In [1]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

import warnings
warnings.filterwarnings('ignore')

%matplotlib inline
```

In [2]:

```
heart=pd.read_csv(r"D:\Full Stack Data Science\18 Aug\18th_resume project\
heart
```

Out[2]:

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	targe
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	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	..
298	57	0	0	140	241	0	1	123	1	0.2	1	0	3	
299	45	1	3	110	264	0	1	132	0	1.2	1	0	3	
300	68	1	0	144	193	1	1	141	0	3.4	1	2	3	
301	57	1	0	130	131	0	1	115	1	1.2	1	1	3	
302	57	0	1	130	236	0	0	174	0	0.0	1	1	2	

303 rows × 14 columns

Dataset Description

The dataset contains several columns which are as follows -

- age : age in years
- sex : (1 = male; 0 = female)
- cp : chest pain type
- trestbps : resting blood pressure (in mm Hg on admission to the hospital)
- chol : serum cholestoral in mg/dl
- fbs : (fasting blood sugar > 120 mg/dl) (1 = true; 0 = false)
- restecg : resting electrocardiographic results
- thalach : maximum heart rate achieved
- exang : exercise induced angina (1 = yes; 0 = no)
- oldpeak : ST depression induced by exercise relative to rest
- slope : the slope of the peak exercise ST segment

- ca : number of major vessels (0-3) colored by flourosopy
- thal : 3 = normal; 6 = fixed defect; 7 = reversable defect
- target : 1 or 0

In [3]:
```
1 heart.shape
```

Out[3]:
```
(303, 14)
```

In [4]:
```
1 heart.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 [5]:
```
1 heart.isnull().sum()
```

Out[5]:
```
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
```

No null values in dataset

In [6]:
```
heart.columns
```

Out[6]:
```
Index(['age', 'sex', 'cp', 'trestbps', 'chol', 'fbs', 'restecg', 'thalach',
       'exang', 'oldpeak', 'slope', 'ca', 'thal', 'target'],
      dtype='object')
```

In [7]:
```
## Statistical properties of numerical dataset.
heart.describe()
```

Out[7]:

	age	sex	cp	trestbps	chol	fbs	restecg	
count	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303
mean	54.366337	0.683168	0.966997	131.623762	246.264026	0.148515	0.528053	149
std	9.082101	0.466011	1.032052	17.538143	51.830751	0.356198	0.525860	22
min	29.000000	0.000000	0.000000	94.000000	126.000000	0.000000	0.000000	71
25%	47.500000	0.000000	0.000000	120.000000	211.000000	0.000000	0.000000	133
50%	55.000000	1.000000	1.000000	130.000000	240.000000	0.000000	1.000000	153
75%	61.000000	1.000000	2.000000	140.000000	274.500000	0.000000	1.000000	166
max	77.000000	1.000000	3.000000	200.000000	564.000000	1.000000	2.000000	202

Statistical properties of character variables

- heart.describe(include='object')

Statistical properties of all variables

- heart.describe(include='all')

Univariate Analysis

- Our dependent variable is target i.e Patient has heart disease or not.

First we check the number of unique values in target variable.

In [9]:
```
heart.target.nunique()
```

Out[9]:
```
2
```

In [8]:
```
heart.target.unique()
```

Out[8]:
```
array([1, 0], dtype=int64)
```

- There is 2 unique values 0 & 1

- Presence of heart disease is denoted by 1 & absence of heart disease is denoted by 0.

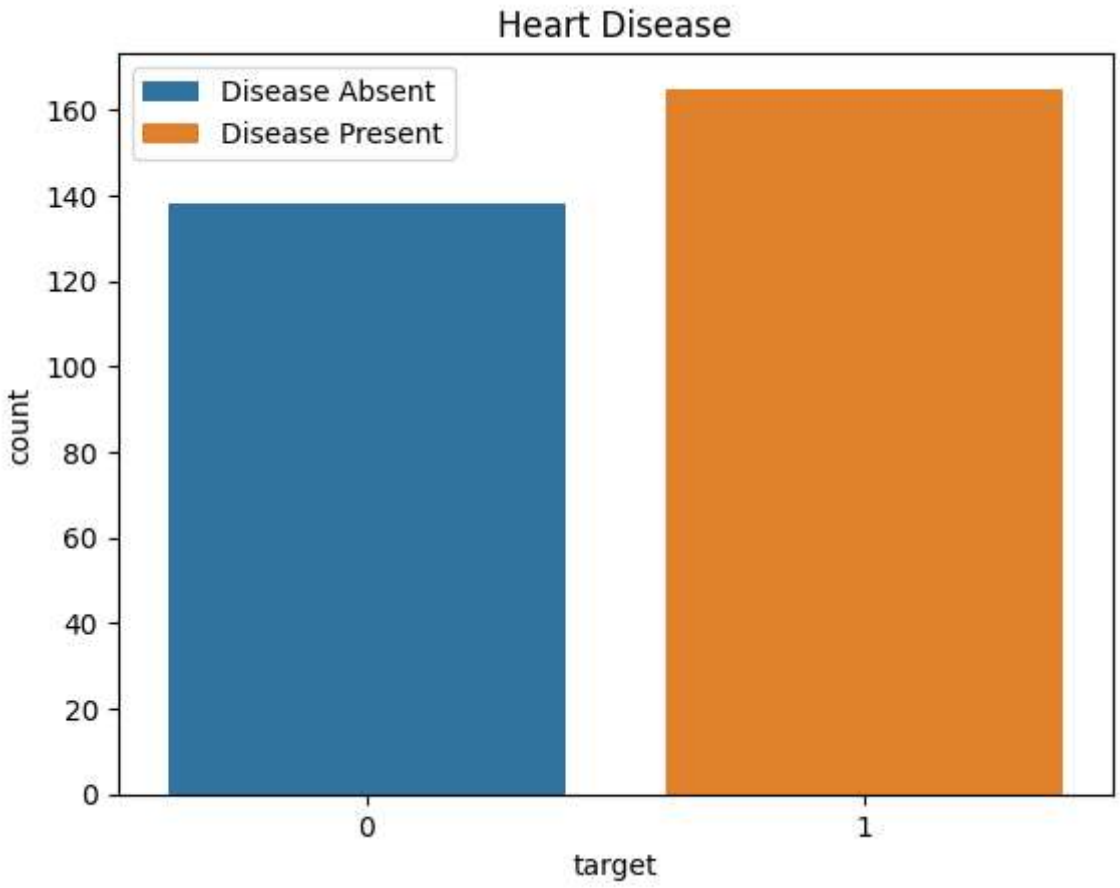
Frequency distribution of target variable.

In [10]:
```
1 heart.target.value_counts()
```

Out[10]:
```
target
1    165
0    138
Name: count, dtype: int64
```

- 1 is stands for presence of disease,there are 165 patients are suffering from heart disease.
- 0 is stands for absence of heart disease, there is 138 patient do not have heart disease.

In [12]:
```
1 f=sns.countplot(data=heart,x='target',label=('Disease Absent','Disease Pr
2 f.set_title('Heart Disease')
3 f=plt.legend()
```



Frequency of target variable with respect gender of patient.

In [13]:
```
1 heart.sex.value_counts()
```

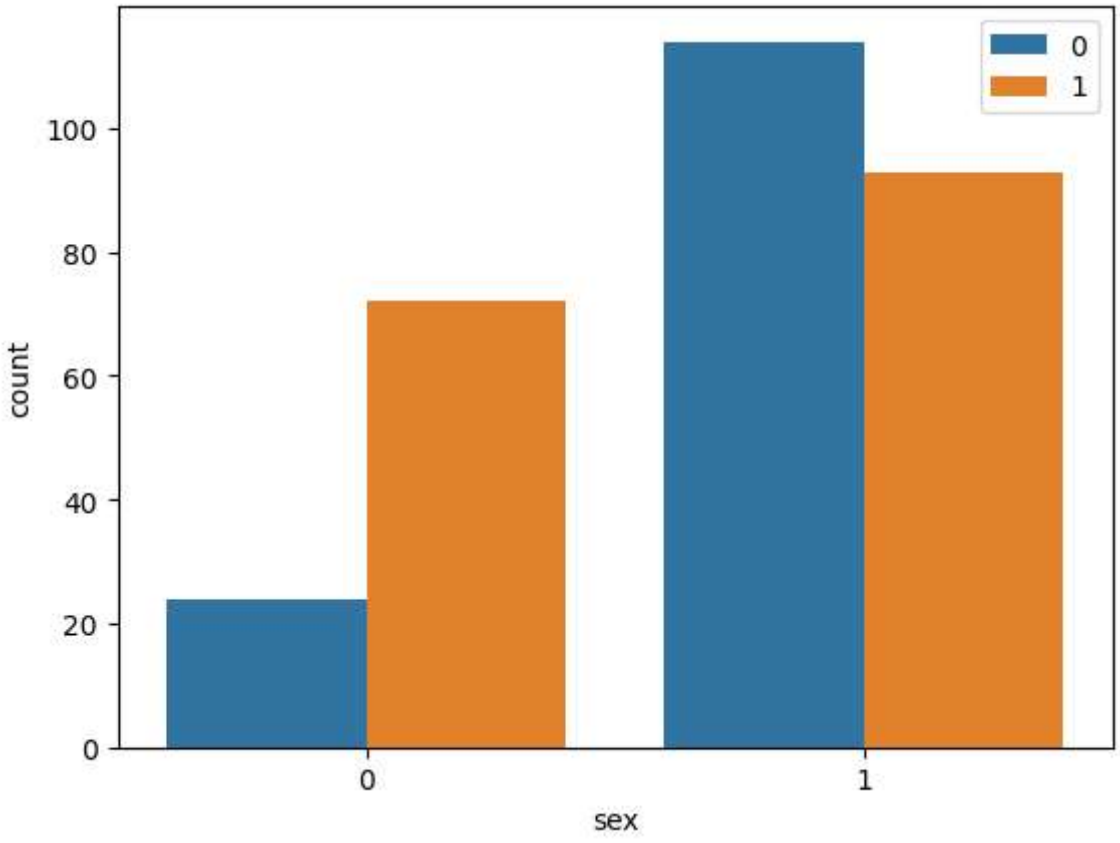
Out[13]:
```
sex
1    207
0     96
Name: count, dtype: int64
```

In [14]:
```
1 heart.groupby('sex')['target'].value_counts()
```

Out[14]:
```
sex  target
0    1          72
     0          24
1    0         114
     1          93
Name: count, dtype: int64
```

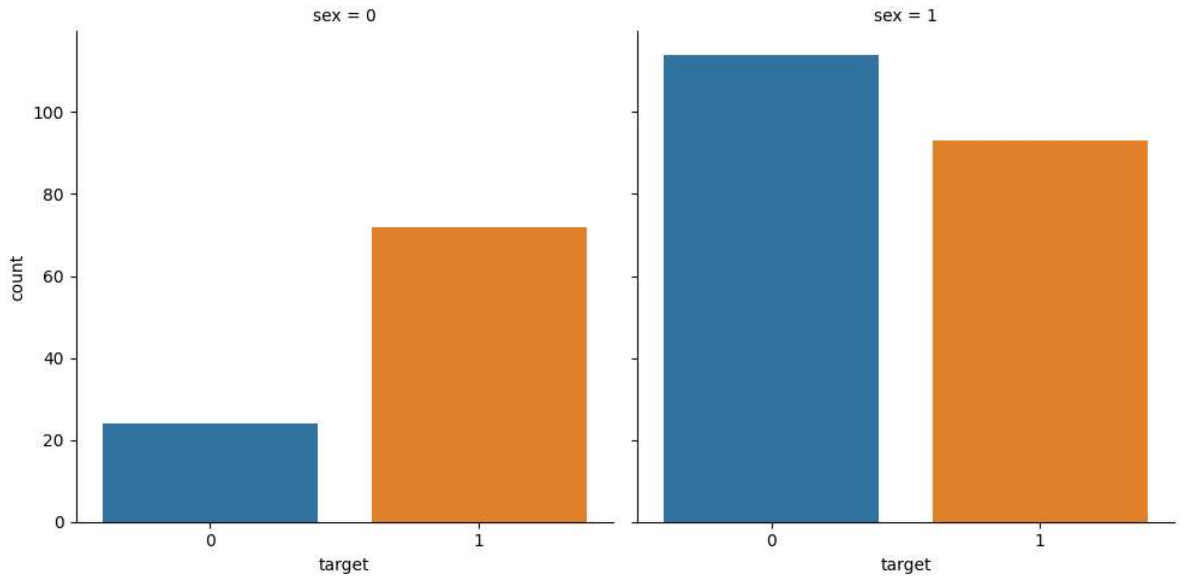
In [17]:
```
1 g=sns.countplot(data=heart,x='sex',hue='target',orient='h')
2 plt.legend()
```

Out[17]:
```
<matplotlib.legend.Legend at 0x2a0a6b56a50>
```

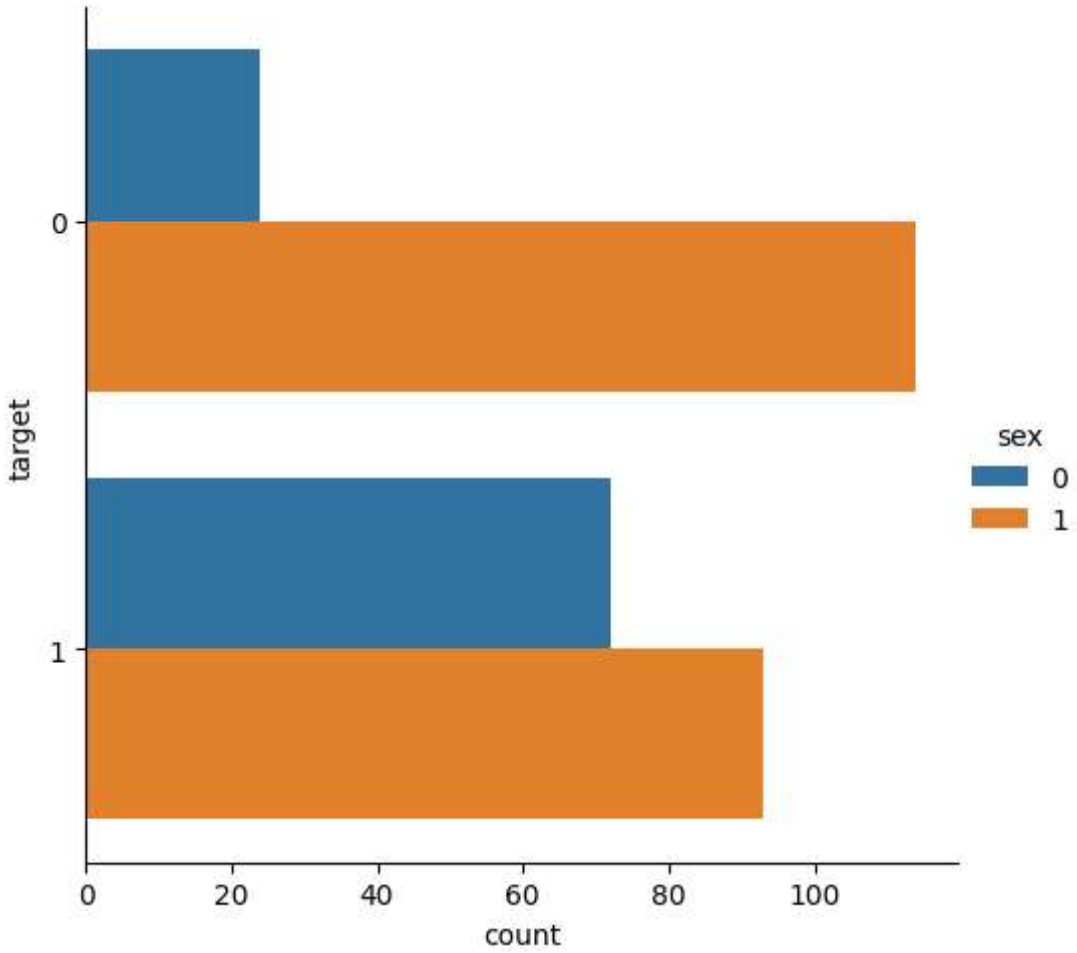


- Total 96 females are in dataset in that 72 females having heart disease remaining 24 do not have heart disease.
- Total 207 males are in dataset in that 93 males having heart disease and remaining 114 do not have heart disease.

