Importing some library's

In [2]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')
```

This model performs binary classification to determine the presence of heart disease.

In [4]:
```
df = pd.read_csv("heart.csv")
df.head()
```

Out[4]:

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal
0	63	1	3	145	233	1	0	150	0	2.3	0	0	1
1	37	1	2	130	250	0	1	187	0	3.5	0	0	2
2	41	0	1	130	204	0	0	172	0	1.4	2	0	2
3	56	1	1	120	236	0	1	178	0	0.8	2	0	2
4	57	0	0	120	354	0	1	163	1	0.6	2	0	2

In [5]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 303 entries, 0 to 302
Data columns (total 14 columns):
 #   Column    Non-Null Count  Dtype  
---  ------    --------------  -----  
 0   age       303 non-null    int64  
 1   sex       303 non-null    int64  
 2   cp        303 non-null    int64  
 3   trestbps  303 non-null    int64  
 4   chol      303 non-null    int64  
 5   fbs       303 non-null    int64  
 6   restecg   303 non-null    int64  
 7   thalach   303 non-null    int64  
 8   exang     303 non-null    int64  
 9   oldpeak   303 non-null    float64
 10  slope     303 non-null    int64  
 11  ca        303 non-null    int64  
 12  thal      303 non-null    int64  
 13  target    303 non-null    int64  
dtypes: float64(1), int64(13)
memory usage: 33.3 KB
```

In [6]:
```
df.describe()
```

Out[6]:

	age	sex	cp	trestbps	chol	fbs	reste
count	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.0000
mean	54.366337	0.683168	0.966997	131.623762	246.264026	0.148515	0.5280
std	9.082101	0.466011	1.032052	17.538143	51.830751	0.356198	0.5258
min	29.000000	0.000000	0.000000	94.000000	126.000000	0.000000	0.0000
25%	47.500000	0.000000	0.000000	120.000000	211.000000	0.000000	0.0000
50%	55.000000	1.000000	1.000000	130.000000	240.000000	0.000000	1.0000
75%	61.000000	1.000000	2.000000	140.000000	274.500000	0.000000	1.0000
max	77.000000	1.000000	3.000000	200.000000	564.000000	1.000000	2.0000

In [7]:
```
df.isnull().sum()
```

Out[7]:
```
age         0
sex         0
cp          0
trestbps    0
chol        0
fbs         0
restecg     0
thalach     0
exang       0
oldpeak     0
slope       0
ca          0
thal        0
target      0
dtype: int64
```

In [8]:
```
cols = df.columns

for i in cols:
    print(df[i].value_counts())
    print("*"*50)
```

```
age
58    19
57    17
54    16
59    14
52    13
51    12
62    11
60    11
44    11
56    11
64    10
41    10
63     9
67     9
65     8
43     8
45     8
55     8
42     8
61     8
53     8
46     7
48     7
66     7
50     7
49     5
47     5
70     4
39     4
35     4
68     4
38     3
71     3
40     3
69     3
34     2
37     2
29     1
74     1
76     1
77     1
Name: count, dtype: int64
**************************************************
sex
1    207
0     96
Name: count, dtype: int64
**************************************************
cp
0    143
2     87
1     50
3     23
Name: count, dtype: int64
**************************************************
trestbps
120    37
130    36
140    32
```

```
110    19
150    17
138    13
128    12
160    11
125    11
112     9
132     8
118     7
124     6
135     6
108     6
152     5
134     5
145     5
122     4
170     4
100     4
105     3
126     3
115     3
180     3
136     3
142     3
102     2
148     2
178     2
94      2
144     2
146     2
200     1
114     1
154     1
123     1
192     1
174     1
165     1
104     1
117     1
101     1
156     1
106     1
155     1
129     1
172     1
164     1
Name: count, dtype: int64
**************************************************
chol
204    6
197    6
234    6
269    5
254    5
      ..
284    1
224    1
167    1
276    1
131    1
```

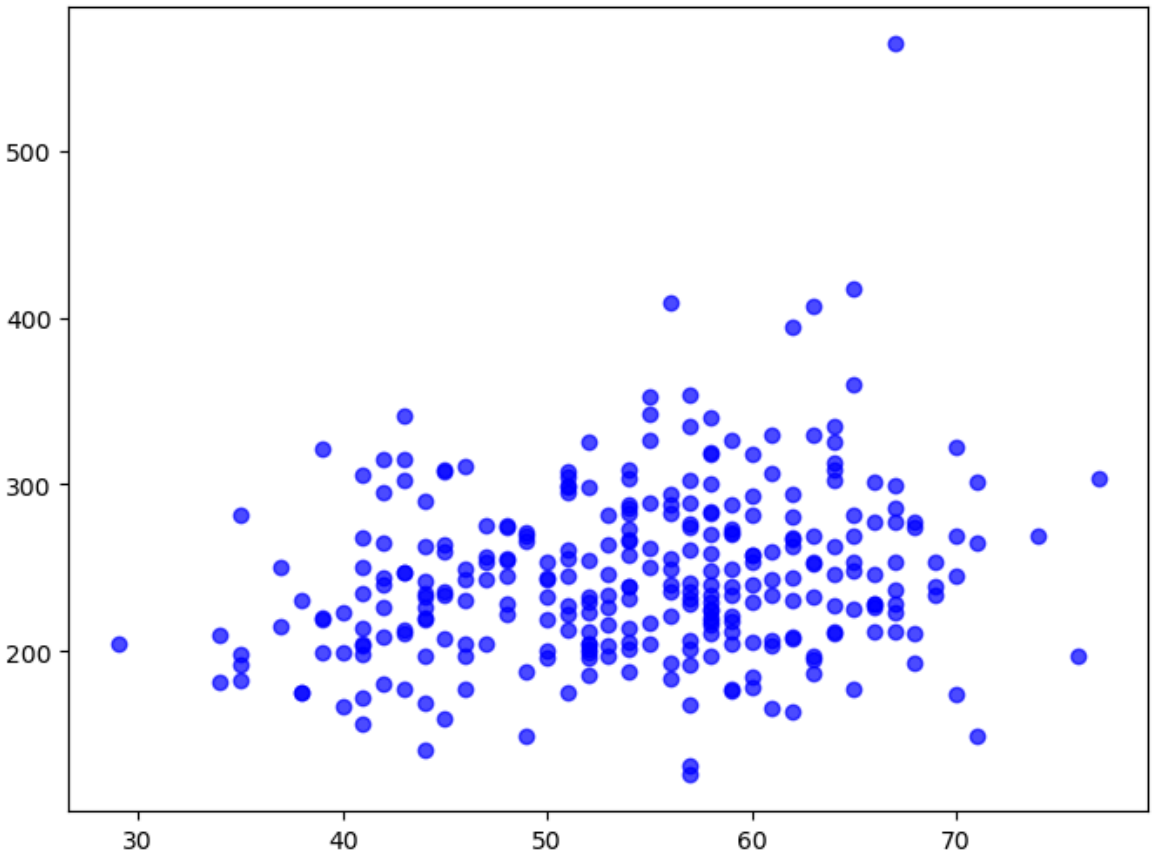
```
Name: count, Length: 152, dtype: int64
**************************************************
fbs
0    258
1     45
Name: count, dtype: int64
**************************************************
restecg
1    152
0    147
2      4
Name: count, dtype: int64
**************************************************
thalach
162    11
160     9
163     9
152     8
173     8
       ..
202     1
184     1
121     1
192     1
90      1
Name: count, Length: 91, dtype: int64
**************************************************
exang
0    204
1     99
Name: count, dtype: int64
**************************************************
oldpeak
0.0    99
1.2    17
1.0    14
0.6    14
1.4    13
0.8    13
0.2    12
1.6    11
1.8    10
0.4     9
2.0     9
0.1     7
2.8     6
2.6     6
1.5     5
3.0     5
1.9     5
0.5     5
3.6     4
2.2     4
2.4     3
0.9     3
3.4     3
4.0     3
0.3     3
2.3     2
3.2     2
```

```
2.5     2
4.2     2
1.1     2
3.1     1
0.7     1
3.5     1
6.2     1
1.3     1
5.6     1
2.9     1
2.1     1
3.8     1
4.4     1
Name: count, dtype: int64
**************************************************
slope
2    142
1    140
0     21
Name: count, dtype: int64
**************************************************
ca
0    175
1     65
2     38
3     20
4      5
Name: count, dtype: int64
**************************************************
thal
2    166
3    117
1     18
0      2
Name: count, dtype: int64
**************************************************
target
1    165
0    138
Name: count, dtype: int64
**************************************************
```

