```
import pandas as pd
import seaborn as sb
import numpy as np
import matplotlib as plt
from sklearn.model_selection import train_test_split as tts
from sklearn.linear_model import LogisticRegression
from sklearn import metrics

heart = pd.read_csv("Heart_dataset.csv")

heart
```

```
     Unnamed: 0  Age  Sex     ChestPain  RestBP  Chol  Fbs  RestECG
MaxHR  \
0              1   63    1       typical     145   233    1        2
150
1              2   67    1  asymptomatic     160   286    0        2
108
2              3   67    1  asymptomatic     120   229    0        2
129
3              4   37    1    nonanginal     130   250    0        0
187
4              5   41    0    nontypical     130   204    0        2
172
..           ...  ...  ...           ...     ...   ...  ...      ...
...
298          299   45    1       typical     110   264    0        0
132
299          300   68    1  asymptomatic     144   193    1        0
141
300          301   57    1  asymptomatic     130   131    0        0
115
301          302   57    0    nontypical     130   236    0        2
174
302          303   38    1    nonanginal     138   175    0        0
173

     ExAng  Oldpeak  Slope   Ca        Thal  AHD
0        0      2.3      3  0.0       fixed   No
1        1      1.5      2  3.0      normal  Yes
2        1      2.6      2  2.0  reversable  Yes
3        0      3.5      3  0.0      normal   No
4        0      1.4      1  0.0      normal   No
..     ...      ...    ...  ...         ...  ...
298      0      1.2      2  0.0  reversable  Yes
299      0      3.4      2  2.0  reversable  Yes
300      1      1.2      2  1.0  reversable  Yes
301      0      0.0      2  1.0      normal  Yes
302      0      0.0      1  NaN      normal   No
```

```
[303 rows x 15 columns]
```

```
heart = heart.drop(columns=['Unnamed: 0'])
```

```
heart
```

```
     Age  Sex     ChestPain  RestBP  Chol  Fbs  RestECG  MaxHR  ExAng
\
0     63    1       typical     145   233    1        2    150      0

1     67    1  asymptomatic     160   286    0        2    108      1

2     67    1  asymptomatic     120   229    0        2    129      1

3     37    1    nonanginal     130   250    0        0    187      0

4     41    0    nontypical     130   204    0        2    172      0

..   ...  ...           ...     ...   ...  ...      ...    ...    ...

298   45    1       typical     110   264    0        0    132      0

299   68    1  asymptomatic     144   193    1        0    141      0

300   57    1  asymptomatic     130   131    0        0    115      1

301   57    0    nontypical     130   236    0        2    174      0

302   38    1    nonanginal     138   175    0        0    173      0


     Oldpeak  Slope   Ca        Thal  AHD
0        2.3      3  0.0       fixed   No
1        1.5      2  3.0      normal  Yes
2        2.6      2  2.0  reversable  Yes
3        3.5      3  0.0      normal   No
4        1.4      1  0.0      normal   No
..       ...    ...  ...         ...  ...
298      1.2      2  0.0  reversable  Yes
299      3.4      2  2.0  reversable  Yes
300      1.2      2  1.0  reversable  Yes
301      0.0      2  1.0      normal  Yes
302      0.0      1  NaN      normal   No

[303 rows x 14 columns]
```

```
heart['AHD'] = np.where(heart["AHD"] == "Yes", 1, 0)
heart
```

```
     Age  Sex     ChestPain  RestBP  Chol  Fbs  RestECG  MaxHR  ExAng
\
```

```
0     63    1       typical     145   233    1        2    150      0

1     67    1  asymptomatic     160   286    0        2    108      1

2     67    1  asymptomatic     120   229    0        2    129      1

3     37    1    nonanginal     130   250    0        0    187      0

4     41    0    nontypical     130   204    0        2    172      0

..   ...  ...           ...     ...   ...  ...      ...    ...    ...

298   45    1       typical     110   264    0        0    132      0

299   68    1  asymptomatic     144   193    1        0    141      0

300   57    1  asymptomatic     130   131    0        0    115      1

301   57    0    nontypical     130   236    0        2    174      0

302   38    1    nonanginal     138   175    0        0    173      0


     Oldpeak  Slope   Ca        Thal  AHD
0        2.3      3  0.0       fixed    0
1        1.5      2  3.0      normal    1
2        2.6      2  2.0  reversable    1
3        3.5      3  0.0      normal    0
4        1.4      1  0.0      normal    0
..       ...    ...  ...         ...  ...
298      1.2      2  0.0  reversable    1
299      3.4      2  2.0  reversable    1
300      1.2      2  1.0  reversable    1
301      0.0      2  1.0      normal    1
302      0.0      1  NaN      normal    0

[303 rows x 14 columns]
```

```
heart['ChestPain'] = heart['ChestPain'].replace(['typical',
'asymptomatic'], [0, 1])

heart['ChestPain'] = heart['ChestPain'].replace(['nonanginal',
'nontypical'], [2, 3])

heart
```

```
     Age  Sex  ChestPain  RestBP  Chol  Fbs  RestECG  MaxHR  ExAng
Oldpeak  \
0     63    1          0     145   233    1        2    150      0
2.3
1     67    1          1     160   286    0        2    108      1
```

```
1.5
2     67    1          1     120   229    0        2    129       1
2.6
3     37    1          2     130   250    0        0    187       0
3.5
4     41    0          3     130   204    0        2    172       0
1.4
..   ...   ...        ...     ...   ...   ...      ...    ...     ...
...
298   45    1          0     110   264    0        0    132       0
1.2
299   68    1          1     144   193    1        0    141       0
3.4
300   57    1          1     130   131    0        0    115       1
1.2
301   57    0          3     130   236    0        2    174       0
0.0
302   38    1          2     138   175    0        0    173       0
0.0

     Slope   Ca        Thal  AHD
0        3  0.0       fixed    0
1        2  3.0      normal    1
2        2  2.0  reversable    1
3        3  0.0      normal    0
4        1  0.0      normal    0
..     ...  ...         ...  ...
298      2  0.0  reversable    1
299      2  2.0  reversable    1
300      2  1.0  reversable    1
301      2  1.0      normal    1
302      1  NaN      normal    0

[303 rows x 14 columns]
```

```
heart['Thal'] = heart['Thal'].replace(['fixed', 'normal',
'reversable'], [0, 1, 2])
```

```
heart
```

```
     Age  Sex  ChestPain  RestBP  Chol  Fbs  RestECG  MaxHR  ExAng
Oldpeak  \
0     63    1          0     145   233    1        2    150       0
2.3
1     67    1          1     160   286    0        2    108       1
1.5
2     67    1          1     120   229    0        2    129       1
2.6
3     37    1          2     130   250    0        0    187       0
3.5
```

```
4     41    0          3     130   204    0        2     172      0
1.4
..   ...  ...        ...     ...   ...  ...      ...     ...    ...
...
298   45    1          0     110   264    0        0     132      0
1.2
299   68    1          1     144   193    1        0     141      0
3.4
300   57    1          1     130   131    0        0     115      1
1.2
301   57    0          3     130   236    0        2     174      0
0.0
302   38    1          2     138   175    0        0     173      0
0.0

     Slope   Ca  Thal  AHD
0        3  0.0   0.0    0
1        2  3.0   1.0    1
2        2  2.0   2.0    1
3        3  0.0   1.0    0
4        1  0.0   1.0    0
..     ...  ...   ...  ...
298      2  0.0   2.0    1
299      2  2.0   2.0    1
300      2  1.0   2.0    1
301      2  1.0   1.0    1
302      1  NaN   1.0    0

[303 rows x 14 columns]
```

```
heart.dtypes
```

```
Age            int64
Sex            int64
ChestPain      int64
RestBP         int64
Chol           int64
Fbs            int64
RestECG        int64
MaxHR          int64
ExAng          int64
Oldpeak      float64
Slope          int64
Ca           float64
Thal         float64
AHD            int64
dtype: object
```

```
heart.isnull().sum()
```

```
Age          0
Sex          0
ChestPain    0
RestBP       0
Chol         0
Fbs          0
RestECG      0
MaxHR        0
ExAng        0
Oldpeak      0
Slope        0
Ca           4
Thal         2
AHD          0
dtype: int64
```

```
heart.mean()
```

```
Age           54.438944
Sex            0.679868
ChestPain      1.537954
RestBP       131.689769
Chol         246.693069
Fbs            0.148515
RestECG        0.990099
MaxHR        149.607261
ExAng          0.326733
Oldpeak        1.039604
Slope          1.600660
Ca             0.672241
Thal           1.328904
AHD            0.458746
dtype: float64
```

```
ca=heart["Ca"].mean()                  #mean of individual  column ca
ca
```

```
0.6722408026755853
```

```
heart["Ca"].fillna(ca, inplace = True)      # h) replace missing data
heart
```

```
     Age  Sex  ChestPain  RestBP  Chol  Fbs  RestECG  MaxHR  ExAng
Oldpeak  \
0     63    1          0     145   233    1        2    150      0
2.3
1     67    1          1     160   286    0        2    108      1
1.5
2     67    1          1     120   229    0        2    129      1
```

```
2.6
3     37    1           2     130   250    0         0    187       0
3.5
4     41    0           3     130   204    0         2    172       0
1.4
..   ...   ...         ...     ...   ...   ...       ...    ...     ...
...
298   45    1           0     110   264    0         0    132       0
1.2
299   68    1           1     144   193    1         0    141       0
3.4
300   57    1           1     130   131    0         0    115       1
1.2
301   57    0           3     130   236    0         2    174       0
0.0
302   38    1           2     138   175    0         0    173       0
0.0

     Slope        Ca  Thal  AHD
0        3  0.000000   0.0    0
1        2  3.000000   1.0    1
2        2  2.000000   2.0    1
3        3  0.000000   1.0    0
4        1  0.000000   1.0    0
..     ...       ...   ...  ...
298      2  0.000000   2.0    1
299      2  2.000000   2.0    1
300      2  1.000000   2.0    1
301      2  1.000000   1.0    1
302      1  0.672241   1.0    0

[303 rows x 14 columns]

th=heart["Thal"].mean()                  #mean of individual  column ca
th

1.3289036544850499

heart["Thal"].fillna(th, inplace = True)      # h) replace missing
data
heart

     Age  Sex  ChestPain  RestBP  Chol  Fbs  RestECG  MaxHR  ExAng
Oldpeak  \
0     63    1           0     145   233    1        2    150       0
2.3
1     67    1           1     160   286    0        2    108       1
1.5
2     67    1           1     120   229    0        2    129       1
2.6
```

```
3     37    1          2     130   250    0        0     187      0
3.5
4     41    0          3     130   204    0        2     172      0
1.4
..   ...   ...        ...     ...   ...   ...      ...     ...    ...
...
298   45    1          0     110   264    0        0     132      0
1.2
299   68    1          1     144   193    1        0     141      0
3.4
300   57    1          1     130   131    0        0     115      1
1.2
301   57    0          3     130   236    0        2     174      0
0.0
302   38    1          2     138   175    0        0     173      0
0.0

     Slope        Ca  Thal  AHD
0        3  0.000000   0.0    0
1        2  3.000000   1.0    1
2        2  2.000000   2.0    1
3        3  0.000000   1.0    0
4        1  0.000000   1.0    0
..     ...       ...   ...  ...
298      2  0.000000   2.0    1
299      2  2.000000   2.0    1
300      2  1.000000   2.0    1
301      2  1.000000   1.0    1
302      1  0.672241   1.0    0

[303 rows x 14 columns]
```

```
heart.isnull().sum()
```

```
Age          0
Sex          0
ChestPain    0
RestBP       0
Chol         0
Fbs          0
RestECG      0
MaxHR        0
ExAng        0
Oldpeak      0
Slope        0
Ca           0
Thal         0
```

```
AHD          0
dtype: int64
```

```
heart.duplicated().sum()
```

```
0
```

```
heart.describe()
```

```
               Age         Sex   ChestPain      RestBP        Chol
Fbs  \
count  303.000000  303.000000  303.000000  303.000000  303.000000
303.000000
mean    54.438944    0.679868    1.537954  131.689769  246.693069
0.148515
std      9.038662    0.467299    0.856053   17.599748   51.776918
0.356198
min     29.000000    0.000000    0.000000   94.000000  126.000000
0.000000
25%     48.000000    0.000000    1.000000  120.000000  211.000000
0.000000
50%     56.000000    1.000000    1.000000  130.000000  241.000000
0.000000
75%     61.000000    1.000000    2.000000  140.000000  275.000000
0.000000
max     77.000000    1.000000    3.000000  200.000000  564.000000
1.000000

          RestECG       MaxHR       ExAng     Oldpeak       Slope
Ca  \
count  303.000000  303.000000  303.000000  303.000000  303.000000
303.000000
mean     0.990099  149.607261    0.326733    1.039604    1.600660
0.672241
std      0.994971   22.875003    0.469794    1.161075    0.616226
0.931209
min      0.000000   71.000000    0.000000    0.000000    1.000000
0.000000
25%      0.000000  133.500000    0.000000    0.000000    1.000000
0.000000
50%      1.000000  153.000000    0.000000    0.800000    2.000000
0.000000
75%      2.000000  166.000000    1.000000    1.600000    2.000000
1.000000
max      2.000000  202.000000    1.000000    6.200000    3.000000
3.000000

             Thal         AHD
count  303.000000  303.000000
mean     1.328904    0.458746
```