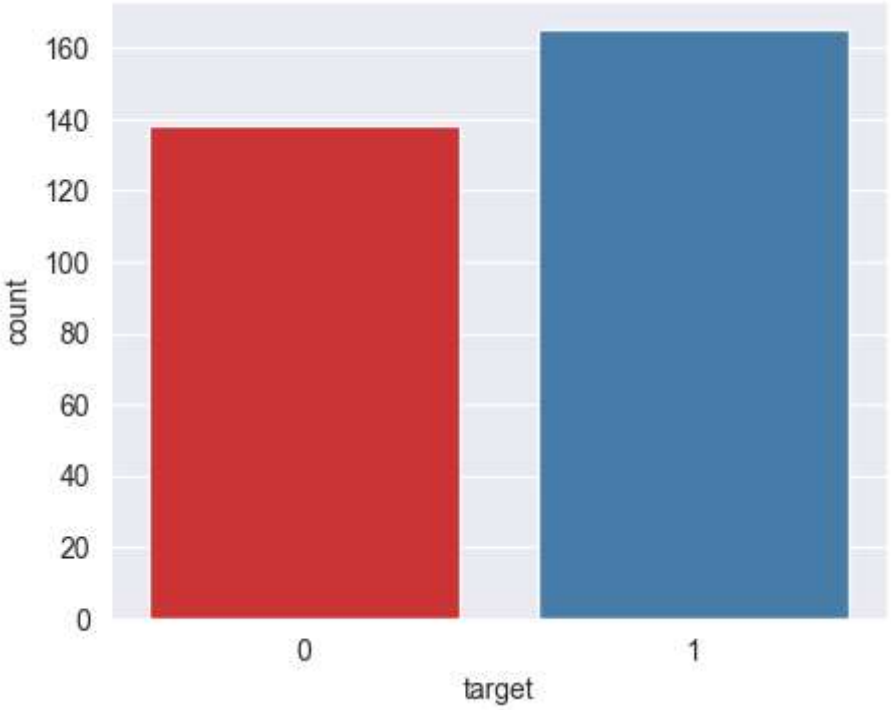
In [18]:
```
1 f=sns.catplot(data=heart,x='target',col='sex',kind='count',height=5,aspec
```



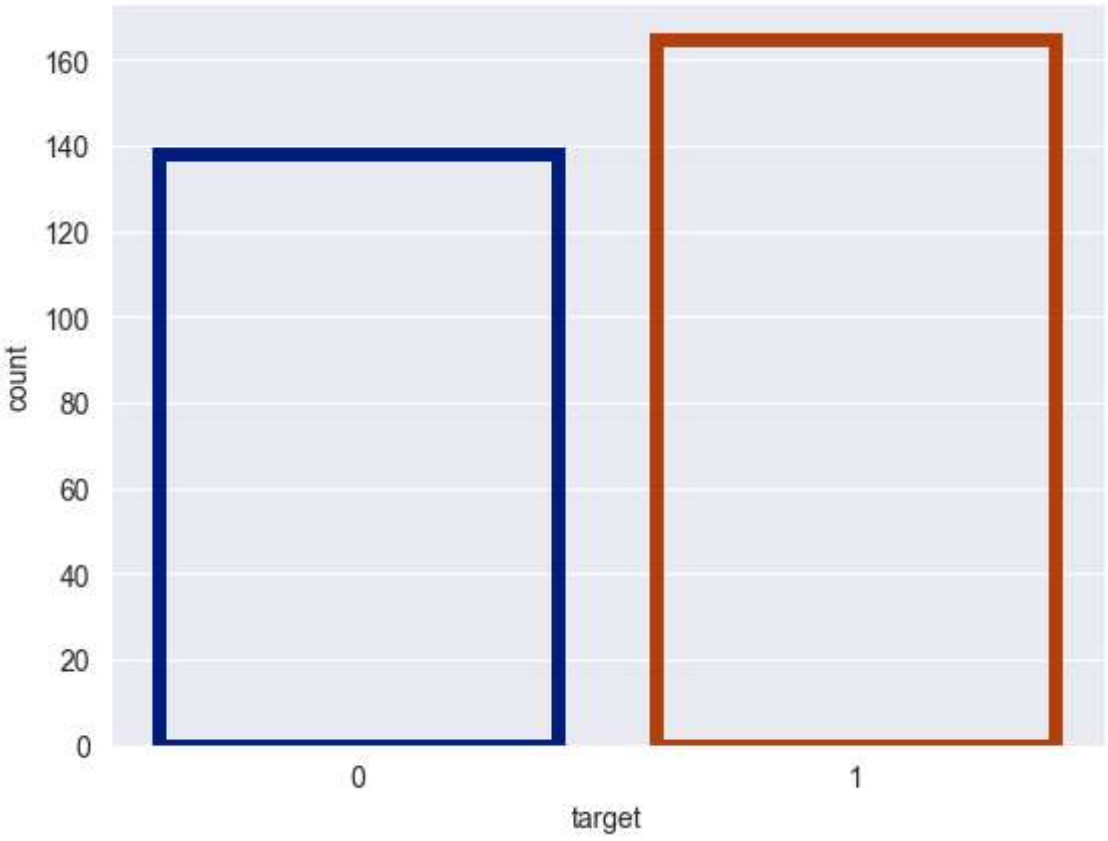
In [19]:
```
1 f=sns.catplot(data=heart,y='target',hue='sex',kind='count',height=5,aspec
```



In [119]:
```
1 f,ax=plt.subplots(figsize=(5,4))
2 s=sns.countplot(data=heart,x='target',palette='Set1')  ## palette- To Cha
```

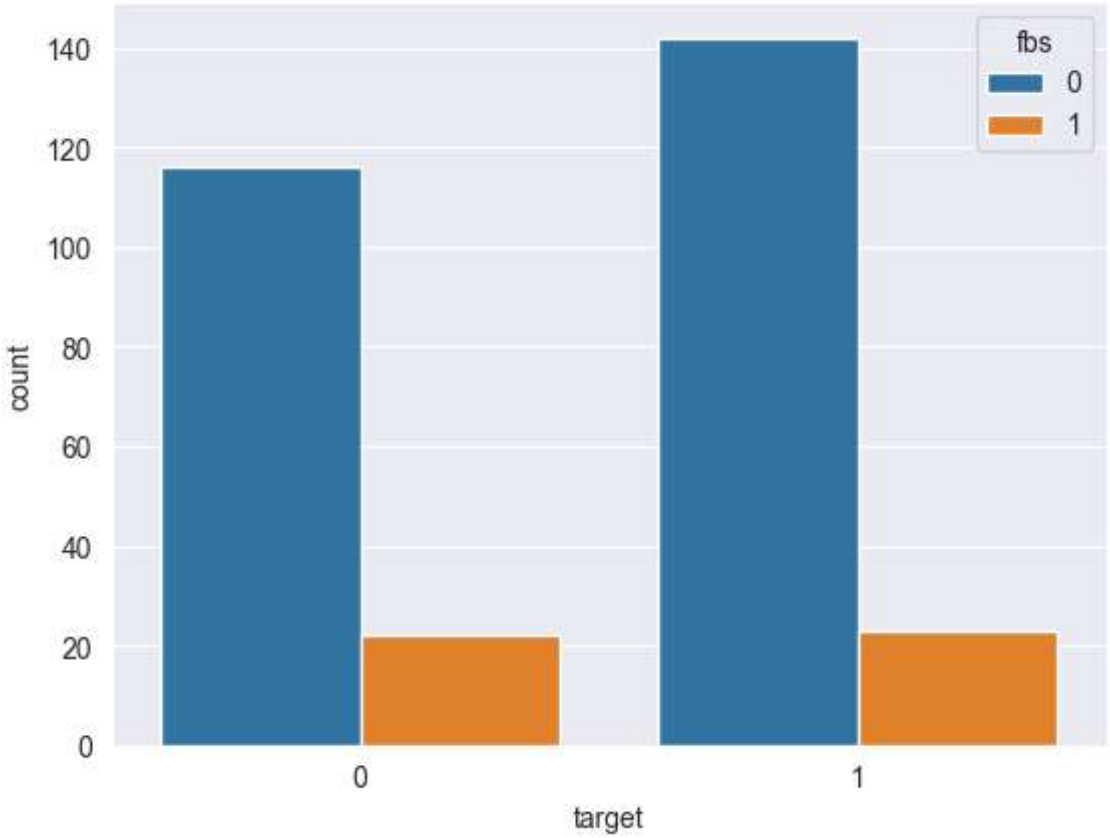


In [134]:
```
1 s=sns.countplot(data=heart,x='target',palette='Set1',facecolor=(0,0,0,0),
```

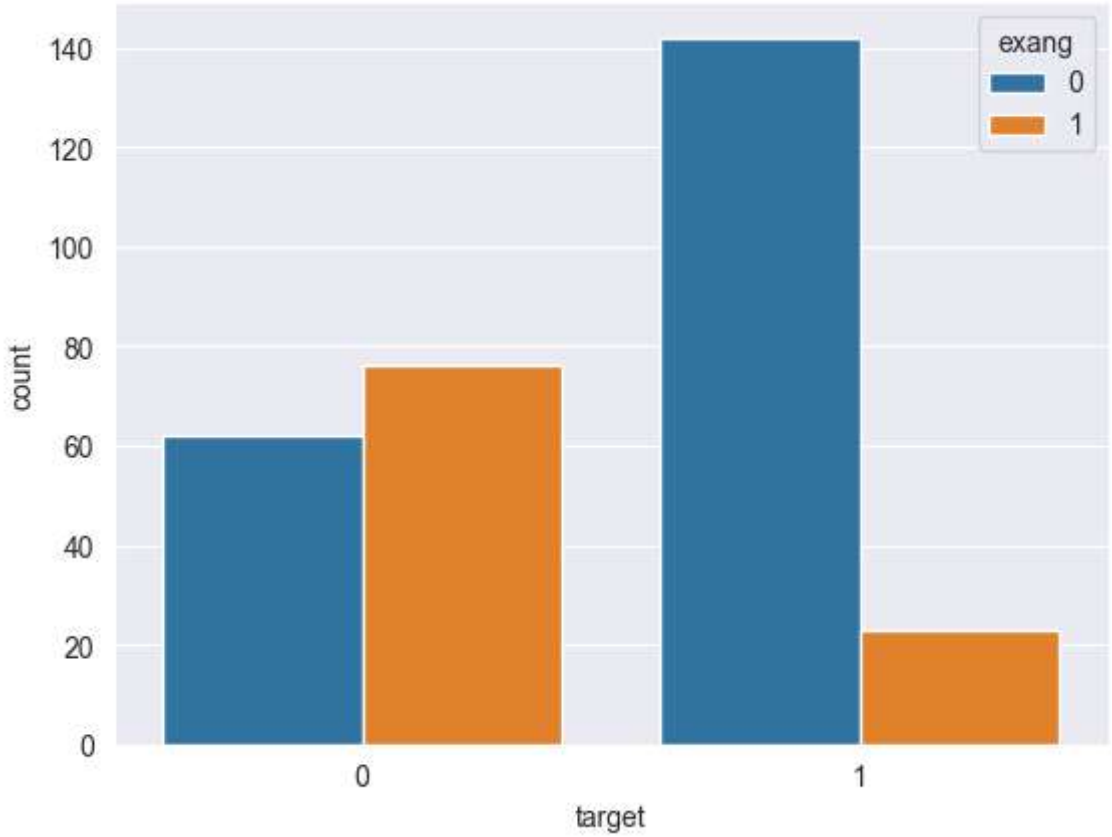


In [121]:

```
## For fbs-fasting blood sugar
ax=sns.countplot(data=heart,x='target',hue='fbs')
```



In [122]:
```
## for exang-exercise induced angina
ax=sns.countplot(data=heart,x='target',hue='exang')
```



Bivariate Analysis

Estimate correlation coefficient

In [25]:
```
correlation=heart.corr()
correlation
```

Out[25]:

	age	sex	cp	trestbps	chol	fbs	restecg	thalach
age	1.000000	-0.098447	-0.068653	0.279351	0.213678	0.121308	-0.116211	-0.398522
sex	-0.098447	1.000000	-0.049353	-0.056769	-0.197912	0.045032	-0.058196	-0.044020
cp	-0.068653	-0.049353	1.000000	0.047608	-0.076904	0.094444	0.044421	0.295762
trestbps	0.279351	-0.056769	0.047608	1.000000	0.123174	0.177531	-0.114103	-0.046698
chol	0.213678	-0.197912	-0.076904	0.123174	1.000000	0.013294	-0.151040	-0.009940
fbs	0.121308	0.045032	0.094444	0.177531	0.013294	1.000000	-0.084189	-0.008567
restecg	-0.116211	-0.058196	0.044421	-0.114103	-0.151040	-0.084189	1.000000	0.044123
thalach	-0.398522	-0.044020	0.295762	-0.046698	-0.009940	-0.008567	0.044123	1.000000
exang	0.096801	0.141664	-0.394280	0.067616	0.067023	0.025665	-0.070733	-0.378812
oldpeak	0.210013	0.096093	-0.149230	0.193216	0.053952	0.005747	-0.058770	-0.344187
slope	-0.168814	-0.030711	0.119717	-0.121475	-0.004038	-0.059894	0.093045	0.386784
ca	0.276326	0.118261	-0.181053	0.101389	0.070511	0.137979	-0.072042	-0.213177
thal	0.068001	0.210041	-0.161736	0.062210	0.098803	-0.032019	-0.011981	-0.096439
target	-0.225439	-0.280937	0.433798	-0.144931	-0.085239	-0.028046	0.137230	0.421741