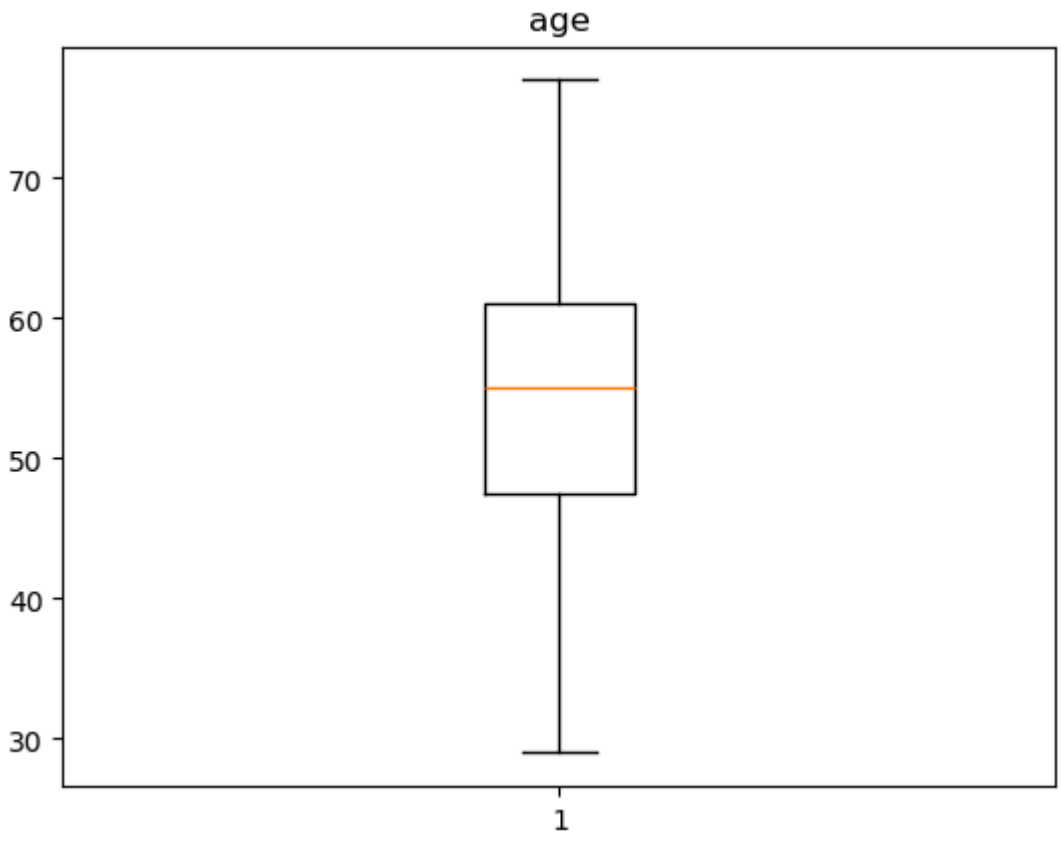
In [ ]:

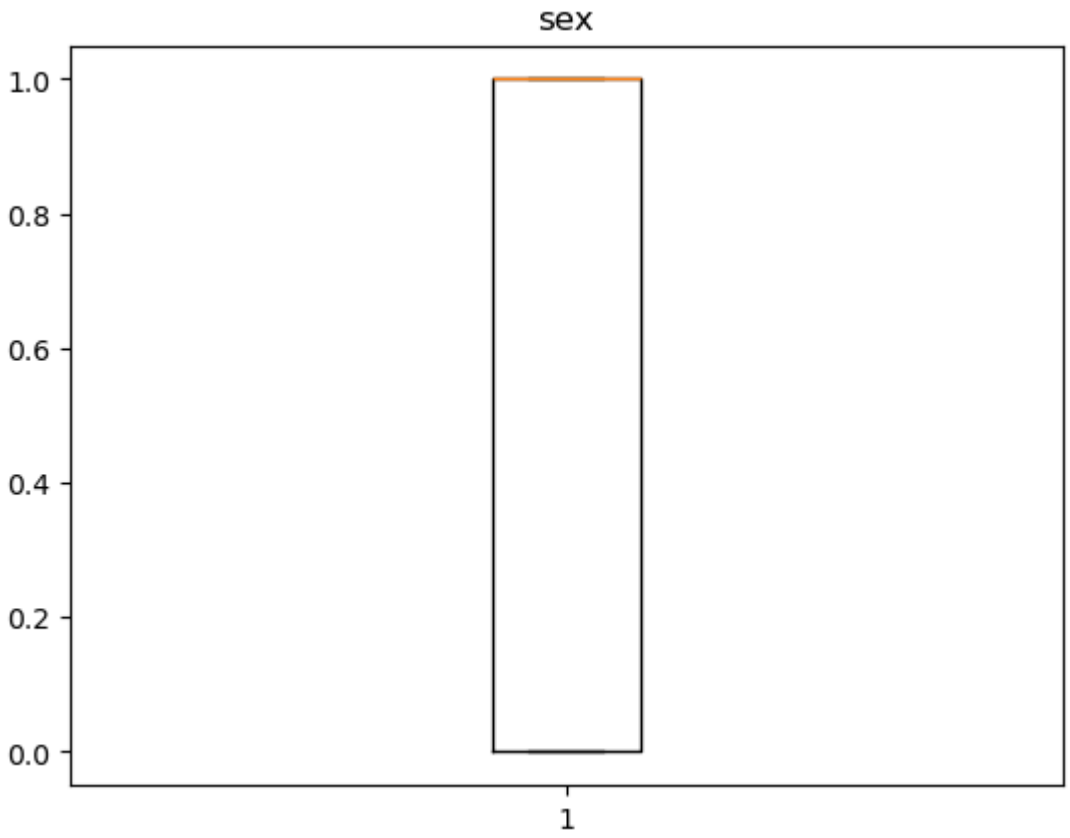
In [9]:
```
plt.figure(figsize=(8, 6))
plt.scatter(df['age'], df['chol'], alpha=0.7, color='blue')
```

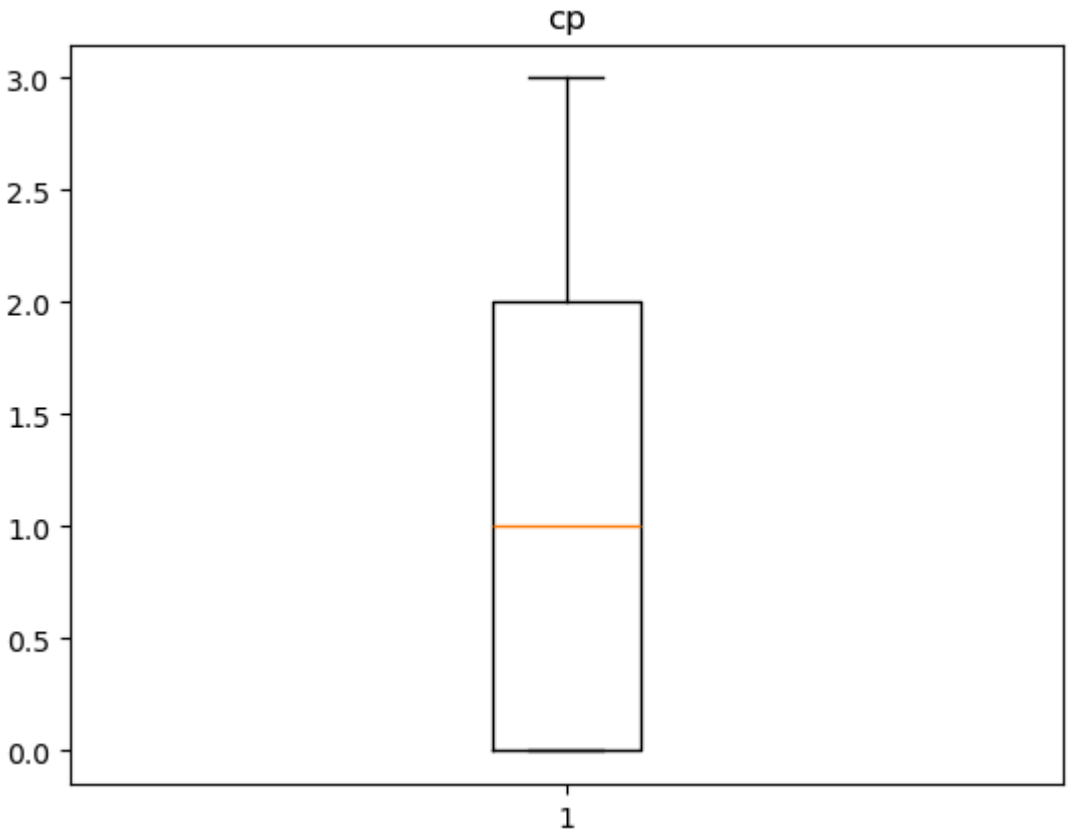
Out[9]: <matplotlib.collections.PathCollection at 0x1b661001ac0>

