In [1]:
```
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
import pandas as pd
sns.set(style='whitegrid')
```

In [2]:
```
import warnings
warnings.filterwarnings('ignore')
```

In [3]:
```
df=pd.read_csv(r"C:\Users\Admin\Desktop\class\resume project\EDA- HEALTHCARE DOM
```

In [4]:
```
df
```

Out[4]:

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	th
0	63	1	3	145	233	1	0	150	0	2.3	0	0	
1	37	1	2	130	250	0	1	187	0	3.5	0	0	
2	41	0	1	130	204	0	0	172	0	1.4	2	0	
3	56	1	1	120	236	0	1	178	0	0.8	2	0	
4	57	0	0	120	354	0	1	163	1	0.6	2	0	
...	...	...	...	...	...	...	...	...	...	...	...	...	
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	
302	57	0	1	130	236	0	0	174	0	0.0	1	1	

303 rows × 14 columns

DATA KEYWORDS

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 [5]:
```
df.shape
```

Out[5]:
```
(303, 14)
```

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

Out[6]:

	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 [7]:
```
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 [8]:
```
df.describe()
```

Out[8]:

	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 [9]:
```
df.dtypes #data types of each column in the data set
```

Out[9]:
```
age           int64
sex           int64
cp            int64
trestbps      int64
chol          int64
fbs           int64
restecg       int64
thalach       int64
exang         int64
oldpeak     float64
slope         int64
ca            int64
thal          int64
target        int64
dtype: object
```

In [10]:
```
df.columns
```

Out[10]:
```
Index(['age', 'sex', 'cp', 'trestbps', 'chol', 'fbs', 'restecg', 'thalach',
       'exang', 'oldpeak', 'slope', 'ca', 'thal', 'target'],
      dtype='object')
```

univariante,bi variante and multi variante analysis of the give data

In [11]:
```
df['target'].unique() #in target 0 is no heart disease and 1 is heart disease
```

Out[11]:
```
array([1, 0], dtype=int64)
```

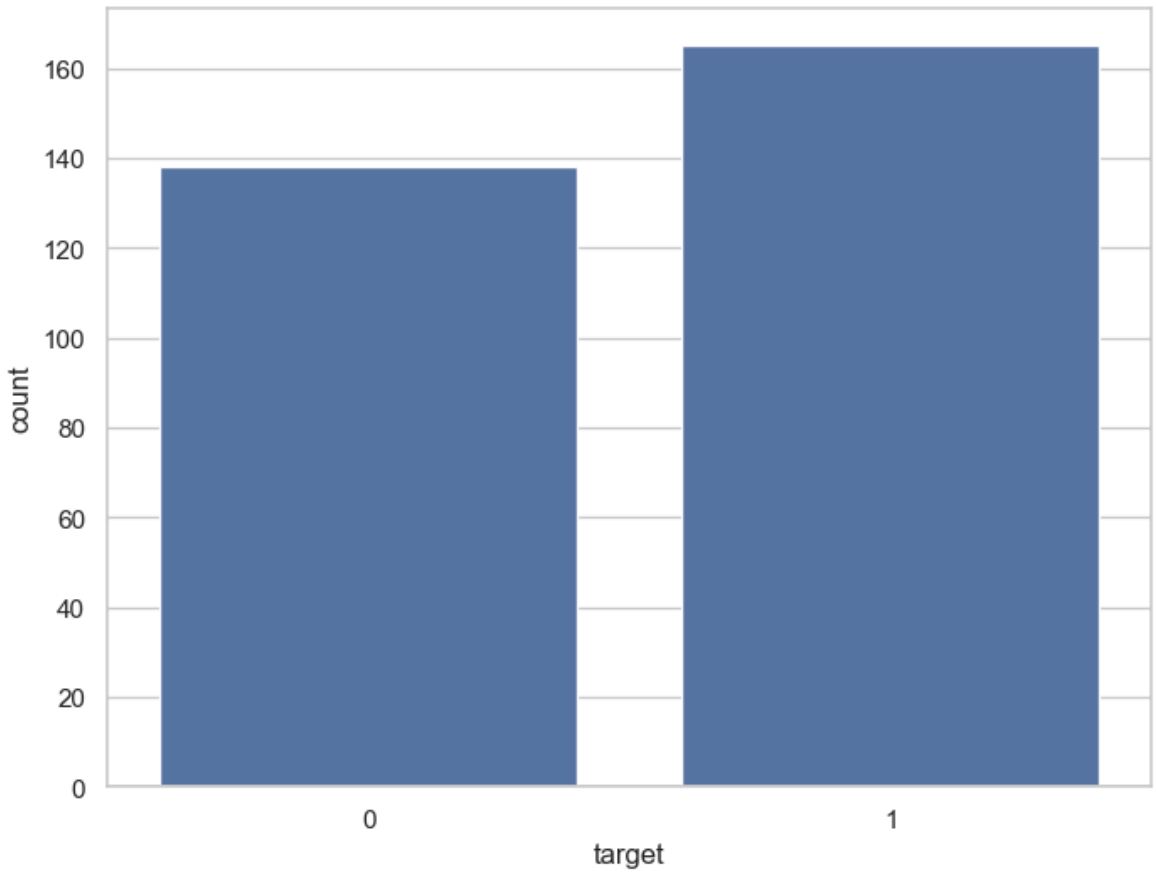
In [12]:
```
df['target'].nunique()