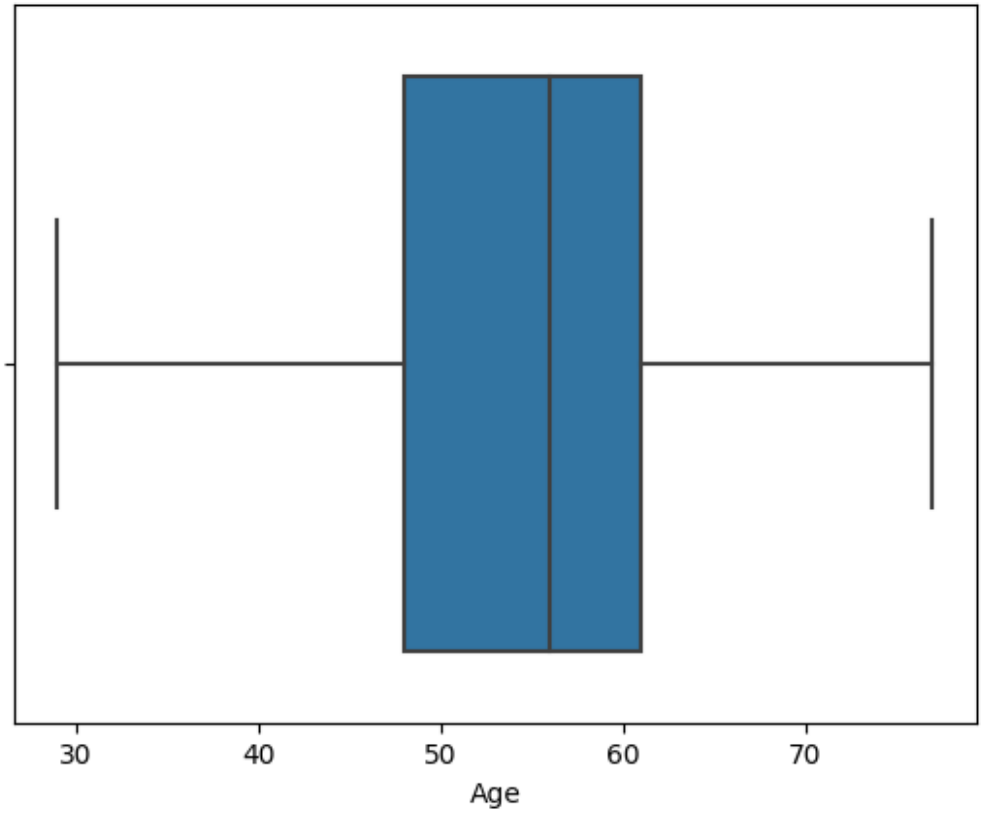
```
std       0.582409    0.499120
min       0.000000    0.000000
25%       1.000000    0.000000
50%       1.000000    0.000000
75%       2.000000    1.000000
max       2.000000    1.000000
```

```
heart.shape
```

```
(303, 14)
```

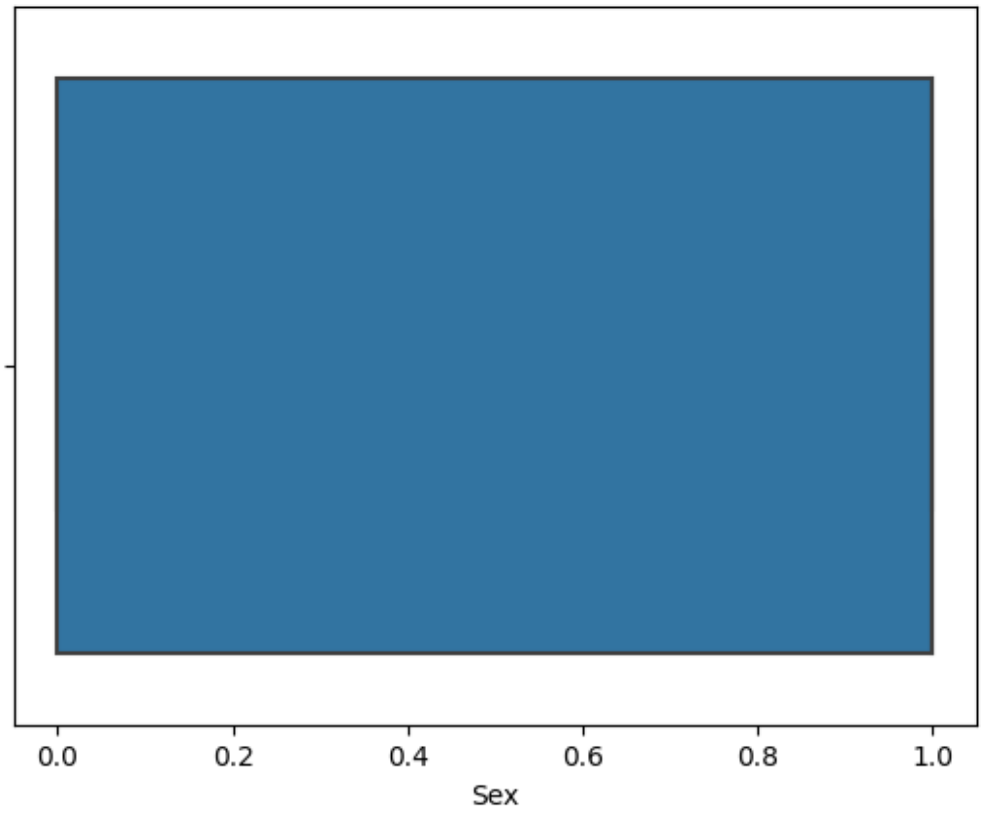
```
sb.boxplot(data=heart, x="Age")
```

```
<Axes: xlabel='Age'>
```



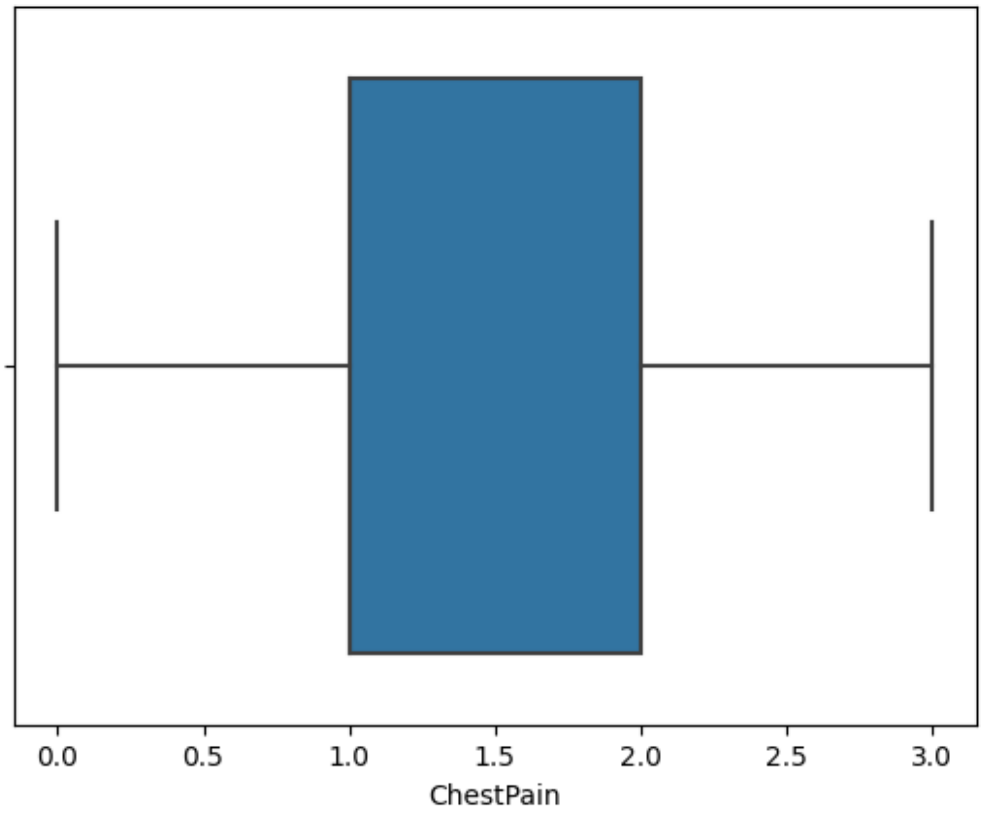

```
sb.boxplot(data=heart, x="Sex")
```

```
<Axes: xlabel='Sex'>
```



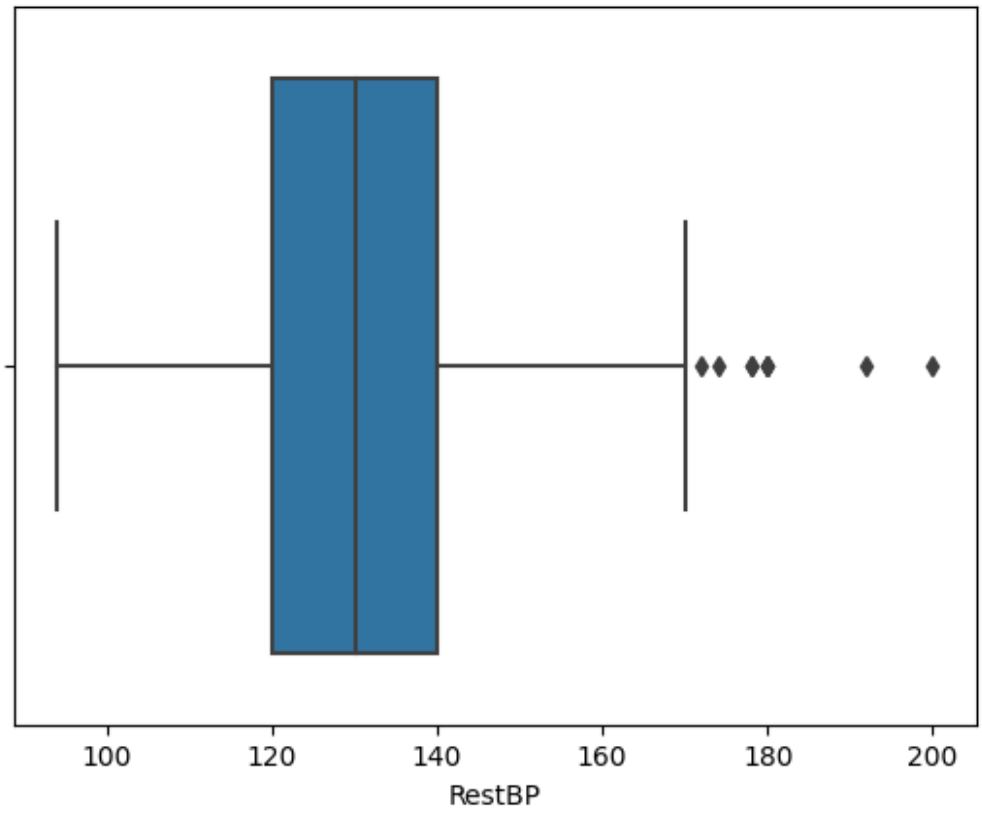

```
sb.boxplot(data=heart, x="ChestPain")
```

```
<Axes: xlabel='ChestPain'>
```



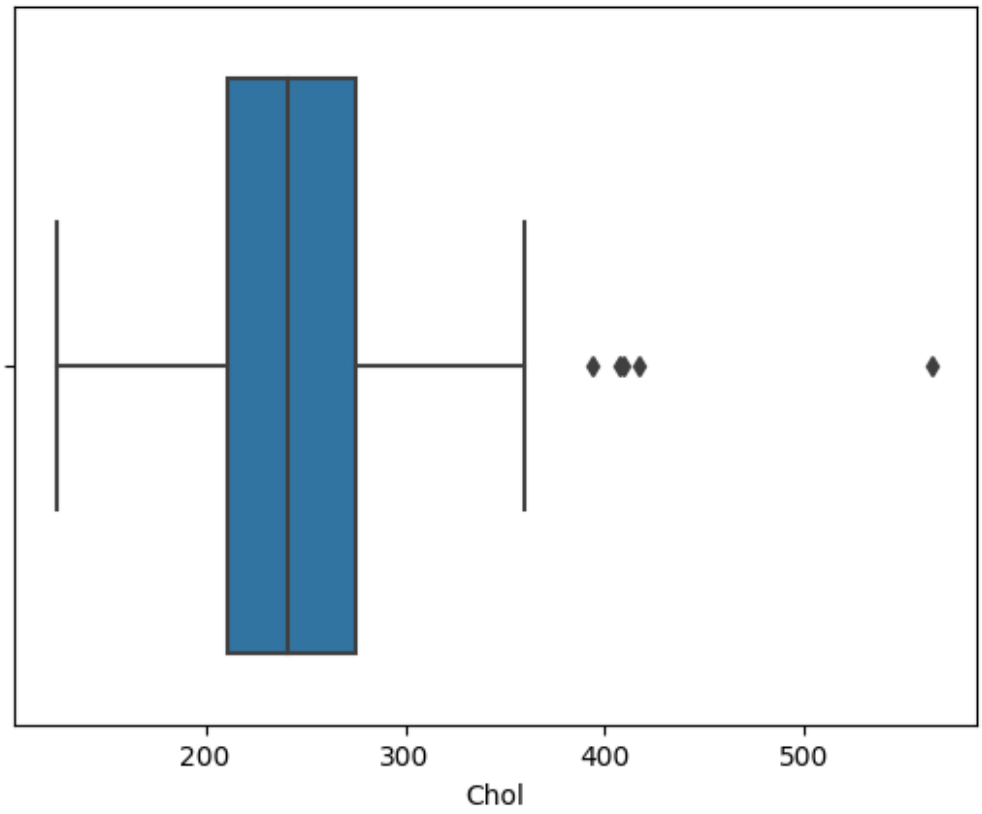

```
sb.boxplot(data=heart, x="RestBP")
```

```
<Axes: xlabel='RestBP'>
```