In [26]:
```
# Correlation of target variable

correlation.target.sort_values(ascending=False)
```

Out[26]:
```
target      1.000000
cp          0.433798
thalach     0.421741
slope       0.345877
restecg     0.137230
fbs        -0.028046
chol       -0.085239
trestbps   -0.144931
age        -0.225439
sex        -0.280937
thal       -0.344029
ca         -0.391724
oldpeak    -0.430696
exang      -0.436757
Name: target, dtype: float64
```

- Correlation between target and cp(chest pain type) is midely postive
- Therfore we analyze the interaction between these features and target variable

Analysis of target and cp (chest pain) variable

In [27]:
```
1 heart.cp.nunique()
```

Out[27]:
```
4
```

In [28]:
```
1 heart.cp.unique()
```

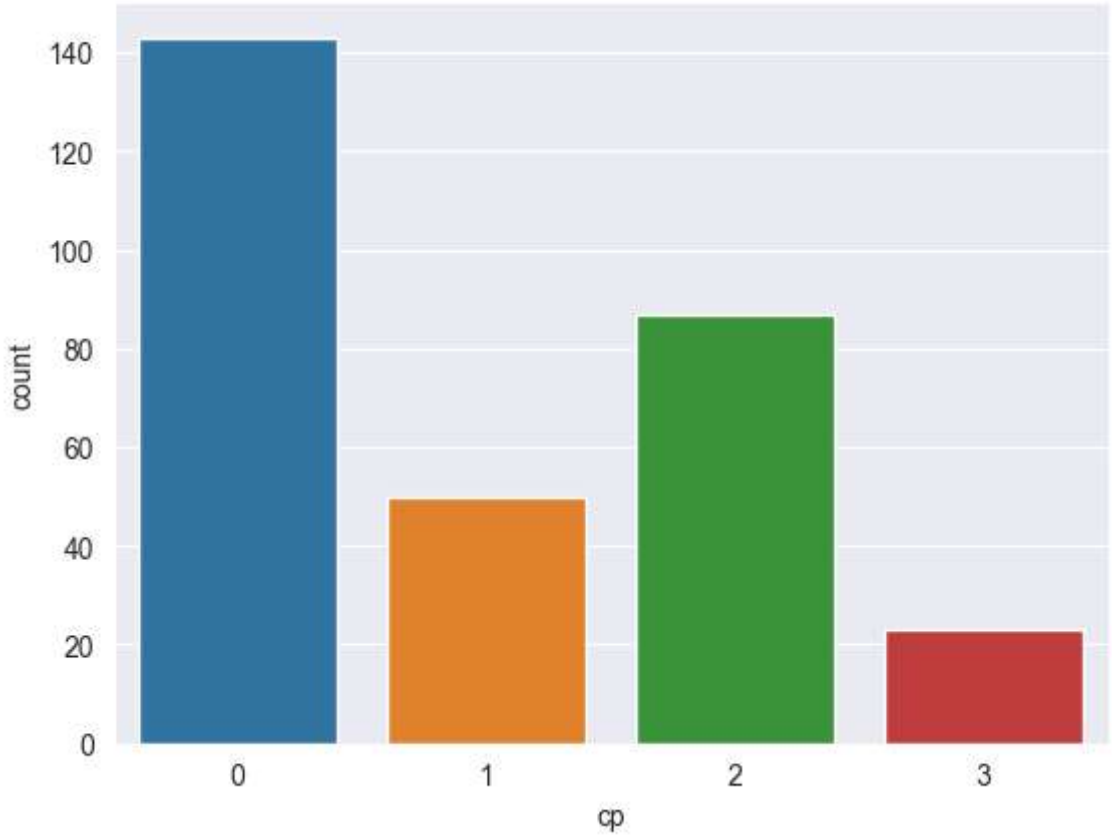
Out[28]:
```
array([3, 2, 1, 0], dtype=int64)
```

In [29]:
```
1 heart.cp.value_counts() ## To count the values
```

Out[29]:
```
cp
0    143
2     87
1     50
3     23
Name: count, dtype: int64
```

Visualize the frequency distribution of cp variable

In [123]:
```
1 cp=sns.countplot(data=heart,x='cp')
```

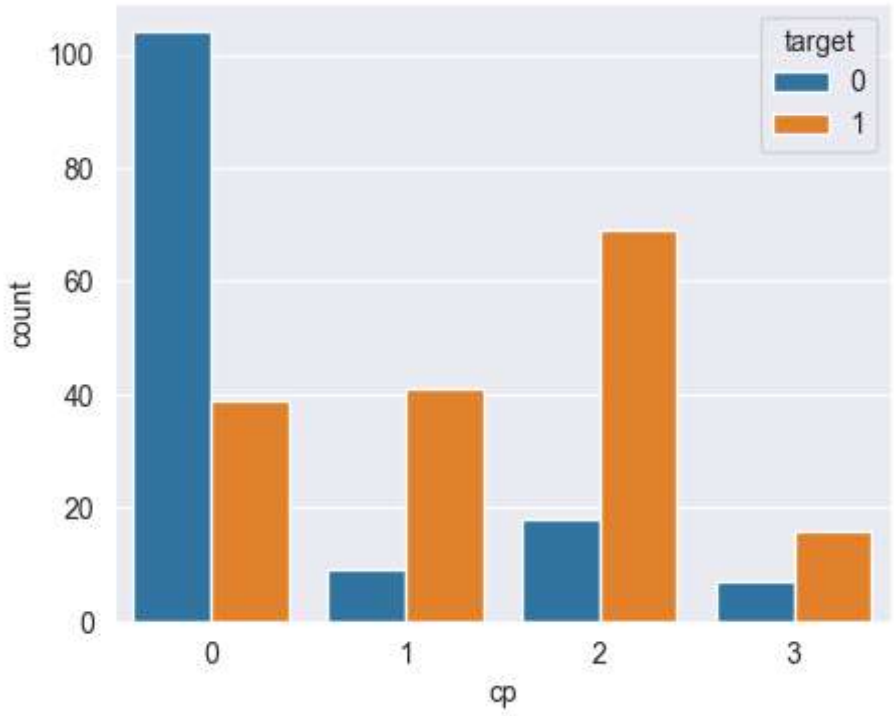


Frequency distribution of target variable wrt cp

In [31]:
```
1 heart.groupby('cp').target.value_counts()
```

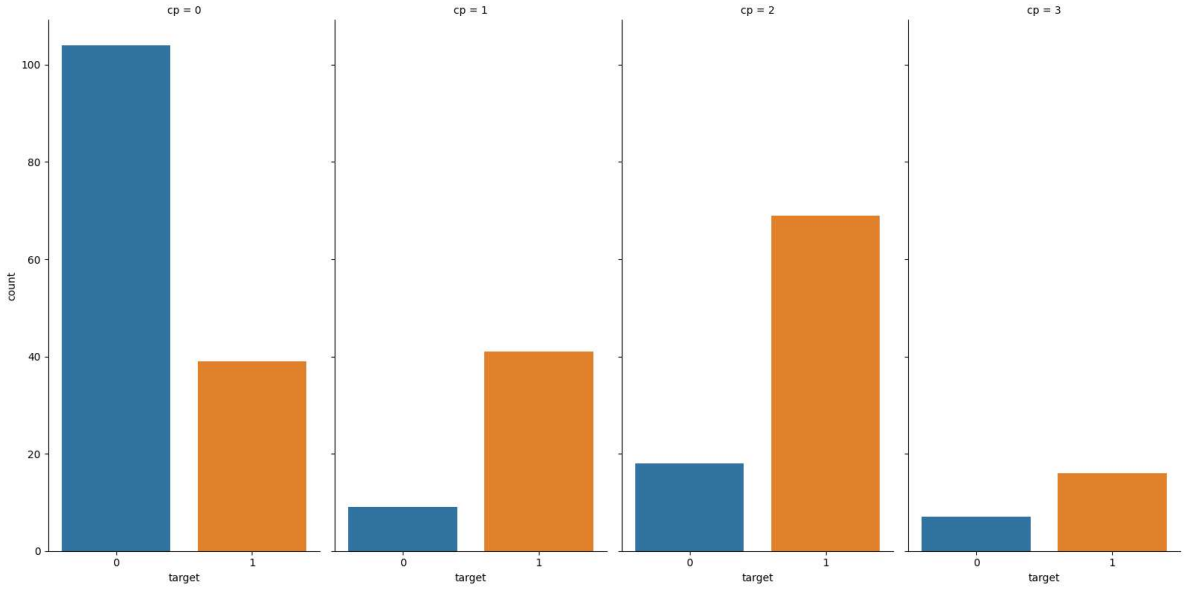
Out[31]:
```
cp  target
0   0         104
    1          39
1   1          41
    0           9
2   1          69
    0          18
3   1          16
    0           7
Name: count, dtype: int64
```

In [125]:
```
1 f,ax=plt.subplots(figsize=(5,4))
2
3 ax=sns.countplot(data=heart,x='cp',hue='target')
```



- When chest pain type is 0 then heart disease not present is not present.
- when chest pain type is 2 the presence rate of heart disease is very high.

In [36]:
```
1 ax=sns.catplot(data=heart,x='target',col='cp',kind='count',height=8,aspec
```



Analysis of target and thalach

- thalach- Maximum heart rate achieved

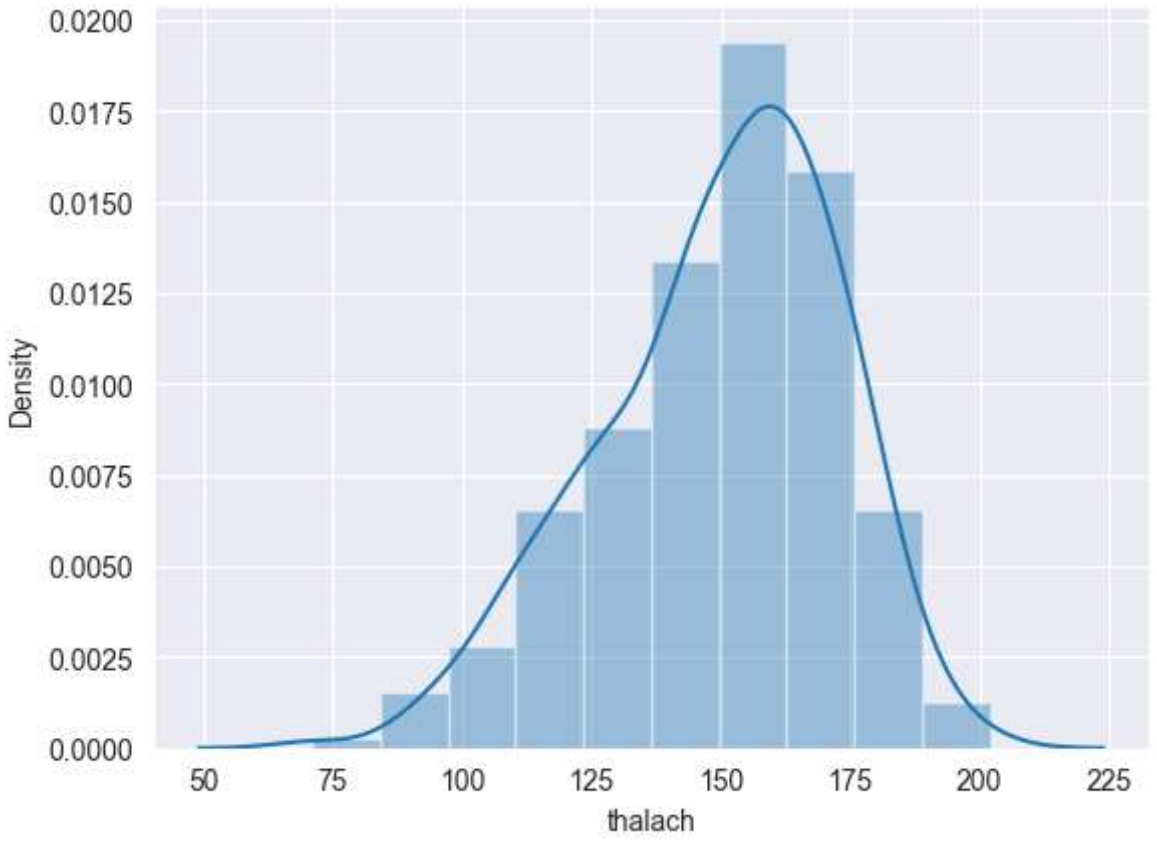
In [37]:
```
1 heart.thalach.nunique()
```

Out[37]: 91

- Number of unique values in thalach variable is 91. Hence it is numerical variable

Visualize the frequency distribution of thalach variable

In [38]:
```
sns.set_style('darkgrid')
x=heart.thalach
d=sns.distplot(x,bins=10)
```