```

Out[12]:
```
2
```

In [13]:
```
df['target'].value_counts()
```

Out[13]:
```
target
1    165
0    138
Name: count, dtype: int64
```

In [14]:
```
f,ax=plt.subplots(figsize=(8,6)) #this visulization tells there are more heart p
ax=sns.countplot(data=df,x='target')
plt.show()
```

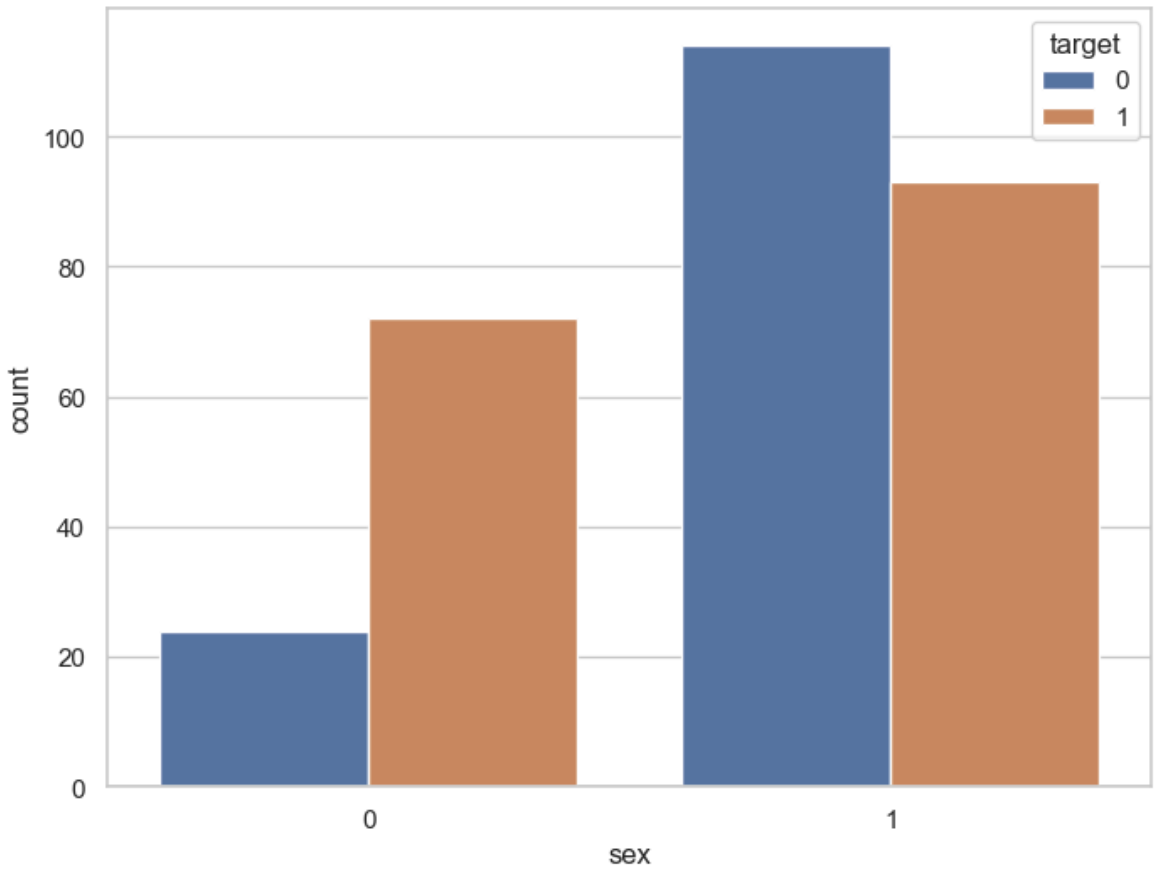


In [15]:
```
df.groupby('sex')['target'].value_counts() # here grouping of sex and target is
```

Out[15]:
```
sex  target
0    1          72
     0          24
1    0         114
     1          93
Name: count, dtype: int64
```

in this graph the 0 and 1 on xticks refers to gender and plot will refer to presence of heart disease

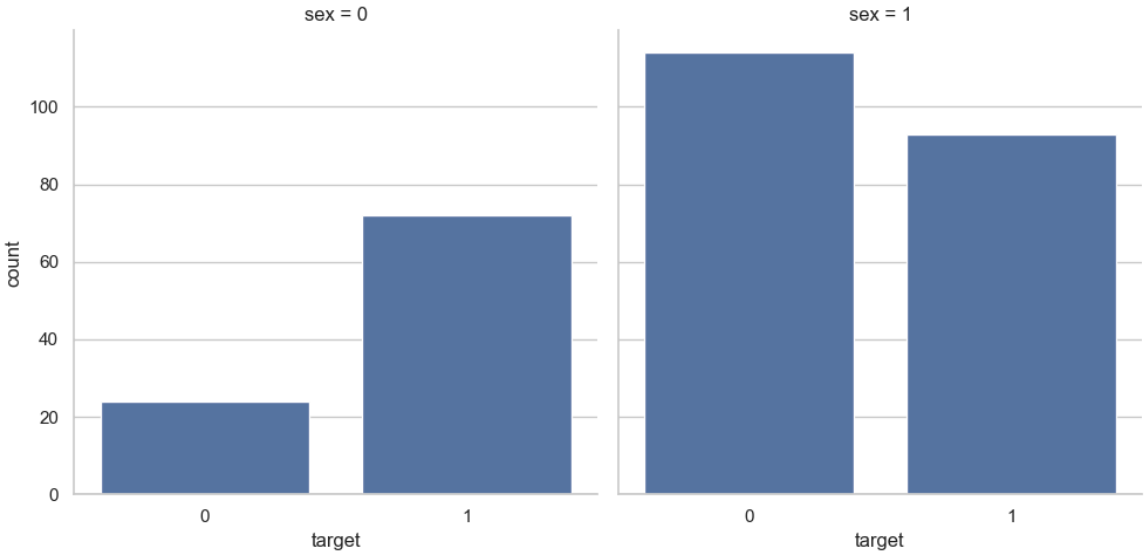
In [16]:
```
ax=plt.subplots(figsize=(8,6)) #in sex 0 refers to  female and 1 refers to male
ax=sns.countplot(x='sex',hue='target',data=df)
plt.show()
```



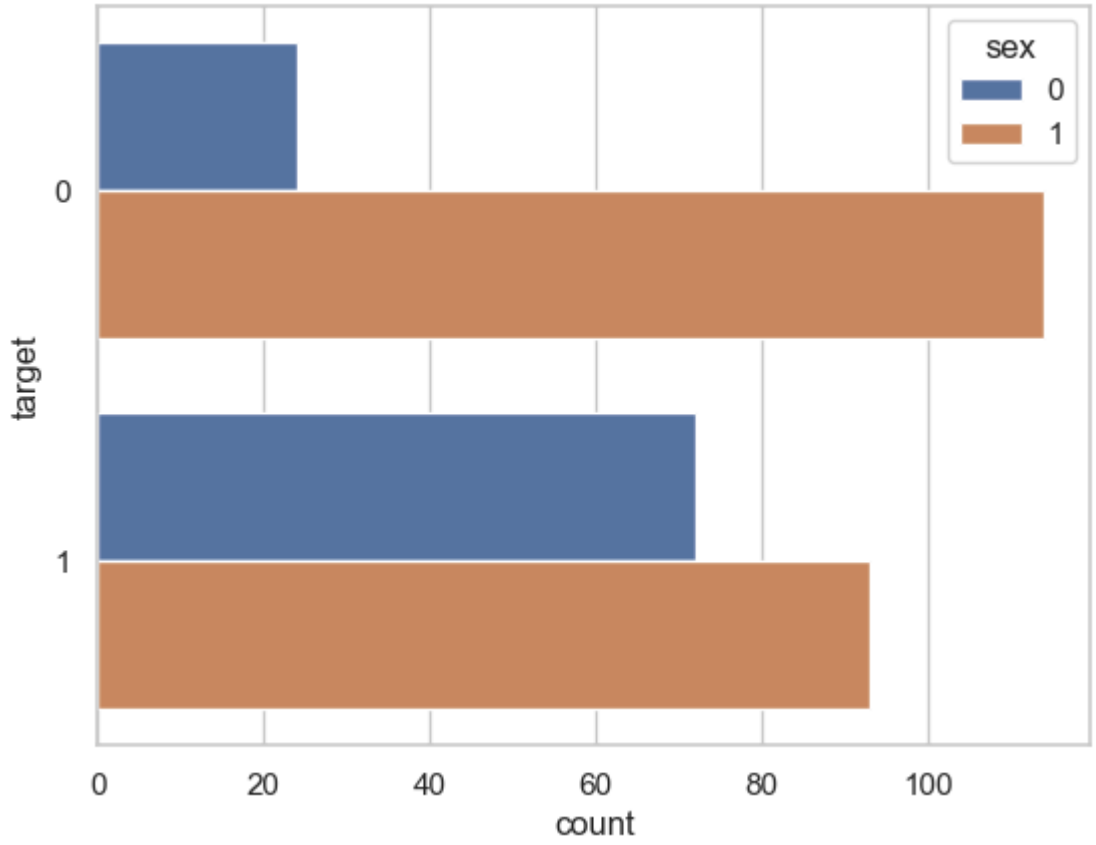
Out of 96 females - 72 have heart disease and 24 do not have heart disease.

