TP2 Heart Disease #Santé (partie 1)

Prédiction des patients atteints de maladie cardiovasculaire

Objectif : appréhender et développer toutes les étapes permettant l’utilisation d’une méthode d’apprentissage automatique supervisée

- Exploration de données
- Découper le jeu de données en une partie pour l’apprentissage et l’autre pour le test
- Évaluation et comparaison des différents algorithmes sur les modèles fournis
- Matrice de confusion
- Courbe ROC

Méthodes :

- Arbre de décision
- Forêts aléatoires

In [1]:

```
import numpy as np
import pandas as pd
```

In [2]:

```
df = pd.read_csv("../input/heart.csv")
```

In [3]:

```
#Afficher les 10 premières lignes
df.head(10)
```

Out[3]:

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
0	63	1	3	145	233	1	0	150	0	2.3	0	0	1	1
1	37	1	2	130	250	0	1	187	0	3.5	0	0	2	1
2	41	0	1	130	204	0	0	172	0	1.4	2	0	2	1
3	56	1	1	120	236	0	1	178	0	0.8	2	0	2	1
4	57	0	0	120	354	0	1	163	1	0.6	2	0	2	1
5	57	1	0	140	192	0	1	148	0	0.4	1	0	1	1
6	56	0	1	140	294	0	0	153	0	1.3	1	0	2	1
7	44	1	1	120	263	0	1	173	0	0.0	2	0	3	1
8	52	1	2	172	199	1	1	162	0	0.5	2	0	3	1
9	57	1	2	150	168	0	1	174	0	1.6	2	0	2	1

13 variables descriptives dont 7 qualitatives (sex, cp, fbs, restecg, exang, slope, thal) et 6 quantitatives (age,

trestbps, chol, thalach, oldpeak, ca)

1 variable cible catégorielle à 2 modalités

- age
- sex (1 = male, 0 = female)
- cp : chest pain type (Value 0: typical angina, Value 1: atypical angina, Value 2: non-anginal pain, Value 3: asymptomatic)
- trestbps : tension artérielle au repos (resting blood pressure) (mm Hg on admission to the hospital)
- chol : serum cholestoral measurement in mg/dl
- fbs : fasting blood sugar (> 120 mg/dl, 1 = true; 0 = false)
- restecg : resting electrocardiographic results (0 = normal, 1 = having ST-T wave abnormality, 2 = showing probable or definite left ventricular hypertrophy by Estes' criteria)
- thalach : fréquence cardiaque maximale atteinte
- exang : exercise induced angina (1 = yes; 0 = no)
- oldpeak : ST depression induced by exercise relative to rest ('ST' relates to positions on the ECG (ElectroCardioGram) plot. See more here (https://litfl.com/st-segment-ecg-library/))
- slope : the slope of the peak exercise ST segment (Value 1: upsloping, Value 2: flat, Value 3: downsloping)
- ca : number of major vessels (0-3) colored by flourosopy
- thal : A blood disorder called thalassemia (3 = normal; 6 = fixed defect; 7 = reversable defect)
- target : Heart disease (0 = no, 1 = yes)

Diagnosis: The diagnosis of heart disease is done on a combination of clinical signs and test results. The types of tests run will be chosen on the basis of what the physician thinks is going on 1 (https://www.mayoclinic.org/diseases-conditions/heart-disease/diagnosis-treatment/drc-20353124), ranging from electrocardiograms and cardiac computerized tomography (CT) scans, to blood tests and exercise stress tests 2 (https://www.heartfoundation.org.au/your-heart/living-with-heart-disease/medical-tests).

Looking at information of heart disease risk factors led me to the following: **high cholesterol, high blood pressure, diabetes, weight, family history and smoking** 3 (https://www.bhf.org.uk/informationsupport/risk-factors). According to another source 4 (https://www.heart.org/en/health-topics/heart-attack/understand-your-risks-to-prevent-a-heart-attack), the major factors that can't be changed are: **increasing age, male gender and heredity**. Note that **thalassemia**, one of the variables in this dataset, is heredity. Major factors that can be modified are: **Smoking, high cholesterol, high blood pressure, physical inactivity, and being overweight and having diabetes**. Other factors include **stress, alcohol and poor diet/nutrition**.

I can see no reference to the 'number of major vessels', but given that the definition of heart disease is **"...what happens when your heart's blood supply is blocked or interrupted by a build-up of fatty substances in the coronary arteries"**, it seems logical the *more* major vessels is a good thing, and therefore will reduce the probability of heart disease.

In [4]:

```
#Renommer les noms de colonnes
df.columns = ['age', 'sex', 'chest_pain_type', 'resting_blood_pressure', 'cholesterol', 'fa
       'exercise_induced_angina', 'st_depression', 'st_slope', 'num_major_vessels', 'thalas
```

In [5]:

```
#Types des variables
df.dtypes
```

Out[5]:

```
age                          int64
sex                          int64
chest_pain_type              int64
resting_blood_pressure       int64
cholesterol                  int64
fasting_blood_sugar          int64
rest_ecg                     int64
max_heart_rate_achieved      int64
exercise_induced_angina      int64
st_depression              float64
st_slope                     int64
num_major_vessels            int64
thalassemia                  int64
target                       int64
dtype: object
```

