                 line_count+=1
77         return b
```

- Les colonnes (les séries des instances des différentes entrées) : pour effectuer une étude exploratoire des caractéristiques descriptives des données.

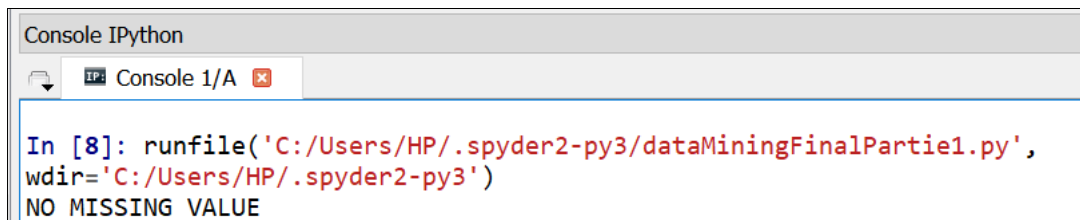
```
81 def readColumn(numColmn):
82     # "this allow to print a column "
83     with open('data_DM.txt') as csv_file:
84         csv_reader = csv.reader(csv_file, delimiter=',')
85         line_count = 0
86         b=[]
87         i=0
88         for row in csv_reader:
89             if(i!=0):
90                 line_count += 1
91                 b.insert(i,float(row[numColmn]))
92                 i=i+1
93         return b
```

➤ 1.3 Valeurs manquantes (Missing Value) :

-Les ensembles de données du monde réels peuvent contenir des valeurs manquantes pour diverses raisons, pour cela nous avons écrit de plus des fonctions de manipulation du dataset, une fonction qui vérifie si ce dernier contient des valeurs manquantes (champs vides). La fonction est la suivante :

```
387 def VerificationMissingValue():
388
389     with open('HEART_Stat_Data.txt') as csv_file:
390         csv_reader = csv.reader(csv_file, delimiter=',')
391         i = 0
392         MissingValue=False
393         s = "NO MISSING VALUE"
394         numColmn=0
395         nbrColumn=13 #nombre de column dans Le dataset
396
397         for i in range(0,nbrColumn):
398             for row in csv_reader: #pour chaque Ligne
399                 if(str(row[numColmn])==''):
400                     MissingValue = True
401                     s="MISSING VALUE"
402                     return s
403         return s
```

Exécution :



```
Console IPython
[IPY] Console 1/A
In [8]: runfile('C:/Users/HP/.spyder2-py3/dataMiningFinalPartie1.py',
wdir='C:/Users/HP/.spyder2-py3')
NO MISSING VALUE
```

-Le dataset ne contient pas de **valeurs manquantes**, du coup on pourrait entamer directement la partie qui suit : « Etude exploratoire des caractéristiques descriptives des données ».

Remarque

-Dans le rapport, on a mis les captures et les interprétations des résultats seulement pour les 3 attributs demandés *de type réel* dont les valeurs sont significatives : « *Resting blood pressure* », « *Serum cholestora* » et « *Maximum heart rate achieved* », En revanche dans la console et l'interface python on pourrait sélectionner et visualiser les résultat pour tous les attributs du dataset.

2 – Etude exploratoire des caractéristiques des données

Avant d'entamer cette partie, on doit d'abord classer les valeurs des séries des différents attributs (les séries statistiques) dans l'ordre croissant afin de pouvoir effectuer nos statistiques ensuite représenter les données avec des différents diagrammes (exploitation de la bibliothèque **matplotlib**).

➤ 2.1 **Statistiques :**

Il s'agit du calcul de la moyenne, la médiane et le mode ; ces derniers sont les mesures principales de tendance centrale d'une série statistique. Elles servent à synthétiser la série étudiée au moyen d'un nombre de valeurs « caractéristiques ».

🚦 2.1.1 Moyenne :

-La moyenne c'est la valeur qui est égale au quotient de la somme de toutes les valeurs (270 valeurs) de la série par le nombre de ces valeurs (l'effectif total).

$$\text{Moyenne} = \frac{\text{somme des valeurs}}{\text{effectif total}} = \frac{\sum xi}{n} = \frac{\sum xi}{270}$$

```
95 def moyenne(col):
96     somme=0
97     nbr=len(col)
98     for var in col:
99         somme=somme+var
100     moy=somme/nbr
101     return moy
```

-Calcul de la moyenne pour les 3 attributs données (att 4, 5 et 8):

$$\text{Resting blood pressure: } \frac{94*2+100*4+101+\dots+200}{270} = 131.34444444444443.$$

$$\text{Serum cholestoral: } \frac{126+141+149*2+\dots+564}{270} = 249.65925925925927.$$

$$\text{Maximum heart rate achieved: } \frac{71+88+95+\dots+202}{270} = 149.67777777777778.$$

🚦 2.1.2 Médiane :

-La médiane c'est la valeur centrale de la série statistique dont les valeurs observées ont été rangées dans l'ordre croissant.

-Dans notre cas le nombre de valeurs des différentes séries est **paire** (270), la médiane donc n'est pas la valeur du milieu (val centrale) mais plutôt la demi somme des deux valeurs du milieu qui est en fait la **moyenne des deux valeurs centrales** ayant les positions $\frac{n}{2}, \frac{n}{2}+1$.i.e. $\frac{270}{2}, \frac{270}{2}+1$.i.e. 135,136 respectivement

$$\text{Médiane} = \frac{\text{somme des deux val centrales}}{2} = \frac{\text{val centrale 1} + \text{val centrale 2}}{2}$$


```

119 def Mediane(col):
120     N=len(col)
121     mediane=0
122     if(N%2==0):
123         pos1=N/2
124         print("pos1")
125         print(pos1)
126         pos2=(N/2)+1
127         print("pos2")
128         print(pos2)
129         mediane=(col[int(pos1)]+col[int(pos2)])/2
130     else:
131         pos=int(N/2)+1
132         mediane=col[int(pos)]
133
134     return mediane

```

-Calcul de la médiane pour les 3 attributs données (att 4, 5 et 8) :

Resting blood pressure: $\frac{130+130}{2} = 130.$

Serum cholestoral: $\frac{245+245}{2} = 245.$

Maximum heart rate achieved: $\frac{145+145}{2} = 145.$

🚩 2.1.3 Mode :

-La valeur la plus fréquente de la série statistique (échantillon)i.e. la ou les valeurs du caractère dont l'effectif est le plus grand.

Le mode est pertinent lorsque dans la série certaines valeurs sont répétées plusieurs fois.

-On parle aussi du type de mode :

Unimodal : s'il existe une valeur distincte dans la série qui a la fréquence max.

Bimodal : s'il existe deux valeurs distinctes dans la série qui ont la même fréquence max ...

```

137 def Mode(col):
138     dict=frequence(col)
139     max=dict[0].get('freq')
140     maxIndx=0
141     pos=0
142     for a in dict :
143         if(a.get('freq')>max):
144             max=a.get('freq')
145             maxIndx=pos
146             pos=pos+1
147     mode=[]
148     for d in dict:
149         if(d.get('freq')==max):
150             mode.append(d.get('value'))
151     size=len(mode)
152     if(size==1):
153         print("UNIMODAL")
154     else:
155         if(mode==2):
156             print("BIMODAL")
157         else:
158             if(mode==3):
159                 print("TRIMODAL")
160             else:
161                 print("MULTIMODAL")
162
163     return(mode)

```

```

104 def frequence(col):
105     liste=[]
106     l=[]
107     col=sorted(col)
108     freq=0
109     pos=0
110     for a in col:
111         if(a not in l ):
112             liste.append({'value':a , 'freq':col.count(a) })
113             freq=freq+col.count(a)
114             pos=pos+1
115             l.insert(pos,a)
116     return(liste)

```

-Calcul du mode pour les 3 attributs données (att 4, 5 et 8):

Resting blood pressure : 120 avec une fréquence max = 34 . Type : UNIMODAL.

Serum cholestoral : 234 avec une fréquence max = 6. Type : UNIMODAL.

Maximum heart rate achieved : 162 avec une fréquence max = 10. Type : UNIMODAL.

➤ 2.2 Symétrie des données :

-On dit que les données sont symétriques lorsque la moyenne = la médiane = le mode.

-Vérification de la symétrie des données des 3 attributs :

Resting blood pressure : 131.34444444444443 \neq 130 \neq 120. ➔ Asymétrie a droite

Serum cholestoral : 249.65925925925927 \neq 245 \neq 234. ➔ Asymétrie a droite

Maximum heart rate achieved : 149.67777777777778 \neq 145 \neq 162. ➔ Asymétrie a gauche

Donc on remarque que les données des 3 attribut « Resting blood pressure », « Serum cholestora » et « Maximum heart rate achieved » **ne sont pas symétriques** ...

Conclusion : Les données sont Asymétriques

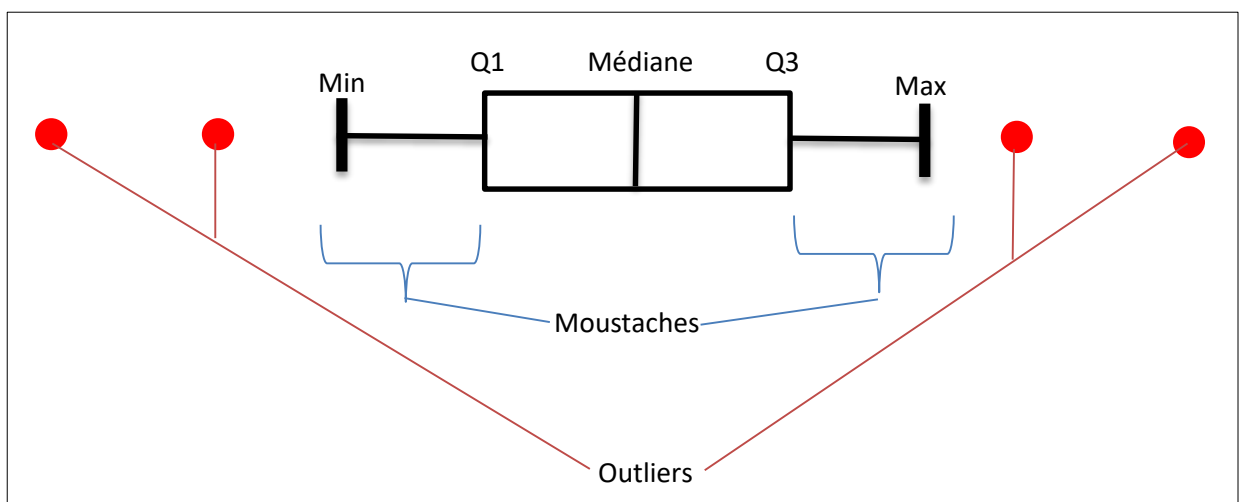
```
207 def symetrie(col,moy,med,mode):
208     if moy==med==mode :
209         S="Les donnees sont Symetriques"
210     else:
211         S="Les donnees sont Asymetriques"
212     return S
```

➤ 2.3 Boite a moustache (diagramme de boite) :

Cette partie met en évidence les cinq paramètres de nos séries et qui sont :

Le minimum	Le 1 ^{er} quartile	La médiane / Le 2 ^{ème} quartile	Le 3 ^{ème} quartile	Le maximum
Min	Q1	Mèd/Q2	Q3	Max

Chaque attribut a sa boite a moustache qui a les caractéristiques suivantes :



2.3.1 Min :

- Initialement c'est la valeur minimale de la série (avant le calcul des valeurs aberrantes faibles).

-Le Min pour les 3 attributs données (att 4, 5 et 8)

Resting blood pressure 94.

Serum cholestoral: 126.

Maximum heart rate achieved: 71.

2.3.2 Q1 :

-C'est la médiane des valeurs strictement inférieures à la médiane de la série, ayant la position $n * 25\%$.i.e $270 * 25\%$.i.e $67.5 \simeq 68$.

-Calcul de Q1 pour les 3 attributs données (att 4, 5 et 8)

Resting blood pressure 120.

Serum cholestoral: 213.

Maximum heart rate achieved: 133.