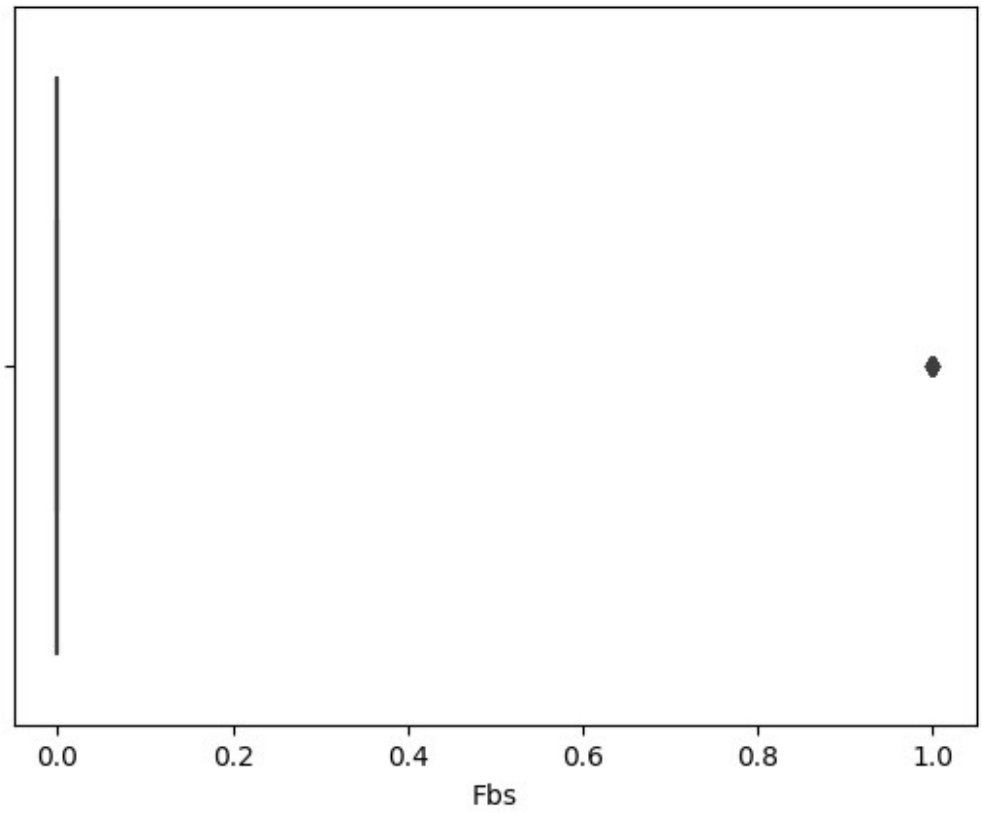

```
sb.boxplot(data=heart, x="Chol")
```

```
<Axes: xlabel='Chol'>
```



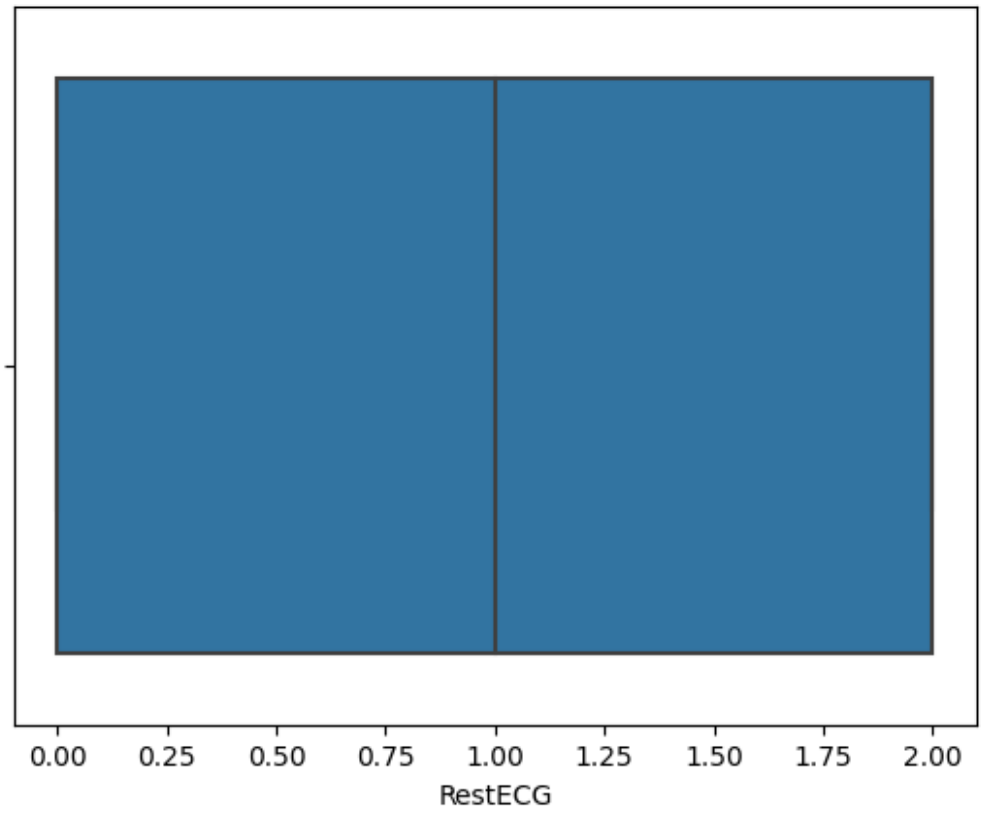

```
sb.boxplot(data=heart, x="Fbs")
```

```
<Axes: xlabel='Fbs'>
```



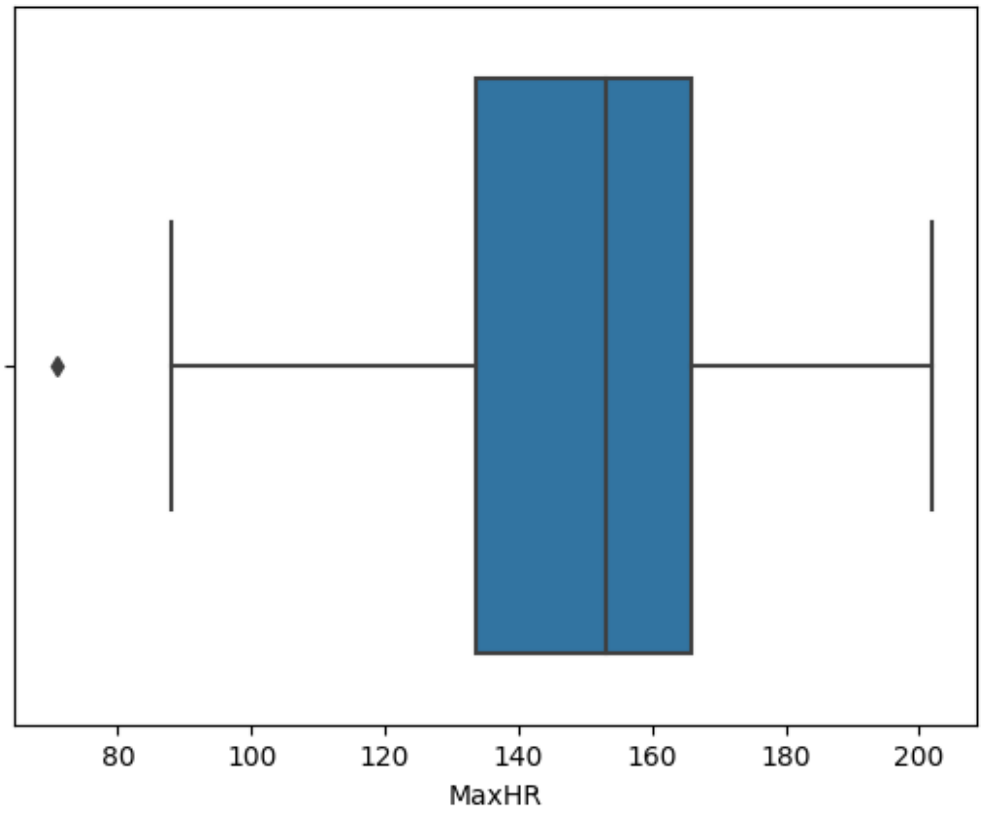

```
sb.boxplot(data=heart, x="RestECG")
```

```
<Axes: xlabel='RestECG'>
```



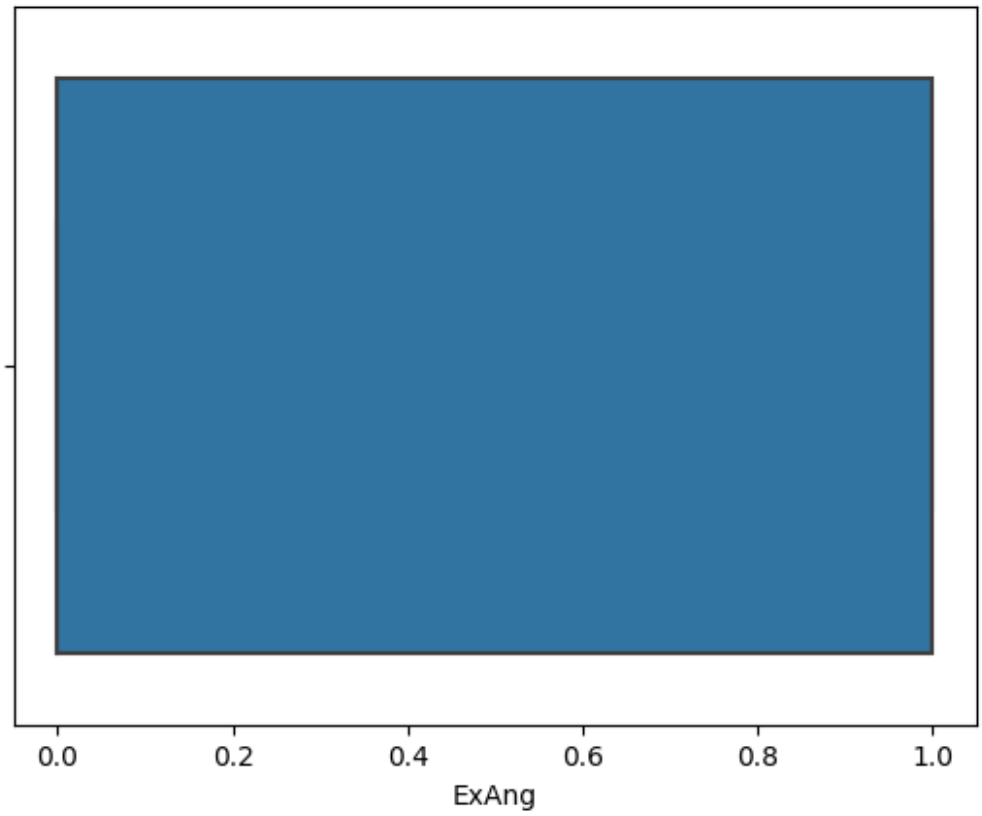

```
sb.boxplot(data=heart, x="MaxHR")
```

```
<Axes: xlabel='MaxHR'>
```