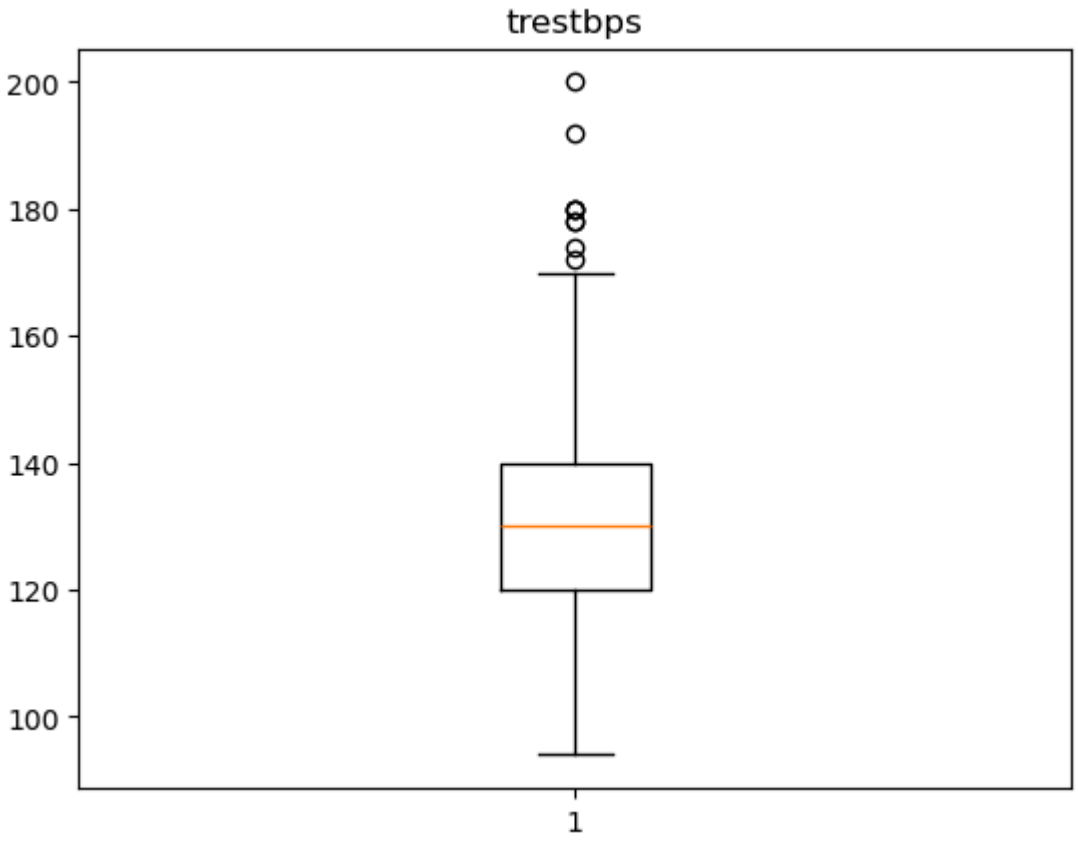


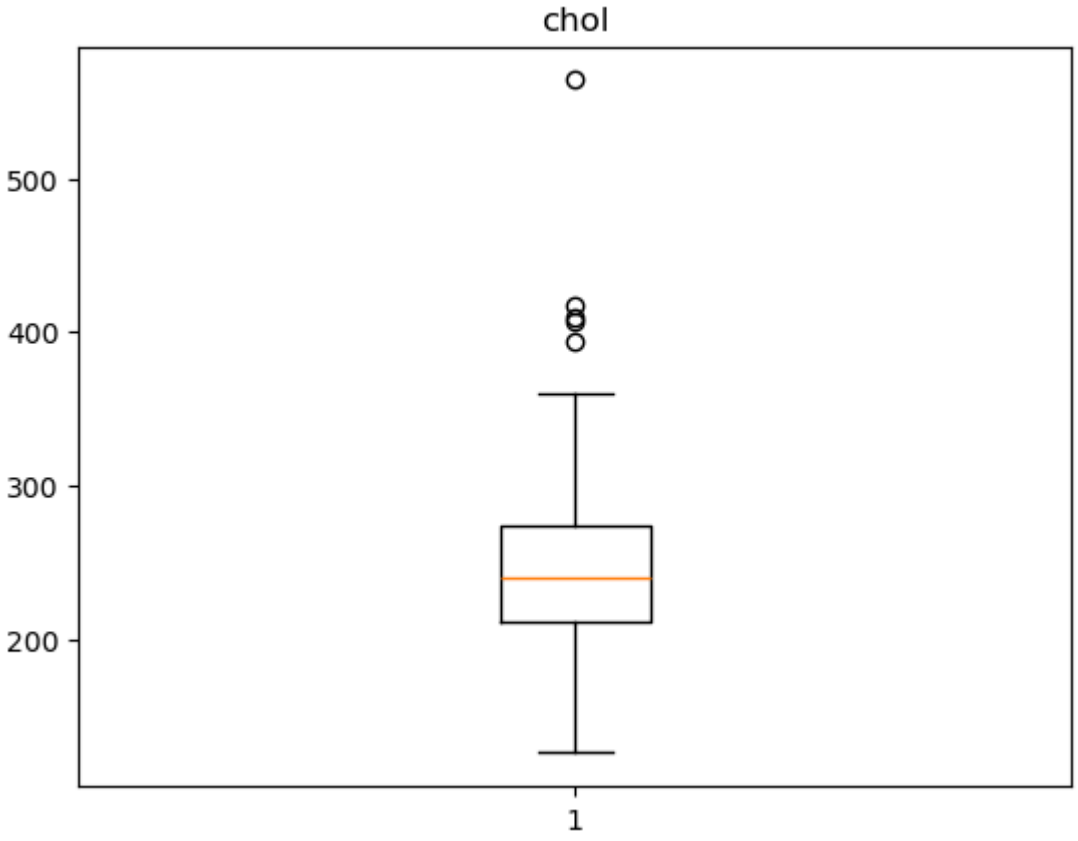
In [10]:
```
for i in cols:
    plt.boxplot(x=df[i])
    plt.title(i)
    plt.show()
```