out of 207 males - 93 have heart disease and 114 do not have heart disease.

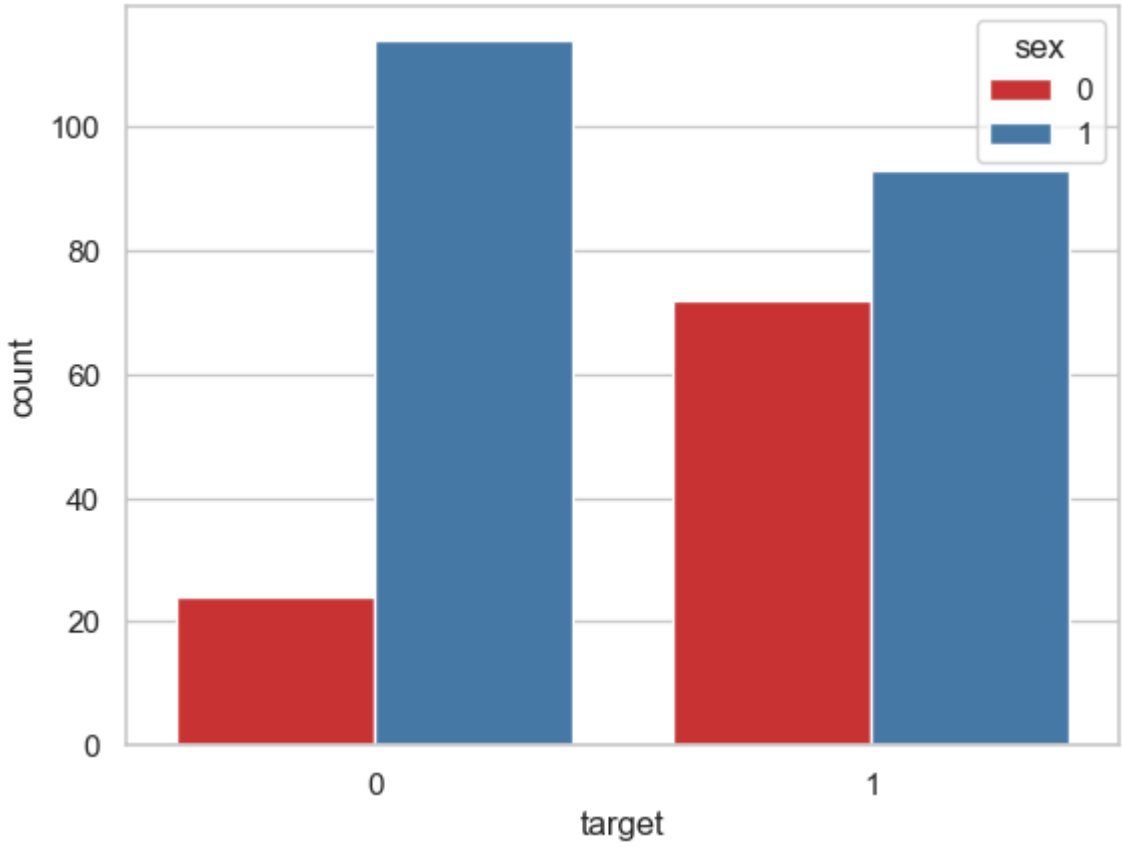
In [17]:
```
ax=sns.catplot(x='target',col='sex',data=df,kind='count') #in this graph it is v
```



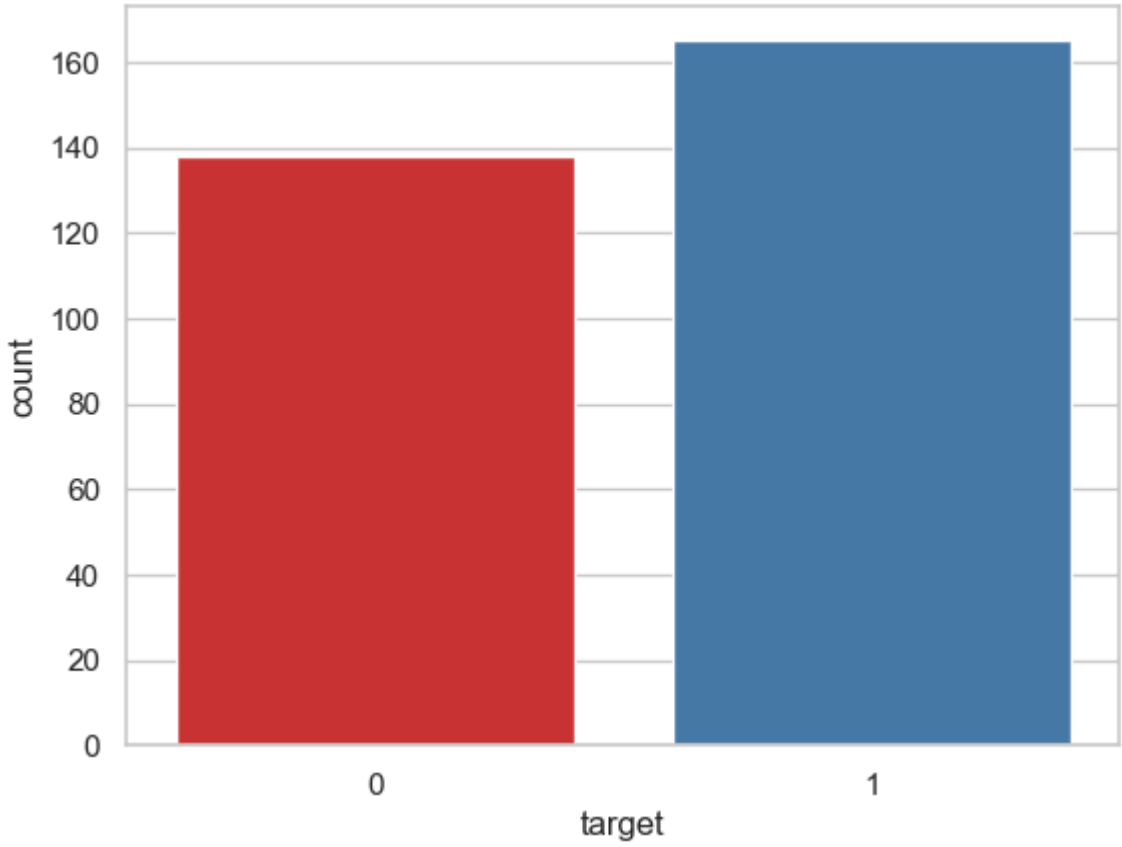
In [18]:
```
ax=sns.countplot(y='target',hue='sex',data=df) # in this graph ,target is visuli
```



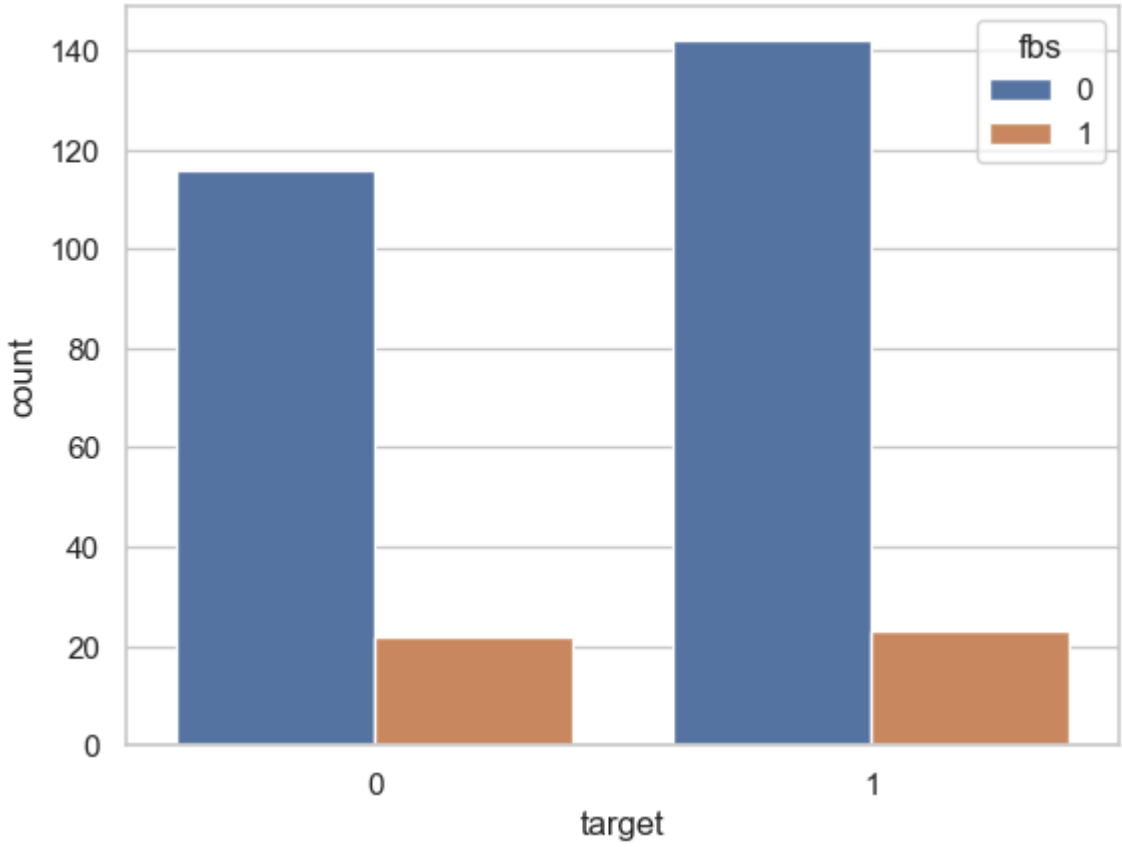
In [19]:
```
ax = sns.countplot(x='target',hue='sex', data=df, palette='Set1') #here palette
                                                                  #there are 3 a
```



In [20]:
```
ax = sns.countplot(x='target', data=df, palette='Set1')
```



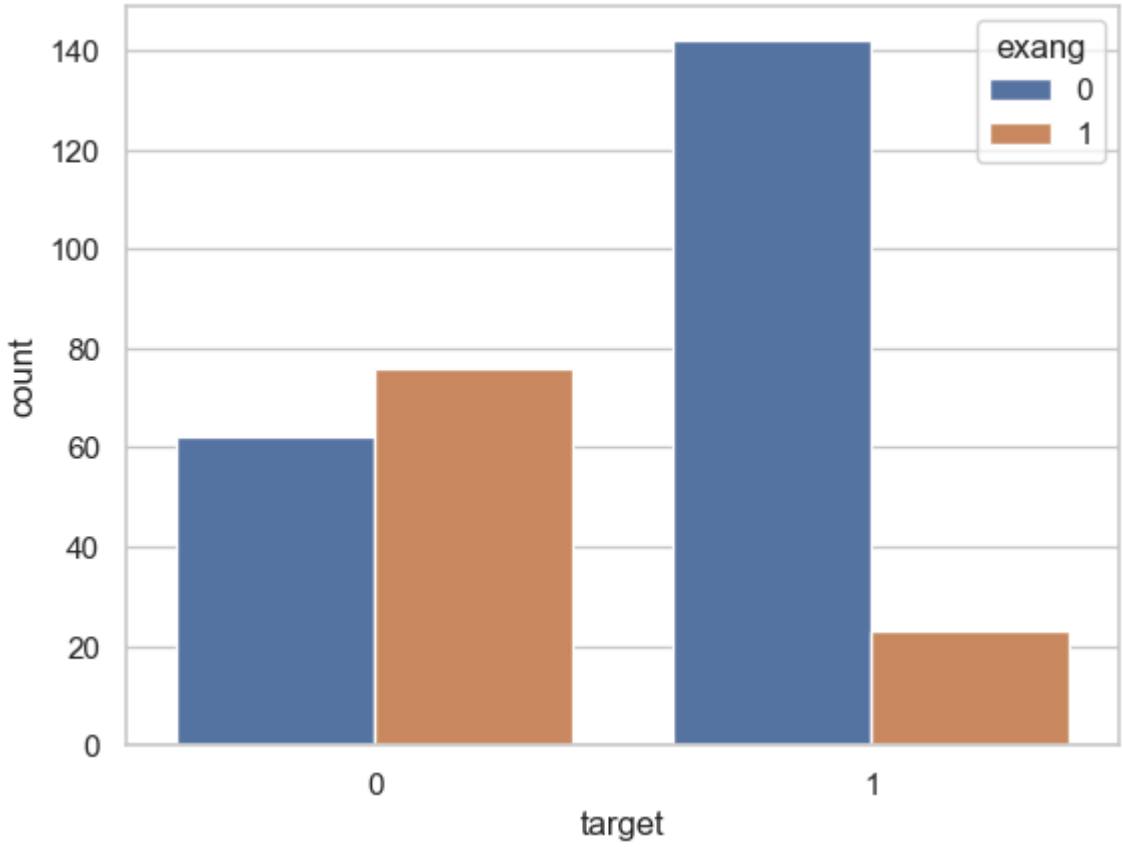
In [21]:
```
ax=sns.countplot(x='target',hue='fbs',data=df)
```