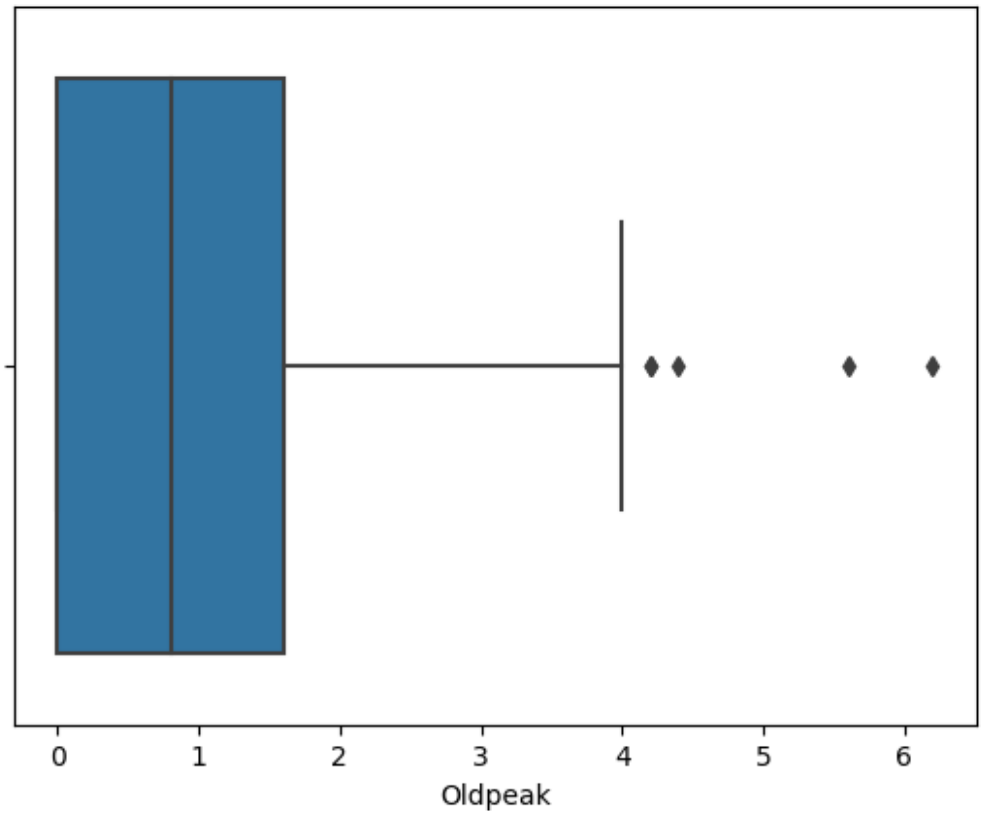

```
sb.boxplot(data=heart, x="ExAng")
```

```
<Axes: xlabel='ExAng'>
```



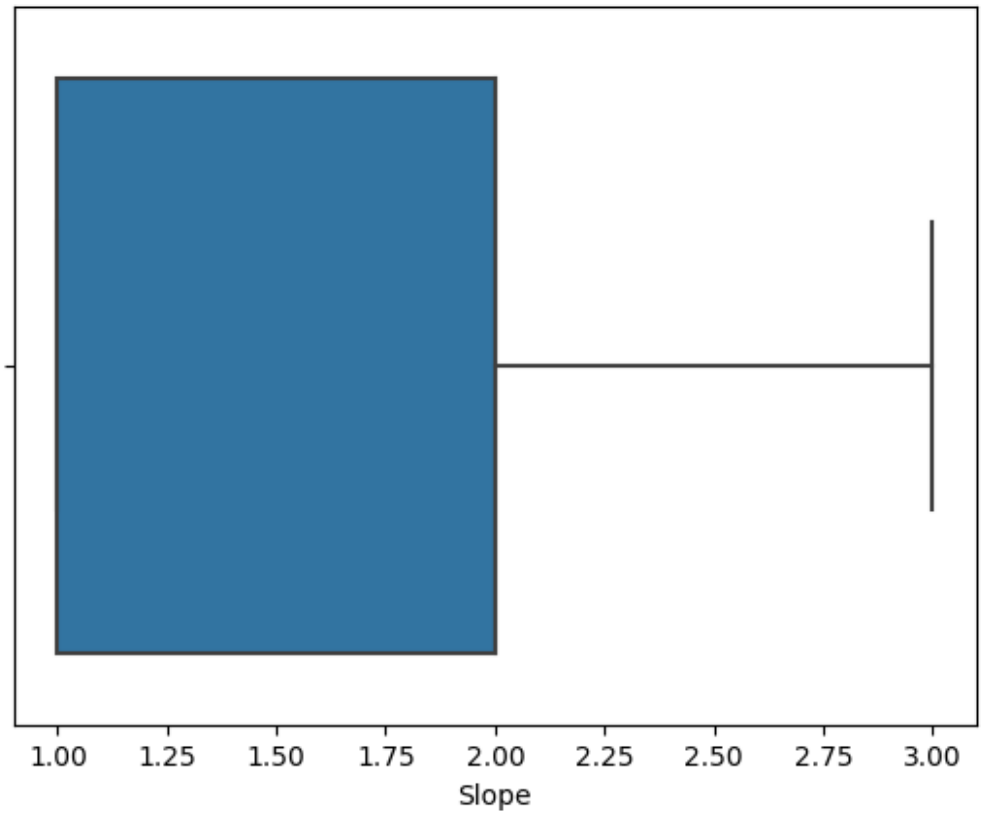

```
sb.boxplot(data=heart, x="Oldpeak")
```

```
<Axes: xlabel='Oldpeak'>
```



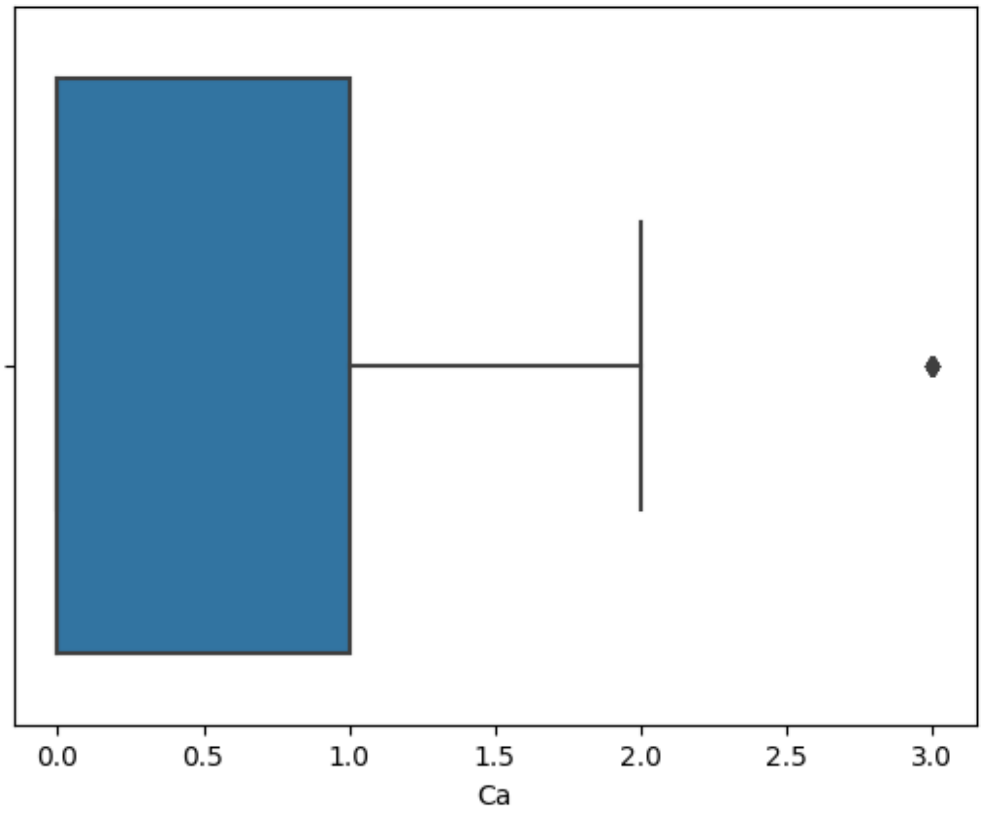

```
sb.boxplot(data=heart, x="Slope")
```

```
<Axes: xlabel='Slope'>
```



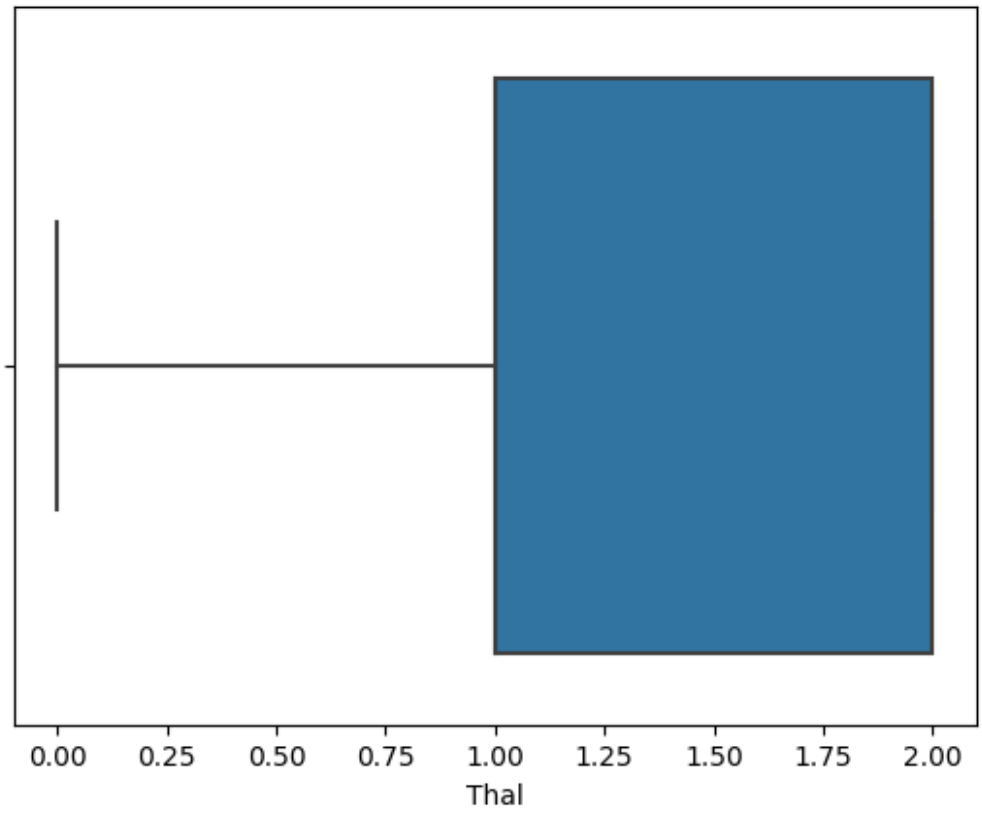

```
sb.boxplot(data=heart, x="Ca")
```

```
<Axes: xlabel='Ca'>
```



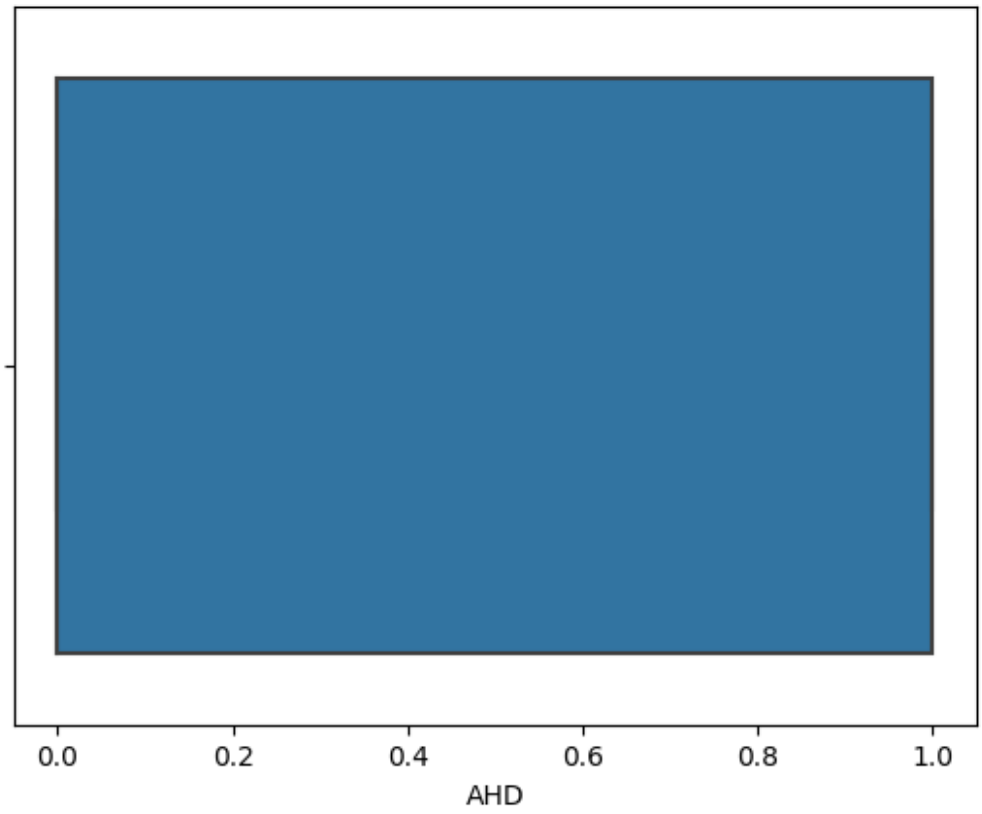

```
sb.boxplot(data=heart, x="Thal")
```

```
<Axes: xlabel='Thal'>
```