In [6]:

```
#Définir les types appropriés : les variables numériques discrètes deviennent de type objec
df['sex'] = df['sex'].astype('object')
df['chest_pain_type'] = df['chest_pain_type'].astype('object')
df['fasting_blood_sugar'] = df['fasting_blood_sugar'].astype('object')
df['rest_ecg'] = df['rest_ecg'].astype('object')
df['exercise_induced_angina'] = df['exercise_induced_angina'].astype('object')
df['st_slope'] = df['st_slope'].astype('object')
df['thalassemia'] = df['thalassemia'].astype('object')
```

In [7]:

```
#Vérification des nouveaux types
df.dtypes
```

Out[7]:

```
age                          int64
sex                         object
chest_pain_type             object
resting_blood_pressure       int64
cholesterol                  int64
fasting_blood_sugar         object
rest_ecg                    object
max_heart_rate_achieved      int64
exercise_induced_angina     object
st_depression              float64
st_slope                    object
num_major_vessels            int64
thalassemia                 object
target                       int64
dtype: object
```

Note : target ne doit pas passer en objet sinon message d'erreur dans l'arbre de décision

Exploration des données

In [8]:

```
df.describe()
```

Out[8]:

	age	resting_blood_pressure	cholesterol	max_heart_rate_achieved	st_depression
count	303.000000	303.000000	303.000000	303.000000	303.000000
mean	54.366337	131.623762	246.264026	149.646865	1.039604
std	9.082101	17.538143	51.830751	22.905161	1.161075
min	29.000000	94.000000	126.000000	71.000000	0.000000
25%	47.500000	120.000000	211.000000	133.500000	0.000000
50%	55.000000	130.000000	240.000000	153.000000	0.800000
75%	61.000000	140.000000	274.500000	166.000000	1.600000
max	77.000000	200.000000	564.000000	202.000000	6.200000

In [9]:

```
#Analyse des moyennes des variables discrétisé par la variable cible target (deux modalité
df.groupby('target').mean()
#Elle sera effectuée uniquement sur les variables quantitatives
```

Out[9]:

	age	resting_blood_pressure	cholesterol	max_heart_rate_achieved	st_depression
target					
0	56.601449	134.398551	251.086957	139.101449	1.585507
1	52.496970	129.303030	242.230303	158.466667	0.583030

Analyse univariable de la variable cible

In [10]:

```
#Nombre d'individus discrétisés par la variable cible
df.target.value_counts()
```

Out[10]:

```
1    165
0    138
Name: target, dtype: int64
```

Dans notre jeu de données, il y a plus de patients atteints de maladie cardiovasculaire (165) que de patients non atteints (138).

In [11]:

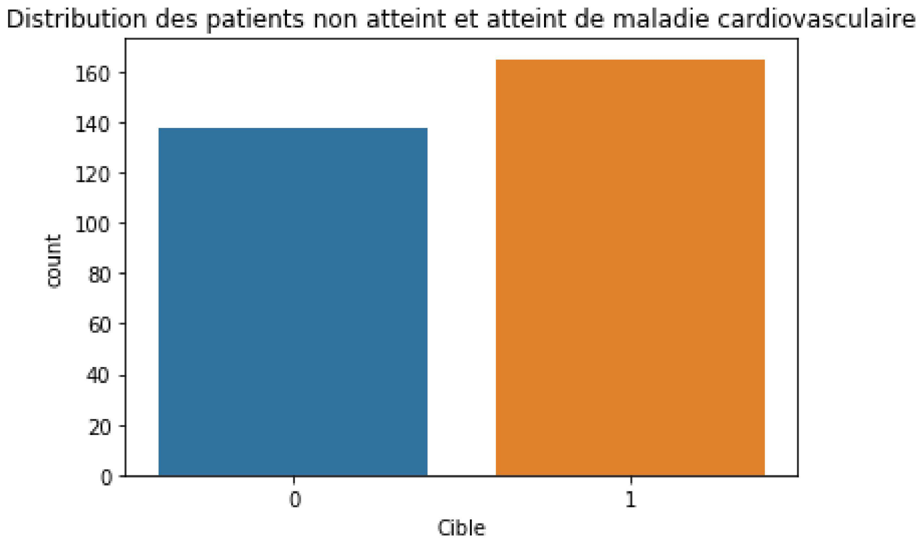
```
import matplotlib.pyplot as plt
import seaborn as sns #for plotting
```

In [12]:

```
#Variable cible
sns.countplot(x="target", data=df)
plt.title('Distribution des patients non atteint et atteint de maladie cardiovasculaire')
plt.xlabel("Cible")
```

Out[12]:

```
Text(0.5, 0, 'Cible')
```



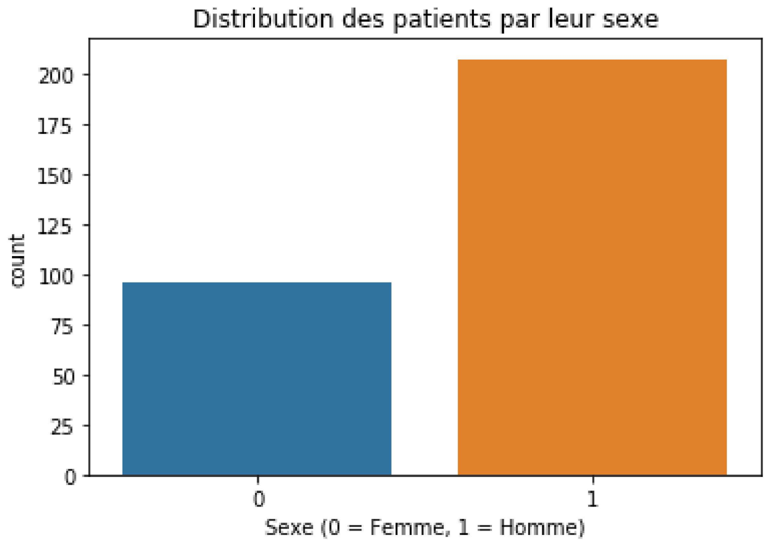
Analyse univariable du sexe

In [13]:

```
sns.countplot(x='sex', data=df)
plt.xlabel("Sexe (0 = Femme, 1 = Homme)")
plt.title("Distribution des patients par leur sexe")
```