fbs refers to food blood sugar in this visualization is the comparion of fbs and presence of heart disease

fbs 0 is false and fbs 1 is true (fasting blood sugar >120 mg/dl)

In [22]:
```
ax=sns.countplot(x='target',hue='exang',data=df)
```



in this visualization comparion of target to exercise induced angina (exang) is done

In [23]:
```
correlation=df.corr() #corr refers to correlation
```

In [24]:
```
correlation['target'].sort_values(ascending=False)
```

Out[24]:
```
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
```

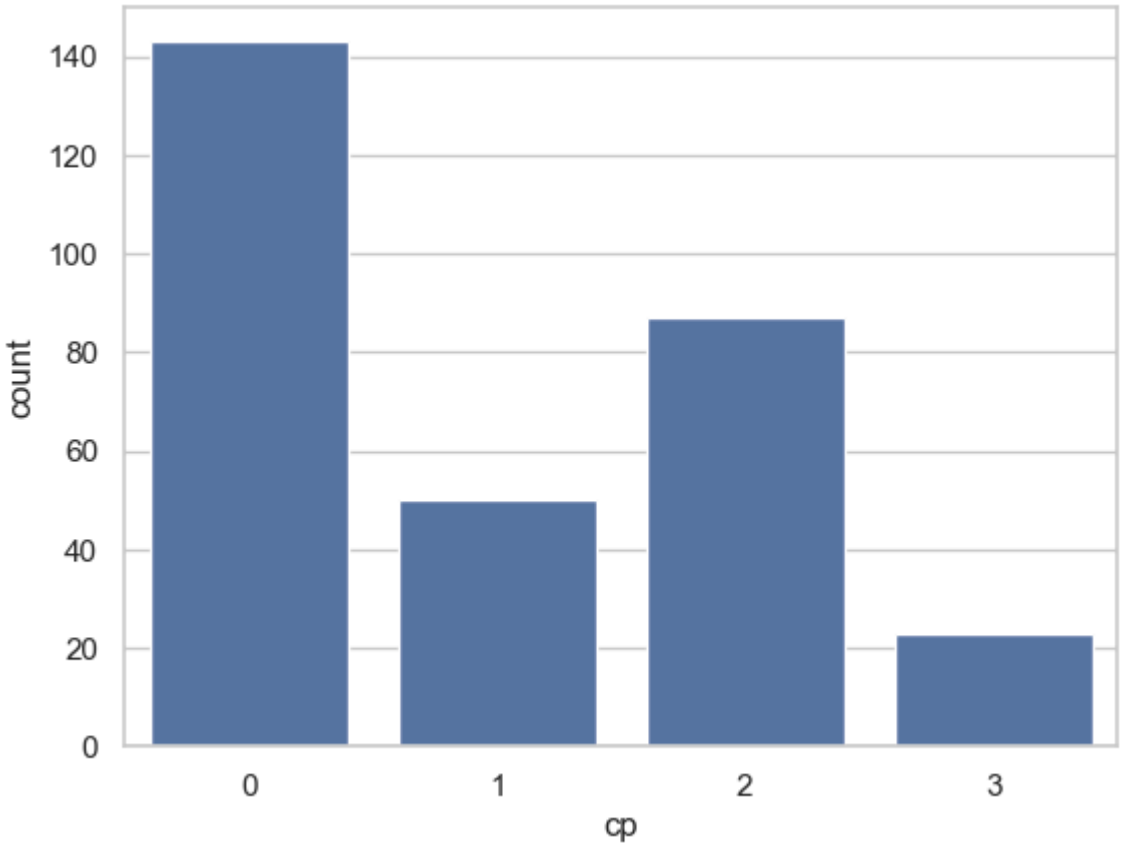
In [25]:
```
df['cp'].nunique() # cp refers to chest pain
```

Out[25]:
```
4
```

In [26]:
```
df['cp'].value_counts() # 0,1,2,3 refers to severity of the chest pain
```

Out[26]:
```
cp
0    143
2     87
1     50
3     23
Name: count, dtype: int64
```

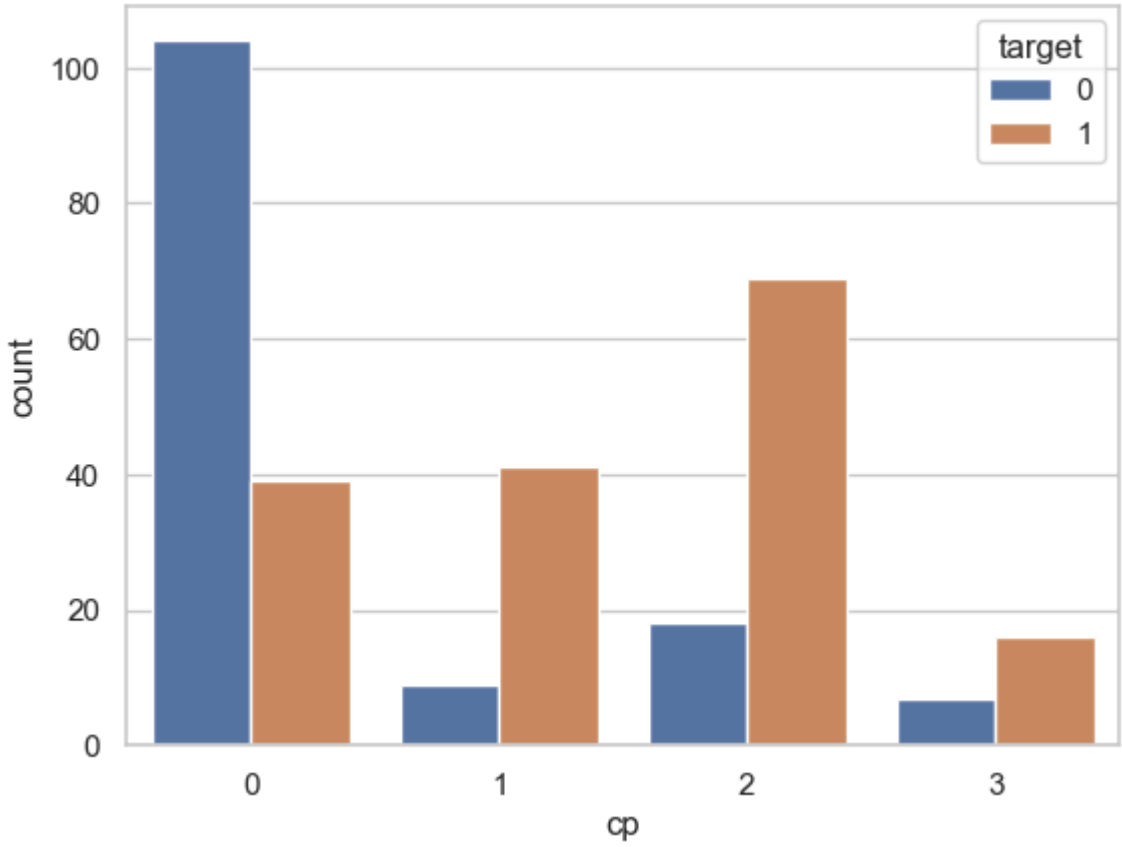