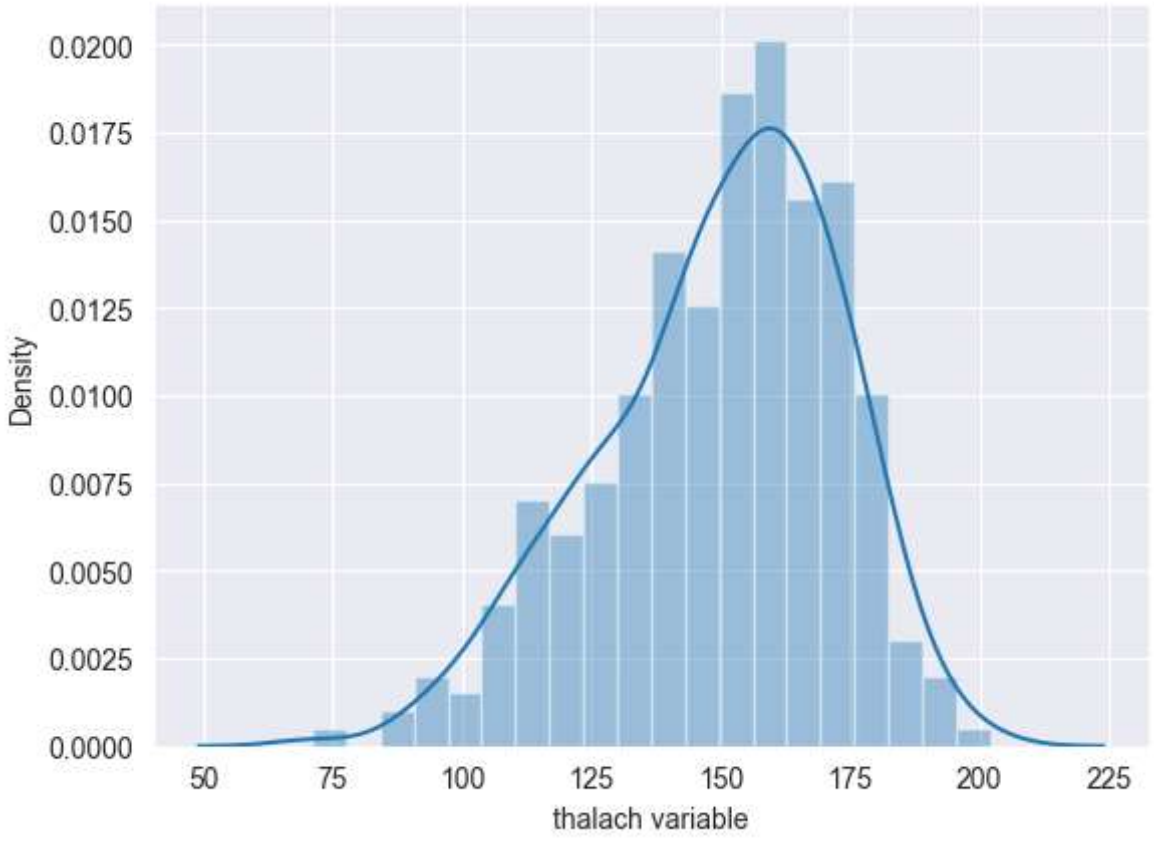


- we can see that thalach variable is slightly negatively skewed

We can use pandas series object to get an informative axis label as follows

In [40]:

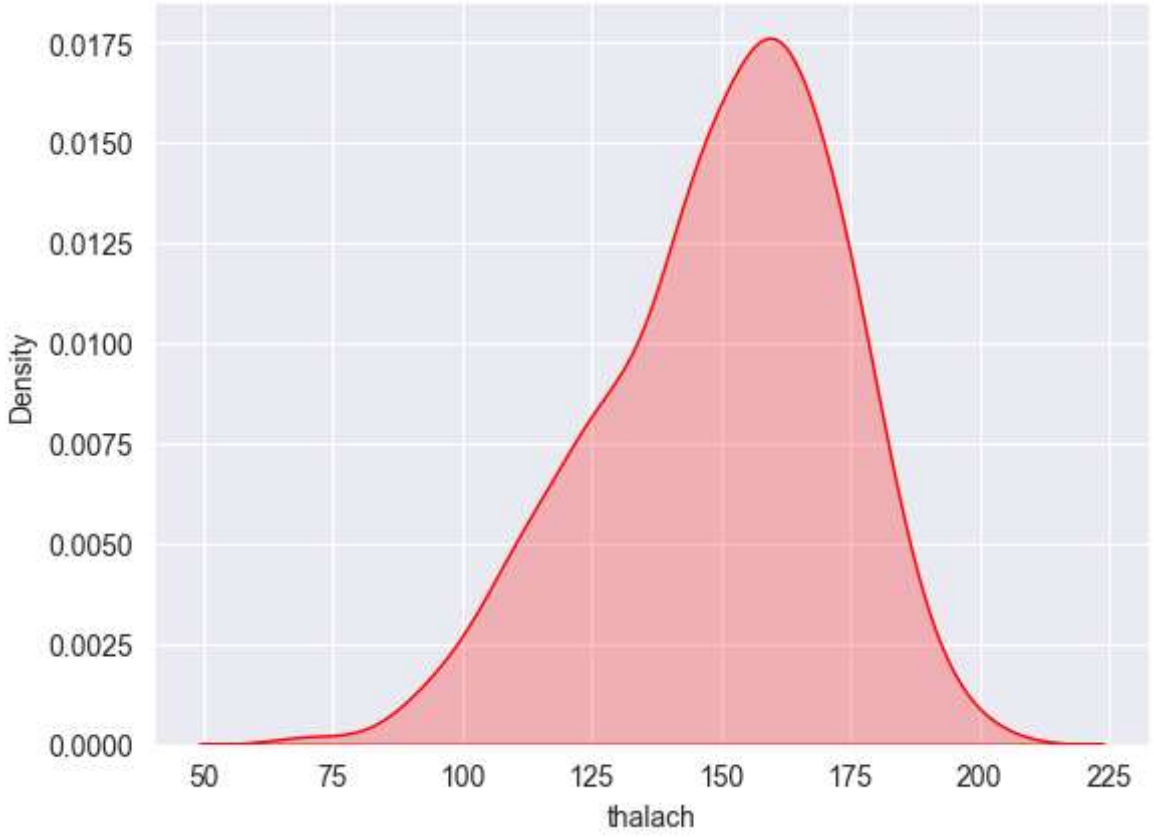
```
x=heart.thalach

x=pd.Series(x,name='thalach variable')
ax=sns.distplot(x,bins=20)
```



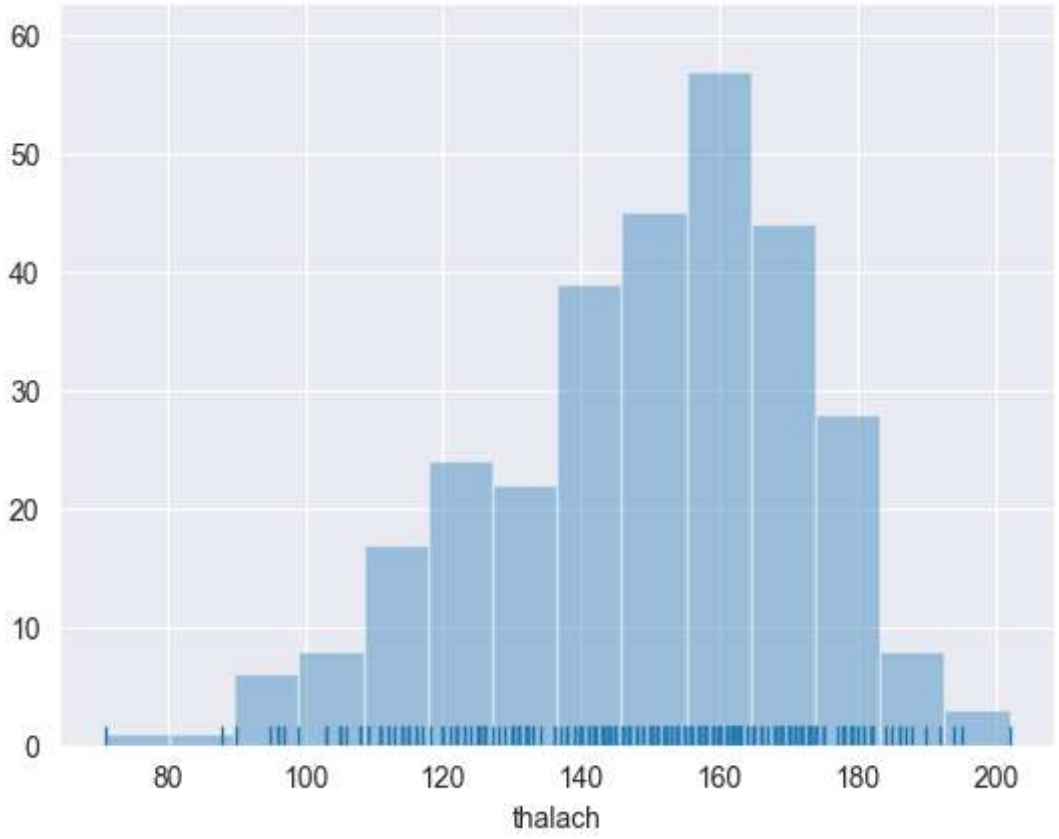
Kernel Density Estimation(KDE) Plot

In [41]:
```
1 sns.kdeplot(heart.thalach,color='r',shade=True)
```

Out[41]: <Axes: xlabel='thalach', ylabel='Density'>

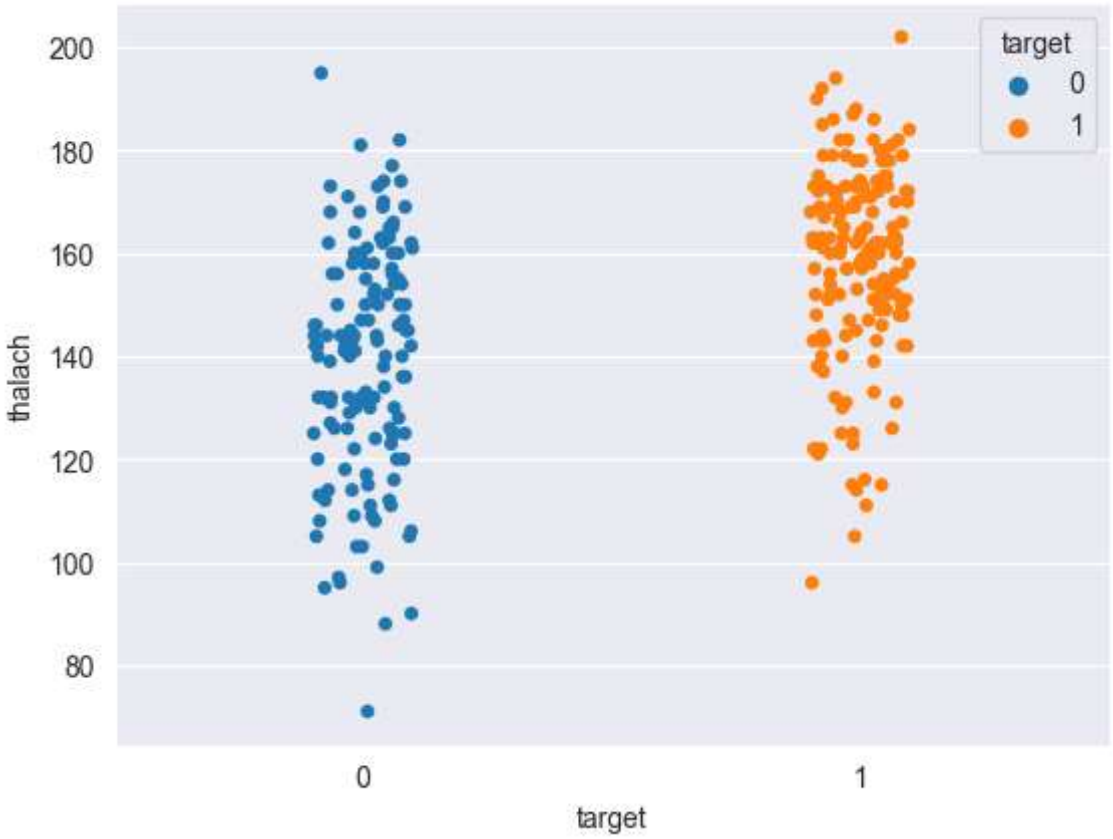


In [42]:
```
1 ax=sns.distplot(heart.thalach,kde=False,rug=True)
```



Visualize frequency distribution of thalach variable wrt target.

In [48]:
```
1 ax=sns.stripplot(data=heart,x='target',y='thalach',hue='target')
```

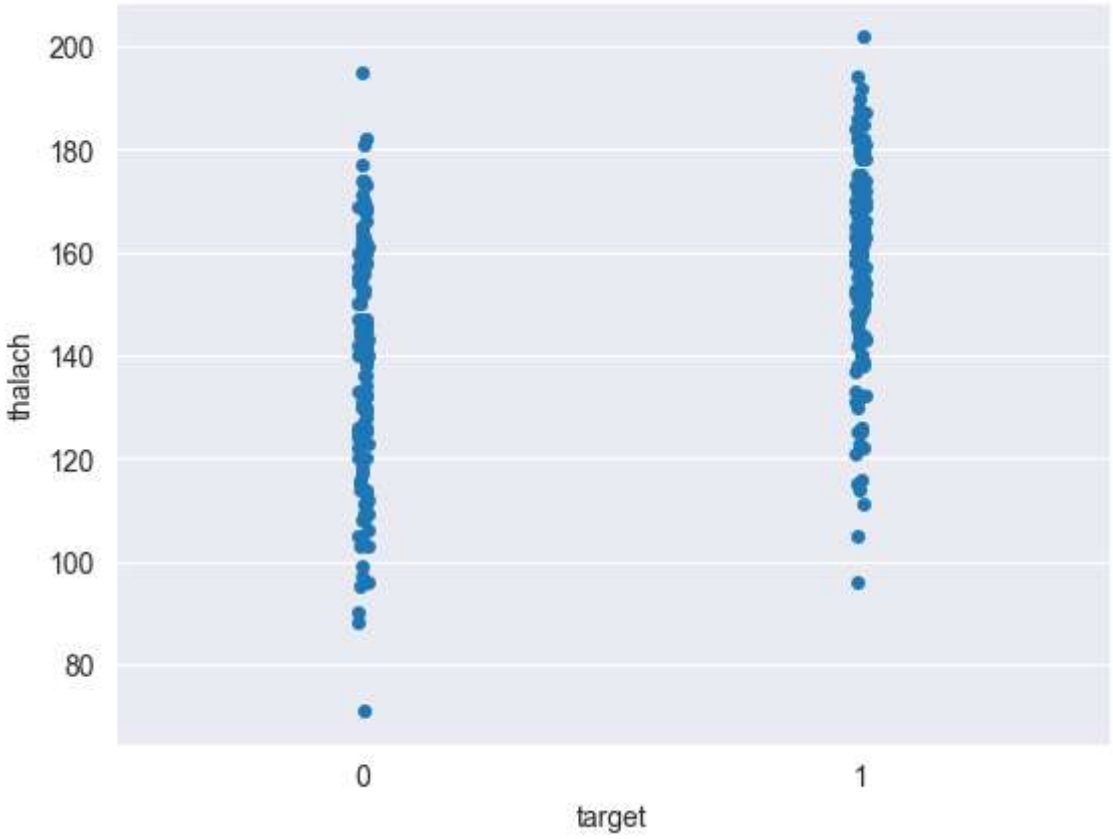


Interpretation

- we can see that those people suffering from heart disease(target=1) have relatively higher heart rate(thalach) as compared to people who are not suffering from heart disease(target=0)