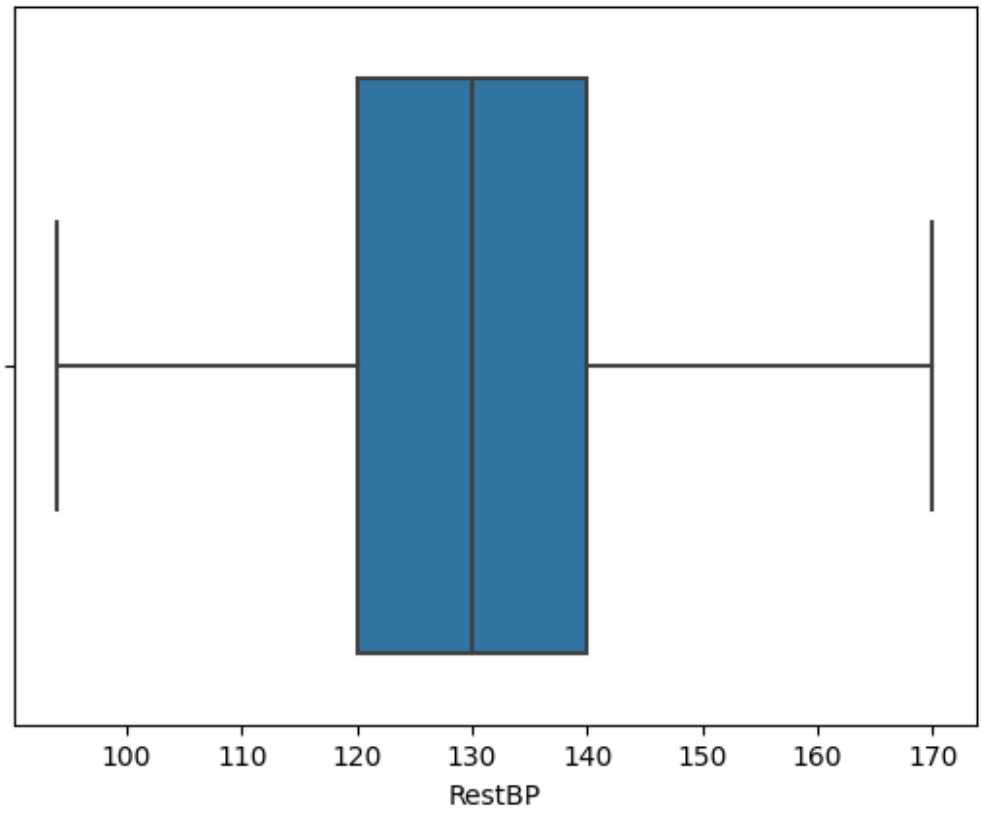

```
sb.boxplot(data=heart, x="AHD")
```

```
<Axes: xlabel='AHD'>
```



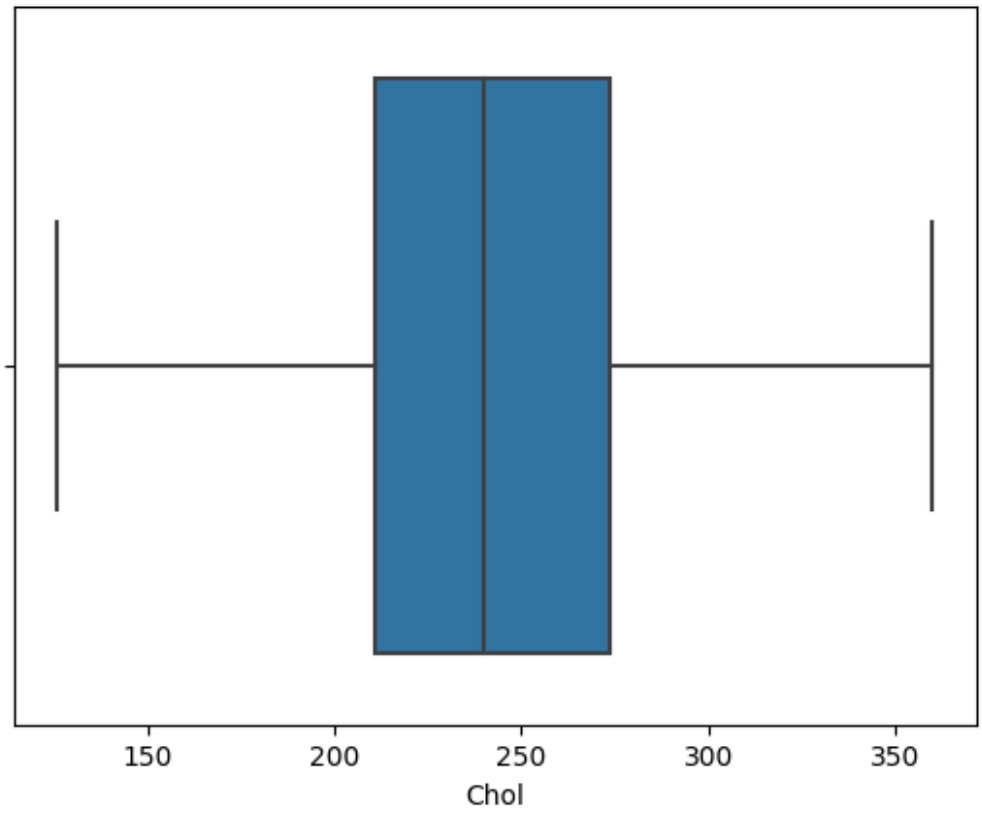

```
for x in ['RestBP']:
    q75,q25 = np.percentile(heart.loc[:,x],[75,25])
    intr_qr = q75-q25

    max = q75+(1.5*intr_qr)
    min = q25-(1.5*intr_qr)

    heart.loc[heart[x] < min,x] = np.nan
    heart.loc[heart[x] > max,x] = np.nan


sb.boxplot(data=heart, x="RestBP")
```

```
<Axes: xlabel='RestBP'>
```



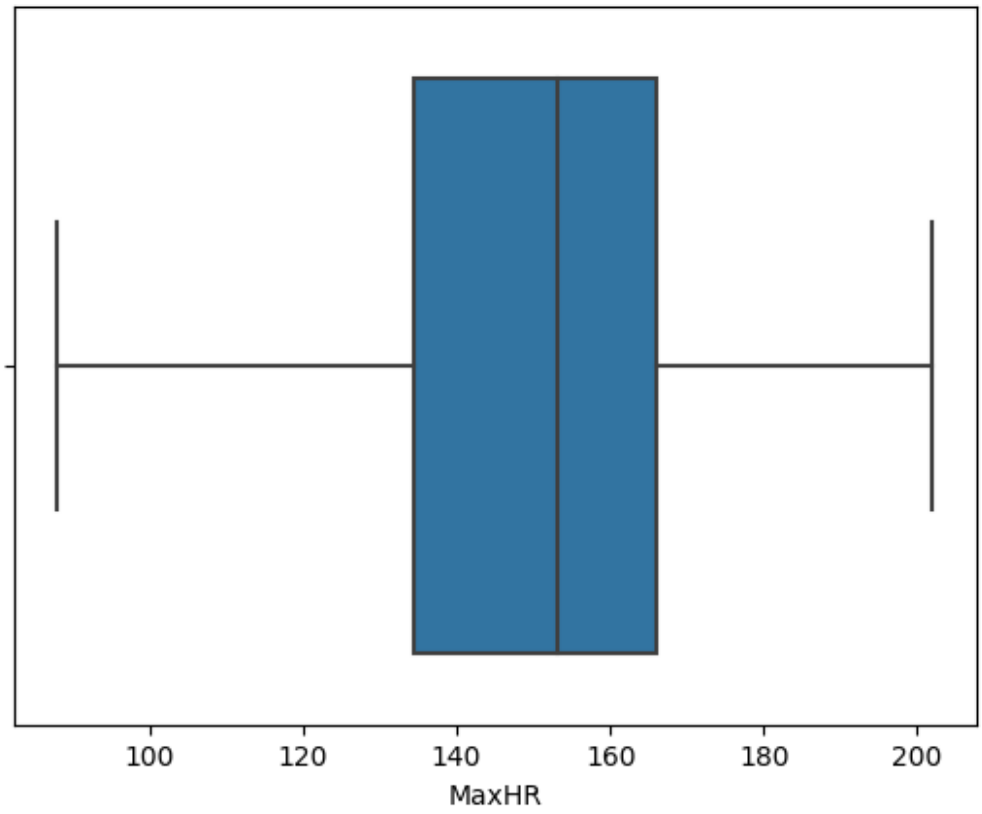

```
for x in ['Chol']:
    q75,q25 = np.percentile(heart.loc[:,x],[75,25])
    intr_qr = q75-q25

    max = q75+(1.5*intr_qr)
    min = q25-(1.5*intr_qr)

    heart.loc[heart[x] < min,x] = np.nan
    heart.loc[heart[x] > max,x] = np.nan

sb.boxplot(data=heart, x="Chol")
```

```
<Axes: xlabel='Chol'>
```



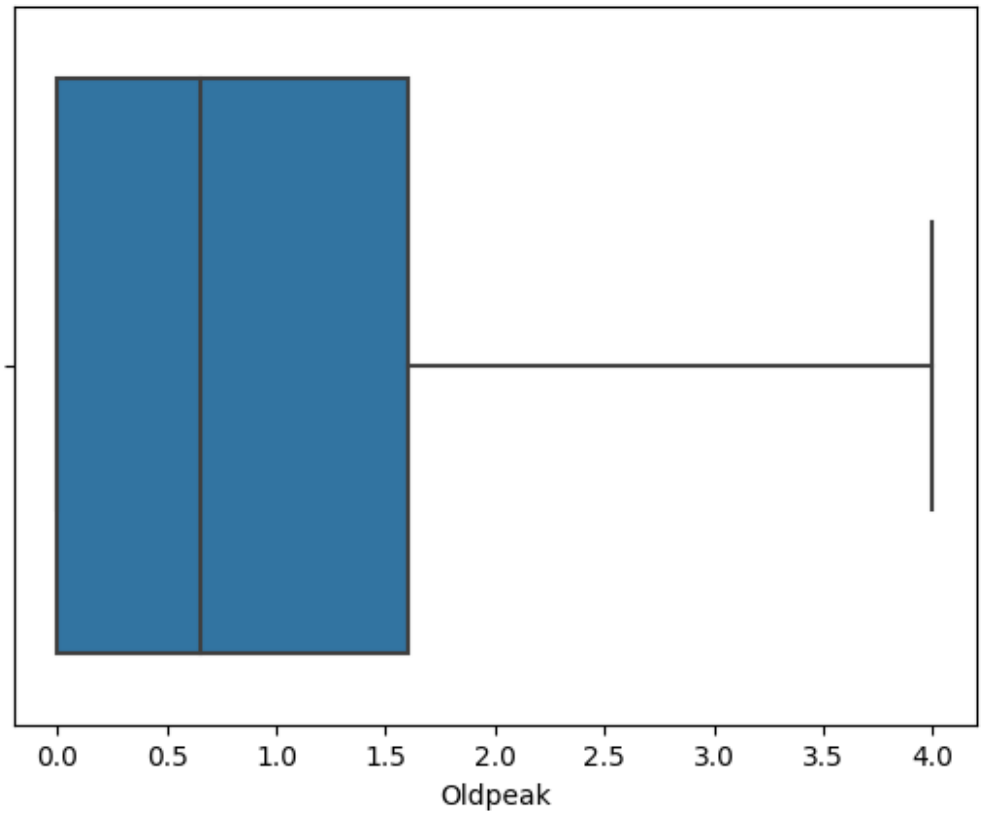

```
for x in ['MaxHR']:
    q75,q25 = np.percentile(heart.loc[:,x],[75,25])
    intr_qr = q75-q25

    max = q75+(1.5*intr_qr)
    min = q25-(1.5*intr_qr)

    heart.loc[heart[x] < min,x] = np.nan
    heart.loc[heart[x] > max,x] = np.nan

sb.boxplot(data=heart, x="MaxHR")
```

```
<Axes: xlabel='MaxHR'>
```