Out[13]:

```
Text(0.5, 1.0, 'Distribution des patients par leur sexe')
```



In [14]:

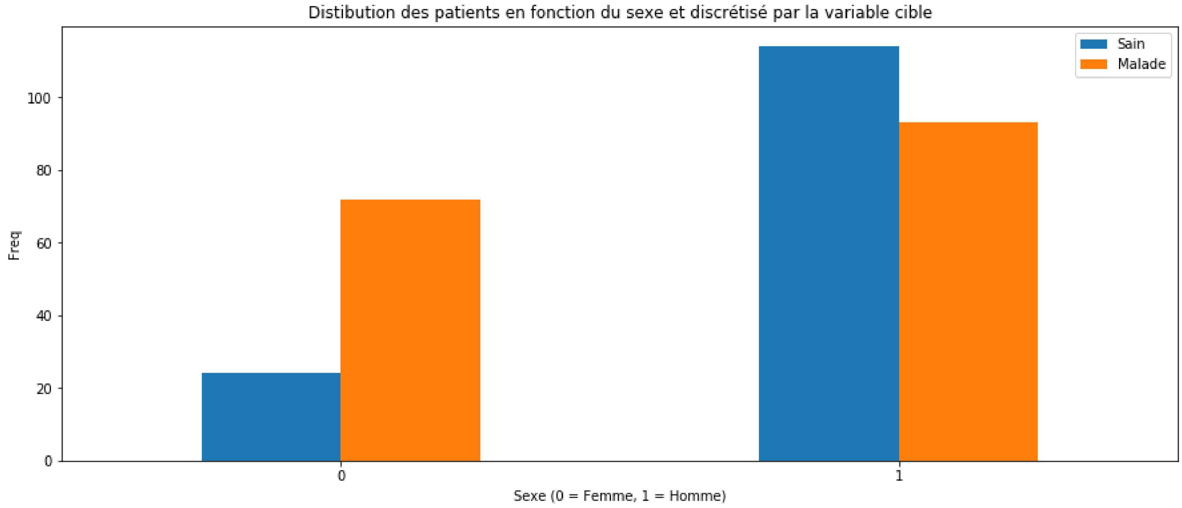
```
#Tableau croisé
pd.crosstab(df["sex"], df["target"])
```

Out[14]:

target	0	1
sex		
0	24	72
1	114	93

In [15]:

```
pd.crosstab(df.sex,df.target).plot(kind="bar",figsize=(15,6))
plt.title('Distibution des patients en fonction du sexe et discrétisé par la variable cible
plt.xlabel('Sexe (0 = Femme, 1 = Homme)')
plt.xticks(rotation=0)
plt.legend(["Sain", "Malade"])
plt.ylabel('Freq')
plt.show()
```



In [16]:

```
#Exemple de double condition
df_femme_pain0 = df [(df["sex"] == 0) & (df['chest_pain_type']==0)]
```

In [17]:

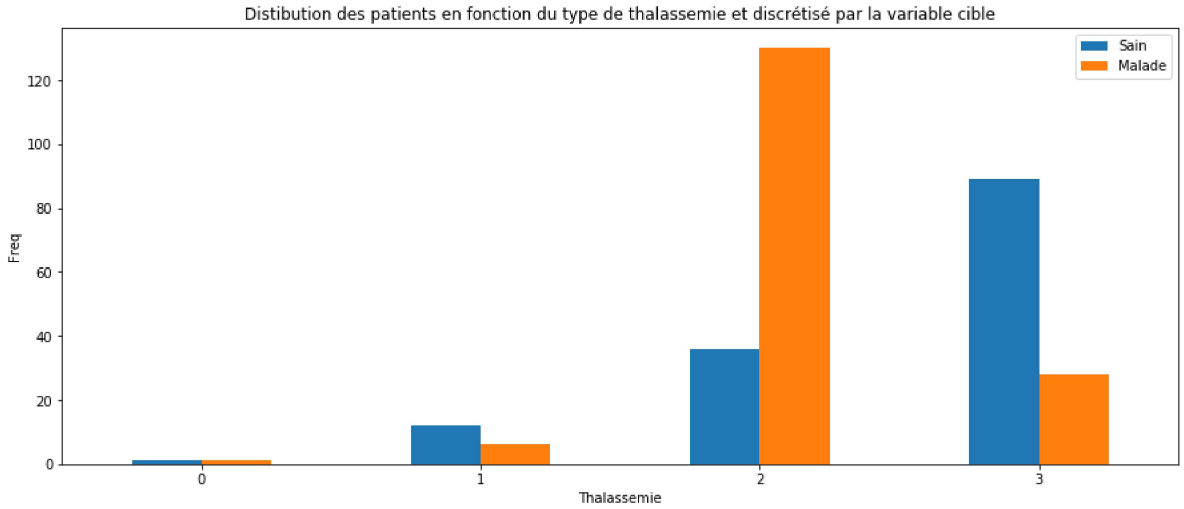
```
df_femme_pain0.head(10)
```

Out[17]:

	age	sex	chest_pain_type	resting_blood_pressure	cholesterol	fasting_blood_sugar	rest_
4	57	0	0	120	354	0	
43	53	0	0	130	264	0	
49	53	0	0	138	234	0	
59	57	0	0	128	303	0	
65	35	0	0	138	183	0	
69	62	0	0	124	209	0	
84	42	0	0	102	265	0	
89	58	0	0	100	248	0	
96	62	0	0	140	394	0	
107	45	0	0	138	236	0	

In [18]:

```
pd.crosstab(df.thalassemia, df.target).plot(kind="bar",figsize=(15,6))
plt.title('Distibution des patients en fonction du type de thalassemie et discrétisé par la
plt.xlabel('Thalassemie')
plt.xticks(rotation=0)
plt.legend(["Sain", "Malade"])
plt.ylabel('Freq')
plt.show()
```

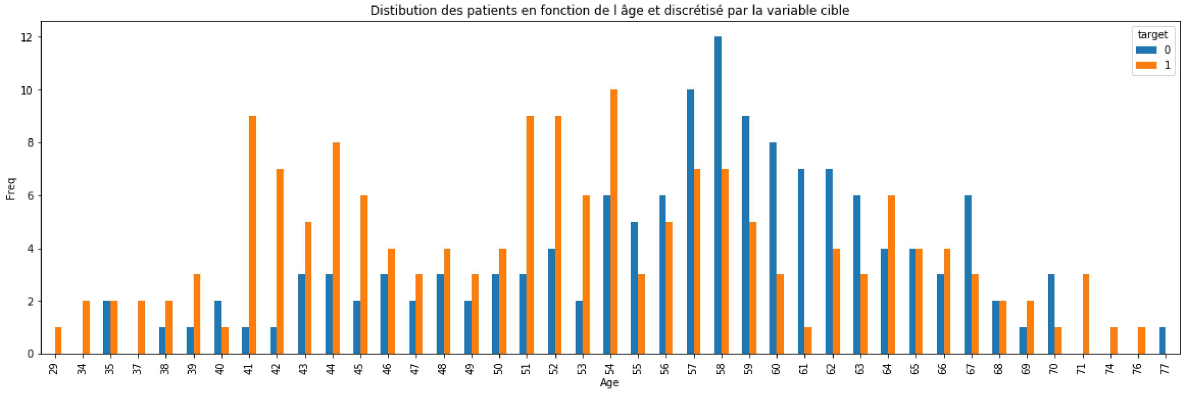


Analyse univariable de l'âge

In [19]:

```
pd.crosstab(df.age,df.target).plot(kind="bar",figsize=(20,6))
plt.title('Distibution des patients en fonction de l âge et discrétisé par la variable cibl
plt.xlabel('Age')
plt.ylabel('Freq')
plt.show()

#plt.savefig('heartDiseaseAndAges.png')
```



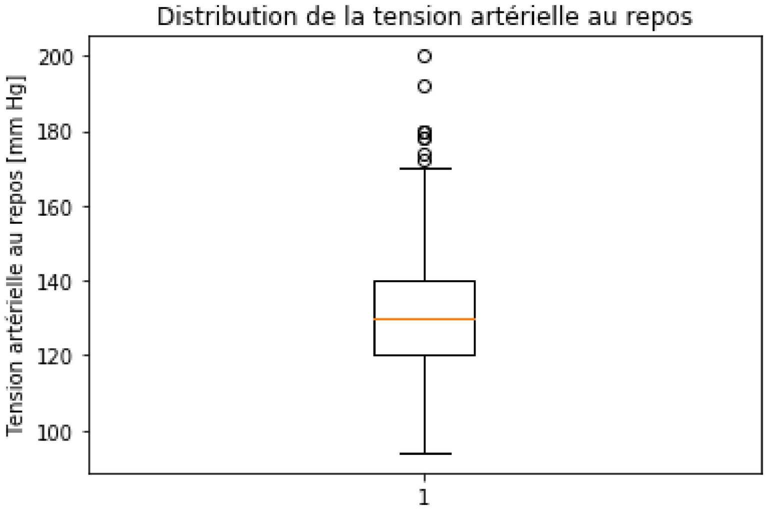
Analyse univariable de la tension artérielle au repos

In [20]:

```
#Variable resting_blood_pressure
bx = plt.boxplot(df['resting_blood_pressure'])
plt.ylabel('Tension artérielle au repos [mm Hg]')
plt.title('Distribution de la tension artérielle au repos')
```

Out[20]:

```
Text(0.5, 1.0, 'Distribution de la tension artérielle au repos')
```



Il y a quelques patients qui ont une tension artérielle au repos "abérrante".