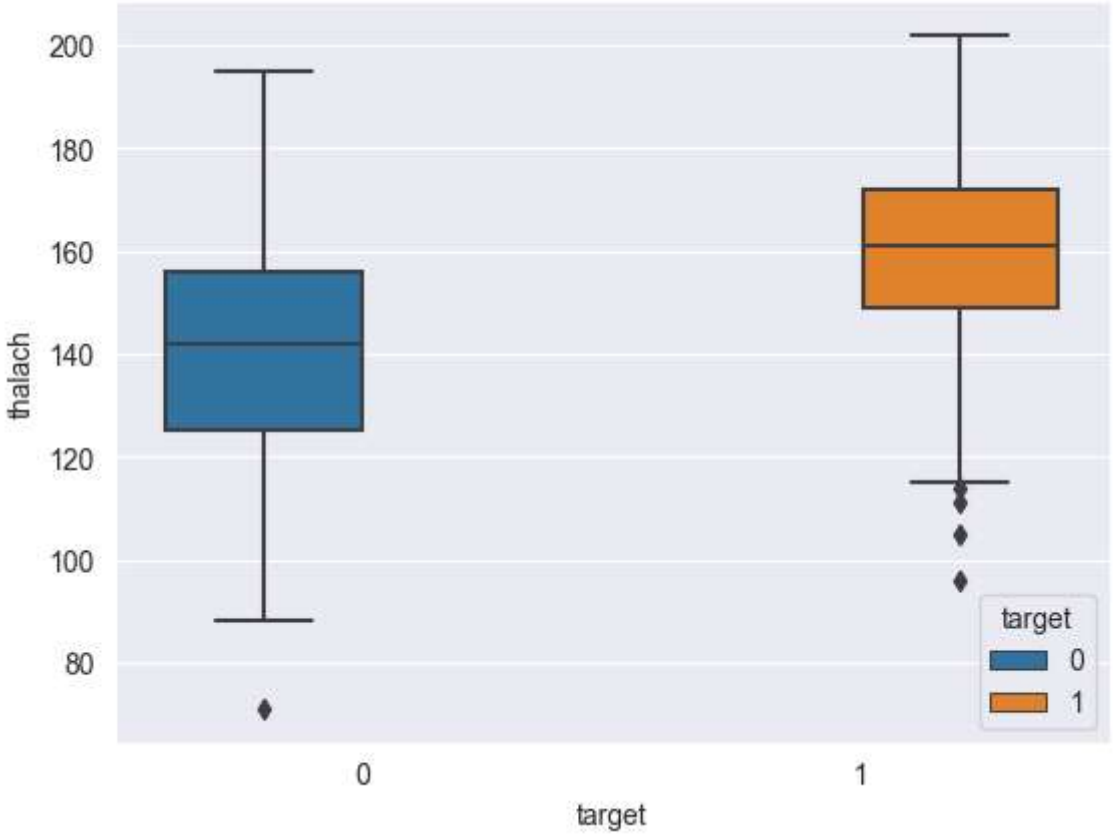
In [50]:
```
# We can add jitter to bring out the distribution of values as folows
sns.stripplot(data=heart,x='target',y='thalach',jitter=0.01)
```

Out[50]: <Axes: xlabel='target', ylabel='thalach'>



Visualize distribution of thalach variable w.r.t target with boxplot

In [51]:
```
1 b=sns.boxplot(data=heart,x='target',y='thalach',hue='target')
```



Interpretation

- The above boxplot confirms our finding that people suffering from heart disease (target = 1) have relatively higher heart rate (thalach) as compared to people who are not suffering from heart disease (target = 0).

Bivariate analysis interpretation

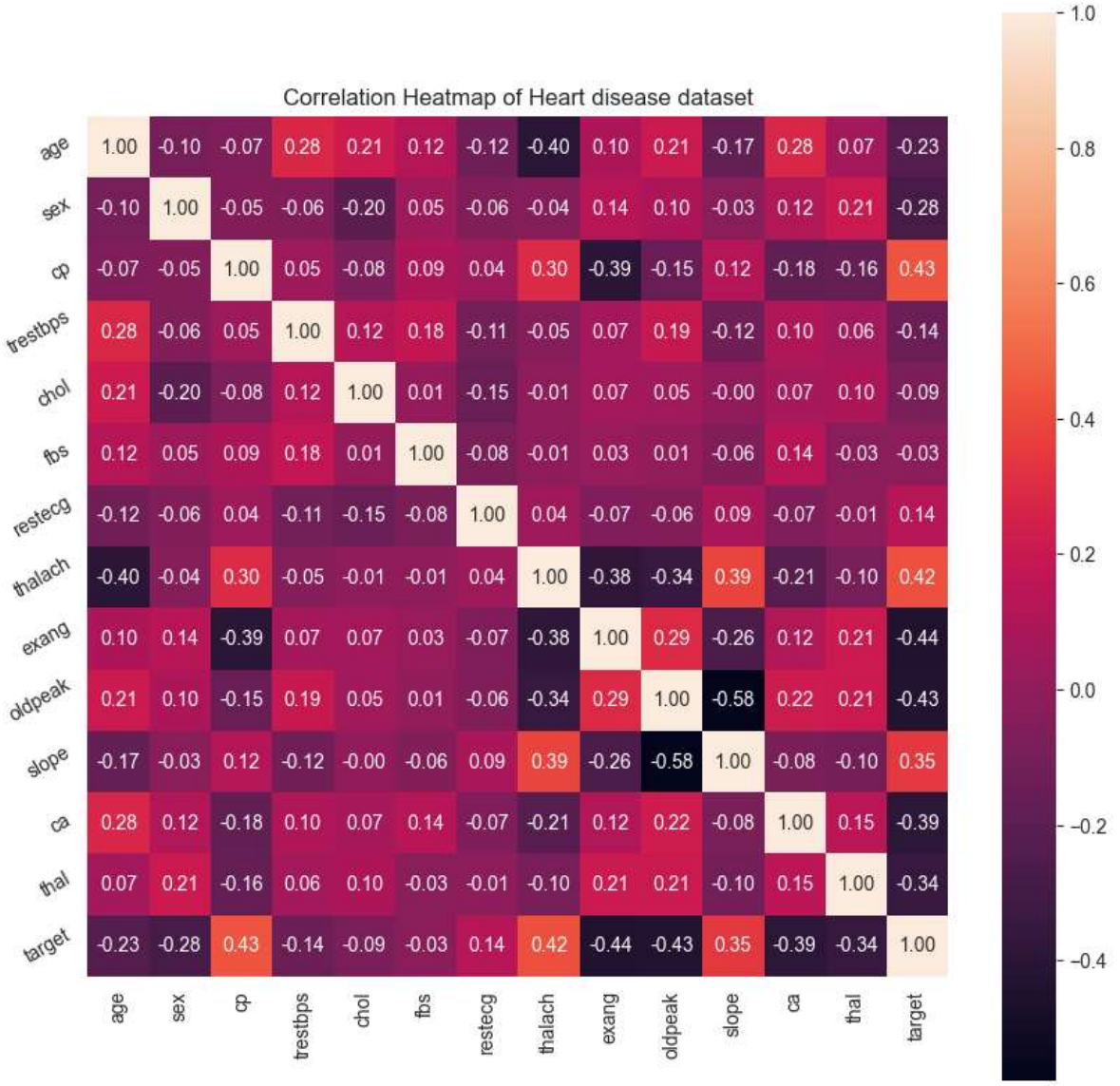
- There is no variable which has strong positive correlation with target variable.
- There is no variable which has strong negative correlation with target variable.
- There is no correlation between target and fbs.
- The cp and thalach variables are mildly positively correlated with target variable.
- We can see that the thalach variable is slightly negatively skewed.
- The people suffering from heart disease (target = 1) have relatively higher heart rate (thalach) as compared to people who are not suffering from heart disease (target = 0).
- The people suffering from heart disease (target = 1) have relatively higher heart rate (thalach) as compared to people who are not suffering from heart disease (target = 0).

Multivariate Analysis

- The objective of the multivariate analysis is to discover patterns and relationship in the dataset.

Heat Map

In [130]:
```
plt.figure(figsize=(10,10))
h=sns.heatmap(correlation,annot=True,square=True,fmt='.2f',linecolor='whi
plt.title('Correlation Heatmap of Heart disease dataset')
h.set_yticklabels(h.get_yticklabels(),rotation=30);
```



Interpretation

From the above correlation heat map, we can conclude that :-

- target and cp variable are mildly positively correlated (correlation coefficient = 0.43).
- target and thalach variable are also mildly positively correlated (correlation coefficient = 0.42).
- target and slope variable are weakly positively correlated (correlation coefficient = 0.35).
- target and exang variable are mildly negatively correlated (correlation coefficient = -0.44).
- target and oldpeak variable are also mildly negatively correlated (correlation coefficient = -0.43).
- target and ca variable are weakly negatively correlated (correlation coefficient = -0.39).
- target and thal variable are also waekly negatively correlated (correlation coefficient = -0.34).

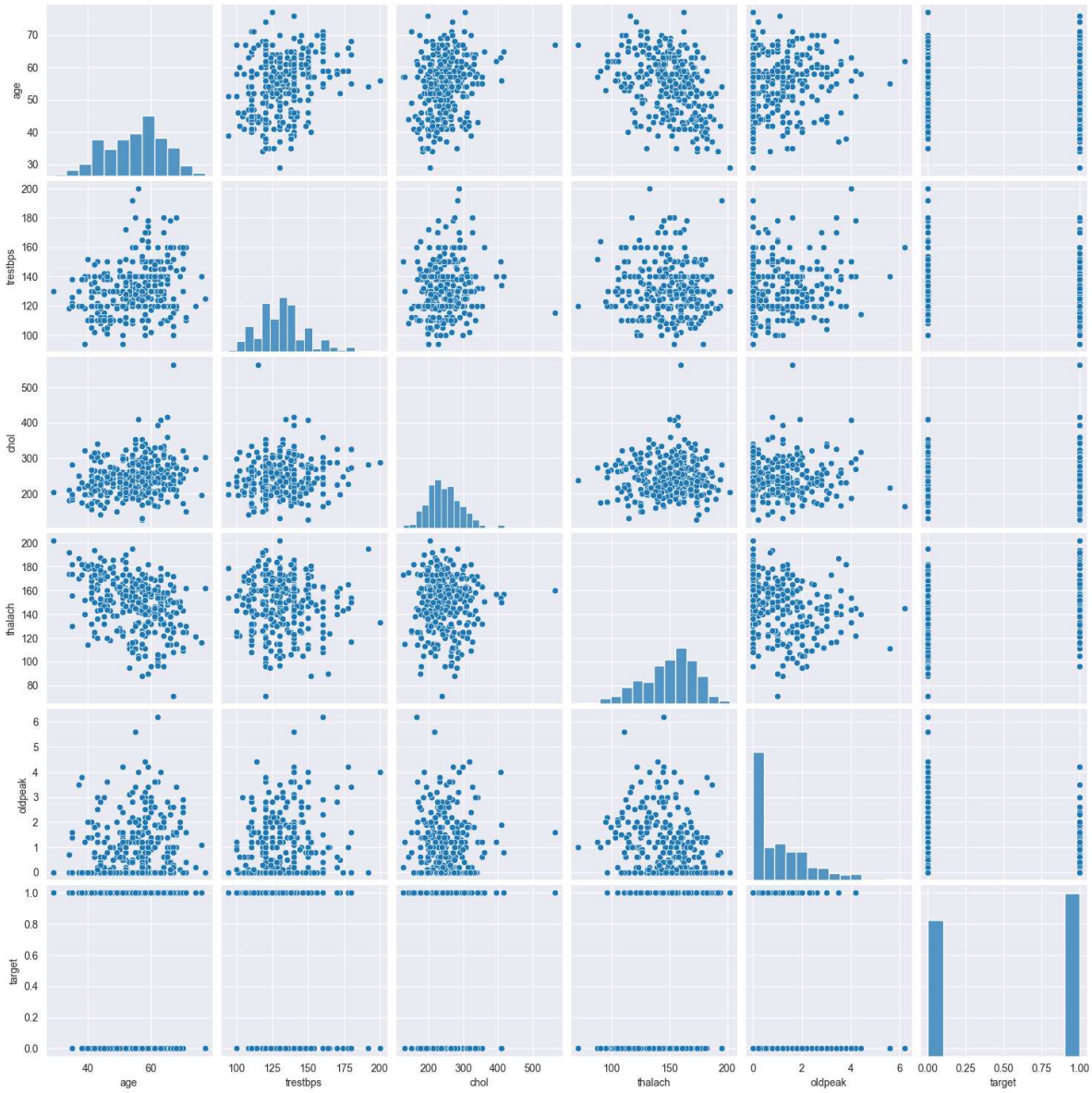
Pair Plot

In [64]:
```
1 heart.columns
```

Out[64]:
```
Index(['age', 'sex', 'cp', 'trestbps', 'chol', 'fbs', 'restecg', 'thalach',
       'exang', 'oldpeak', 'slope', 'ca', 'thal', 'target'],
      dtype='object')
```

In [71]:
```
## To check the relationship between target and other numerical variables

numerical_var=['age','trestbps','chol','thalach','oldpeak','target']
sns.pairplot(heart[numerical_var],kind='scatter')
```

Out[71]: <seaborn.axisgrid.PairGrid at 0x2a0af48c290>



Analysis of age variable

In [72]:
```
heart.age.nunique()
```

Out[72]: 41

In [73]:
```
1 heart.age.unique()
```

Out[73]:
```
array([63, 37, 41, 56, 57, 44, 52, 54, 48, 49, 64, 58, 50, 66, 43, 69, 59,
       42, 61, 40, 71, 51, 65, 53, 46, 45, 39, 47, 62, 34, 35, 29, 55, 60,
       67, 68, 74, 76, 70, 38, 77], dtype=int64)
```

In [74]:
```
1 heart.age.describe()
```

Out[74]:
```
count    303.000000
mean      54.366337
std        9.082101
min       29.000000
25%       47.500000
50%       55.000000
75%       61.000000
max       77.000000
Name: age, dtype: float64
```