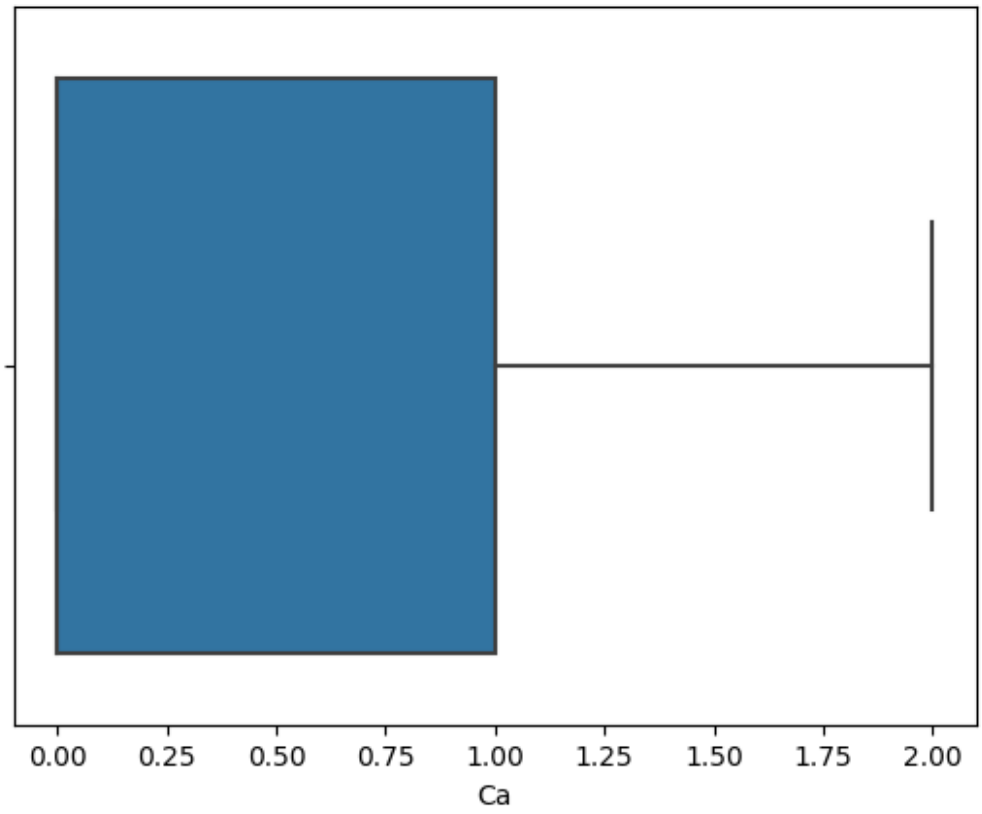



```
for x in ['Oldpeak']:
    q75,q25 = np.percentile(heart.loc[:,x],[75,25])
    intr_qr = q75-q25

    max = q75+(1.5*intr_qr)
    min = q25-(1.5*intr_qr)

    heart.loc[heart[x] < min,x] = np.nan
    heart.loc[heart[x] > max,x] = np.nan

sb.boxplot(data=heart, x="Oldpeak")
```

```
<Axes: xlabel='Oldpeak'>
```




```
for x in ['Ca']:
    q75,q25 = np.percentile(heart.loc[:,x],[75,25])
    intr_qr = q75-q25

    max = q75+(1.5*intr_qr)
    min = q25-(1.5*intr_qr)

    heart.loc[heart[x] < min,x] = np.nan
    heart.loc[heart[x] > max,x] = np.nan

sb.boxplot(data=heart, x="Ca")


<Axes: xlabel='Ca'>
```




```
heart.isnull().sum()
```

```
Age          0
Sex          0
ChestPain    0
RestBP       9
Chol         5
Fbs          0
RestECG      0
MaxHR        1
ExAng        0
Oldpeak      5
Slope        0
Ca          20
Thal         0
AHD          0
dtype: int64
```

```
data = heart.dropna(axis = 0)
```

```
data.isnull().sum()
```

```
Age          0
Sex          0
ChestPain    0
```

```
RestBP       0
Chol         0
Fbs          0
RestECG      0
MaxHR        0
ExAng        0
Oldpeak      0
Slope        0
Ca           0
Thal         0
AHD          0
dtype: int64
```

```
data.shape
```

```
(268, 14)
```

```
data.head()
```

```
   Age  Sex  ChestPain  RestBP   Chol  Fbs  RestECG  MaxHR  ExAng
Oldpeak  \
0   63    1          0   145.0  233.0    1        2  150.0      0
2.3
2   67    1          1   120.0  229.0    0        2  129.0      1
2.6
3   37    1          2   130.0  250.0    0        0  187.0      0
3.5
4   41    0          3   130.0  204.0    0        2  172.0      0
1.4
5   56    1          3   120.0  236.0    0        0  178.0      0
0.8

   Slope   Ca  Thal  AHD
0      3  0.0   0.0    0
2      2  2.0   2.0    1
3      3  0.0   1.0    0
4      1  0.0   1.0    0
5      1  0.0   1.0    0
```

```
subset1 = data[data["Age"]>40].iloc[0:30,0:6]
```

```
sb2 = data[["RestBP", "Sex", "ChestPain", "Age", "Ca", "Thal", "AHD"]]
```

```
subset1
```

```
    Age  Sex  ChestPain  RestBP   Chol  Fbs
0    63    1          0   145.0  233.0    1
2    67    1          1   120.0  229.0    0
4    41    0          3   130.0  204.0    0
5    56    1          3   120.0  236.0    0
6    62    0          1   140.0  268.0    0
```

7	57	0	1	120.0	354.0	0
8	63	1	1	130.0	254.0	0
9	53	1	1	140.0	203.0	1
10	57	1	1	140.0	192.0	0
11	56	0	3	140.0	294.0	0
12	56	1	2	130.0	256.0	1
13	44	1	3	120.0	263.0	0
15	57	1	2	150.0	168.0	0
16	48	1	3	110.0	229.0	0
17	54	1	1	140.0	239.0	0
18	48	0	2	130.0	275.0	0
19	49	1	3	130.0	266.0	0
20	64	1	0	110.0	211.0	0
21	58	0	0	150.0	283.0	1
22	58	1	3	120.0	284.0	0
23	58	1	2	132.0	224.0	0
24	60	1	1	130.0	206.0	0
25	50	0	2	120.0	219.0	0
26	58	0	2	120.0	340.0	0
27	66	0	0	150.0	226.0	0
28	43	1	1	150.0	247.0	0
30	69	0	0	140.0	239.0	0
31	60	1	1	117.0	230.0	1
32	64	1	2	140.0	335.0	0
33	59	1	1	135.0	234.0	0

```
subset2 = data[data["RestBP"]<140].iloc[0:40,3:9]

subset2
```

	RestBP	Chol	Fbs	RestECG	MaxHR	ExAng
2	120.0	229.0	0	2	129.0	1
3	130.0	250.0	0	0	187.0	0
4	130.0	204.0	0	2	172.0	0
5	120.0	236.0	0	0	178.0	0
7	120.0	354.0	0	0	163.0	1
8	130.0	254.0	0	2	147.0	0
12	130.0	256.0	1	2	142.0	1
13	120.0	263.0	0	0	173.0	0
16	110.0	229.0	0	0	168.0	0
18	130.0	275.0	0	0	139.0	0
19	130.0	266.0	0	0	171.0	0
20	110.0	211.0	0	2	144.0	1
22	120.0	284.0	0	2	160.0	0
23	132.0	224.0	0	2	173.0	0
24	130.0	206.0	0	2	132.0	1
25	120.0	219.0	0	0	158.0	0
26	120.0	340.0	0	0	172.0	0
29	110.0	167.0	0	2	114.0	1
31	117.0	230.0	1	0	160.0	1

```
33   135.0  234.0    0        0  161.0      0
34   130.0  233.0    0        0  179.0      1
36   120.0  177.0    0        2  120.0      1
38   132.0  353.0    0        0  132.0      1
44   130.0  330.0    0        2  169.0      0
45   112.0  230.0    0        2  165.0      0
46   110.0  175.0    0        0  123.0      0
49   130.0  197.0    1        2  152.0      0
50   105.0  198.0    0        0  168.0      0
51   120.0  177.0    0        0  140.0      0
52   112.0  290.0    0        2  153.0      0
53   130.0  219.0    0        2  188.0      0
54   130.0  253.0    0        0  144.0      1
55   124.0  266.0    0        2  109.0      1
57   110.0  172.0    0        2  158.0      0
58   125.0  273.0    0        2  152.0      0
59   125.0  213.0    0        2  125.0      1
60   130.0  305.0    0        0  142.0      1
63   135.0  304.0    1        0  170.0      0
64   120.0  188.0    0        0  113.0      0
71   125.0  254.0    1        0  163.0      0
```

```
subset1.merge(subset2,how="inner")
```

```
    Age  Sex  ChestPain  RestBP   Chol  Fbs  RestECG  MaxHR  ExAng
0    67    1          1   120.0  229.0    0        2  129.0      1
1    41    0          3   130.0  204.0    0        2  172.0      0
2    56    1          3   120.0  236.0    0        0  178.0      0
3    57    0          1   120.0  354.0    0        0  163.0      1
4    63    1          1   130.0  254.0    0        2  147.0      0
5    56    1          2   130.0  256.0    1        2  142.0      1
6    44    1          3   120.0  263.0    0        0  173.0      0
7    48    1          3   110.0  229.0    0        0  168.0      0
8    48    0          2   130.0  275.0    0        0  139.0      0
9    49    1          3   130.0  266.0    0        0  171.0      0
10   64    1          0   110.0  211.0    0        2  144.0      1
11   58    1          3   120.0  284.0    0        2  160.0      0
12   58    1          2   132.0  224.0    0        2  173.0      0
13   60    1          1   130.0  206.0    0        2  132.0      1
14   50    0          2   120.0  219.0    0        0  158.0      0
15   58    0          2   120.0  340.0    0        0  172.0      0
16   60    1          1   117.0  230.0    1        0  160.0      1
17   59    1          1   135.0  234.0    0        0  161.0      0
```

```
subset1.merge(subset2,how="cross")
```

```
      Age  Sex  ChestPain  RestBP_x  Chol_x  Fbs_x  RestBP_y  Chol_y
Fbs_y  \
0      63    1          0     145.0   233.0      1     120.0   229.0
0
```

```
1      63    1          0     145.0   233.0      1     130.0   250.0
0
2      63    1          0     145.0   233.0      1     130.0   204.0
0
3      63    1          0     145.0   233.0      1     120.0   236.0
0
4      63    1          0     145.0   233.0      1     120.0   354.0
0
...   ...  ...        ...       ...     ...    ...       ...     ...
...
1195   59    1          1     135.0   234.0      0     125.0   213.0
0
1196   59    1          1     135.0   234.0      0     130.0   305.0
0
1197   59    1          1     135.0   234.0      0     135.0   304.0
1
1198   59    1          1     135.0   234.0      0     120.0   188.0
0
1199   59    1          1     135.0   234.0      0     125.0   254.0
1

      RestECG  MaxHR  ExAng
0           2  129.0      1
1           0  187.0      0
2           2  172.0      0
3           0  178.0      0
4           0  163.0      1
...       ...    ...    ...
1195        2  125.0      1
1196        0  142.0      1
1197        0  170.0      0
1198        0  113.0      0
1199        0  163.0      0

[1200 rows x 12 columns]

pd.concat([subset1,subset2])

     Age  Sex  ChestPain  RestBP   Chol  Fbs  RestECG  MaxHR  ExAng
0   63.0  1.0        0.0   145.0  233.0    1      NaN    NaN    NaN
2   67.0  1.0        1.0   120.0  229.0    0      NaN    NaN    NaN
4   41.0  0.0        3.0   130.0  204.0    0      NaN    NaN    NaN
5   56.0  1.0        3.0   120.0  236.0    0      NaN    NaN    NaN
6   62.0  0.0        1.0   140.0  268.0    0      NaN    NaN    NaN
..   ...  ...        ...     ...    ...  ...      ...    ...    ...
59   NaN  NaN        NaN   125.0  213.0    0      2.0  125.0    1.0
60   NaN  NaN        NaN   130.0  305.0    0      0.0  142.0    1.0
63   NaN  NaN        NaN   135.0  304.0    1      0.0  170.0    0.0
64   NaN  NaN        NaN   120.0  188.0    0      0.0  113.0    0.0
71   NaN  NaN        NaN   125.0  254.0    1      0.0  163.0    0.0
```

```
[70 rows x 9 columns]
```

```
pd.melt(subset1,id_vars="Age")
```

```
     Age variable  value
0     63      Sex    1.0
1     67      Sex    1.0
2     41      Sex    0.0
3     56      Sex    1.0
4     62      Sex    0.0
..   ...      ...    ...
145   43      Fbs    0.0
146   69      Fbs    0.0
147   60      Fbs    1.0
148   64      Fbs    0.0
149   59      Fbs    0.0

[150 rows x 3 columns]
```

```
pd.pivot_table(data,index="RestECG",values="AHD",columns="ChestPain",a
ggfunc="sum")
```

```
ChestPain    0     1    2    3
RestECG
0          4.0  34.0  8.0  2.0
1          NaN   1.0  0.0  NaN
2          3.0  50.0  6.0  5.0
```

```
data.corr()
```