Est-ce que ces tensions anormalement très élevé auraient un impact sur la variable cible ?

```
In [21]:
```

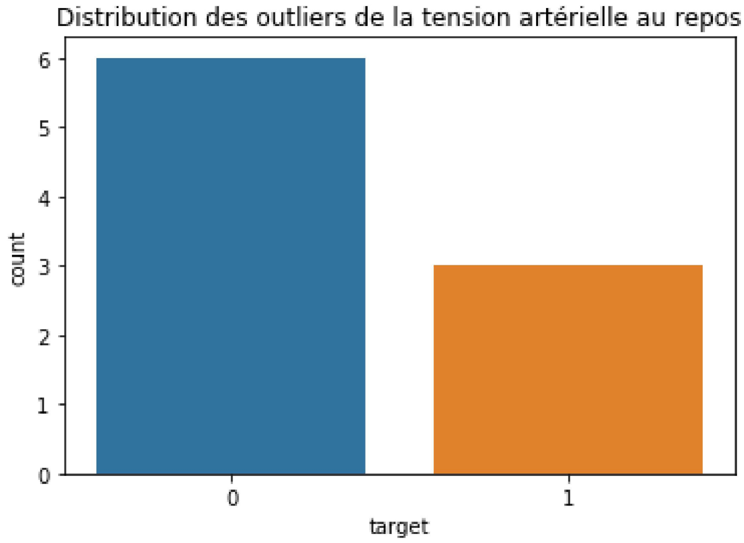
```
#Récupérer les outliers
seuil = bx['whiskers'][1]._yorig[1]
outliers = df[df["resting_blood_pressure"]> seuil]
print(len(outliers))

#Figure
sns.countplot(x='target', data=outliers)
plt.title('Distribution des outliers de la tension artérielle au repos')
```

```
9
```

```
Out[21]:
```

```
Text(0.5, 1.0, 'Distribution des outliers de la tension artérielle au repo
s')
```



Nous analysons plus particulièrement ces 9 outliers, il s'avère que 6 sont des patients sains et 3 sont des patients atteints de maladie cardiovasculaire.

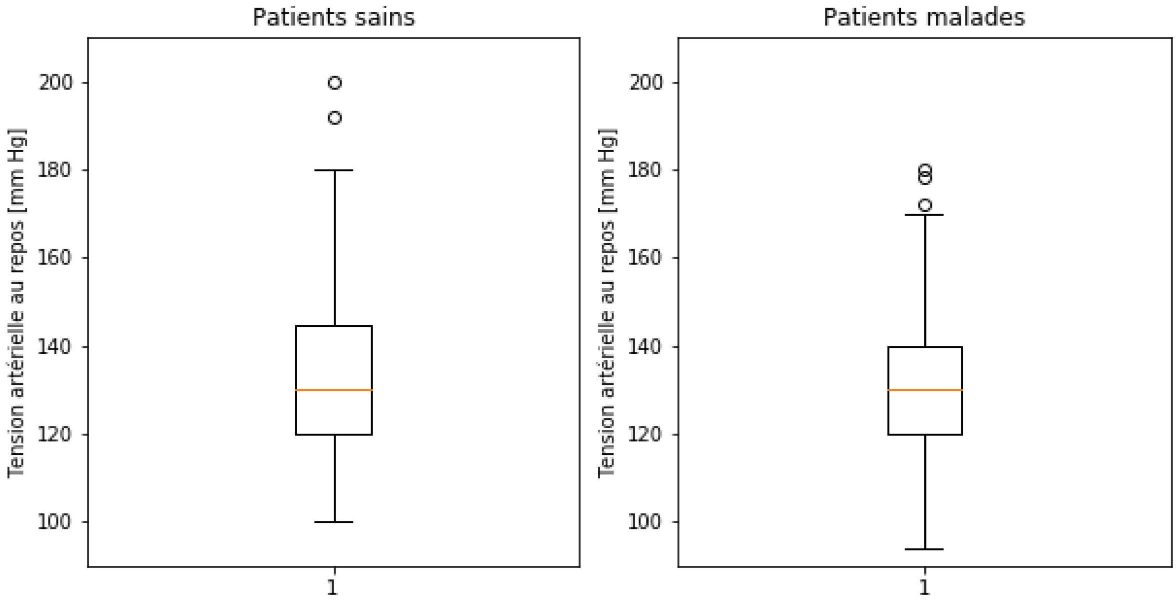
In [22]:

```
#Figure distribution des tension artérielles au repos discrétisé par variable cible
plt.figure(figsize=(10,5))
plt.subplot(1,2,1)
plt.boxplot(df[df['target']==0]['resting_blood_pressure'])
plt.ylim([90,210])
plt.ylabel('Tension artérielle au repos [mm Hg]')
plt.title('Patients sains')

plt.subplot(1,2,2)
plt.boxplot(df[df['target']==1]['resting_blood_pressure'])
plt.ylim([90,210])
plt.ylabel('Tension artérielle au repos [mm Hg]')
plt.title('Patients malades')
```