```
166 def Quartile1(col):
167     N=len(col)
168     diff=((N*25)/100)-int(((N*25)/100))
169     pos = int((N * 25) / 100)
170     if(diff>=0.5):
171         pos =pos+1
172     return(col[pos])
```

2.3.3 Q3 :

-C'est la médiane des valeurs strictement supérieures à la médiane de la série, ayant la position $n * 75\%$.i.e $270 * 75\%$.i.e $202.5 \simeq 203$.

-Calcul de Q3 pour les 3 attributs données (att 4, 5 et 8)

Resting blood pressure 140.

Serum cholestoral: 282.

Maximum heart rate achieved: 167.

```
184 def Quartile3(col):
185     N = len(col)
186     diff = ((N * 75) / 100) - int(((N * 75) / 100))
187     pos = int((N * 75) / 100)
188     if (diff >= 0.5):
189         pos = pos + 1
190     return(col[pos])
```

2.3.4 Max :

-Initialement c'est la valeur maximale de la série (avant le calcul des valeurs aberrantes élevées).

-Le Max pour les 3 attributs données (att 4, 5 et 8)

Resting blood pressure : 200.

Serum cholestoral : 564.

Maximum heart rate achieved : 202.

✓ 2.3.5 Déduction des valeurs aberrantes (Outliers):

-Il s'agit des valeurs qui s'écartent fortement des autres observations, anormalement **faibles** ou **élevée**, on peut les déduire en utilisant la boîte à moustache (les valeurs qui sont en **dessous** et en **dessus** des moustaches), ces valeurs sont calculées automatiquement lorsque on trace la boîte à moustaches en python.

-Pour confirmer ces valeurs aberrantes retournées dans le diagramme à boîte on peut effectuer les calculs suivants sur chacun des attributs demandés :

En gros modo toute valeur inférieure à : $Q1 - 1.5 * \text{Ecart interquartile}$ ou supérieure à : $Q3 + 1.5 * \text{Ecart interquartile}$.

Notons que l'**Ecart interquartile** = $IQR = Q3 - Q1$.

- 2.3.5.1 valeurs aberrantes faibles :

Resting blood pressure : $Q1 - 1.5 * (Q3 - Q1) = 120 - 1.5 * (140 - 120) = 90$ c.à.d. toutes les valeurs de cet attribut qui sont inférieures ou égales à 90 seront considérées comme valeurs aberrantes faibles = [] «ici pas de valeurs aberrantes faibles».

Serum cholestoral : $Q1 - 1.5 * (Q3 - Q1) = 213 - 1.5 * (282 - 213) = 109,5$ c.à.d. toutes les valeurs de cet attribut qui sont inférieures ou égales à 109,5 seront considérées comme valeurs aberrantes faibles = [] «ici pas de valeurs aberrantes faibles».

Maximum heart rate achieved : $Q1 - 1.5 * (Q3 - Q1) = 133 - 1.5 * (167 - 133) = 82$ c.à.d. toutes les valeurs de cet attribut qui sont inférieures ou égales à 82 seront considérées comme valeurs aberrantes faibles = [71] d'où le nouveau min = 88 (successeur de 71 dans la série).

```
193 def OutliersInf(col,Q1,Q3,min):
194     outliers=[]
195     IQR=Q3-Q1
196     BorneInf=Q1-(1.5*IQR)
197
198     if(BorneInf<min):
199         print(" Pas d' Outliers anormalement faibles ")
200     else:
201         for i in col:
202             if(i<=BorneInf):
203                 outliers.append(i)
204
205     return outliers
```

- 2.3.5.2 valeurs aberrantes élevées :

Resting blood pressure : $Q3 + 1.5 * (Q3 - Q1) = 140 + 1.5 * (140 - 120) = 170$ c.à.d. toutes les valeurs de cet attribut qui sont supérieures ou égales à 170 seront considérées comme valeurs aberrantes élevées = [170, 170, 172, 174, 178, 178, 180, 180, 180, 192, 200] d'où le nouveau max = 165 (prédécesseur de 170 dans la série) .

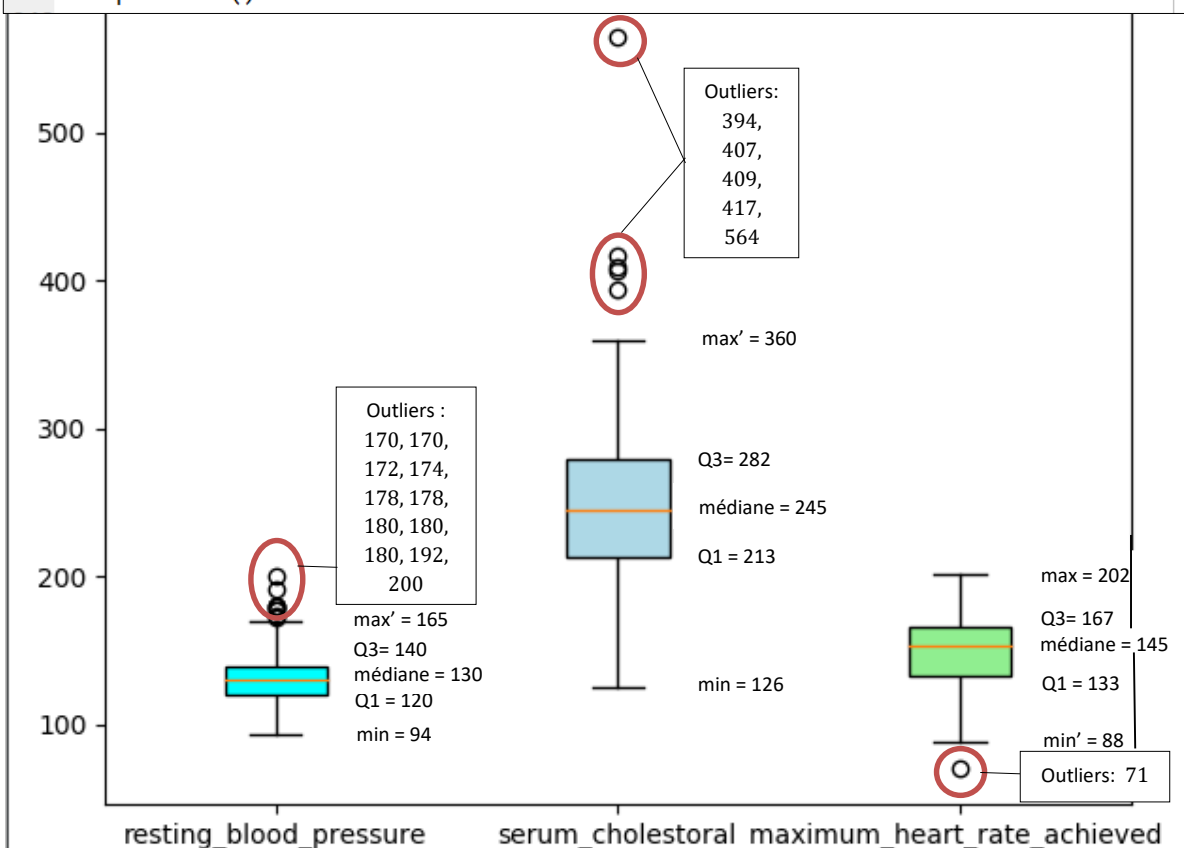
Serum cholestoral : $Q3 + 1.5 * (Q3 - Q1) = 282 + 1.5 * (282 - 213) = 385,5$ c.à.d. toutes les valeurs de cet attribut qui sont supérieures ou égales à 385,5 seront considérées comme valeurs aberrantes élevées = [394, 407, 409, 417, 564] d'où le nouveau max = 360 (prédécesseur de 394 dans la série) .

Maximum heart rate achieved : $Q3 + 1.5 * (Q3 - Q1) = 167 + 1.5 * (167 - 133) = 218$ c.à.d. toutes les valeurs de cet attribut qui sont supérieures ou égales à 218 seront considérées comme valeurs aberrantes élevées = [] «ici pas de valeurs aberrantes élevées» .