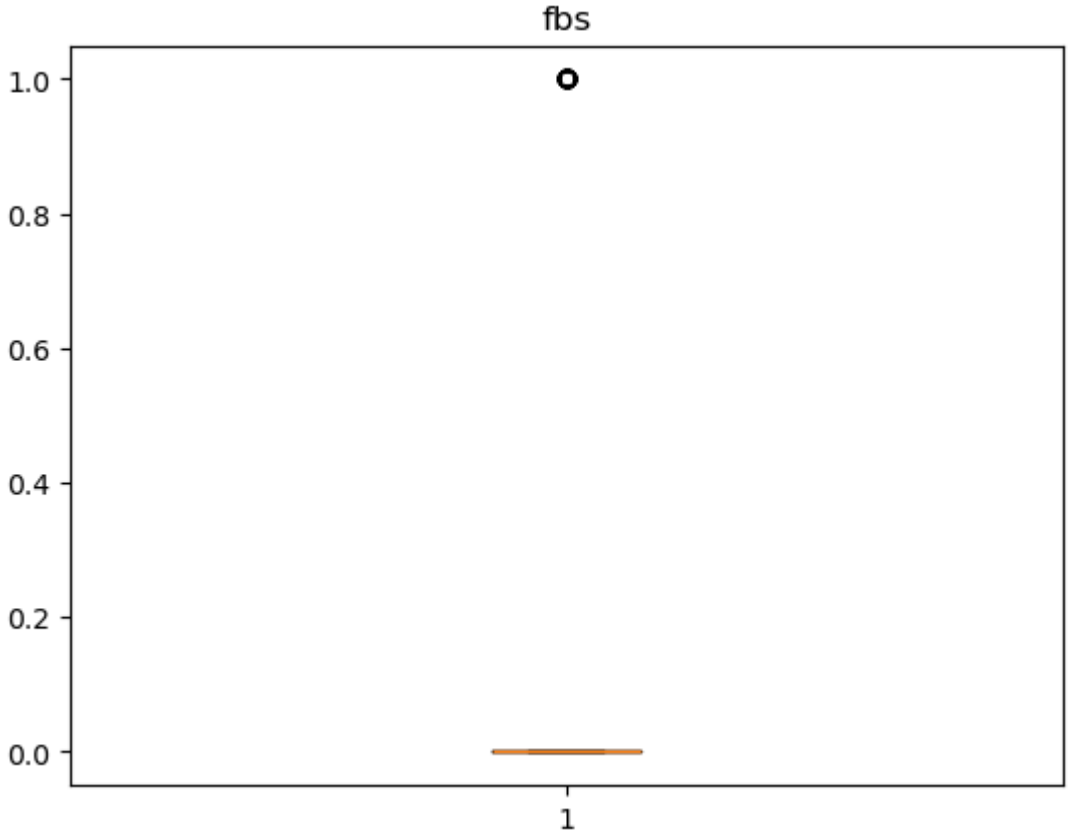


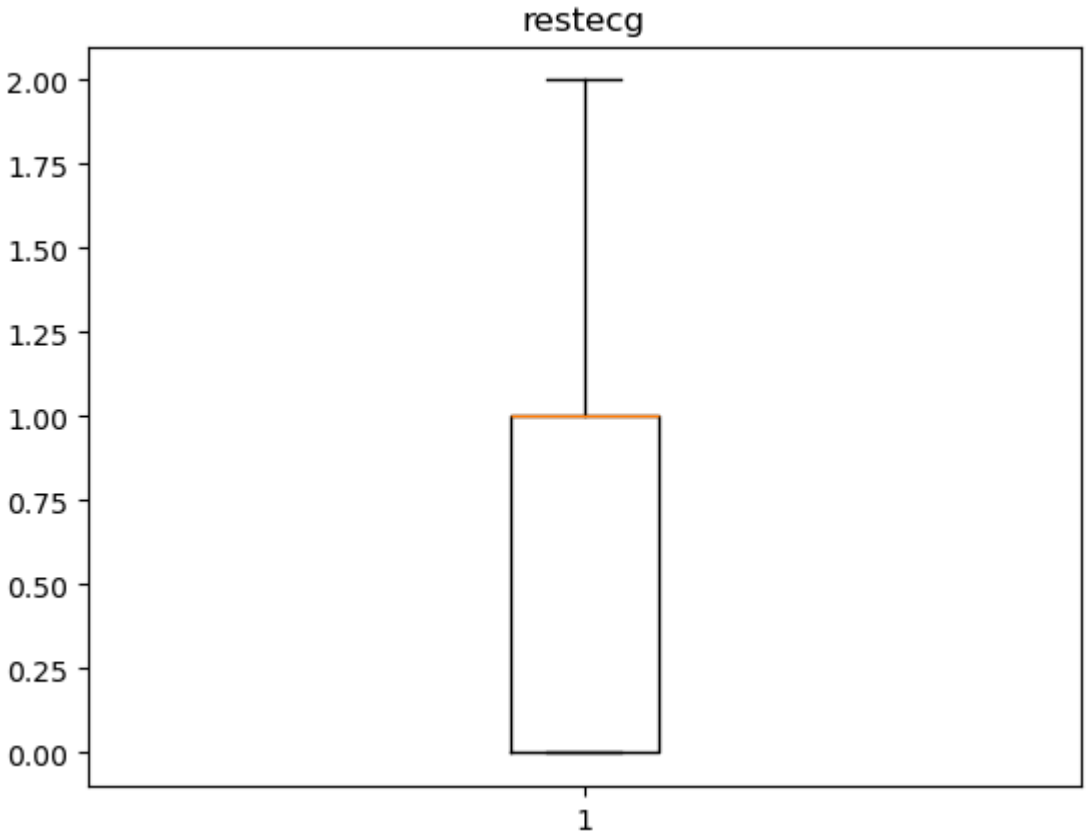

sex
1.0
0.8
0.6
0.4
0.2
0.0
1

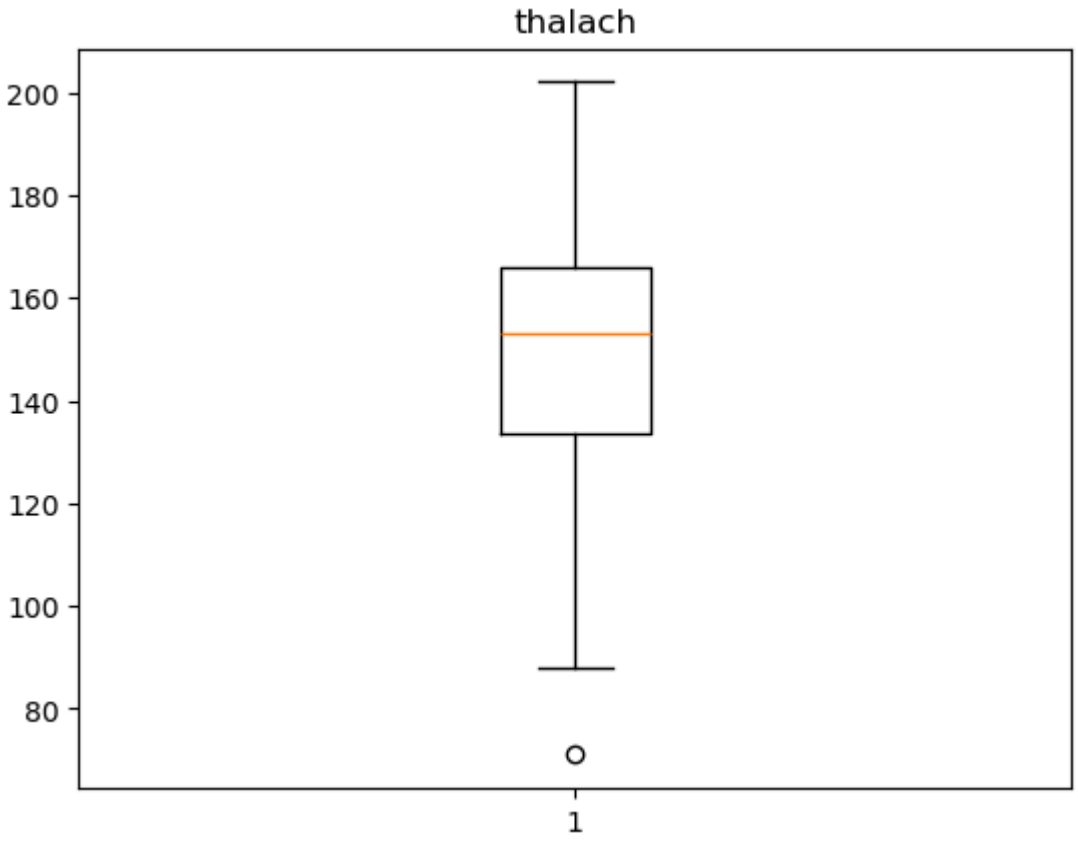

cp
3.0
2.5
2.0
1.5
1.0
0.5
0.0
1

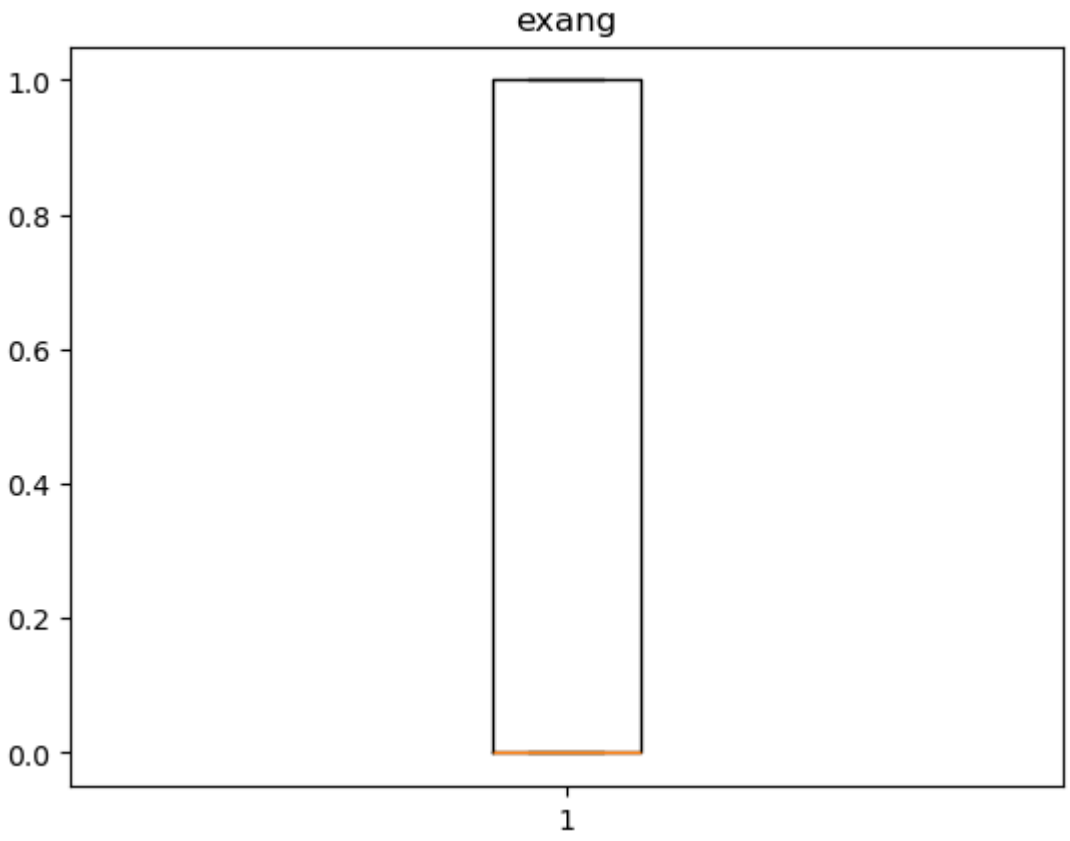

trestbps
200
180
160
140
120
100
1

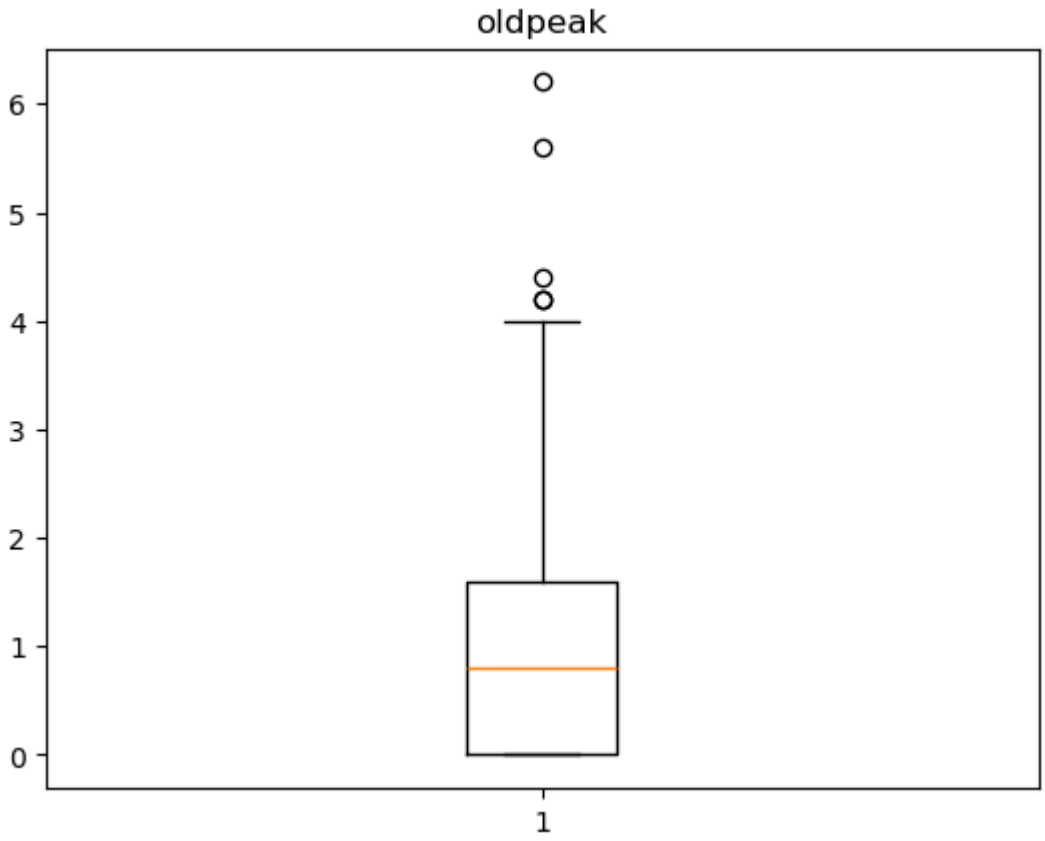

chol
500
400
300
200
1

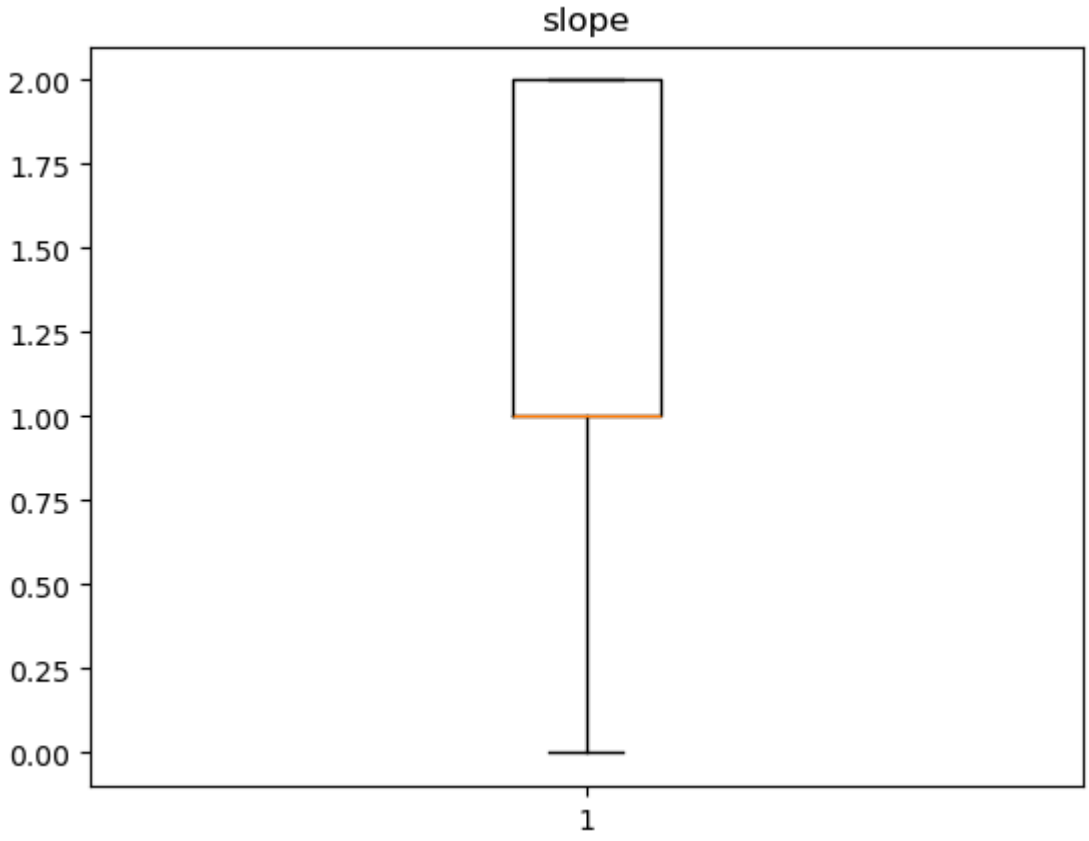

fbs
1.0
0.8
0.6
0.4
0.2
0.0
1

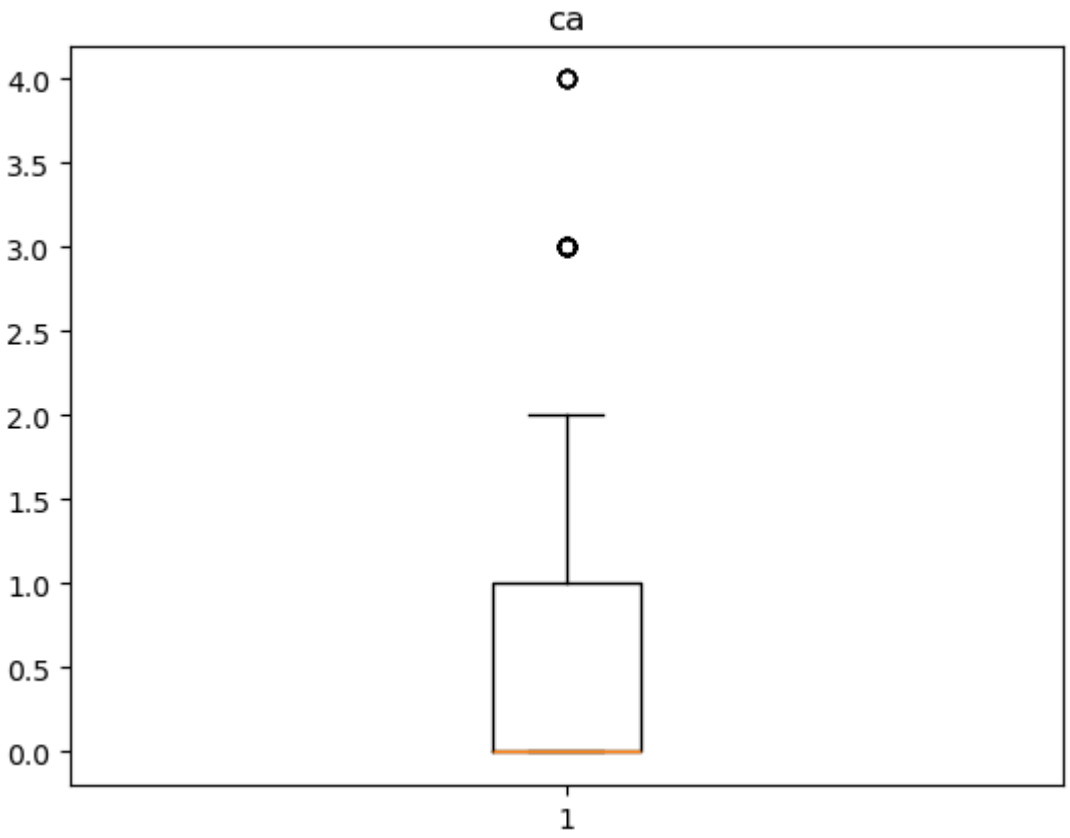