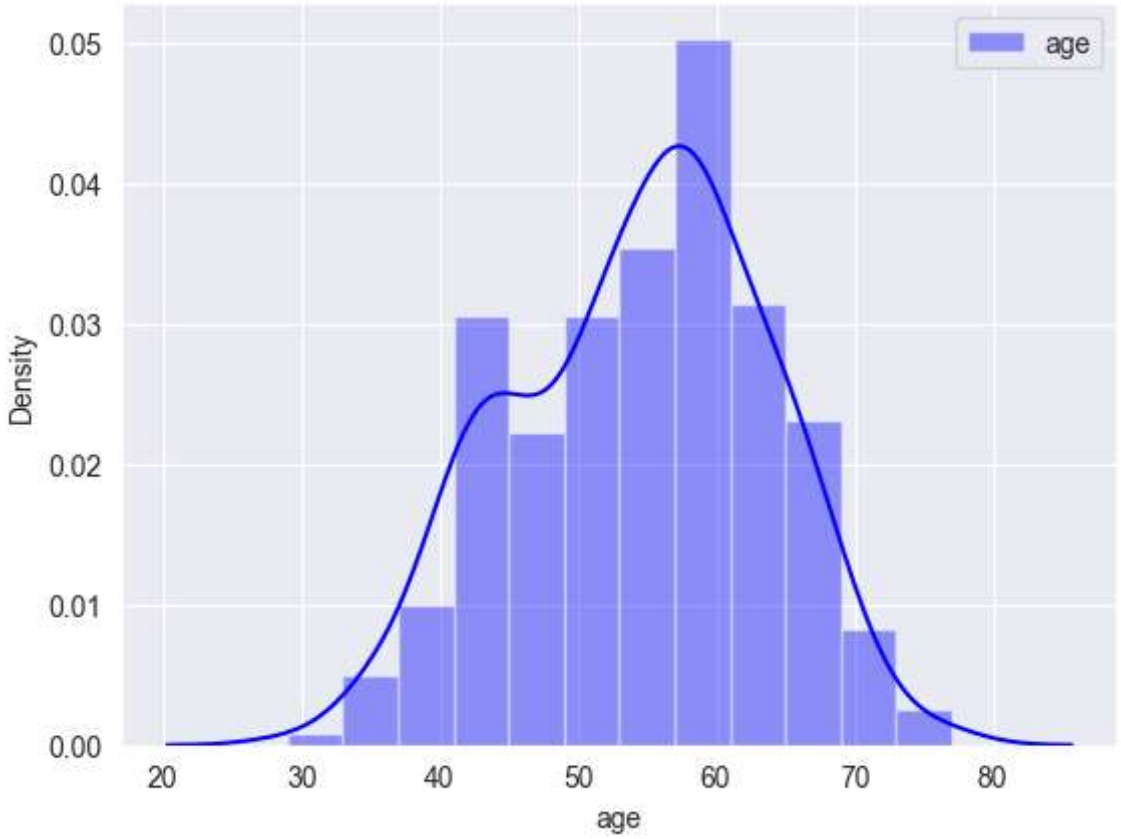
Interpretation

- Then min age is 29 and maximum age is 77
- The mean value of age is 54 years.

Plot the distribution of age variable

In [85]:
```
age=heart.age
a=sns.distplot(age,color='b',label='age')
plt.legend()
```

Out[85]:
```
<matplotlib.legend.Legend at 0x2a0be2f2a50>
```

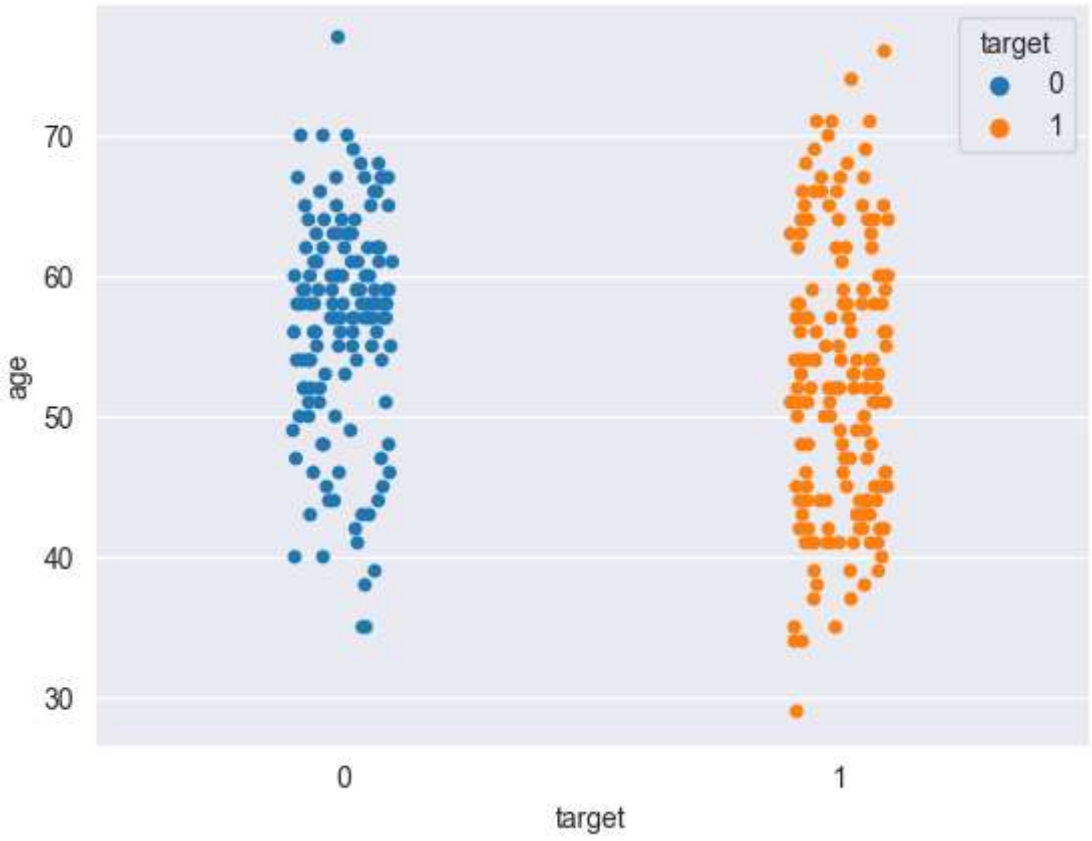


Interpretation

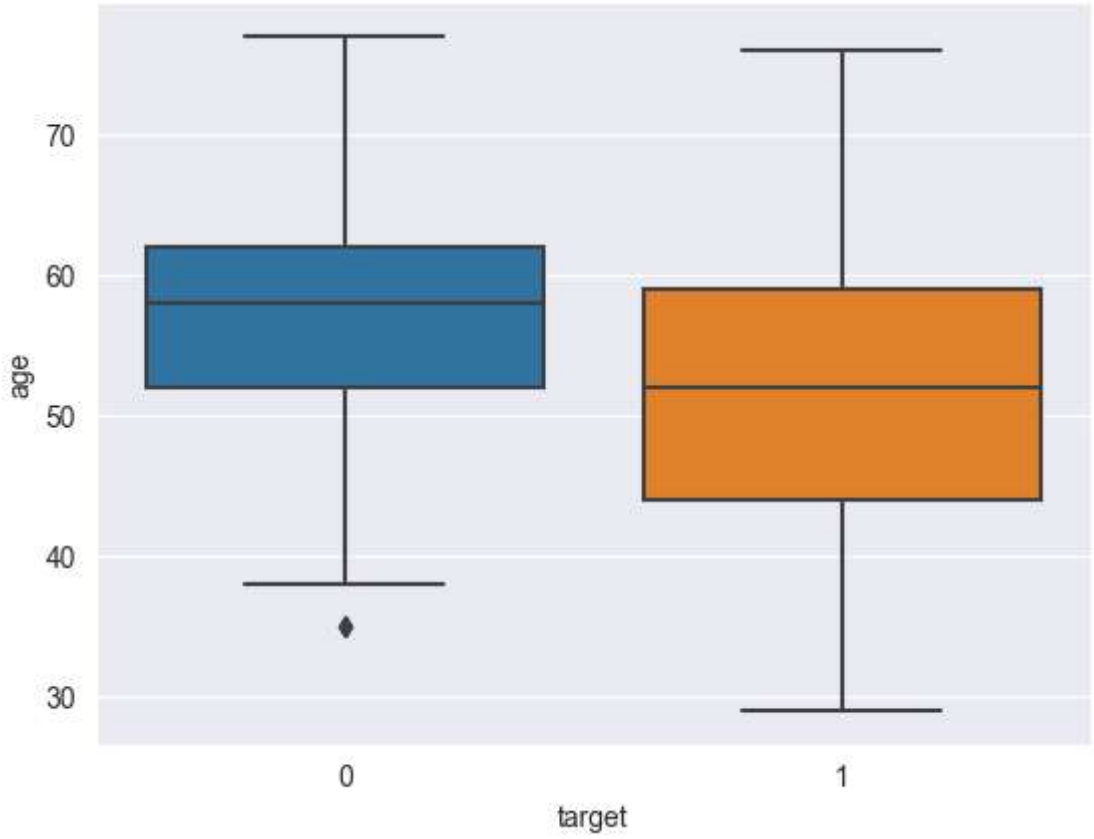
- The age variables distribution is approximately normal.

Analyze age and target variable

In [89]:
```
1 vis=sns.stripplot(data=heart,x='target',y='age',hue='target')
```



In [90]:
```
# Boxplot
b=sns.boxplot(data=heart,x='target',y='age')
```



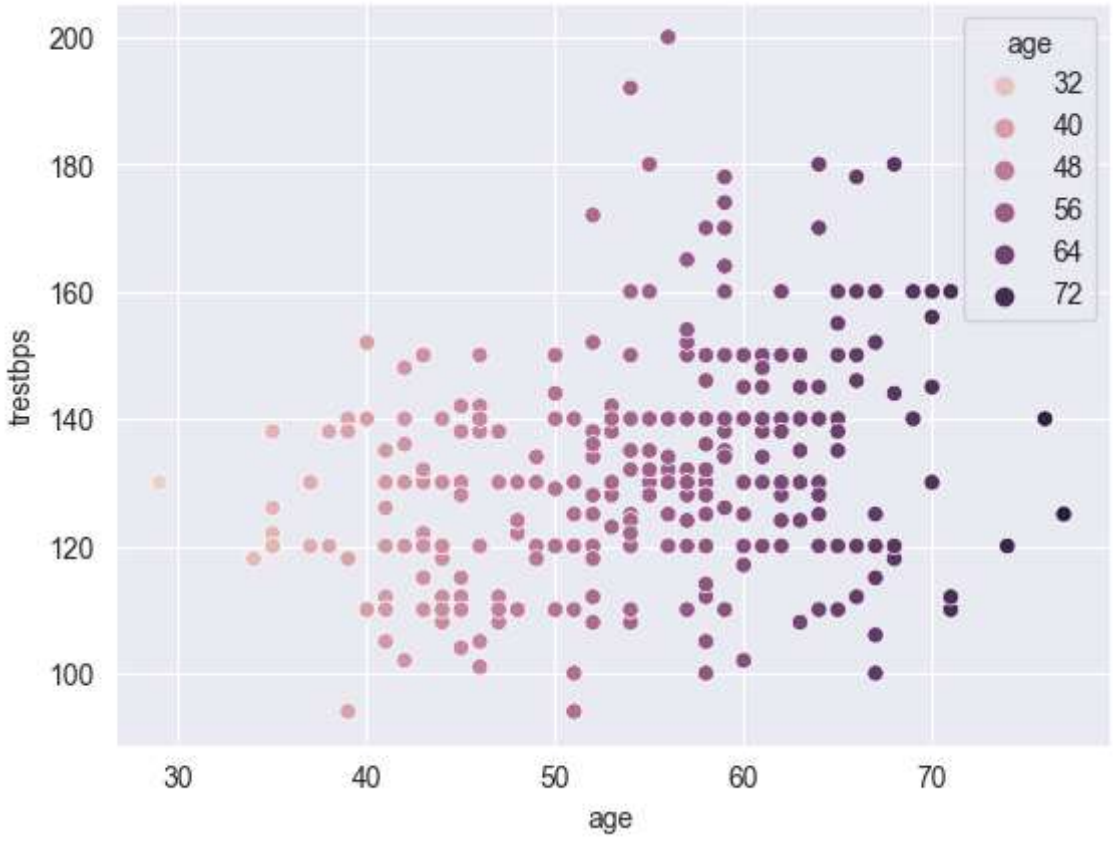
Interpretation

The above boxplot tells two different things :

- The mean age of the people who have heart disease is less than the mean age of the people who do not have heart disease.
- The dispersion or spread of age of the people who have heart disease is greater than the dispersion or spread of age of the people who do not have heart disease.

Analyse age and trestbps variable

In [94]:
```
1 s=sns.scatterplot(data=heart,x='age',y='trestbps',hue='age')
```

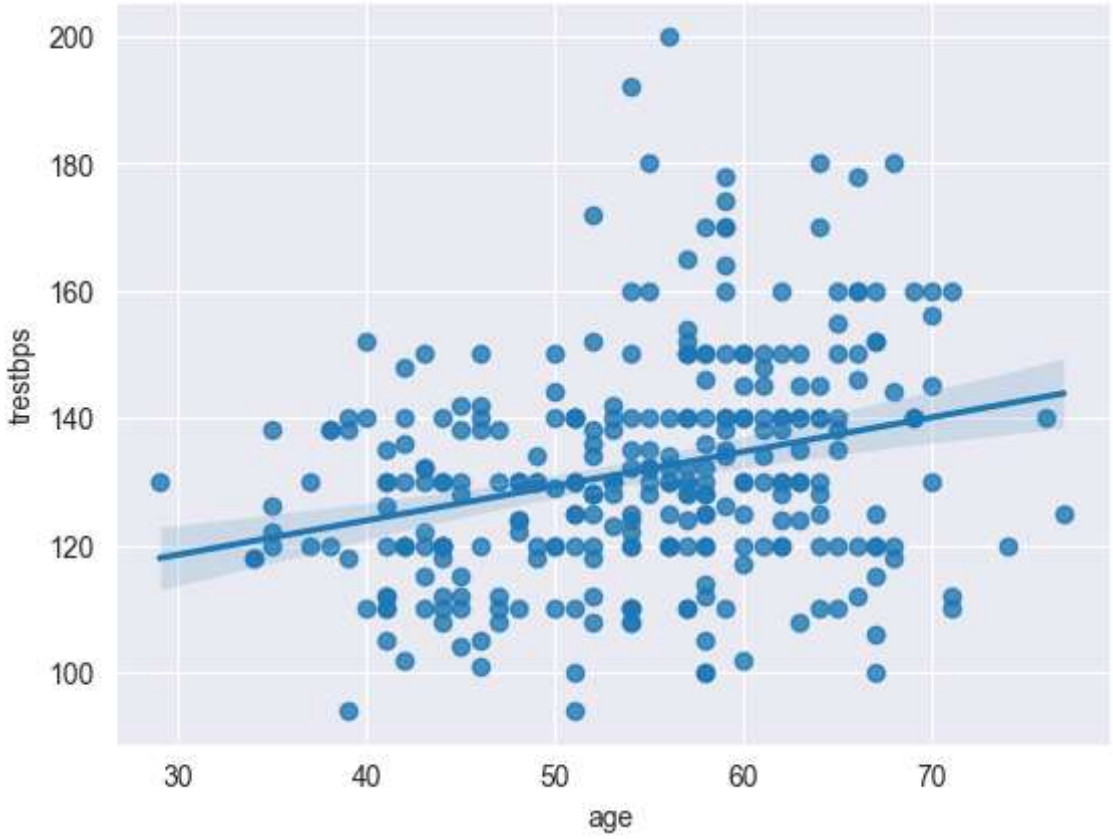


Interpretation

- There is no correlation between age and trestbps

In [95]:

```
## Regression Plot

r=sns.regplot(data=heart,x='age',y='trestbps')
```