```
220 def OutliersSup(col,Q1,Q3,max):
221     outliers=[]
222     IQR=Q3-Q1
223     BorneSup=Q3+(1.5*IQR)
224
225     if(BorneSup>max):
226         print(" Pas d' Outliers anormalement élevées ")
227     else:
228         for i in col:
229             if(i>=BorneSup):
230                 outliers.append(i)
231
232     return outliers
```

✓ 2.3.6 Code et exécution des boîtes à moustaches :

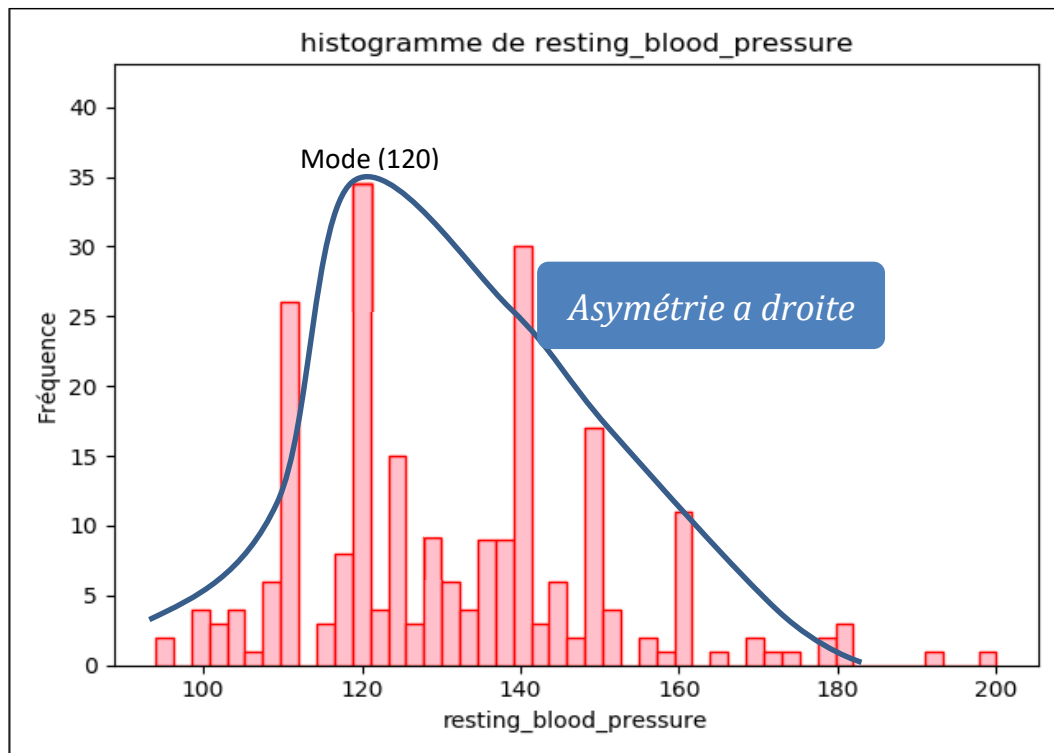
```
338 def boxplot3(num1,num2,num3):
339     c1 = sorted(readColumn(num1))
340     c2 = sorted(readColumn(num2))
341     c3 = sorted(readColumn(num3))
342     box_plot_data=[c1,c2,c3]
343     box=plt.boxplot(box_plot_data,patch_artist=True,labels=[str(df.columns[num1])],
344     colors=['cyan','lightblue','lightgreen']
345     for patch,color in zip(box['boxes'],colors):
346         patch.set_facecolor(color)
347     plt.show()
```



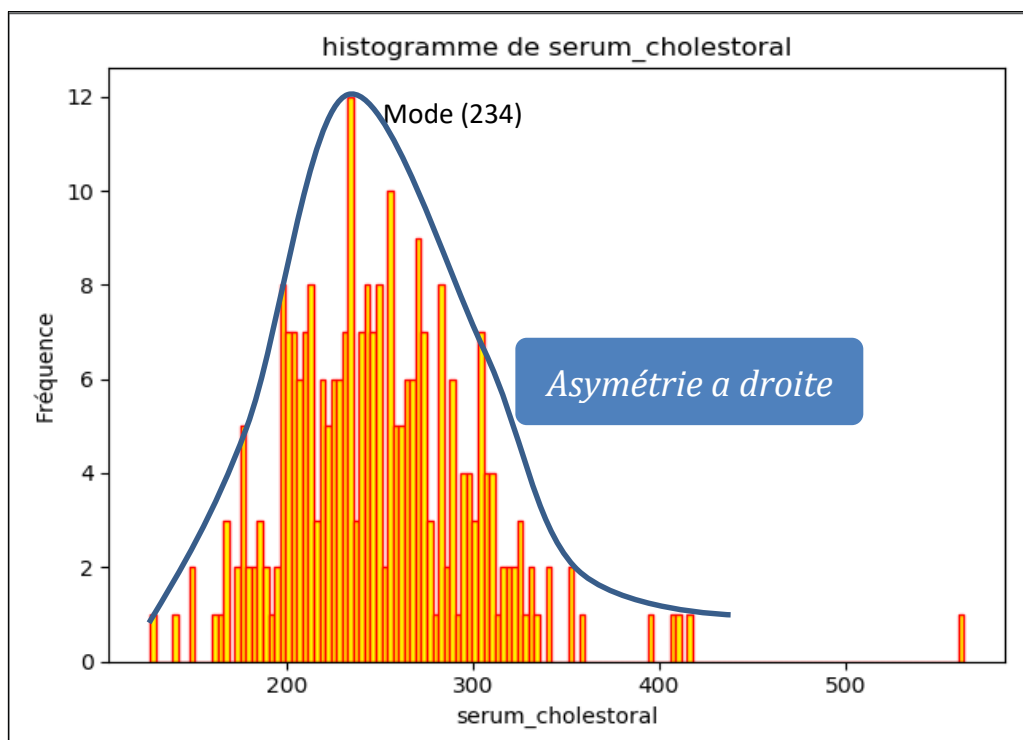
➤ 2.4 Histogramme :

L'histogramme nous a permis de représenter les séries continues de chaque attribut, tel que l'axe des X représente les valeurs et l'axe des Y représente les fréquences. Chaque valeur est représentée par un rectangle de hauteur égale à sa fréquence dans la série. Ainsi avec ces histogrammes on a pu résumer facilement la distribution des données, l'asymétrie ainsi que le mode. i.e. valeur ayant la plus grande fréquence qui est en fait la valeur correspondante au plus haut rectangle. On a effectué 3 histogrammes, un pour chaque attribut réel

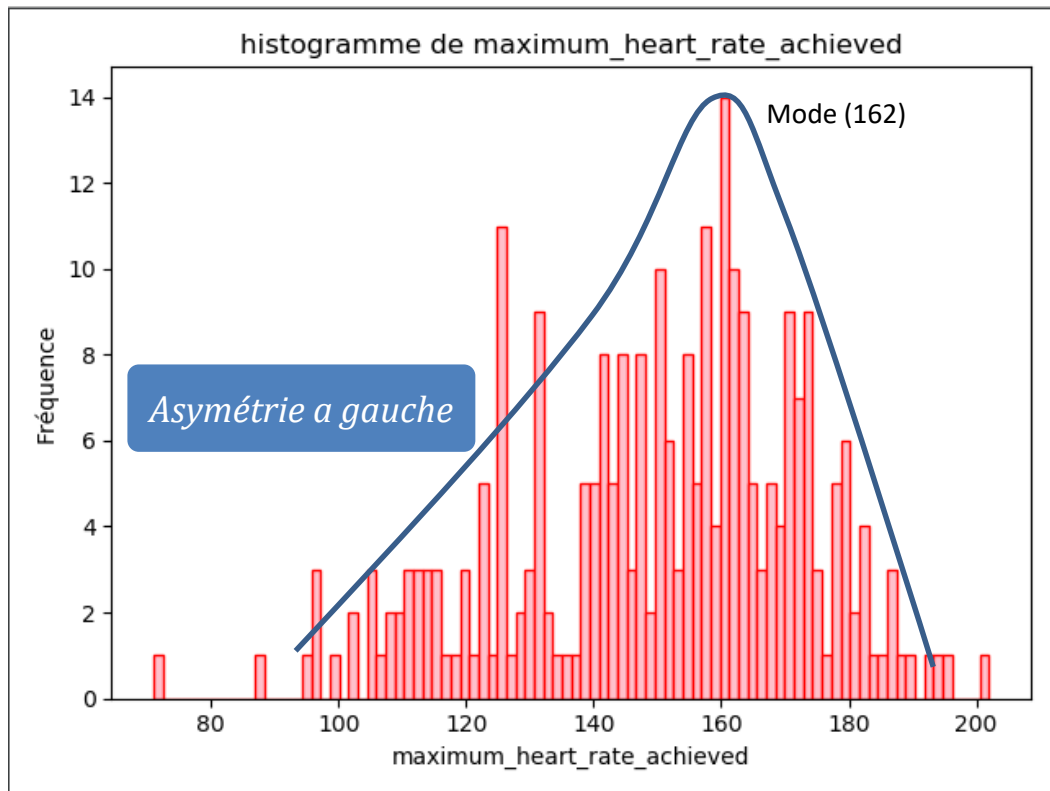
❖ *Resting blood pressure* :



❖ *Serum cholestoral* :



❖ *Maximum heart rate achieved:*



✓ [Code de l'histogramme :](#)