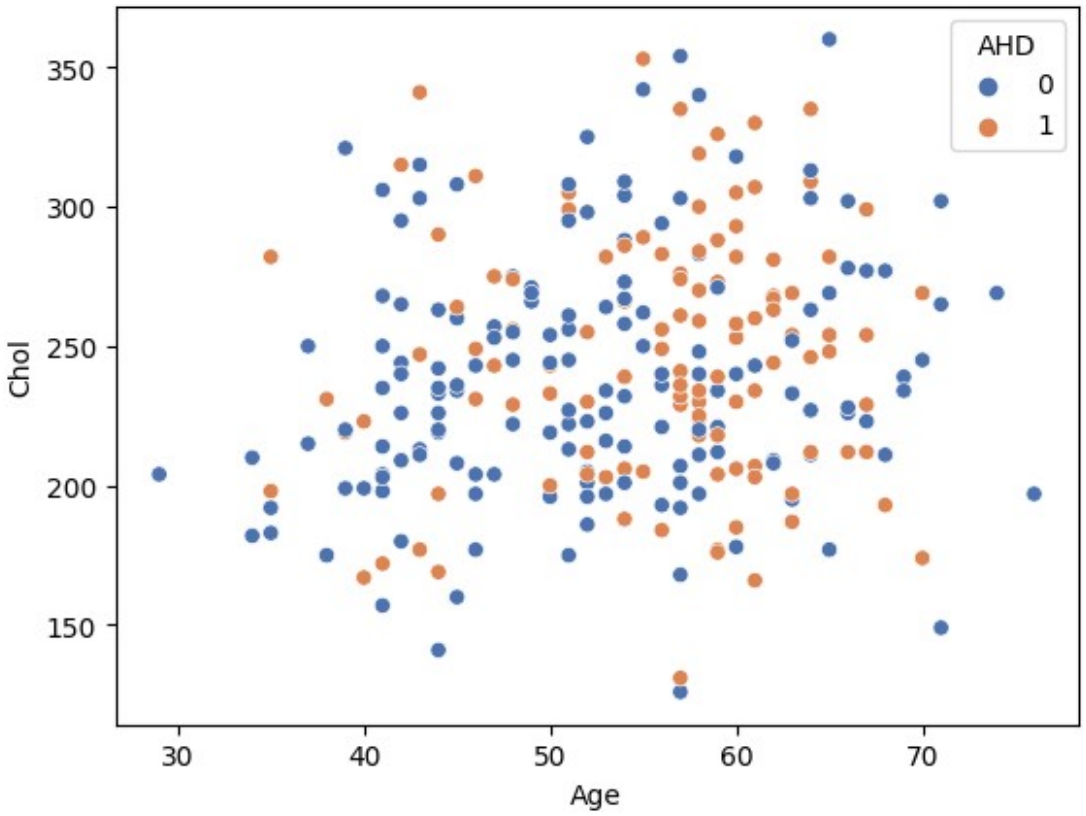
```
                Age       Sex  ChestPain    RestBP      Chol       Fbs
\
Age        1.000000 -0.082313  -0.167917  0.273692  0.144322  0.112548

Sex       -0.082313  1.000000  -0.162160 -0.003163 -0.145834  0.070901

ChestPain -0.167917 -0.162160   1.000000 -0.149945  0.020044 -0.021731

RestBP     0.273692 -0.003163  -0.149945  1.000000  0.100947  0.129918

Chol       0.144322 -0.145834   0.020044  0.100947  1.000000 -0.024842

Fbs        0.112548  0.070901  -0.021731  0.129918 -0.024842  1.000000

RestECG    0.127263  0.034258  -0.159273  0.143882  0.115783  0.073069

MaxHR     -0.399756 -0.046418   0.260866 -0.031096  0.002969 -0.010502

ExAng      0.080525  0.179089  -0.324184 -0.021488  0.049062  0.024584

Oldpeak    0.200569  0.167998  -0.327163  0.142862 -0.020917  0.021831
```

```
Slope      0.142068  0.070904  -0.225903  0.048212 -0.054274  0.096725

Ca         0.376330  0.096271  -0.213618  0.052188  0.084114  0.108020

Thal       0.063050  0.252077  -0.164349 -0.013388  0.076371 -0.061952

AHD        0.197965  0.316289  -0.367801  0.092991  0.092157  0.027863


              RestECG     MaxHR     ExAng   Oldpeak     Slope         Ca
\
Age        0.127263 -0.399756  0.080525  0.200569  0.142068  0.376330

Sex        0.034258 -0.046418  0.179089  0.167998  0.070904  0.096271

ChestPain -0.159273  0.260866 -0.324184 -0.327163 -0.225903 -0.213618

RestBP     0.143882 -0.031096 -0.021488  0.142862  0.048212  0.052188

Chol       0.115783  0.002969  0.049062 -0.020917 -0.054274  0.084114

Fbs        0.073069 -0.010502  0.024584  0.021831  0.096725  0.108020

RestECG    1.000000 -0.107348  0.103136  0.101184  0.134971  0.077498

MaxHR     -0.107348  1.000000 -0.411041 -0.333459 -0.369553 -0.241072

ExAng      0.103136 -0.411041  1.000000  0.347531  0.275077  0.182204

Oldpeak    0.101184 -0.333459  0.347531  1.000000  0.527522  0.276203

Slope      0.134971 -0.369553  0.275077  0.527522  1.000000  0.051044

Ca         0.077498 -0.241072  0.182204  0.276203  0.051044  1.000000

Thal      -0.048357 -0.144927  0.253427  0.179483  0.084955  0.123461

AHD        0.159887 -0.393276  0.432982  0.433068  0.316661  0.450196


                Thal       AHD
Age         0.063050  0.197965
Sex         0.252077  0.316289
ChestPain -0.164349 -0.367801
RestBP    -0.013388  0.092991
Chol       0.076371  0.092157
Fbs       -0.061952  0.027863
RestECG   -0.048357  0.159887
MaxHR     -0.144927 -0.393276
ExAng      0.253427  0.432982
```

```
Oldpeak    0.179483  0.433068
Slope      0.084955  0.316661
Ca         0.123461  0.450196
Thal       1.000000  0.386752
AHD        0.386752  1.000000

data

     Age  Sex  ChestPain  RestBP   Chol  Fbs  RestECG  MaxHR  ExAng
Oldpeak  \
0     63    1          0   145.0  233.0    1        2  150.0      0
2.3
2     67    1          1   120.0  229.0    0        2  129.0      1
2.6
3     37    1          2   130.0  250.0    0        0  187.0      0
3.5
4     41    0          3   130.0  204.0    0        2  172.0      0
1.4
5     56    1          3   120.0  236.0    0        0  178.0      0
0.8
..   ...  ...        ...     ...    ...  ...      ...    ...    ...
...
298   45    1          0   110.0  264.0    0        0  132.0      0
1.2
299   68    1          1   144.0  193.0    1        0  141.0      0
3.4
300   57    1          1   130.0  131.0    0        0  115.0      1
1.2
301   57    0          3   130.0  236.0    0        2  174.0      0
0.0
302   38    1          2   138.0  175.0    0        0  173.0      0
0.0

     Slope        Ca  Thal  AHD
0        3  0.000000   0.0    0
2        2  2.000000   2.0    1
3        3  0.000000   1.0    0
4        1  0.000000   1.0    0
5        1  0.000000   1.0    0
..     ...       ...   ...  ...
298      2  0.000000   2.0    1
299      2  2.000000   2.0    1
300      2  1.000000   2.0    1
301      2  1.000000   1.0    1
302      1  0.672241   1.0    0

[268 rows x 14 columns]
```

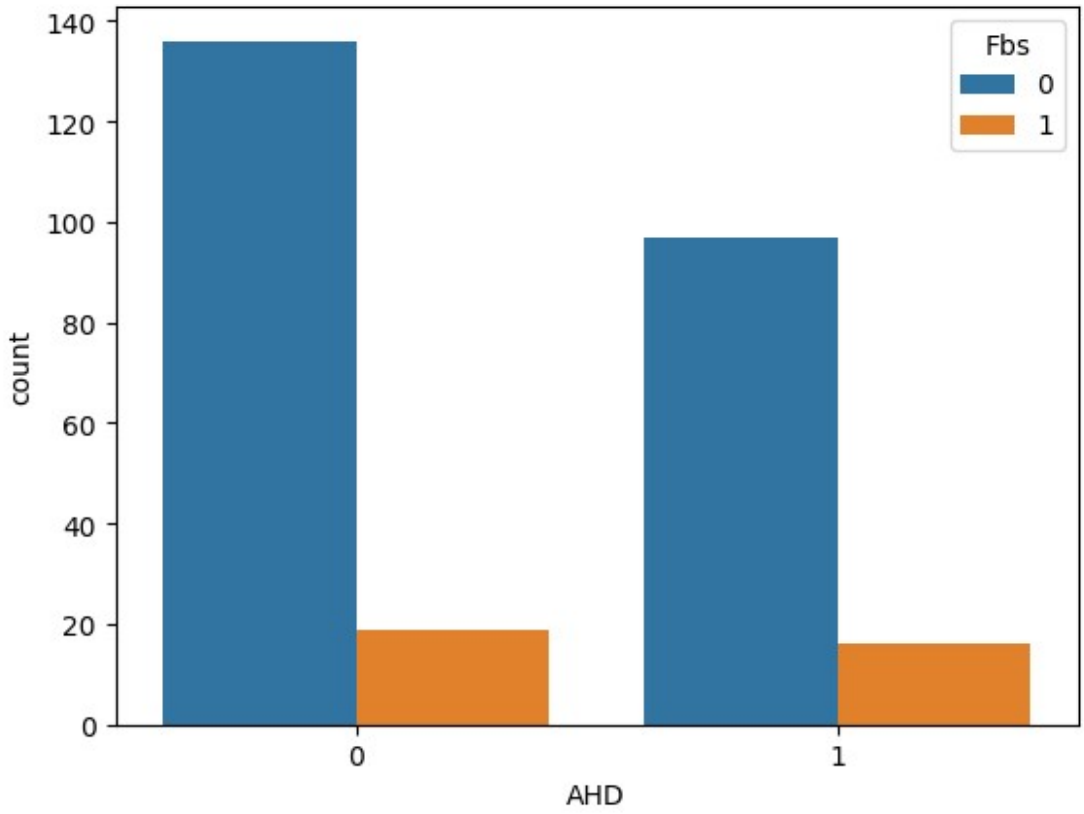
```
sb.scatterplot(data=data, x='Age', y='Chol', hue="AHD",palette="deep")
```

```
<Axes: xlabel='Age', ylabel='Chol'>
```



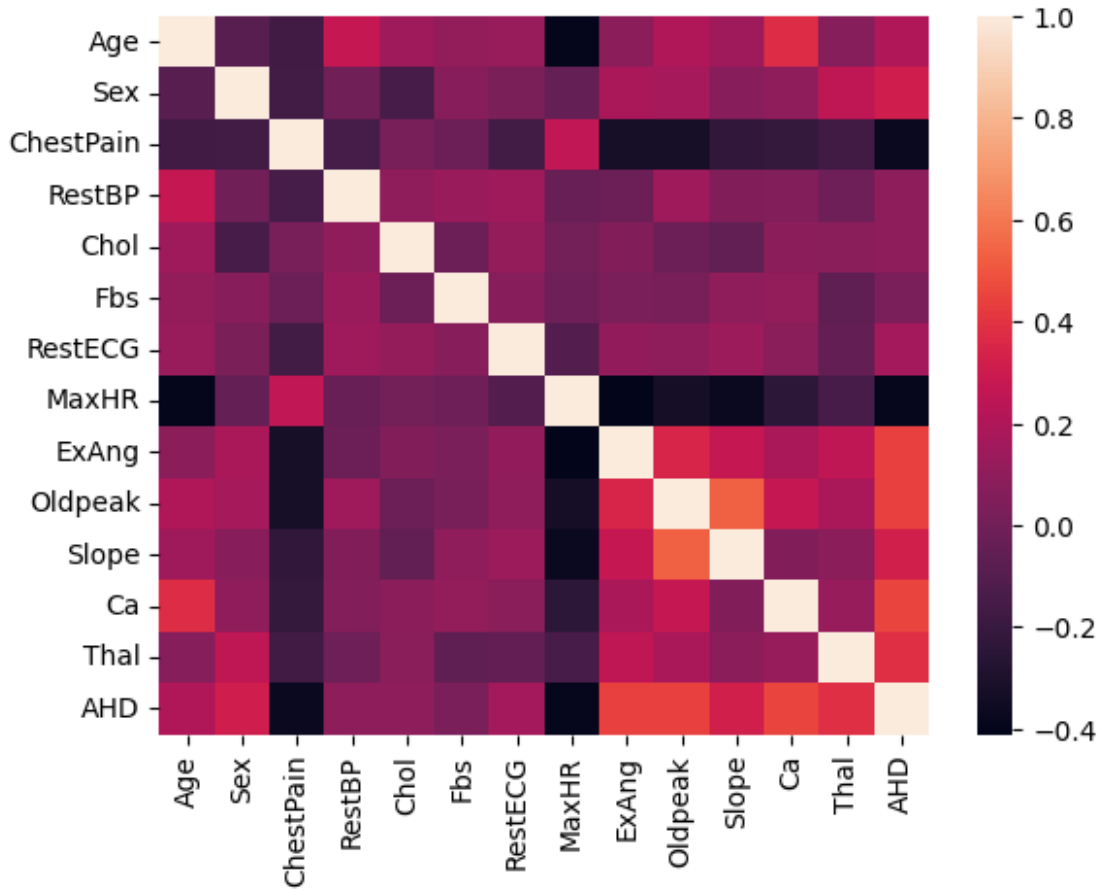

```
sb.countplot(data=data,x="AHD",hue="Fbs")
```

```
<Axes: xlabel='AHD', ylabel='count'>
```