restecg
2.00
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00
1

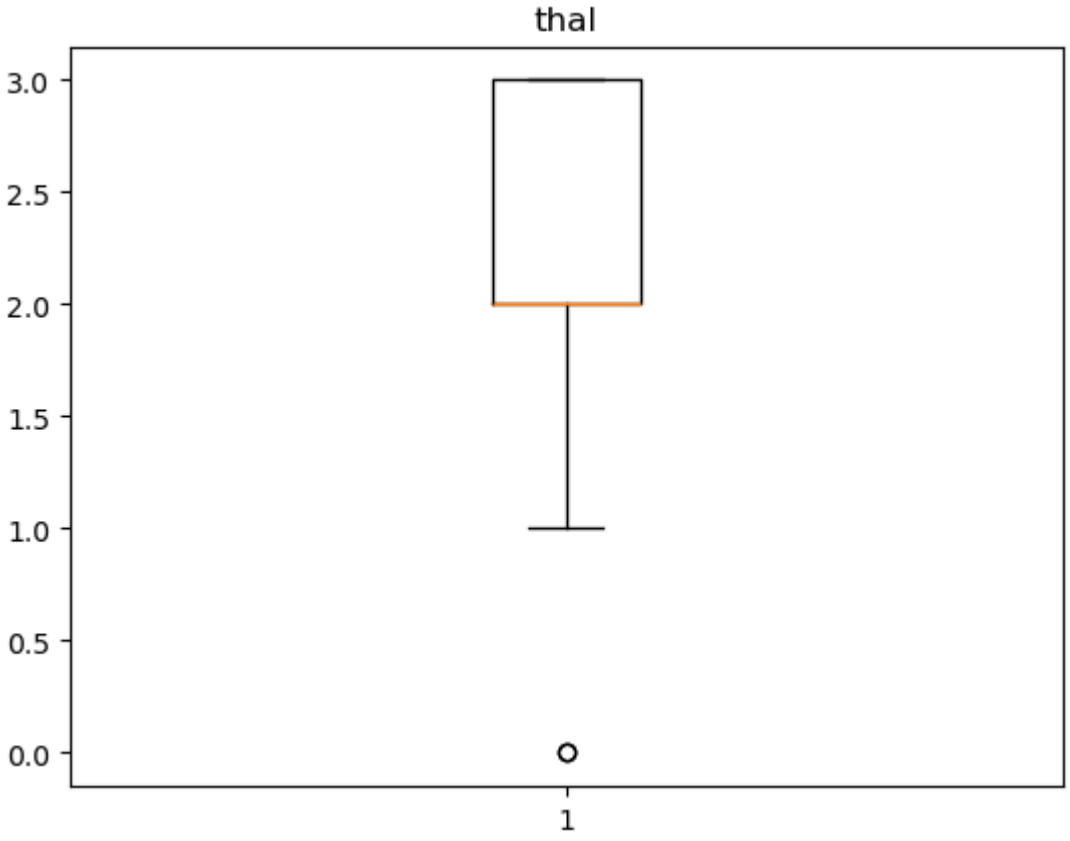

thalach
200
180
160
140
120
100
80
1

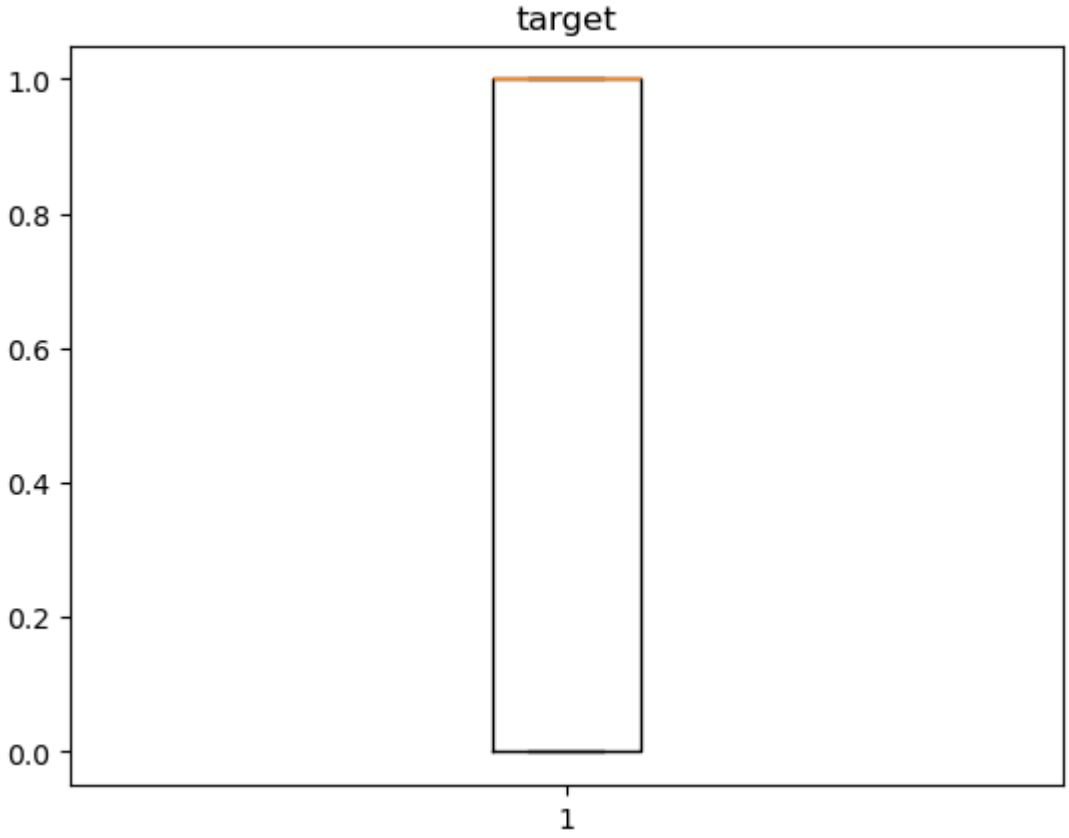

exang
1.0
0.8
0.6
0.4
0.2
0.0
1


oldpeak
6
5
4
3
2
1
0
1


slope
2.00
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00
1


ca
4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
1

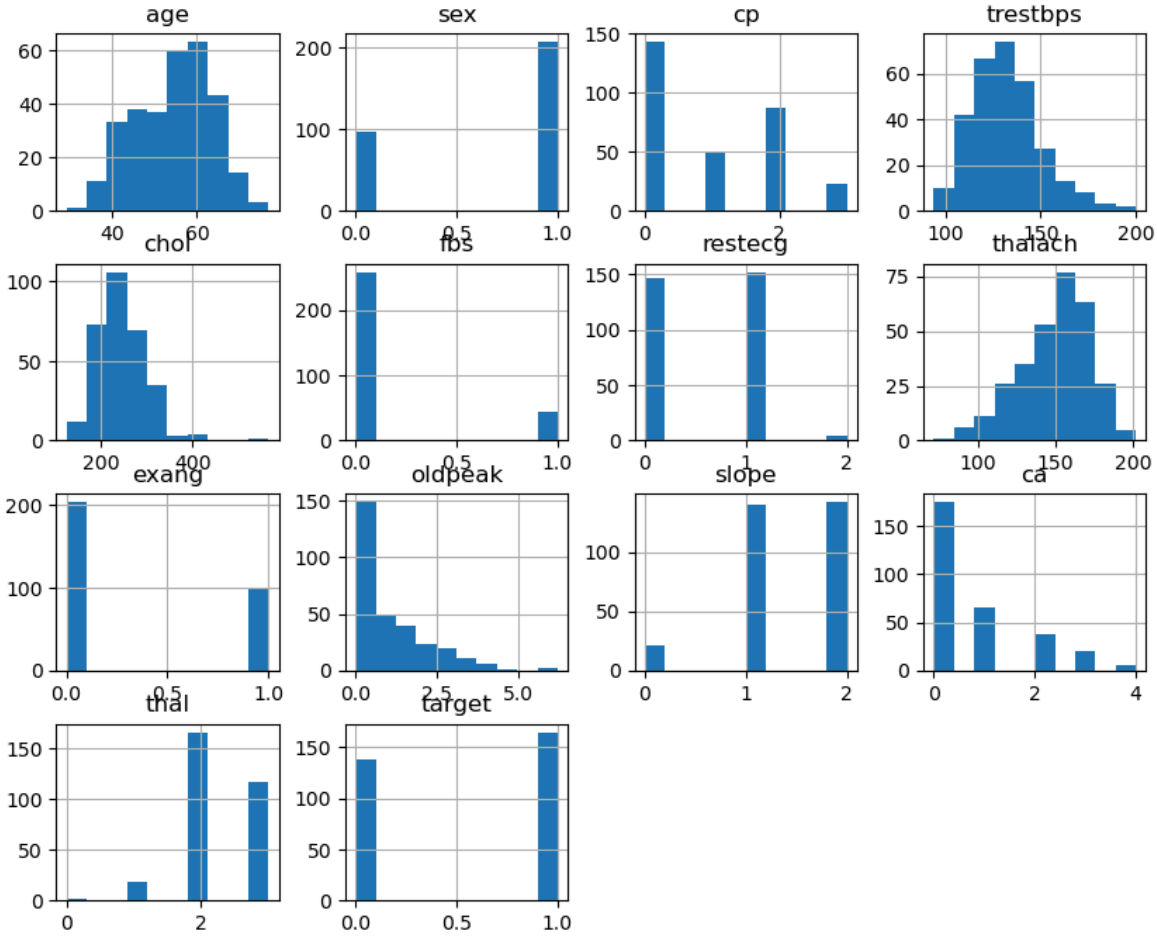

thal
3.0
2.5
2.0
1.5
1.0
0.5
0.0
1



In [11]:
```
max_chol = df['chol'].max()
max_chol

# we would have handled outliers in Column Cholestrol but there is possibilty of
# however this is extremely high and considered a severe health risk

# same goes with resting blood pressure

## Before cheking:
# only in column Thal, we will remove values below 1 as it is not relevant

## after checking:
# there are 0 values present.
```