```
373 def histogramme(num):
374
375     c=sorted(readColumn(num))
376     nbr_val1 = len(frequency(c,,
377     plt.hist(c,color = 'pink',
378             edgecolor = 'red',bins=nbr_val1)
379     plt.xlabel(str(df.columns[num]))
380
381     plt.ylabel("Fréquence")
382     plt.title("histogramme de "+str(df.columns[num]))
383     plt.show()
```



➤ 2.5 Diagramme de dispersion (Nuage de points) :

Le diagramme de dispersion est une représentation graphique d'une série statistique a deux variables, il nous a permis d'observer la corrélation des relations entre nos variables deux a deux. Nous avons effectué un ajustement des points ce diagrammes par une courbe afin d'effectuer des prévisions, calculer le coefficient de corrélation (Pearson), extraire des regroupements, de corrélations, outliers...

Coefficient de Corrélation : c'est une mesure de l'intensité et du sens de la relation linéaire entre deux variables, sa valeur est comprise entre -1 et 1.

-Calcul :

$$\text{Cor}(X,Y) = \frac{\text{Cov}(X,Y)}{\partial X * \partial Y}$$

Cov(X,Y) : covariance entre X et Y

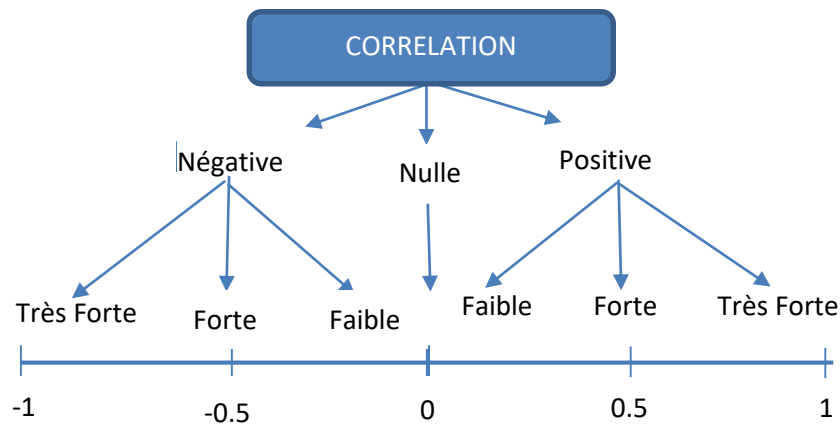
∂X, ∂Y : l'écart type de X, l'écart type de Y respectivement

```
435 def Covariance(c1,c2):
436
437     size=len(c1)
438     moyx=moyenne(c1)
439     moyy=moyenne(c2)
440     somme=0
441     for j in range(0,size):
442         somme+=(c1[j]-moyx)*(c2[j]-moyy)
443
444     return (somme/size)
445
446 def CoefCorrelation(numc1,numc2):
447     return(Covariance(numc1,numc2)/(EcartType(numc1)*EcartType(numc2)))
```