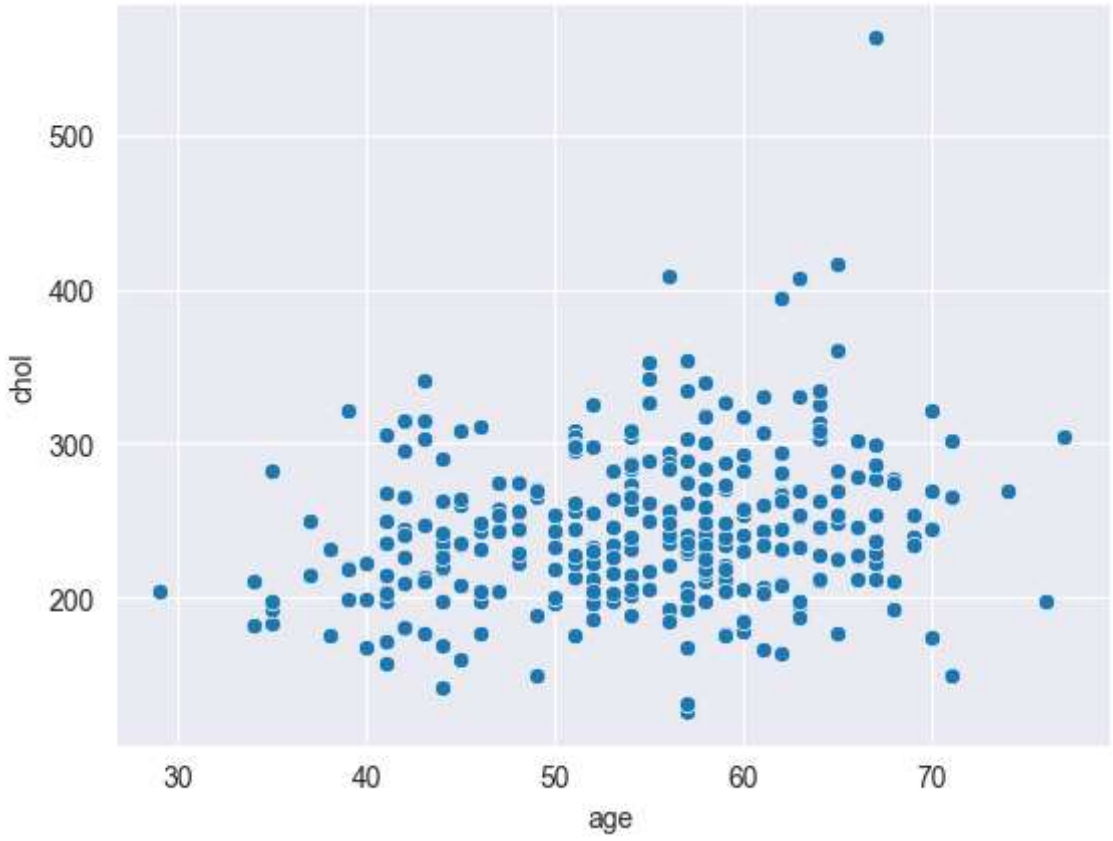


Interpretation

- The above line shows that linear regression model is not good fit to the data.

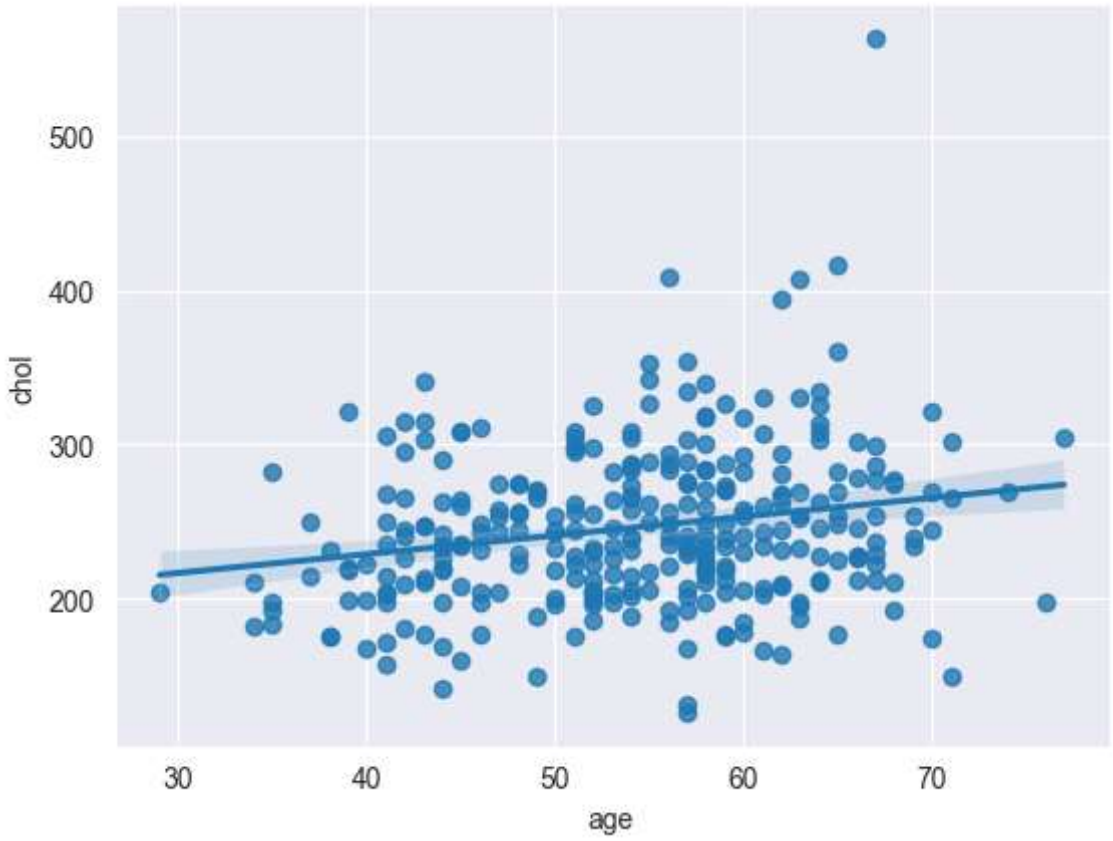
Analyze age and chol variable

In [98]:
```
1 ax=sns.scatterplot(data=heart,x='age',y='chol')
```



In [99]:
```
1 sns.regplot(data=heart,x='age',y='chol')
```

Out[99]: <Axes: xlabel='age', ylabel='chol'>

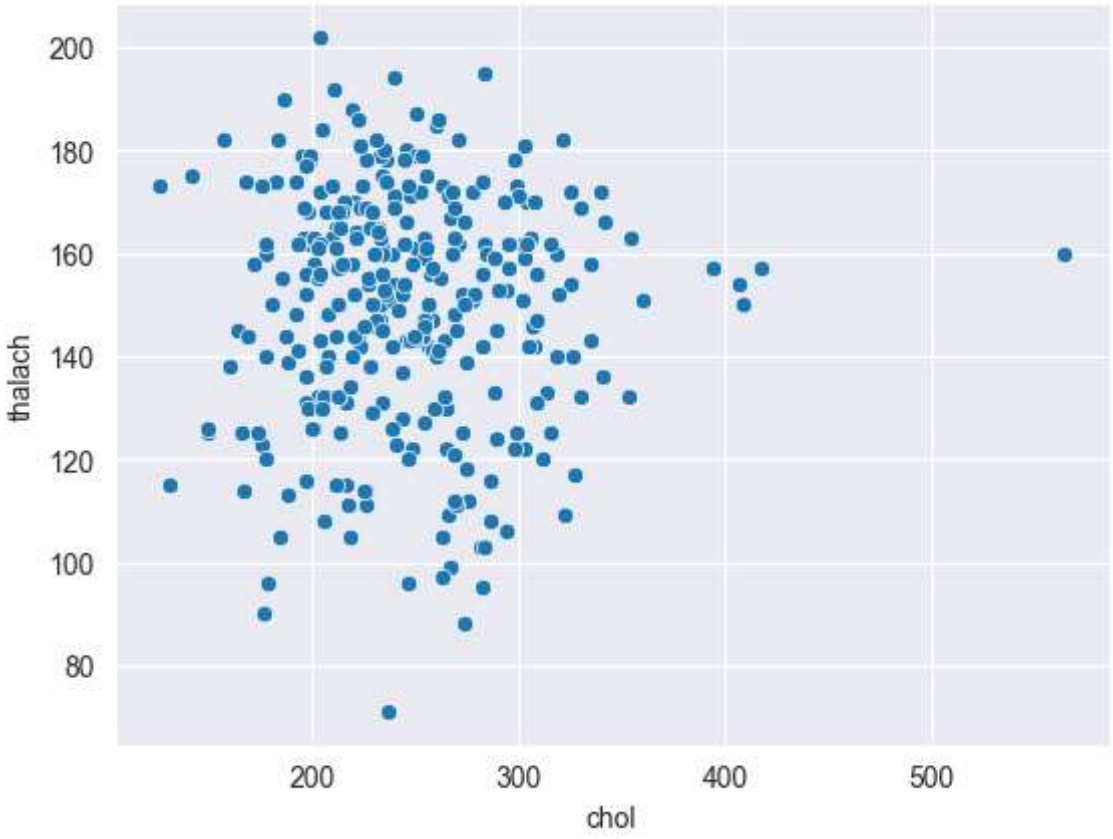


Interpretation

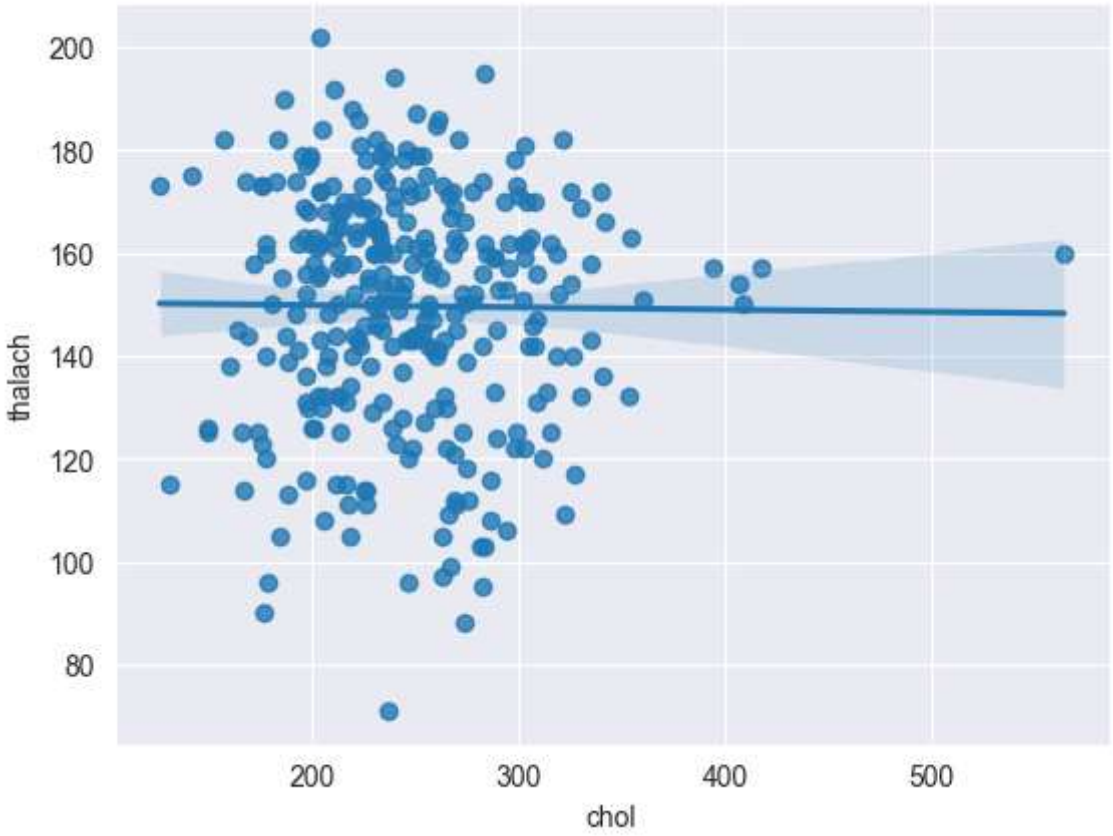
- The above plot confirms that there is a slightly correlation between age and chol variables.

Analyze chol and thalach variable

In [100]:
```
1 ax=sns.scatterplot(data=heart,x='chol',y='thalach')
```



In [101]:
```
1 ax=sns.regplot(data=heart,x='chol',y='thalach')
```



Interpretation

- The above plot shows that there is no correlation between chol and thalach variable

Check with ASSERT Statement

In [104]:
```
1 ## assert that there are no missing values in the dataset
2 assert pd.notnull(heart).all().all()
```

In [105]:
```
1 ## assert all values are greater than or equal to 0
2 assert (heart>=0).all().all()
```

Interpretation

- The above command do not show any error. Hence it is confirmed that there are no missing or negative values in the dataset.
- All the values are greater than or equal to zero.