Out[22]:

```
Text(0.5, 1.0, 'Patients malades')
```

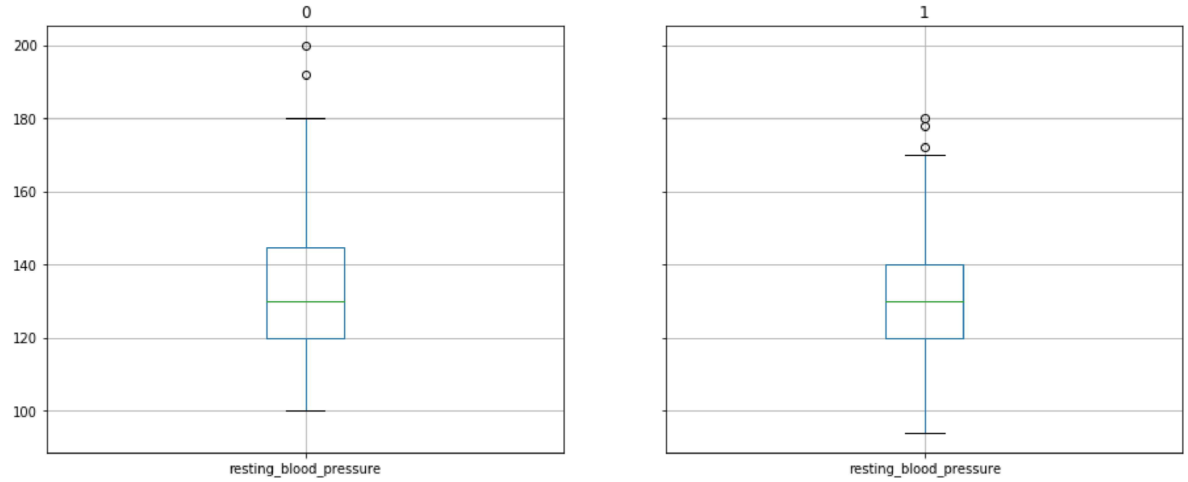


In [23]:

```
#Equivalent
df.groupby('target').boxplot(column='resting_blood_pressure', figsize=(15,6))
```

Out[23]:

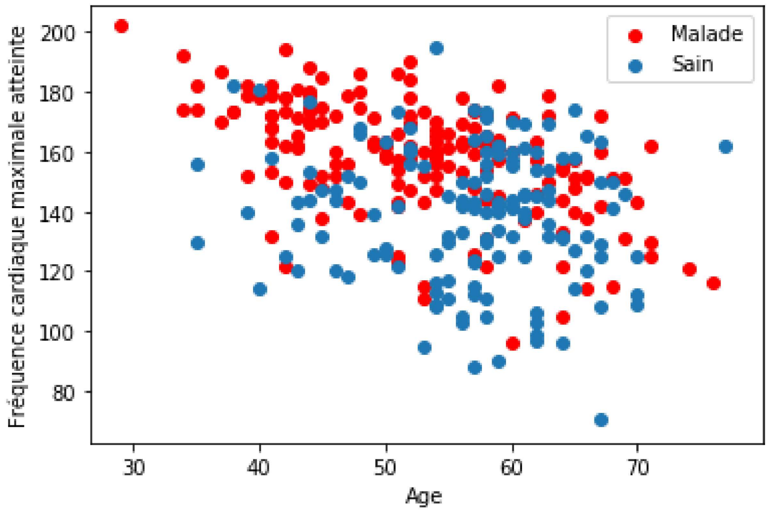
```
0         AxesSubplot(0.1,0.15;0.363636x0.75)
1    AxesSubplot(0.536364,0.15;0.363636x0.75)
dtype: object
```



Analyse bivariable : age et fréquence cardiaque maximale atteinte

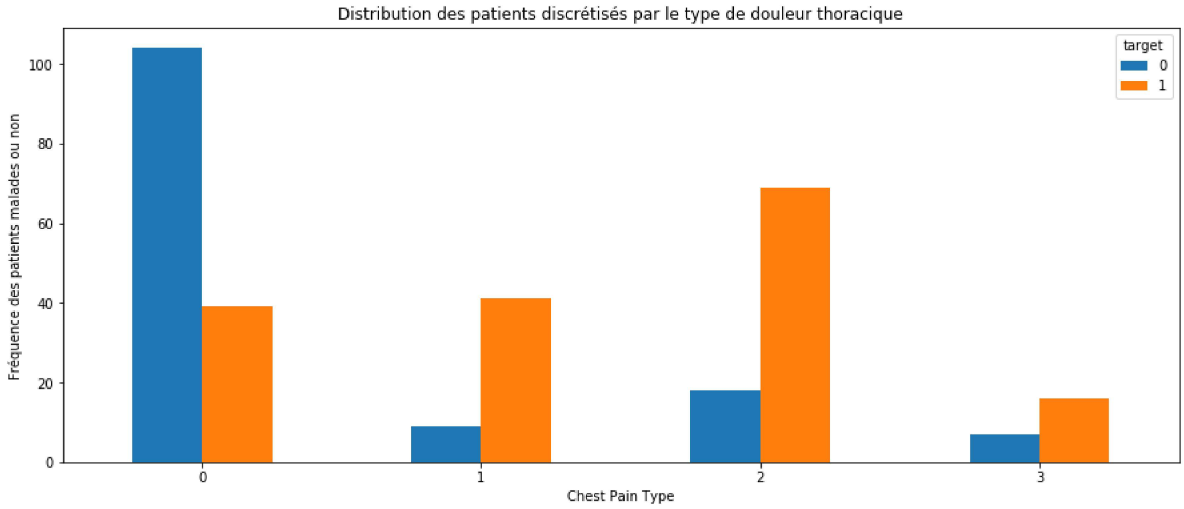
In [24]:

```
plt.scatter(x=df.age[df.target==1], y=df.max_heart_rate_achieved[(df.target==1)], c="red")
plt.scatter(x=df.age[df.target==0], y=df.max_heart_rate_achieved[(df.target==0)])
plt.legend(["Malade", "Sain"])
plt.xlabel("Age")
plt.ylabel("Fréquence cardiaque maximale atteinte")
plt.show()
```