```
424 def EcartType(c):
425
426     moy=moyenne(c)
427     somme=0
428     size=len(c)
429     for i in range(0,size):
430         somme+=(c[i]-moy)*(c[i]-moy)
431
432
433     return (math.sqrt(somme/size))
```

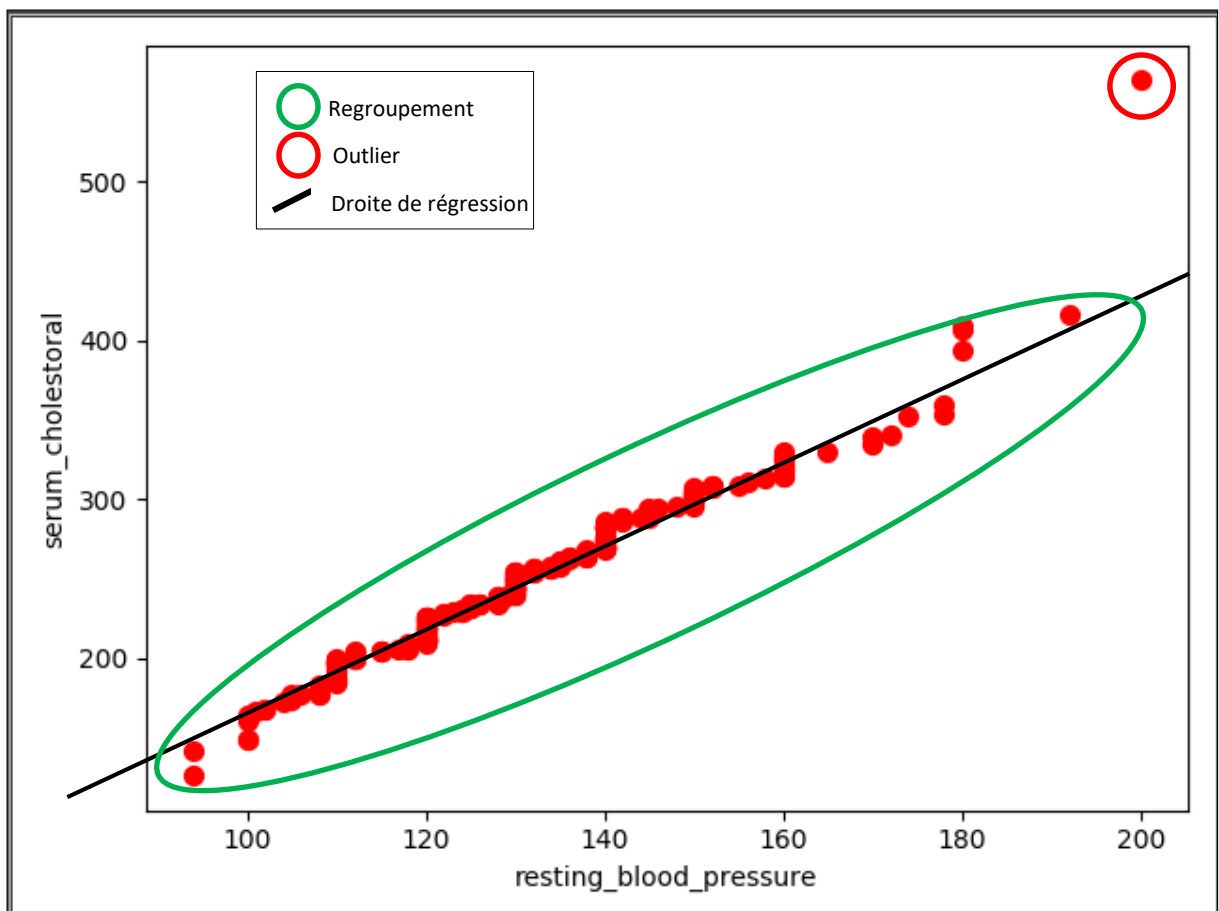
```
446 def CoefCorrelation(numc1,numc2):
447     return(Covariance(numc1,numc2)/(EcartType(numc1)*EcartType(numc2)))
```

-Interprétation :



-On a effectué 3 diagrammes de dispersion, chacun entre 2 attributs réels :

❖ *Resting blood pressure* (att4) & *Serum cholestoral* (att5) :



Regroupement : le nuage est plat et ses points se groupent autour d'une ligne (la droite de régression) dont les coordonnées de ses points étant liés par une relation $y=ax+b$, du coup on peut facilement révéler qu'il y a une forte relation entre l'attribut 4 et 5, pour le confirmer calculons le coefficient de corrélation :

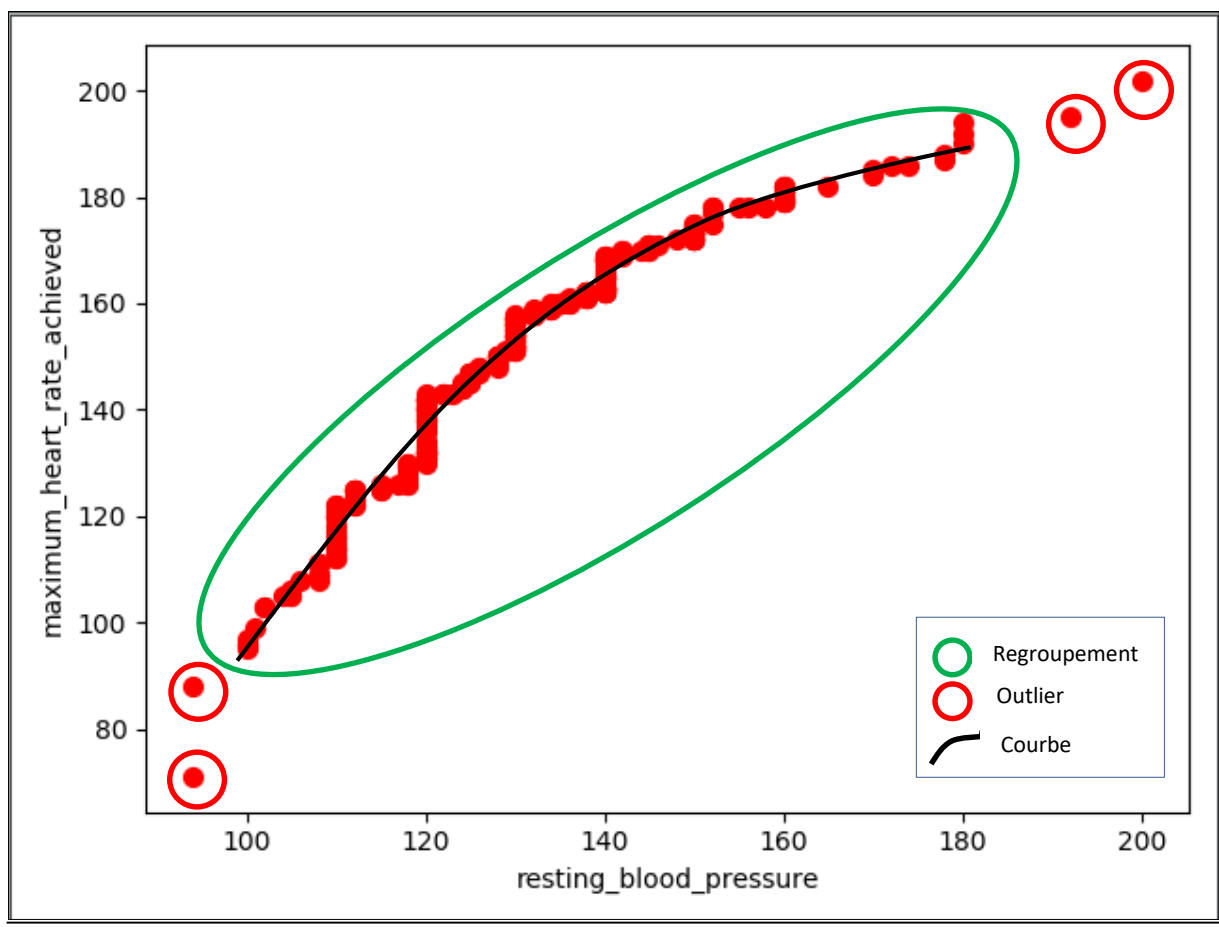
Coefficient de Corrélation : $\frac{Cov(\text{resting_blood_pressure}, \text{serum_cholestoral})}{\sigma_{\text{resting_blood_pressure}} * \sigma_{\text{serum_cholestoral}}} = 0.982186044956698.$

Corrélation : corrélation **linéaire, positive** (lorsque l'attribut 4 croît, l'attribut 5 croît également), **très forte** (coef de corrélation converge vers 1).

Outliers : il s'agit de tous les points qui n'appartiennent pas au regroupement, ainsi les outliers obtenus dans les diagrammes de dispersions correspondent aux outliers obtenus dans la boîte à moustaches.

❖ *Resting blood pressure* (att4) & *Maximum heart rate achieved* (at8) :

On ajuste les points de ce diagramme d'une façon à avoir une certaine courbe afin de décrire la relation entre l'attribut 4 (resting blood pressure) et l'attribut 8 (maximum heart rate achieved).



Regroupement : le nuage est plat et ses points se groupent autour d'une légère courbure, du coup on peut facilement révéler qu'il y a une forte relation entre l'attribut 4 et 8 du dataset, pour le prouver calculons le coefficient de corrélation :

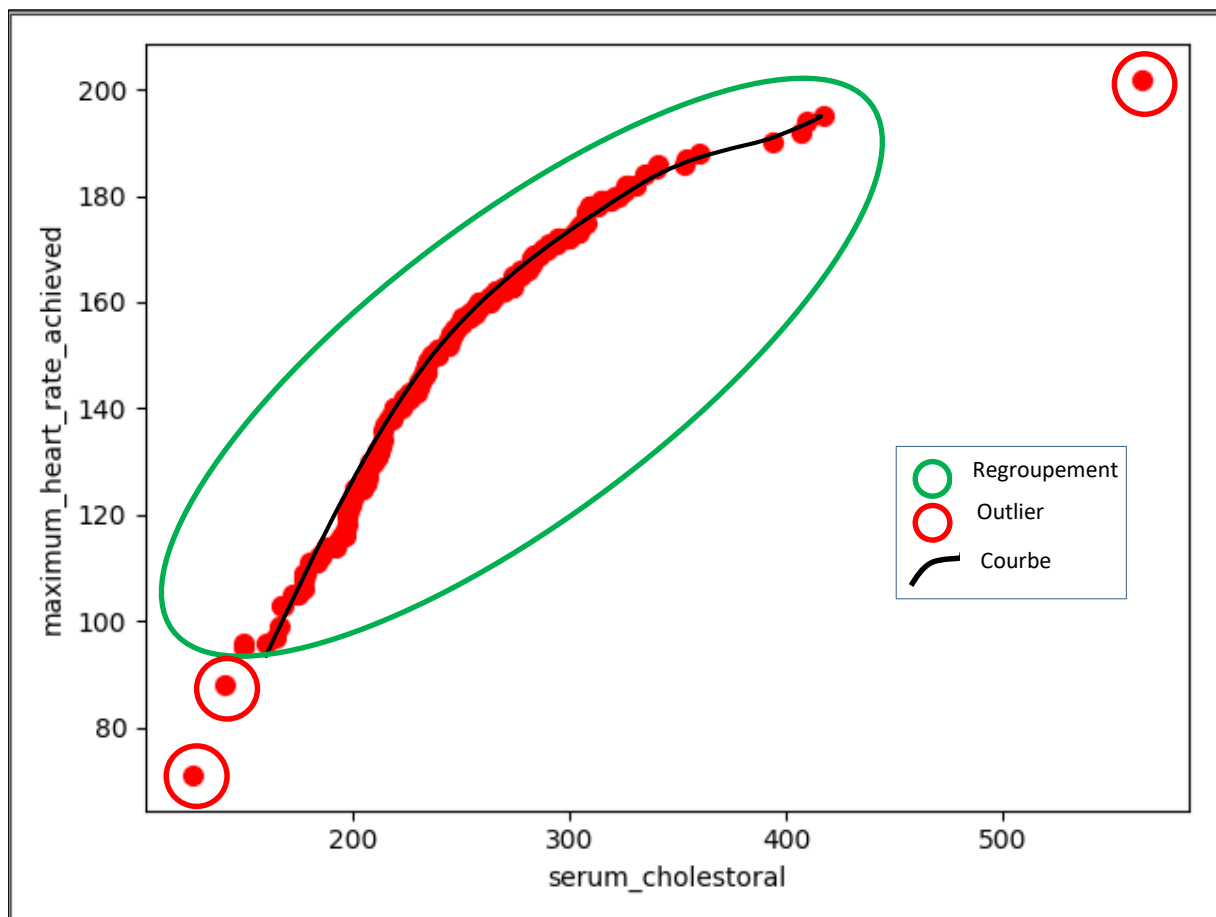
Coefficient de Corrélation : $\frac{\text{Cov}(\text{resting_blood_pressure}, \text{maximum_heart_rate_achieved})}{\sigma_{\text{resting_blood_pressure}} * \sigma_{\text{maximum_heart_rate_achieved}}} = 0.9478371771008656.$

Corrélation : corrélation **positive** (lorsque l'attribut 4 croît, l'attribut 8 croît également), **très forte** (coef de corrélation converge vers 1).

Outliers : il s'agit de tous les points qui n'appartiennent pas au regroupement, ainsi les outliers obtenus dans les diagrammes de dispersions correspondent aux outliers obtenus dans la boîte à moustaches.

❖ *Serum cholestoral* (att5) & *Maximum heart rate achieved* (att8) :

On ajuste les points de ce diagramme d'une façon à avoir une certaine courbe afin de décrire la relation entre l'attribut 5 (serum cholestoral) et l'attribut 8 (maximum heart rate achieved).



Regroupement : le nuage est plat et ses points se groupent autour d'une légère courbure, du coup on peut facilement révéler qu'il y a une forte relation entre l'attribut 5 et 8 du dataset, pour le prouver calculons le coefficient de corrélation :

Coefficient de Corrélation : $\frac{\text{Cov}(\text{serum_cholestoral}, \text{maximum_heart_rate_achieved})}{\sigma_{\text{serum_cholestoral}} * \sigma_{\text{maximum_heart_rate_achieved}}} = 0.9357798819137487.$

Corrélation : corrélation **positive** (lorsque l'attribut 5 croît, l'attribut 8 croît également), **très forte** (coef de corrélation converge vers 1).

Outliers : il s'agit de tous les points qui n'appartiennent pas au regroupement, ainsi les outliers obtenus dans les diagrammes de dispersions correspondent aux outliers obtenus dans la boîte à moustaches.

✓ Code du diagramme de dispersion :