Outlier Detection

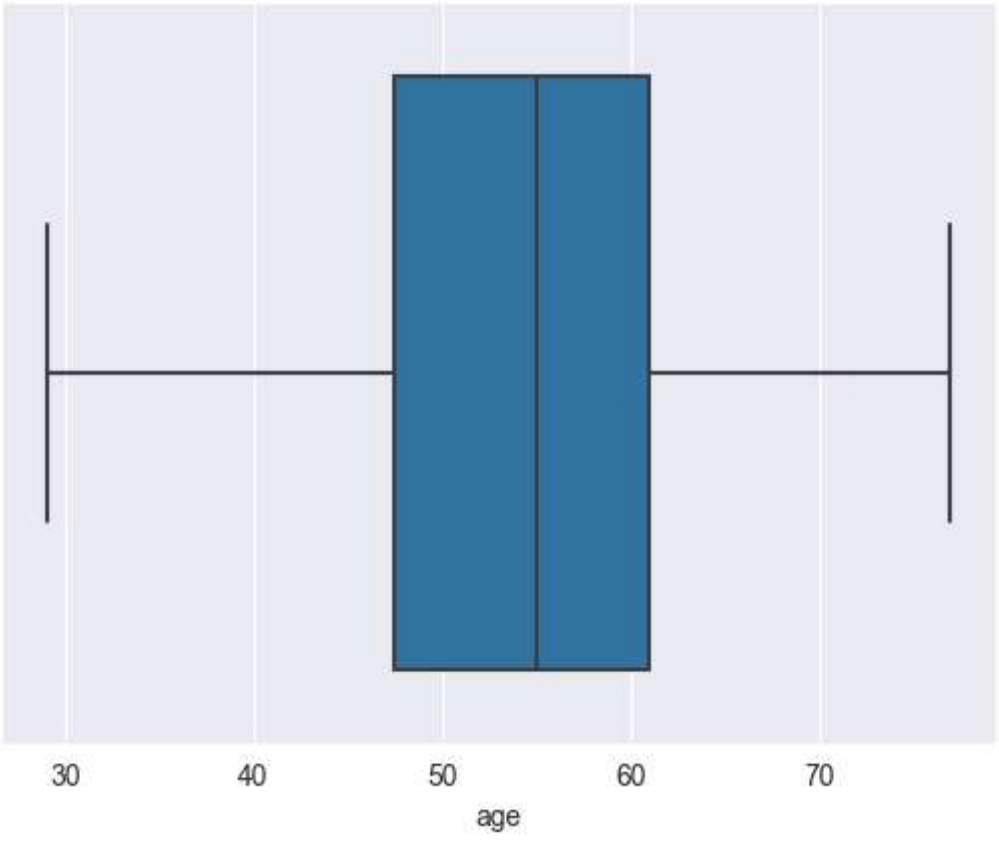
Age

In [106]:
```
1 heart.age.describe()
```

Out[106]:
```
count    303.000000
mean      54.366337
std        9.082101
min       29.000000
25%       47.500000
50%       55.000000
75%       61.000000
max       77.000000
Name: age, dtype: float64
```

Boxplot of AGE variable

In [110]:
```
1 a=sns.boxplot(x=heart.age)
```



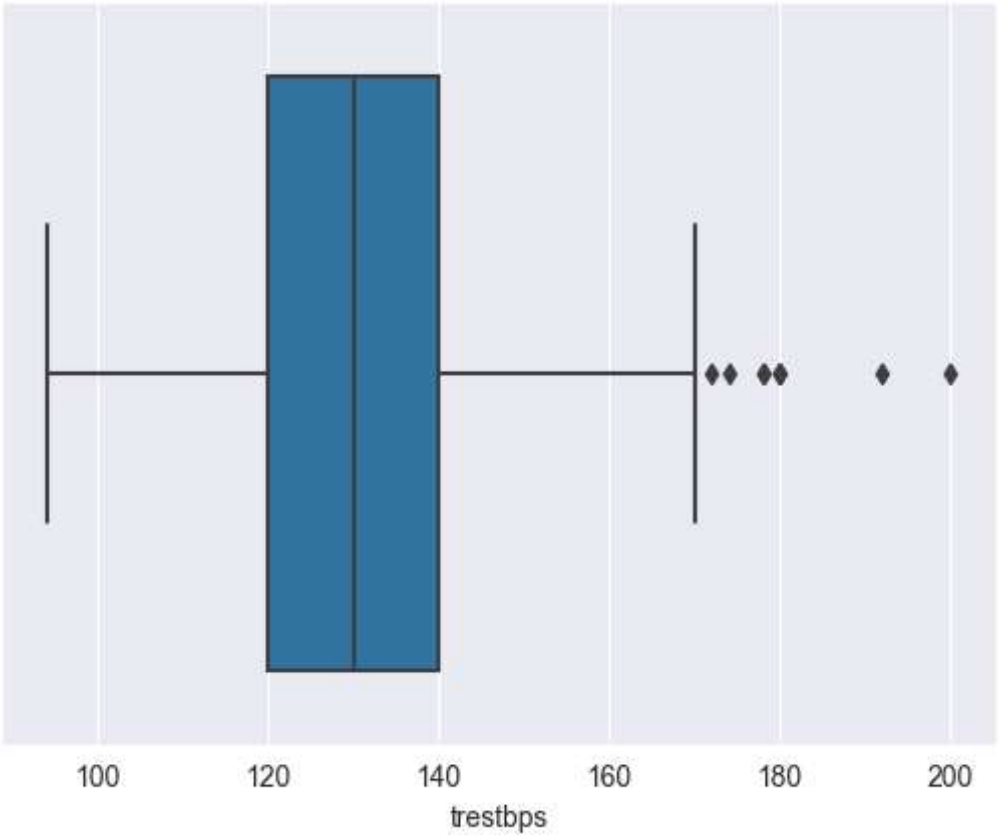
trestbps variable

In [111]:
```
1 heart.trestbps.describe()
```

Out[111]:
```
count    303.000000
mean     131.623762
std       17.538143
min       94.000000
25%      120.000000
50%      130.000000
75%      140.000000
max      200.000000
Name: trestbps, dtype: float64
```

In [112]:
```
1 sns.boxplot(x=heart.trestbps)
```

Out[112]:
```
<Axes: xlabel='trestbps'>
```



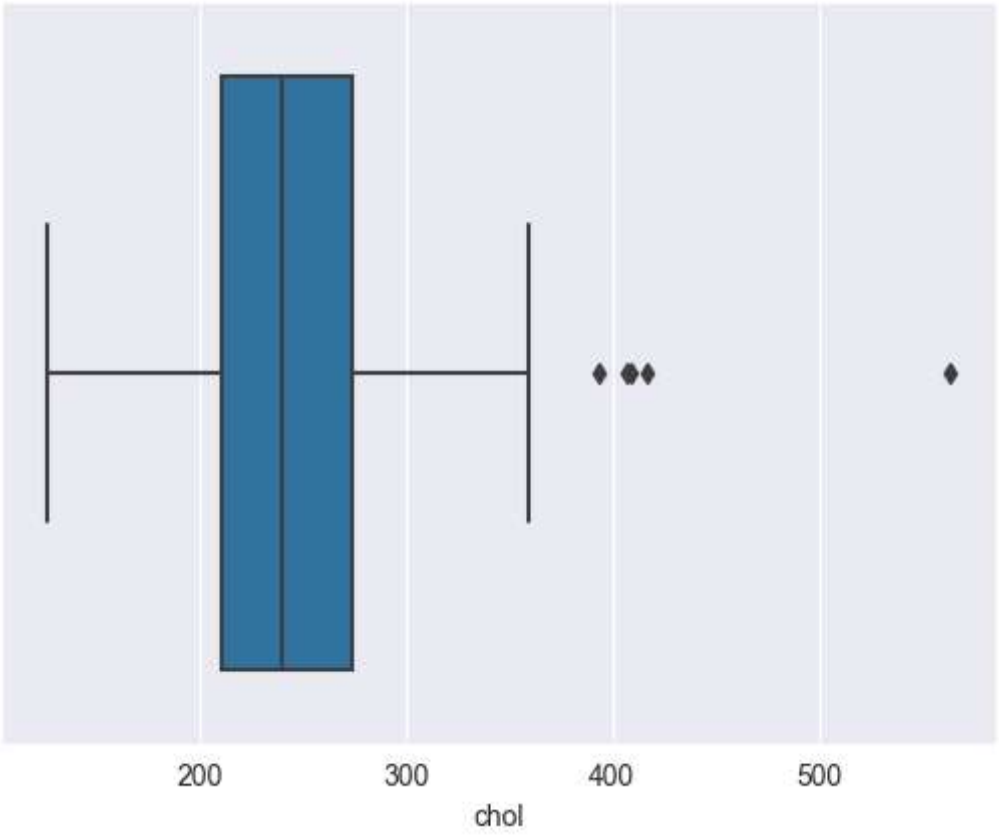
Chol variable

In [113]:
```
1 heart.chol.describe()
```

Out[113]:
```
count    303.000000
mean     246.264026
std       51.830751
min      126.000000
25%      211.000000
50%      240.000000
75%      274.500000
max      564.000000
Name: chol, dtype: float64
```

In [114]:
```
1 sns.boxplot(x=heart.chol)
```

Out[114]:
```
<Axes: xlabel='chol'>
```



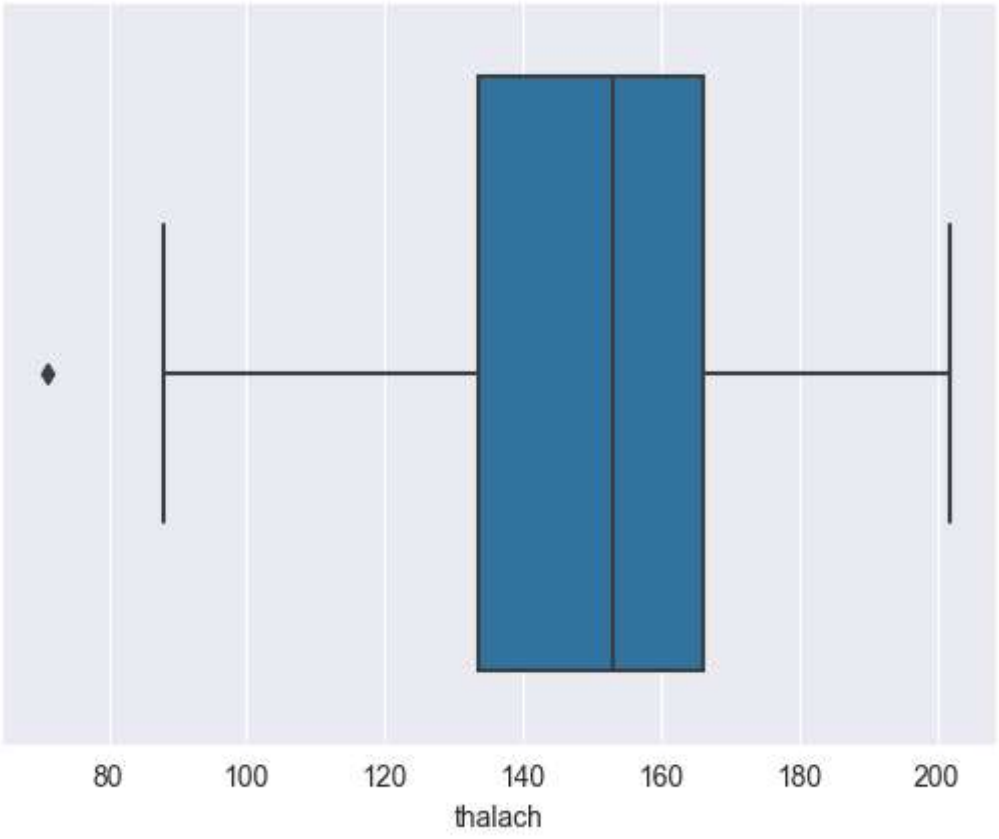
thalach variable

In [115]:
```
1 heart.thalach.describe()
```

Out[115]:
```
count    303.000000
mean     149.646865
std       22.905161
min       71.000000
25%      133.500000
50%      153.000000
75%      166.000000
max      202.000000
Name: thalach, dtype: float64
```

In [116]:
```
1 sns.boxplot(x=heart.thalach)
```

Out[116]:
```
<Axes: xlabel='thalach'>
```



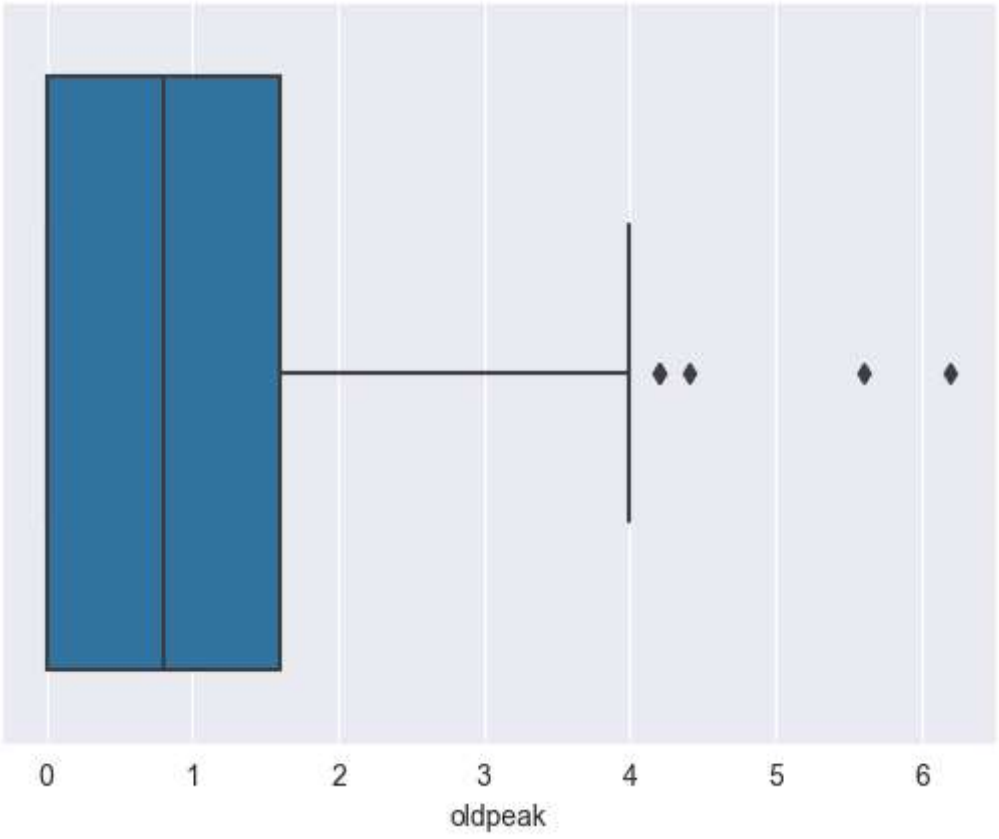
oldpeak

In [117]:
```
1 heart.oldpeak.describe()
```

Out[117]:
```
count    303.000000
mean       1.039604
std        1.161075
min        0.000000
25%        0.000000
50%        0.800000
75%        1.600000
max        6.200000
Name: oldpeak, dtype: float64
```

In [118]:
```
1 sns.boxplot(x=heart.oldpeak)
```

Out[118]:
```
<Axes: xlabel='oldpeak'>
```



Interpretation

- The age variable does not contain any outlier.
- trestbps variable contains outliers to the right side.
- chol variable also contains outliers to the right side.
- thalach variable contains a single outlier to the left side.
- oldpeak variable contains outliers to the right side.
- Those variables containing outliers needs further investigation.

In [ ]:

```
1
```