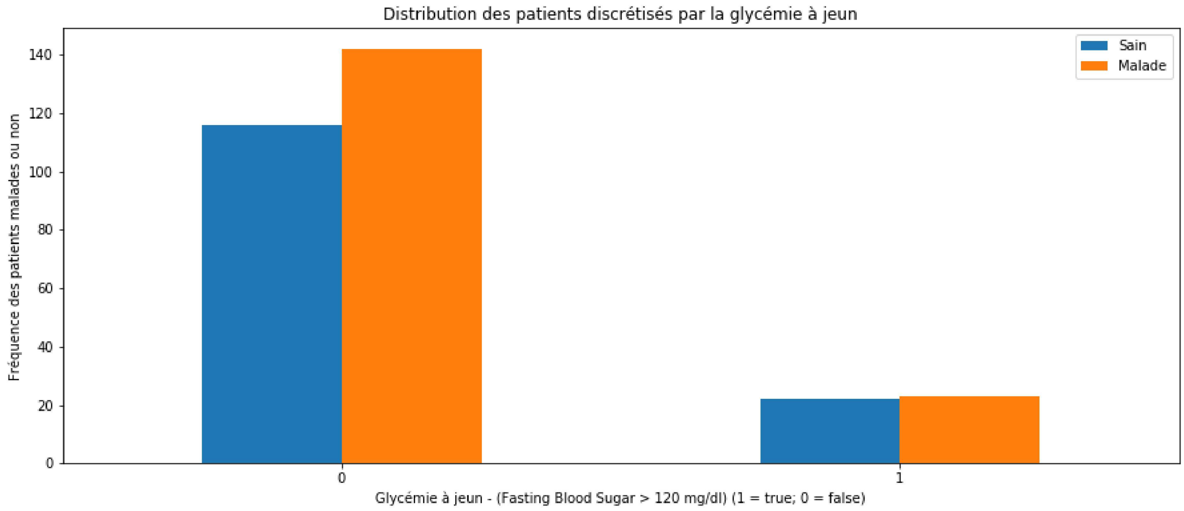
In [25]:

```
pd.crosstab(df.chest_pain_type,df.target).plot(kind="bar",figsize=(15,6))
plt.title('Distribution des patients discrétisés par le type de douleur thoracique')
plt.xlabel('Chest Pain Type')
plt.xticks(rotation = 0)
plt.ylabel('Fréquence des patients malades ou non')
plt.show()
```



In [26]:

```
pd.crosstab(df.fasting_blood_sugar, df.target).plot(kind="bar",figsize=(15,6))
plt.title('Distribution des patients discrétisés par la glycémie à jeun')
plt.xlabel('Glycémie à jeun - (Fasting Blood Sugar > 120 mg/dl) (1 = true; 0 = false)')
plt.xticks(rotation = 0)
plt.legend(["Sain", "Malade"])
plt.ylabel('Fréquence des patients malades ou non')
plt.show()
```