```
315 def dispersion(num1,num2):
316     c1=sorted(readColumn(num1))
317     c2=sorted(readColumn(num2))
318     plt.scatter(c1, c2, label='myPlot', color='red', s=50)
319     plt.xlabel(str(df.columns[num1]))
320     plt.ylabel(str(df.columns[num2]))
321     plt.show()
```

➤ 2.6 Existence des corrélations entre différents attributs :

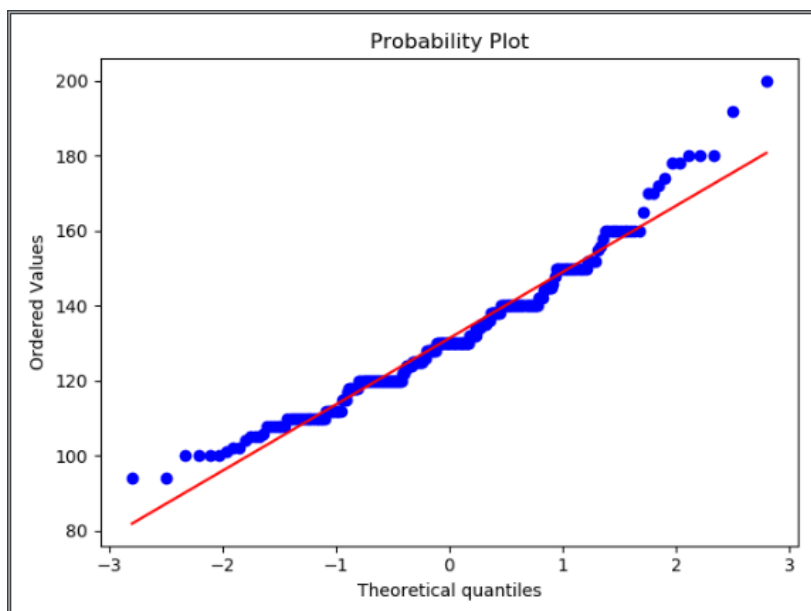
Il est clair qu'il y a des relations entre les trois attributs demandés du dataset HEART_Stat.txt, ce qui est logique étant donné que :

- D'après les coefficients de corrélation entre ces attributs (convergence vers 1)
- D'après les diagrammes de corrélations obtenus,
- Les valeurs réelles des attributs (**Tension artérielle au repos, Serum cholestoral, Fréquence cardiaque maximale atteinte**) sont significatives.
- Les attributs ont une relation (CULTURE GENERALE) car ils sont tous liés à la maladie cardiaque et qui dit un de ces attribut dit la maladie cardiaque.
- On a demandé l'avis d'un Médecin spécialiste en cardiologie (terrain réel) et ce dernier nous a confirmé qu'il y a une grande relation entre ces attributs tels qu'en se basant sur eux on pourrait trancher à 80% si le patient est atteint de la maladie ou pas.

➤ 2.7 QQplot (Droite de Henry) :

Il s'agit de la courbe des fréquences cumulées (fonction de répartition) donnant une droite dans le cas d'une distribution normale, ce qui permet un diagnostic visuel simple de la normalité d'une distribution. Pour tracer une droite de Henry, on range l'ensemble des valeurs par ordre croissant et on attribue à chaque valeur la fréquence cumulée « p » obtenue en divisant le rang de la valeur par $N + 1$ (taille de l'échantillon augmentée d'une unité). Si la distribution est normale, tous les points sont alignés sur une droite parfaite inclinée à 45°. Outre le diagnostic de normalité, la droite de Henry permet d'apprécier la forme de la distribution

❖ *Resting blood pressure :*

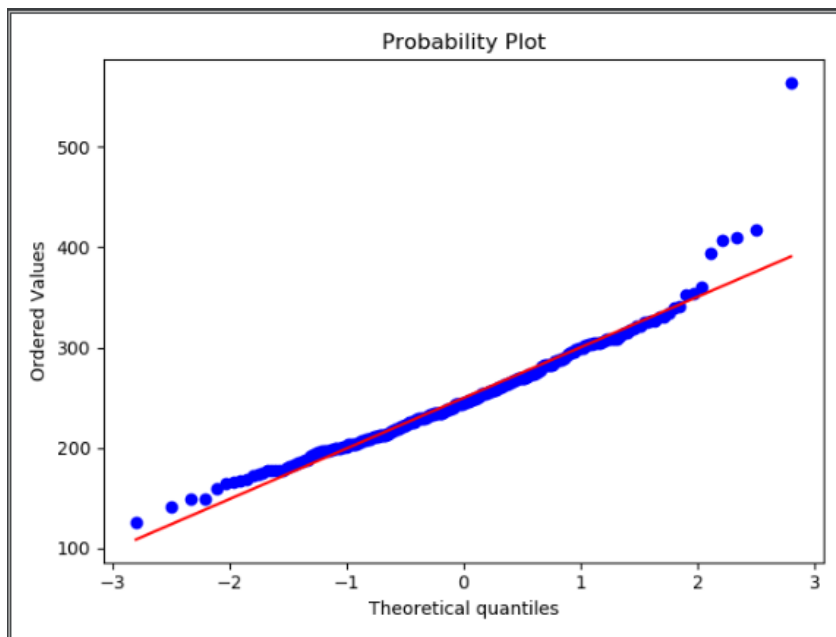


Les points sont alignés sur une **droite** parfaite inclinée à 45°



On déduit que la distribution est **normale** pour l'attribut resting blood pressure.

❖ *Serum cholesterol :*

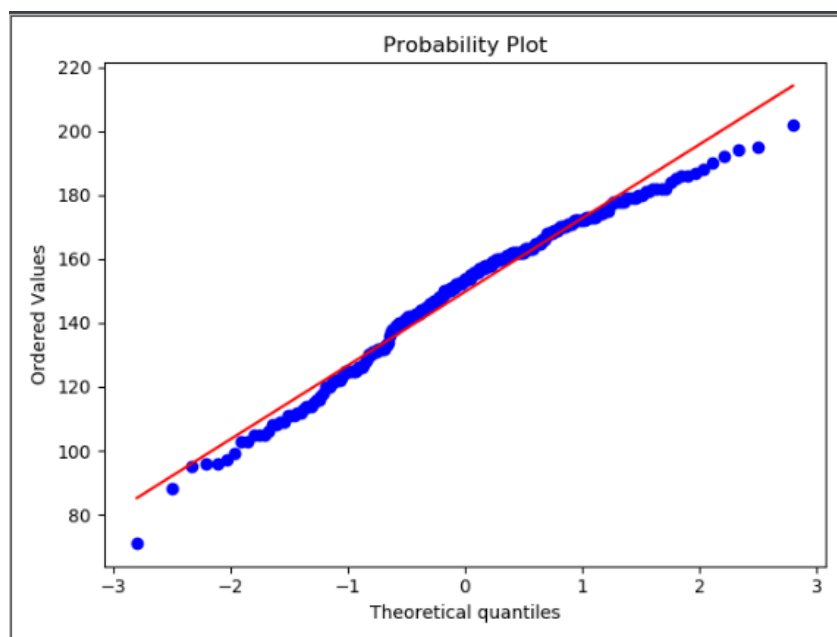


Les points sont alignés sur une **droite** inclinée légèrement en dessous de 45°



On déduit que la distribution est **normale** pour l'attribut serum cholesterol.

❖ *Maximum heart rate achieved :*



Les points sont alignés sur une **droite** parfaite inclinée à 45°



On déduit que la distribution est **normale** pour l'attribut maximum heart rate achieved.

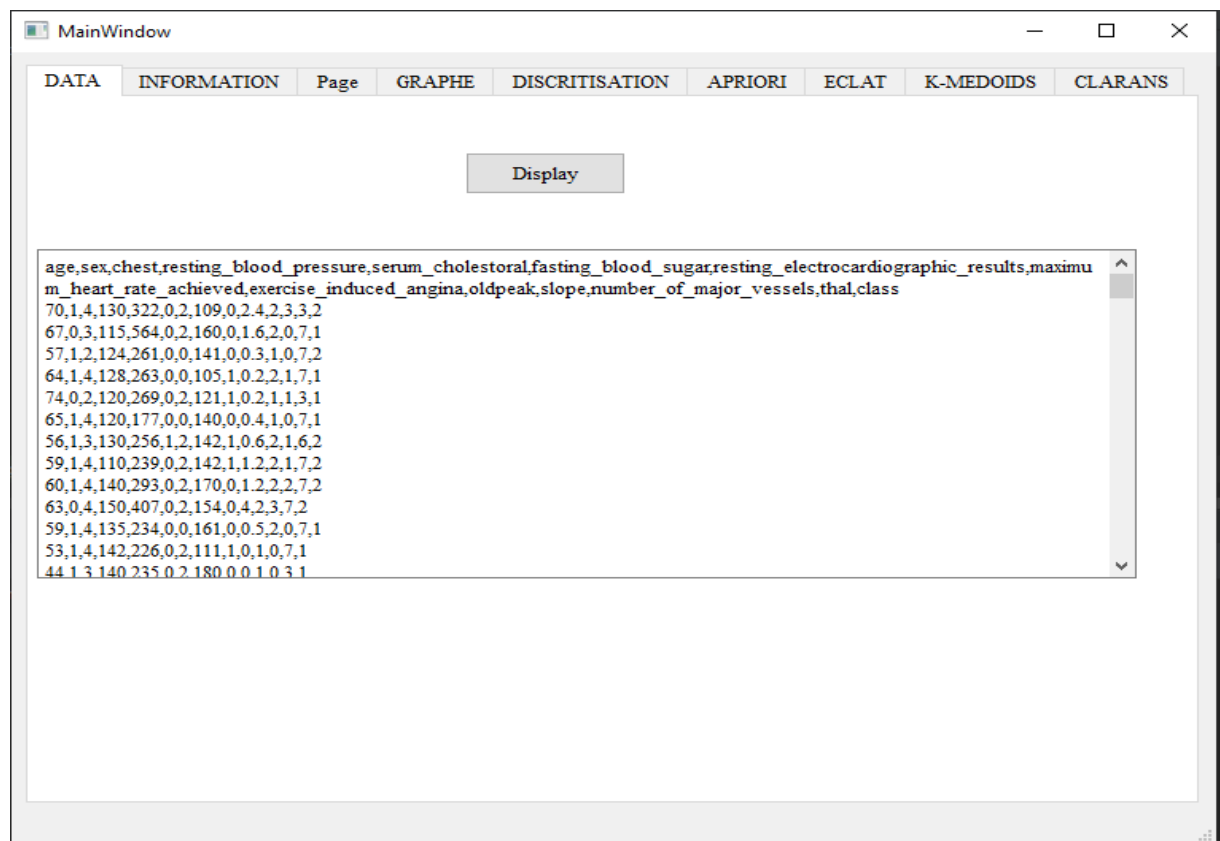
✓ [Code QQplot :](#)