Out[11]: 564

In [12]:
```
df.hist(figsize=(10,8))
plt.show()
```



In [13]:
```
for i in cols:

    print(i,":")
    q1=df[i].quantile(0.25)
    q3=df[i].quantile(0.75)
    iqr=q3-q1
    upper_tail=q3+1.5*iqr
    lower_tail=q1-1.5*iqr

    print("q1 --->",q1)
    print("q3 --->",q3)
    print("iqr --->",iqr)
    print("upper tail --->",upper_tail)
    print("lower_tail --->",lower_tail)
    print("*"*100)
```

```
age :
q1 ---> 47.5
q3 ---> 61.0
iqr ---> 13.5
upper tail ---> 81.25
lower_tail ---> 27.25
**************************************************************************************
******************
sex :
q1 ---> 0.0
q3 ---> 1.0
iqr ---> 1.0
upper tail ---> 2.5
lower_tail ---> -1.5
**************************************************************************************
******************
cp :
q1 ---> 0.0
q3 ---> 2.0
iqr ---> 2.0
upper tail ---> 5.0
lower_tail ---> -3.0
**************************************************************************************
******************
trestbps :
q1 ---> 120.0
q3 ---> 140.0
iqr ---> 20.0
upper tail ---> 170.0
lower_tail ---> 90.0
**************************************************************************************
******************
chol :
q1 ---> 211.0
q3 ---> 274.5
iqr ---> 63.5
upper tail ---> 369.75
lower_tail ---> 115.75
**************************************************************************************
******************
fbs :
q1 ---> 0.0
q3 ---> 0.0
iqr ---> 0.0
upper tail ---> 0.0
lower_tail ---> 0.0
**************************************************************************************
******************
restecg :
q1 ---> 0.0
q3 ---> 1.0
iqr ---> 1.0
upper tail ---> 2.5
lower_tail ---> -1.5
**************************************************************************************
******************
thalach :
q1 ---> 133.5
q3 ---> 166.0
iqr ---> 32.5
```

```
upper tail ---> 214.75
lower_tail ---> 84.75
********************************************************************************
******************
exang :
q1 ---> 0.0
q3 ---> 1.0
iqr ---> 1.0
upper tail ---> 2.5
lower_tail ---> -1.5
********************************************************************************
******************
oldpeak :
q1 ---> 0.0
q3 ---> 1.6
iqr ---> 1.6
upper tail ---> 4.0
lower_tail ---> -2.4000000000000004
********************************************************************************
******************
slope :
q1 ---> 1.0
q3 ---> 2.0
iqr ---> 1.0
upper tail ---> 3.5
lower_tail ---> -0.5
********************************************************************************
******************
ca :
q1 ---> 0.0
q3 ---> 1.0
iqr ---> 1.0
upper tail ---> 2.5
lower_tail ---> -1.5
********************************************************************************
******************
thal :
q1 ---> 2.0
q3 ---> 3.0
iqr ---> 1.0
upper tail ---> 4.5
lower_tail ---> 0.5
********************************************************************************
******************
target :
q1 ---> 0.0
q3 ---> 1.0
iqr ---> 1.0
upper tail ---> 2.5
lower_tail ---> -1.5
********************************************************************************
******************
```

In [14]:
```
(df['thal'] < lower_tail).sum()
```

Out[14]:
```
0
```

In [15]:
```
df[df['thal'] > upper_tail]
```

Out[15]:

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	th
7	44	1	1	120	263	0	1	173	0	0.0	2	0	
8	52	1	2	172	199	1	1	162	0	0.5	2	0	
20	59	1	0	135	234	0	1	161	0	0.5	1	0	
24	40	1	3	140	199	0	1	178	1	1.4	2	0	
31	65	1	0	120	177	0	1	140	0	0.4	2	0	
...	...	...	...	...	...	...	...	...	...	...	...	...	
295	63	1	0	140	187	0	0	144	1	4.0	2	2	
298	57	0	0	140	241	0	1	123	1	0.2	1	0	
299	45	1	3	110	264	0	1	132	0	1.2	1	0	
300	68	1	0	144	193	1	1	141	0	3.4	1	2	
301	57	1	0	130	131	0	1	115	1	1.2	1	1	

117 rows × 14 columns

In [16]:
```
df.skew()
```

Out[16]:
```
age        -0.202463
sex        -0.791335
cp          0.484732
trestbps    0.713768
chol        1.143401
fbs         1.986652
restecg     0.162522
thalach    -0.537410
exang       0.742532
oldpeak     1.269720
slope      -0.508316
ca          1.310422
thal       -0.476722
target     -0.179821
dtype: float64
```

In [17]:
```
cols_to_transform = ['chol', 'fbs', 'oldpeak', 'ca']
for col in cols_to_transform:
    df[col] = np.sqrt(df[col])
```

In [18]:
```
df.skew()
```

Out[18]:
```
age        -0.202463
sex        -0.791335
cp          0.484732
trestbps    0.713768
chol        0.561800
fbs         1.986652
restecg     0.162522
thalach    -0.537410
exang       0.742532
oldpeak     0.160467
slope      -0.508316
ca          0.615547
thal       -0.476722
target     -0.179821
dtype: float64
```

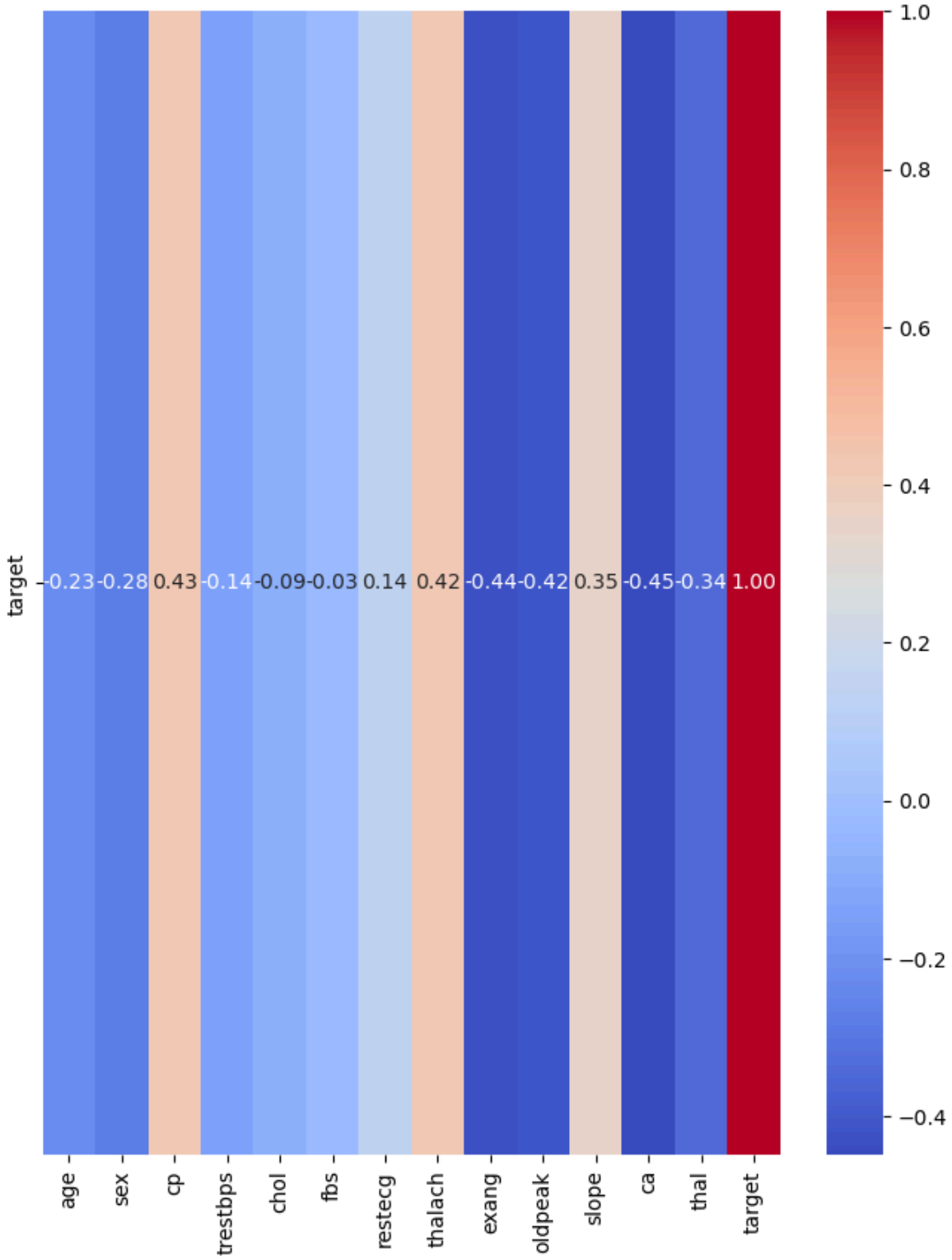
In [19]:
```
df.corr().tail(1)
```

Out[19]:

	age	sex	cp	trestbps	chol	fbs	restecg	thalacl
target	-0.225439	-0.280937	0.433798	-0.144931	-0.090264	-0.028046	0.13723	0.42174