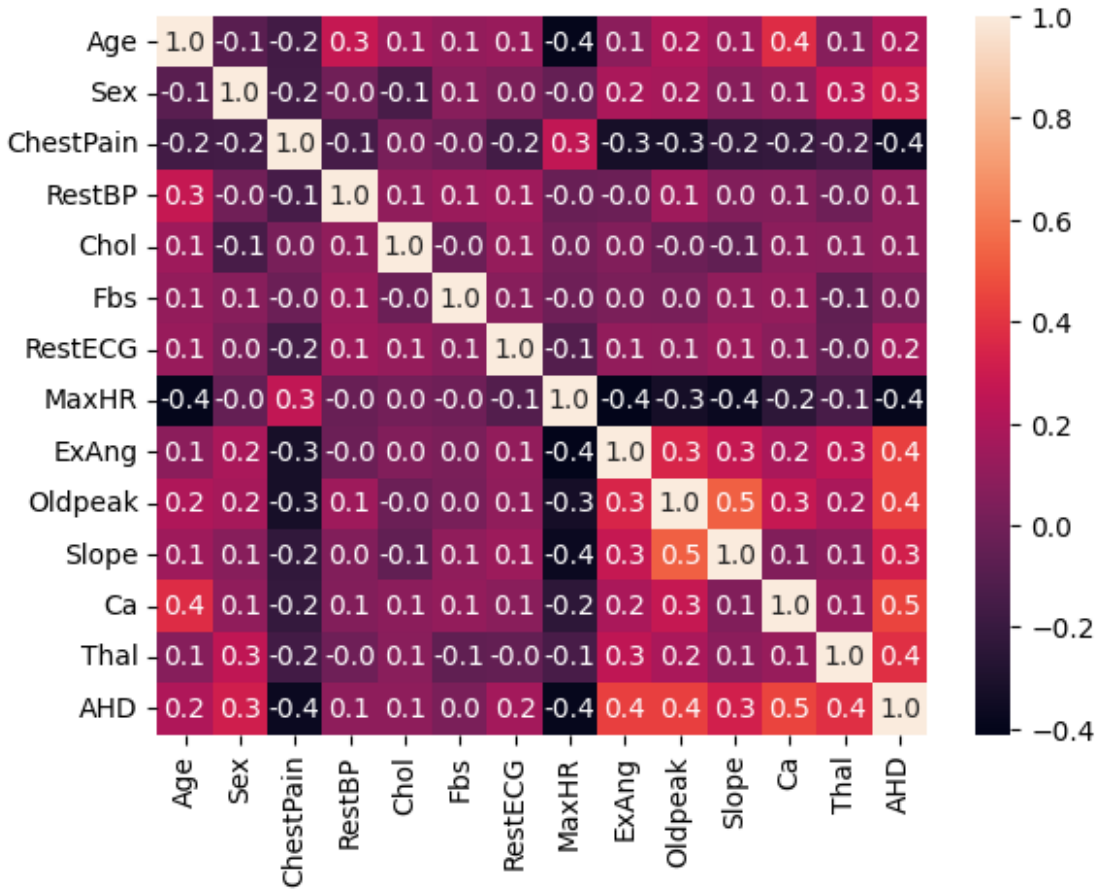

```
c = data.corr()
sb.heatmap(c)
```

```
<Axes: >
```




```
sb.heatmap(c, annot = True, fmt = '0.1f')
```

```
<Axes: >
```




```
data.corr()
```

	Age	Sex	ChestPain	RestBP	Chol	Fbs
\						
Age	1.000000	-0.082313	-0.167917	0.273692	0.144322	0.112548
Sex	-0.082313	1.000000	-0.162160	-0.003163	-0.145834	0.070901
ChestPain	-0.167917	-0.162160	1.000000	-0.149945	0.020044	-0.021731
RestBP	0.273692	-0.003163	-0.149945	1.000000	0.100947	0.129918
Chol	0.144322	-0.145834	0.020044	0.100947	1.000000	-0.024842
Fbs	0.112548	0.070901	-0.021731	0.129918	-0.024842	1.000000
RestECG	0.127263	0.034258	-0.159273	0.143882	0.115783	0.073069
MaxHR	-0.399756	-0.046418	0.260866	-0.031096	0.002969	-0.010502
ExAng	0.080525	0.179089	-0.324184	-0.021488	0.049062	0.024584
Oldpeak	0.200569	0.167998	-0.327163	0.142862	-0.020917	0.021831

```
Slope      0.142068  0.070904  -0.225903  0.048212 -0.054274  0.096725

Ca         0.376330  0.096271  -0.213618  0.052188  0.084114  0.108020

Thal       0.063050  0.252077  -0.164349 -0.013388  0.076371 -0.061952

AHD        0.197965  0.316289  -0.367801  0.092991  0.092157  0.027863


              RestECG     MaxHR     ExAng   Oldpeak     Slope         Ca
\
Age        0.127263 -0.399756  0.080525  0.200569  0.142068  0.376330

Sex        0.034258 -0.046418  0.179089  0.167998  0.070904  0.096271

ChestPain -0.159273  0.260866 -0.324184 -0.327163 -0.225903 -0.213618

RestBP     0.143882 -0.031096 -0.021488  0.142862  0.048212  0.052188

Chol       0.115783  0.002969  0.049062 -0.020917 -0.054274  0.084114

Fbs        0.073069 -0.010502  0.024584  0.021831  0.096725  0.108020

RestECG    1.000000 -0.107348  0.103136  0.101184  0.134971  0.077498

MaxHR     -0.107348  1.000000 -0.411041 -0.333459 -0.369553 -0.241072

ExAng      0.103136 -0.411041  1.000000  0.347531  0.275077  0.182204

Oldpeak    0.101184 -0.333459  0.347531  1.000000  0.527522  0.276203

Slope      0.134971 -0.369553  0.275077  0.527522  1.000000  0.051044

Ca         0.077498 -0.241072  0.182204  0.276203  0.051044  1.000000

Thal      -0.048357 -0.144927  0.253427  0.179483  0.084955  0.123461

AHD        0.159887 -0.393276  0.432982  0.433068  0.316661  0.450196


                Thal       AHD
Age         0.063050  0.197965
Sex         0.252077  0.316289
ChestPain -0.164349 -0.367801
RestBP    -0.013388  0.092991
Chol       0.076371  0.092157
Fbs       -0.061952  0.027863
RestECG   -0.048357  0.159887
MaxHR     -0.144927 -0.393276
ExAng      0.253427  0.432982
```

```
Oldpeak    0.179483  0.433068
Slope      0.084955  0.316661
Ca         0.123461  0.450196
Thal       1.000000  0.386752
AHD        0.386752  1.000000
```

```
data=data.drop("Fbs",axis=1)
```

```
X = data.iloc[:,0:11]
```

```
X
```

```
     Age  Sex  ChestPain  RestBP   Chol  RestECG  MaxHR  ExAng
Oldpeak  \
0     63    1          0   145.0  233.0        2  150.0      0
2.3
2     67    1          1   120.0  229.0        2  129.0      1
2.6
3     37    1          2   130.0  250.0        0  187.0      0
3.5
4     41    0          3   130.0  204.0        2  172.0      0
1.4
5     56    1          3   120.0  236.0        0  178.0      0
0.8
..   ...  ...        ...     ...    ...      ...    ...    ...      ..
.
298   45    1          0   110.0  264.0        0  132.0      0
1.2
299   68    1          1   144.0  193.0        0  141.0      0
3.4
300   57    1          1   130.0  131.0        0  115.0      1
1.2
301   57    0          3   130.0  236.0        2  174.0      0
0.0
302   38    1          2   138.0  175.0        0  173.0      0
0.0

     Slope        Ca
0        3  0.000000
2        2  2.000000
3        3  0.000000
4        1  0.000000
5        1  0.000000
..     ...       ...
298      2  0.000000
299      2  2.000000
300      2  1.000000
301      2  1.000000
302      1  0.672241
```

```
[268 rows x 11 columns]
```

```
Y = data["AHD"].values
```

```
Y
```

```
array([0, 1, 0, 0, 0, 1, 0, 1, 1, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 1,
1,
       1, 0, 0, 0, 0, 1, 0, 1, 1, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 1, 1,
0,
       1, 0, 0, 0, 1, 0, 1, 1, 1, 1, 0, 0, 1, 0, 0, 1, 1, 1, 0, 1, 1,
0,
       1, 1, 1, 1, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
1,
       1, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 0, 1, 1, 0, 0, 0,
1,
       1, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 0, 0, 1, 0, 1,
0,
       1, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0,
1,
       0, 1, 1, 1, 0, 1, 0, 1, 0, 0, 0, 1, 0, 1, 0, 0, 0, 1, 0, 0, 0,
0,
       1, 1, 0, 1, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0,
1,
       1, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 0, 1, 0, 1, 1, 1, 0, 0, 1, 0,
0,
       0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 1, 1, 1, 1, 1, 0, 1, 0, 1, 0, 1,
0,
       0, 0, 1, 0, 1, 0, 1, 0, 1, 1, 0, 0, 0, 1, 0, 1, 1, 1, 0, 1, 1,
1,
       1, 1, 1, 0])
```

```
X_train,X_test,Y_train,Y_test = tts(X,Y)
```

```
X_train
```

```
     Age  Sex  ChestPain  RestBP   Chol  RestECG  MaxHR  ExAng
Oldpeak  \
297   57    0          1   140.0  241.0        0  123.0      1
0.2
68    59    1          1   170.0  326.0        2  140.0      1
3.4
143   64    1          2   125.0  309.0        0  131.0      1
1.8
299   68    1          1   144.0  193.0        0  141.0      0
3.4
264   61    1          1   138.0  166.0        2  125.0      1
3.6
..   ...  ...        ...     ...    ...      ...    ...    ...      ..
.
```

```
208   55    1          3   130.0  262.0        0  155.0      0
0.0
216   46    0          3   105.0  204.0        0  172.0      0
0.0
292   44    1          1   120.0  169.0        0  144.0      1
2.8
128   44    1          3   120.0  220.0        0  170.0      0
0.0
207   50    1          1   144.0  200.0        2  126.0      1
0.9

     Slope   Ca
297      2  0.0
68       3  0.0
143      2  0.0
299      2  2.0
264      2  1.0
..     ...  ...
208      1  0.0
216      1  0.0
292      3  0.0
128      1  0.0
207      2  0.0

[201 rows x 11 columns]
```

X_test

```
     Age  Sex  ChestPain  RestBP   Chol  RestECG  MaxHR  ExAng
Oldpeak  \
144   58    1          2   105.0  240.0        2  154.0      1
0.6
256   67    0          1   106.0  223.0        0  142.0      0
0.3
151   42    0          1   102.0  265.0        2  122.0      0
0.6
287   58    1          3   125.0  220.0        0  144.0      0
0.4
109   39    1          1   118.0  219.0        0  140.0      0
1.2
..   ...  ...        ...     ...    ...      ...    ...    ...      ..
.
7     57    0          1   120.0  354.0        0  163.0      1
0.6
273   71    0          1   112.0  149.0        0  125.0      0
1.6
210   37    0          2   120.0  215.0        0  170.0      0
0.0
47    50    1          1   150.0  243.0        2  128.0      0
2.6
```

```
112   52    1          0   118.0  186.0        2  190.0      0
0.0

     Slope        Ca
144      2  0.000000
256      1  2.000000
151      2  0.000000
287      2  0.672241
109      2  0.000000
..     ...       ...
7        1  0.000000
273      2  0.000000
210      1  0.000000
47       2  0.000000
112      2  0.000000

[67 rows x 11 columns]

Y_train

array([1, 1, 1, 1, 1, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0,
0,
       0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 0, 0, 1, 1, 0, 1, 0, 0,
0,
       0, 0, 1, 1, 1, 1, 0, 1, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 1, 0, 1,
1,
       0, 0, 1, 0, 0, 0, 1, 0, 1, 0, 1, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0,
0,
       0, 1, 0, 1, 0, 0, 0, 0, 1, 1, 0, 0, 1, 1, 0, 1, 0, 1, 1, 0, 0,
1,
       1, 0, 1, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1,
1,
       0, 0, 1, 0, 1, 1, 0, 0, 0, 0, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 0,
0,
       1, 0, 1, 0, 0, 0, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1, 0, 1,
1,
       0, 1, 0, 0, 0, 1, 1, 0, 0, 1, 0, 1, 0, 1, 0, 1, 1, 0, 0, 0, 0,
0,
       1, 0, 1])

Y_test

array([0, 0, 0, 0, 1, 0, 0, 1, 1, 1, 1, 0, 0, 0, 1, 1, 1, 0, 1, 0, 0,
0,
       1, 1, 1, 0, 1, 1, 1, 0, 0, 0, 0, 0, 0, 1, 1, 0, 1, 1, 1, 1, 0,
0,
       1, 0, 1, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0,
1,
       0])

lrm = LogisticRegression(max_iter=1000).fit(X_train,Y_train)
```

```
Y_pred = lrm.predict(X_test)

metrics.confusion_matrix(Y_test,Y_pred)

array([[34,  5],
       [ 8, 20]])

metrics.accuracy_score(Y_test,Y_pred)

0.8059701492537313

metrics.recall_score(Y_test,Y_pred)

0.7142857142857143

metrics.precision_score(Y_test,Y_pred)

0.8

df1 = pd.DataFrame({
    "Age":[63],
    "Sex":[1],
    "ChestPain":[1],
    "RestBP":[145],
    "RestECG":[2],
    "MaxHR":[150],
    "ExAng":[0],
    "Oldpeak":[2.3],
    "Slope":[3],
    "Ca":[0.0],
    "Thal":[1]
})
```