```
328 def Qplot(num2):  
329     c=sorted(readColumn(num2))  
330     #measurements = np.random.normal(loc = 20, scale = 5, size=100)  
331     stats.probplot(c, dist="norm", plot=pylab)  
332     pylab.show()
```

3 – INTERFACE

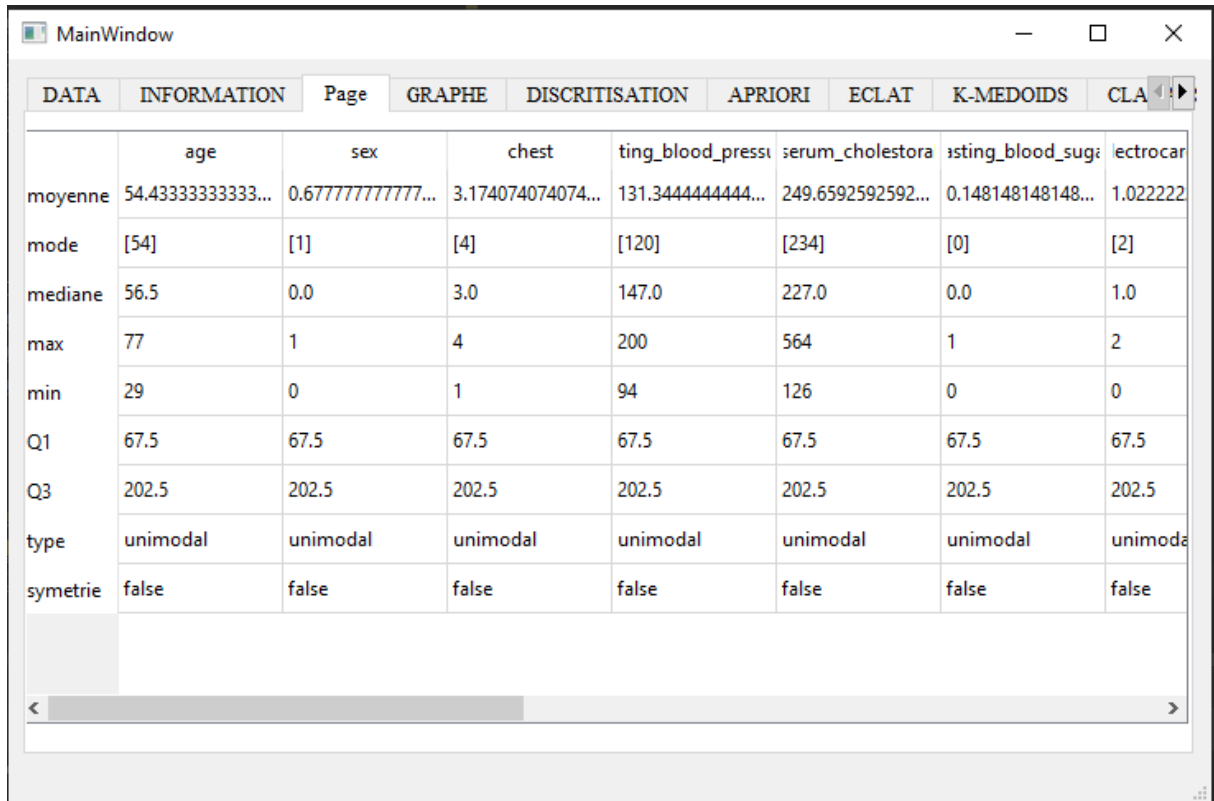
- ✓ Dans cette Partie nous allons visualiser tous les résultats obtenus (statistiques, diagrammes ...) dans une interface python (une sorte de résumé des points abordés dans le projet) afin de faciliter sa compréhension ainsi que pour avoir une idée globale sur les données du dataset HEART_Stat surtout pour les cardiologues qui ne s'y connaissent pas dans le domaine informatique et qui veulent comprendre le contenu du dataset ou exploiter ses données pour des besoins professionnels.

>> **VISUALISER LES DONNEES DU DATASET :**



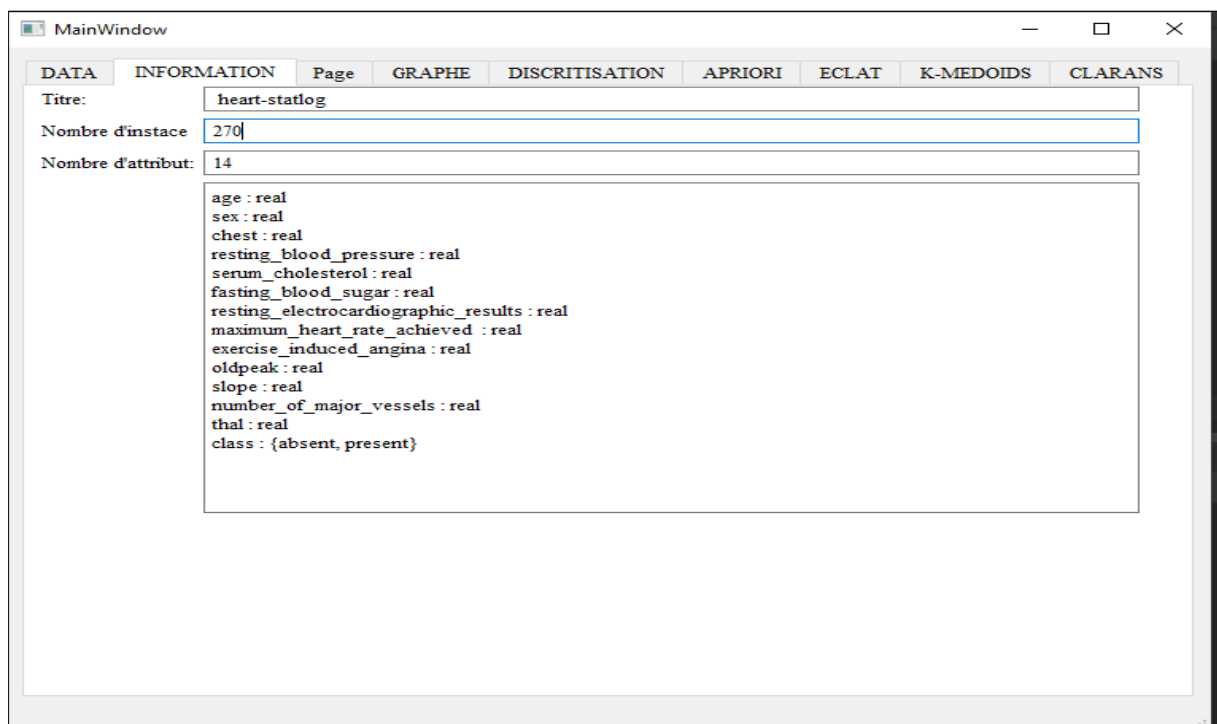
>> **Résultats pour tous les attributs :**

On a les colonnes représentent les attributs, les lignes représentent les différents résultats (moyenne, mode, médiane, max, min, Q1, Q3, type mode, et la symétrie)



	age	sex	chest	ting_blood_pressi	serum_cholestora	isting_blood_suga	lectrocar
moyenne	54.43333333333...	0.67777777777...	3.174074074074...	131.3444444444...	249.6592592592...	0.148148148148...	1.022222
mode	[54]	[1]	[4]	[120]	[234]	[0]	[2]
mediane	56.5	0.0	3.0	147.0	227.0	0.0	1.0
max	77	1	4	200	564	1	2
min	29	0	1	94	126	0	0
Q1	67.5	67.5	67.5	67.5	67.5	67.5	67.5
Q3	202.5	202.5	202.5	202.5	202.5	202.5	202.5
type	unimodal	unimodal	unimodal	unimodal	unimodal	unimodal	unimodal
symetrie	false	false	false	false	false	false	false

>> **INFORMATION SUR LE DATASET :**



Titre: heart-statlog

Nombre d'instance: 270

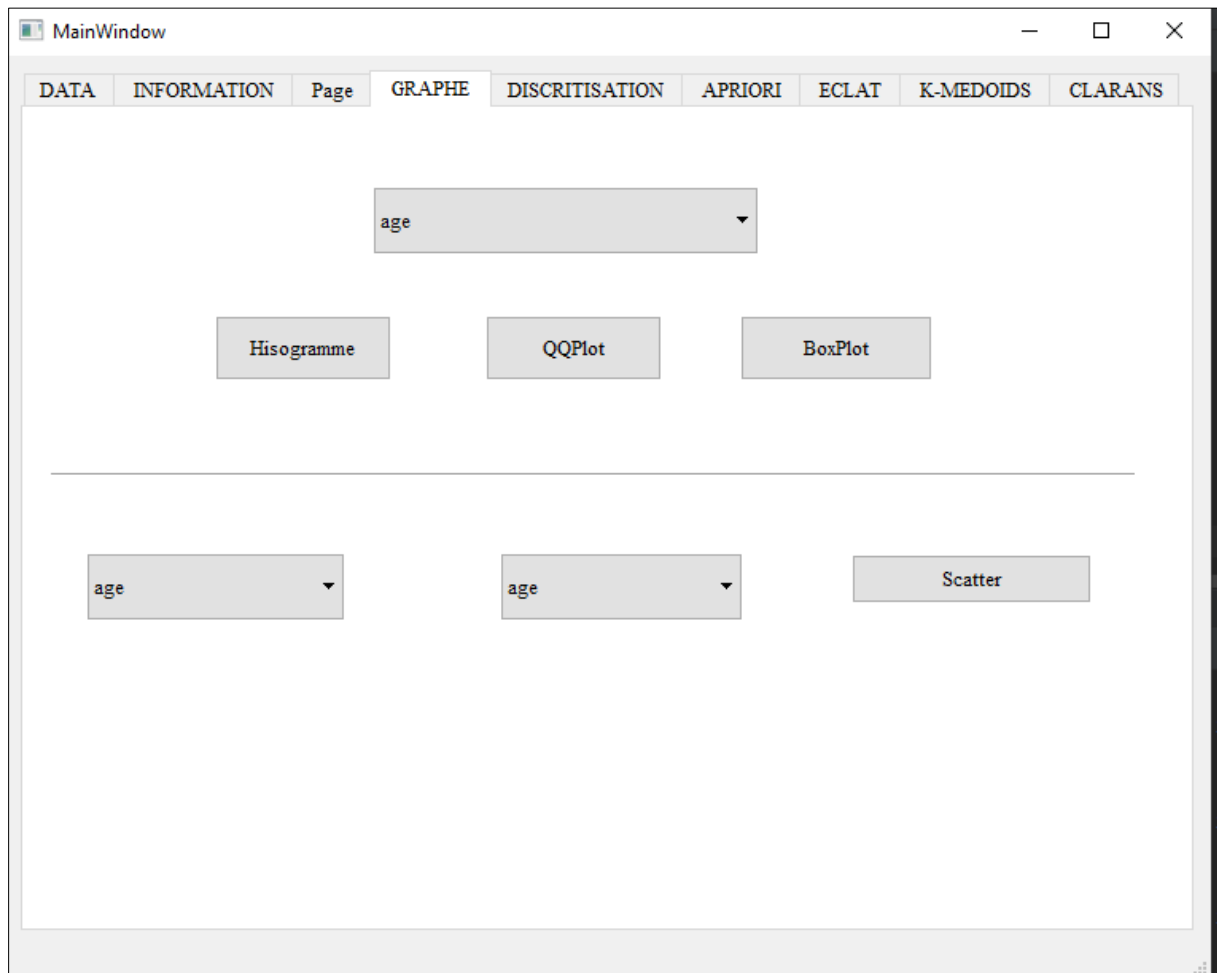
Nombre d'attribut: 14