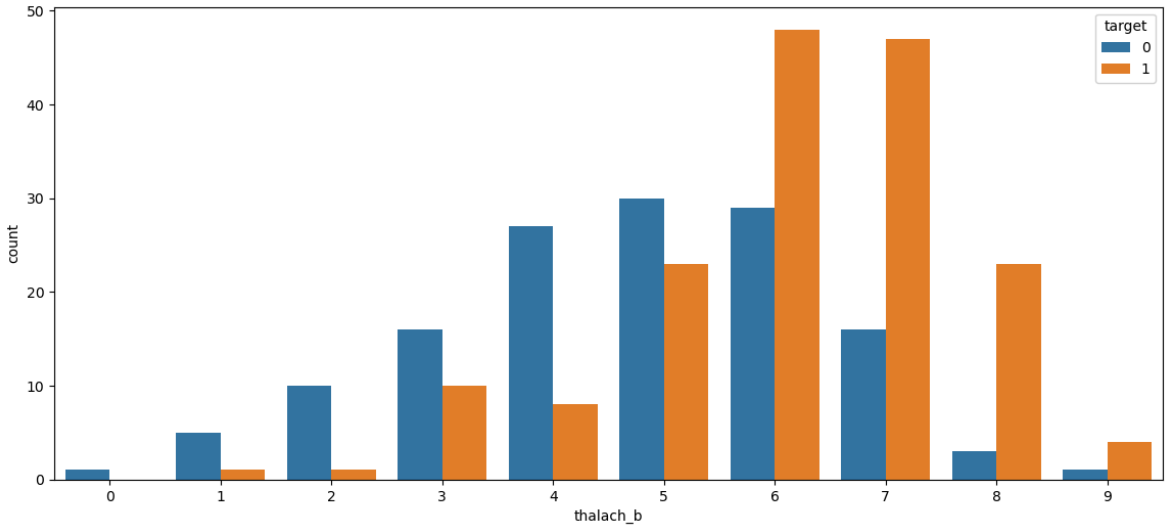
In [20]:
```
plt.figure(figsize=(8,10))
sns.heatmap(df.corr().tail(1),cmap = 'coolwarm',fmt = '.2f',annot = True)
```

Out[20]:
```
<Axes: >
```

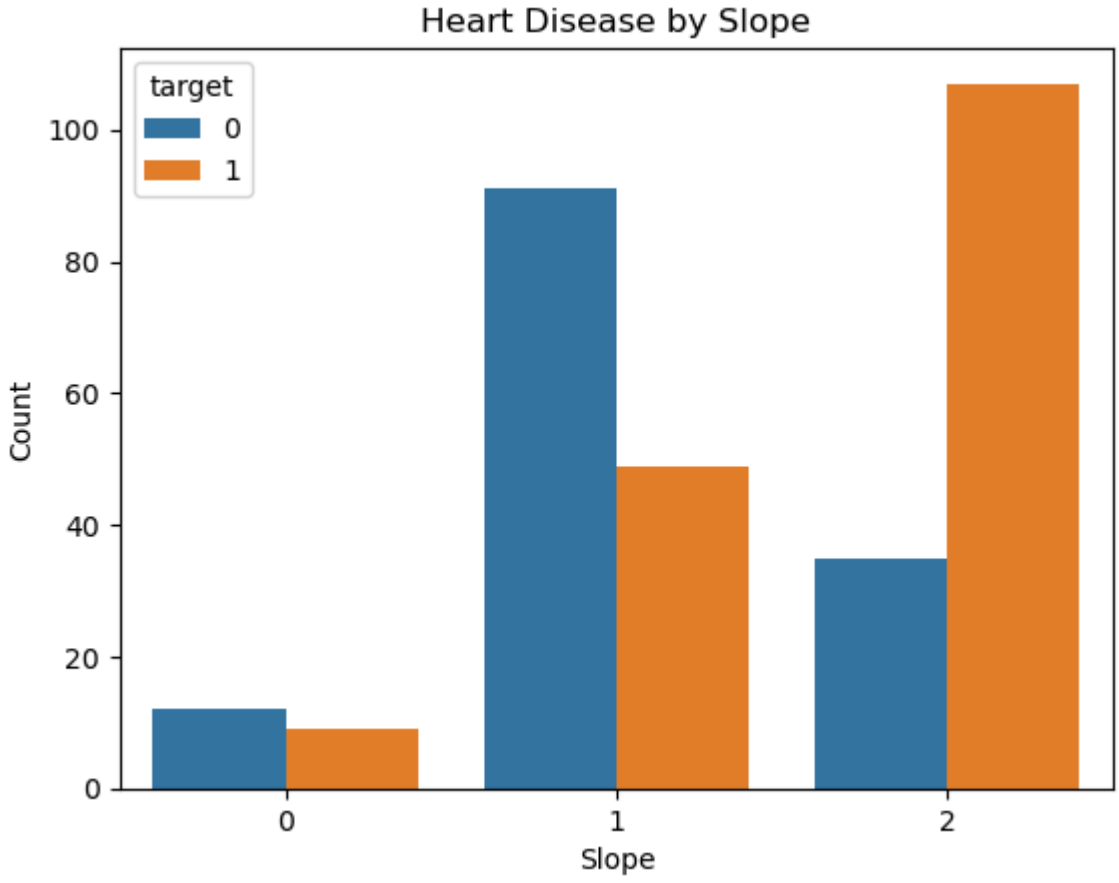


In [21]:
```
# Create bins for thalach
df['thalach_b'] = pd.cut(df['thalach'], bins=10,labels=False)

# Create the countplot
plt.figure(figsize=(14,6))
sns.countplot(x='thalach_b', hue='target', data=df)
```

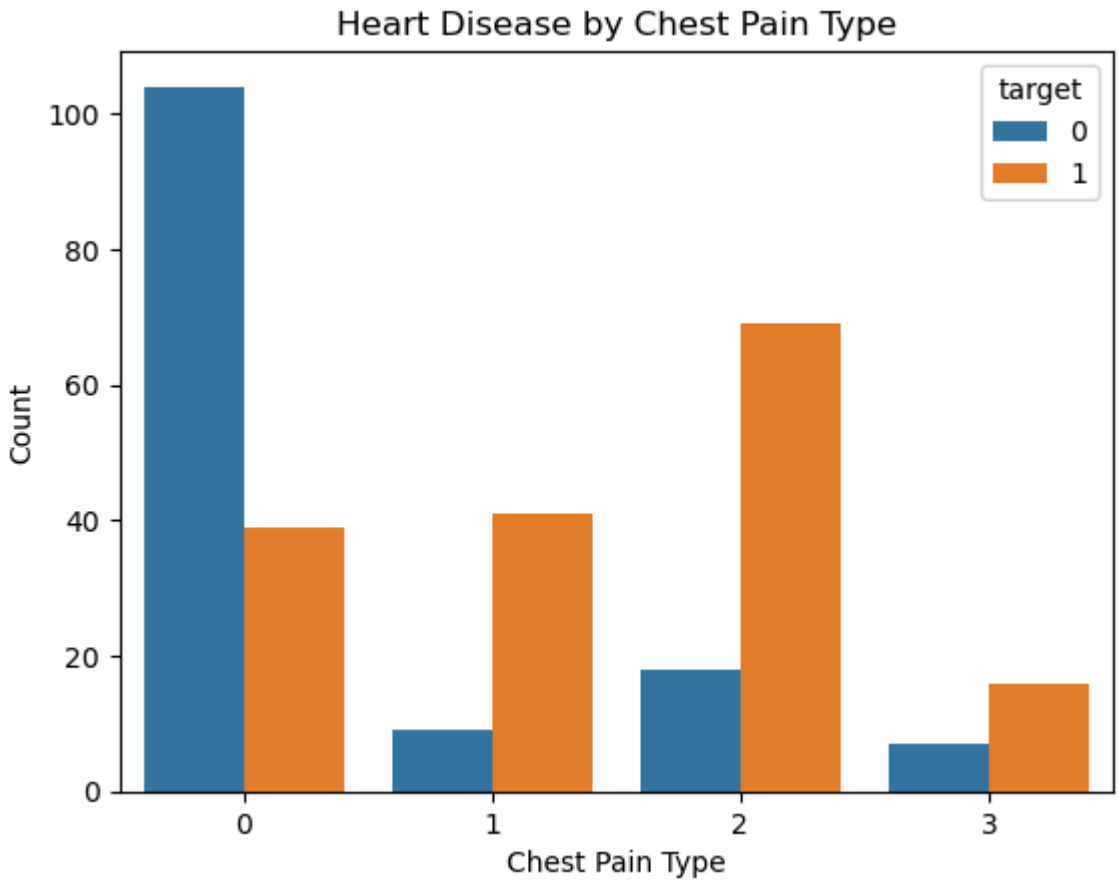
Out[21]:
```
<Axes: xlabel='thalach_b', ylabel='count'>
```



In [22]:
```
sns.countplot(x='slope', hue='target', data=df)
plt.xlabel('Slope')
plt.ylabel('Count')
plt.title('Heart Disease by Slope')
plt.show()
```



In [23]:
```
sns.countplot(x='cp', hue='target', data=df)
plt.xlabel('Chest Pain Type')
plt.ylabel('Count')
plt.title('Heart Disease by Chest Pain Type')
plt.show()
```


Heart Disease by Chest Pain Type
target
0
1
Count
100
80
60
40
20
0
0
1
2
3
Chest Pain Type