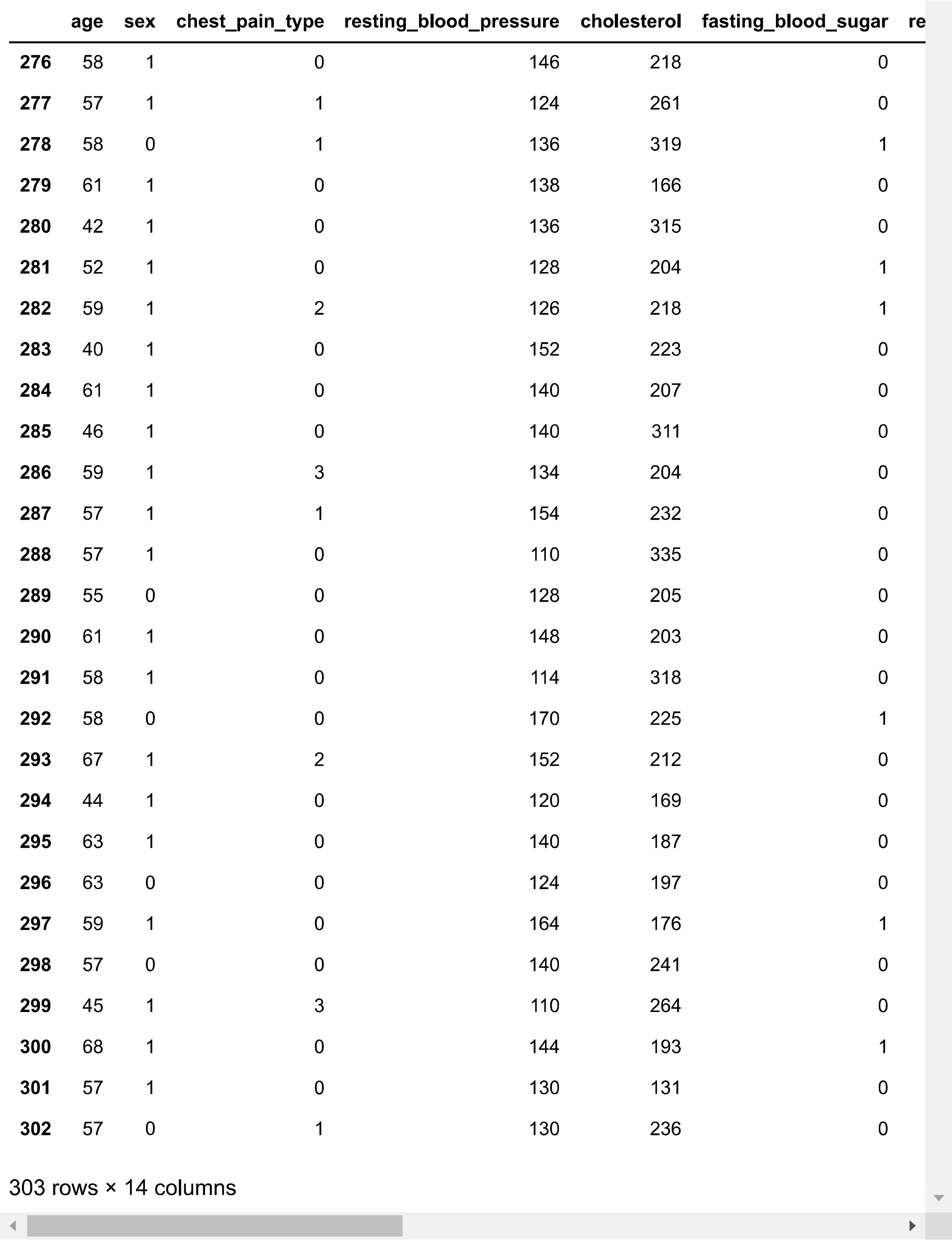


In [27]:

```
df
```

Out[27]:

	age	sex	chest_pain_type	resting_blood_pressure	cholesterol	fasting_blood_sugar	re
0	63	1	3	145	233	1	
1	37	1	2	130	250	0	
2	41	0	1	130	204	0	
3	56	1	1	120	236	0	
4	57	0	0	120	354	0	
5	57	1	0	140	192	0	
6	56	0	1	140	294	0	
7	44	1	1	120	263	0	
8	52	1	2	172	199	1	
9	57	1	2	150	168	0	
10	54	1	0	140	239	0	
11	48	0	2	130	275	0	
12	49	1	1	130	266	0	
13	64	1	3	110	211	0	
14	58	0	3	150	283	1	
15	50	0	2	120	219	0	
16	58	0	2	120	340	0	
17	66	0	3	150	226	0	
18	43	1	0	150	247	0	
19	69	0	3	140	239	0	
20	59	1	0	135	234	0	
21	44	1	2	130	233	0	
22	42	1	0	140	226	0	
23	61	1	2	150	243	1	
24	40	1	3	140	199	0	
25	71	0	1	160	302	0	
26	59	1	2	150	212	1	
27	51	1	2	110	175	0	
28	65	0	2	140	417	1	
29	53	1	2	130	197	1	
...	...	...	...	...	...	...	
273	58	1	0	100	234	0	
274	47	1	0	110	275	0	
275	52	1	0	125	212	0	



	age	sex	chest_pain_type	resting_blood_pressure	cholesterol	fasting_blood_sugar	re
276	58	1	0	146	218	0	
277	57	1	1	124	261	0	
278	58	0	1	136	319	1	
279	61	1	0	138	166	0	
280	42	1	0	136	315	0	
281	52	1	0	128	204	1	
282	59	1	2	126	218	1	
283	40	1	0	152	223	0	
284	61	1	0	140	207	0	
285	46	1	0	140	311	0	
286	59	1	3	134	204	0	
287	57	1	1	154	232	0	
288	57	1	0	110	335	0	
289	55	0	0	128	205	0	
290	61	1	0	148	203	0	
291	58	1	0	114	318	0	
292	58	0	0	170	225	1	
293	67	1	2	152	212	0	
294	44	1	0	120	169	0	
295	63	1	0	140	187	0	
296	63	0	0	124	197	0	
297	59	1	0	164	176	1	
298	57	0	0	140	241	0	
299	45	1	3	110	264	0	
300	68	1	0	144	193	1	
301	57	1	0	130	131	0	
302	57	0	1	130	236	0	

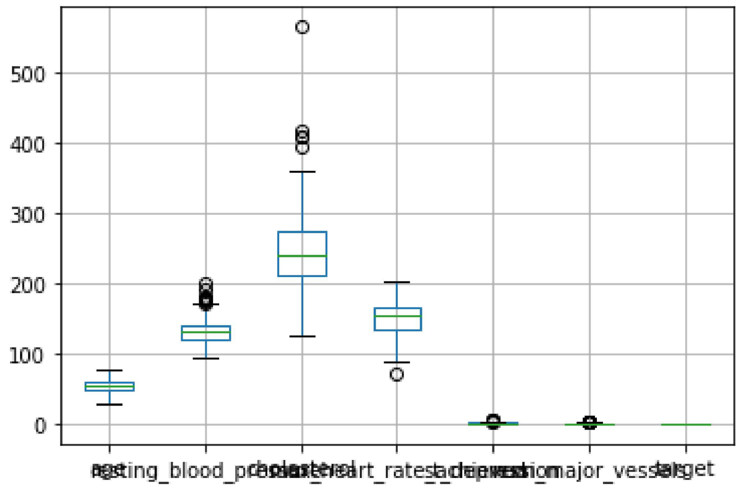
303 rows × 14 columns

In [28]:

```
#Boites à moustaches
df.boxplot()
```

Out[28]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x13d8b36e860>
```


age
resting_blood_p
cholesterol
heart_rate
depression
major_vess
target