```

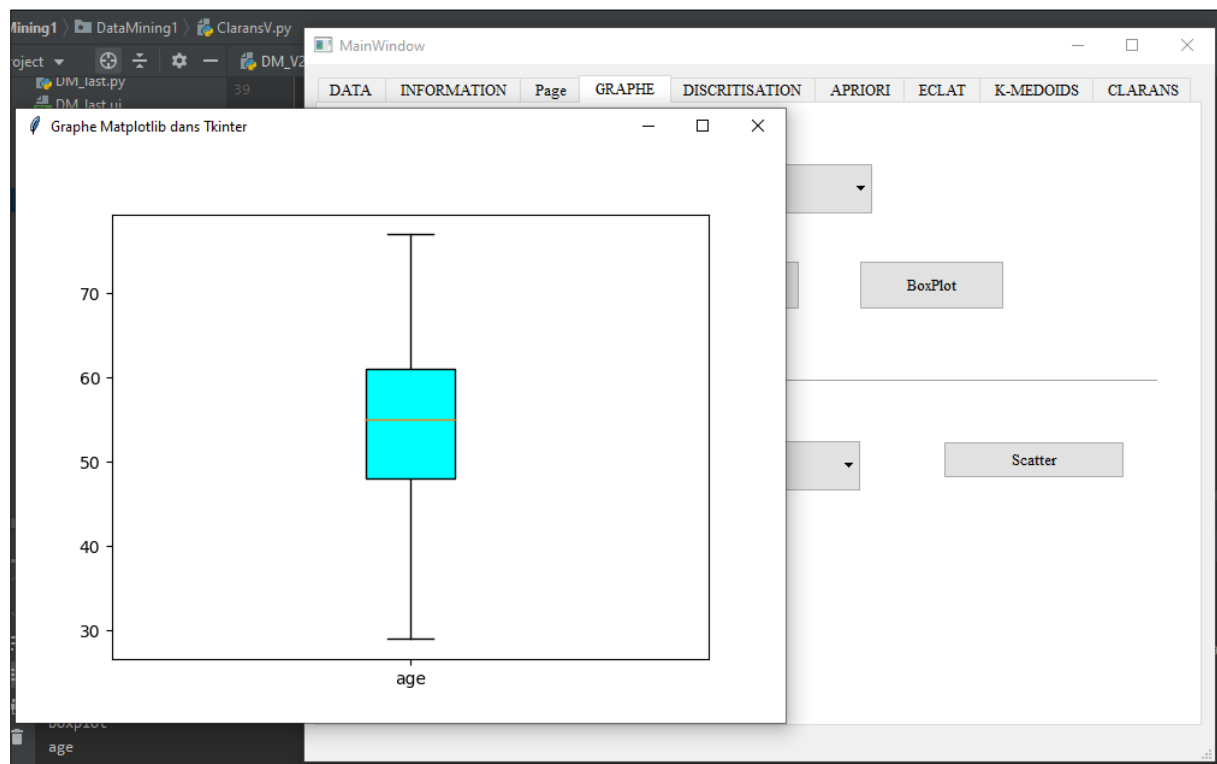
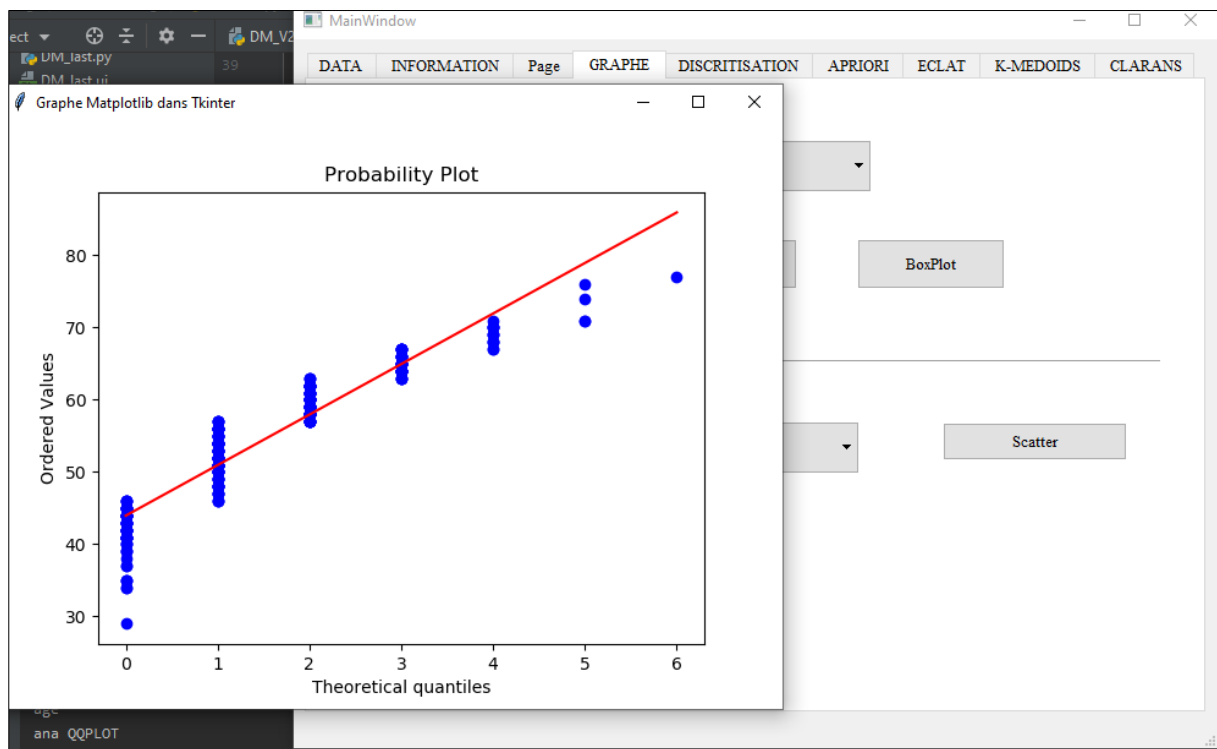
age : real
sex : real
chest : real
resting_blood_pressure : real
serum_cholesterol : real
fasting_blood_sugar : real
resting_electrocardiographic_results : real
maximum_heart_rate_achieved : real
exercise_induced_angina : real
oldpeak : real
slope : real
number_of_major_vessels : real
thal : real
class : {absent, present}
  
```

>> Affichage des graphes :

On sélectionne l'attribut désiré et on clique sur le graphe qu'on veut afficher (partie haute), sauf pour le diagramme de dispersion on sélectionne deux attributs différents (partie basse).



Exemple d'affichage de la boite a moustache et QQPlot pour l'attribut âge :



CONCLUSION GENERALE

Pour conclure, on dit que le **prétraitement des données** concernant l'extraction des caractéristiques descriptives des données du dataset HEART_Stat nous a permis d'avoir une vue globale sur les données ainsi que sur leur distribution.

Ainsi, grâce à l'analyse des données et aux différentes méthodes développées sous python, on a pu révéler que :

- ✓ Les données du dataset HEART_Stat sont **Complètes** i.e. ce dernier ne contient pas de valeurs manquantes.
- ✓ Les données des différents attributs du dataset ne sont pas symétriques (i.e. les données sont **asymétriques**) tel que quelques attributs sont asymétriques à gauche (à l'instar *Maximum heart rate achieved*), d'autres sont asymétriques à droite (comme *Serum cholesterol*, *Resting blood pressure*).
- ✓ Les données des différents attributs du dataset sont **bruitées** et erronés (i.e. existence des valeurs aberrantes faibles et élevées).
- ✓ Certains attributs sont **corrélés** i.e. il existe une relation positive et très forte entre eux, surtout pour les attributs de type réel ayant des valeurs significatives (*Resting blood pressure*, *Serum cholesterol*, *Maximum heart rate achieved*).
- ✓ Les données du dataset sont **distribuées** d'une façon **normale** surtout pour les attributs de type réel ayant des valeurs significatives (*Resting blood pressure*, *Serum cholesterol*, *Maximum heart rate achieved*) → Données suivent une loi normale.

Maintenant, on pourrait excéder à la prochaine étape, qui est la mise en forme des résultats, en passant par le plus important : la classification ainsi qu'aux différents algorithmes de datamining :

-Apprentissage supervisé :

- **Arbres de décision** : ID3, C4.5, Gini index(CART).
- **Les K- proches voisins** (K-NN ou K nearest neighbors).
- **Réseaux de neurones**.

-Apprentissage non supervisé :

- **Clustering** :
 - >> **Partitionnement** : K-Means, K-Medoids (&PAM), CLARA, CLARANS...
 - >> **Hiérarchique** : Diana, Agnes, CHAMELEON...
 - >> **Basé sur la densité** : DBScan...
- **Extraction des motifs fréquents et règles d'association** : Apriori, Eclat, FP-Growth
- **Séquence mining**.

-**Apprentissage Incrémental / semi-supervisé** : méthodes bayésiennes, Séparateurs à Vastes Marges...