In [27]:
```
ax=sns.countplot(x='cp',data=df)
```



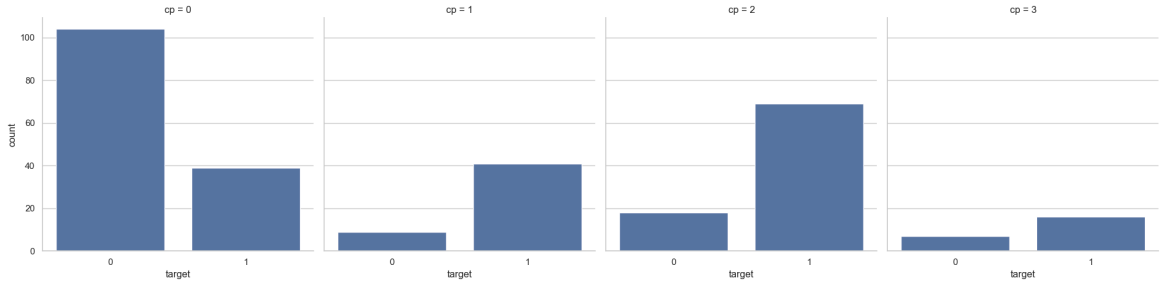
In [28]:
```
df.groupby('cp')['target'].value_counts() #grouping of target and chest pain is
```

Out[28]:
```
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 [29]:
```
ax=sns.countplot(x='cp',hue='target',data=df)
```



In [30]:
```
ax=sns.catplot(x='target',col='cp',data=df,kind='count')
```



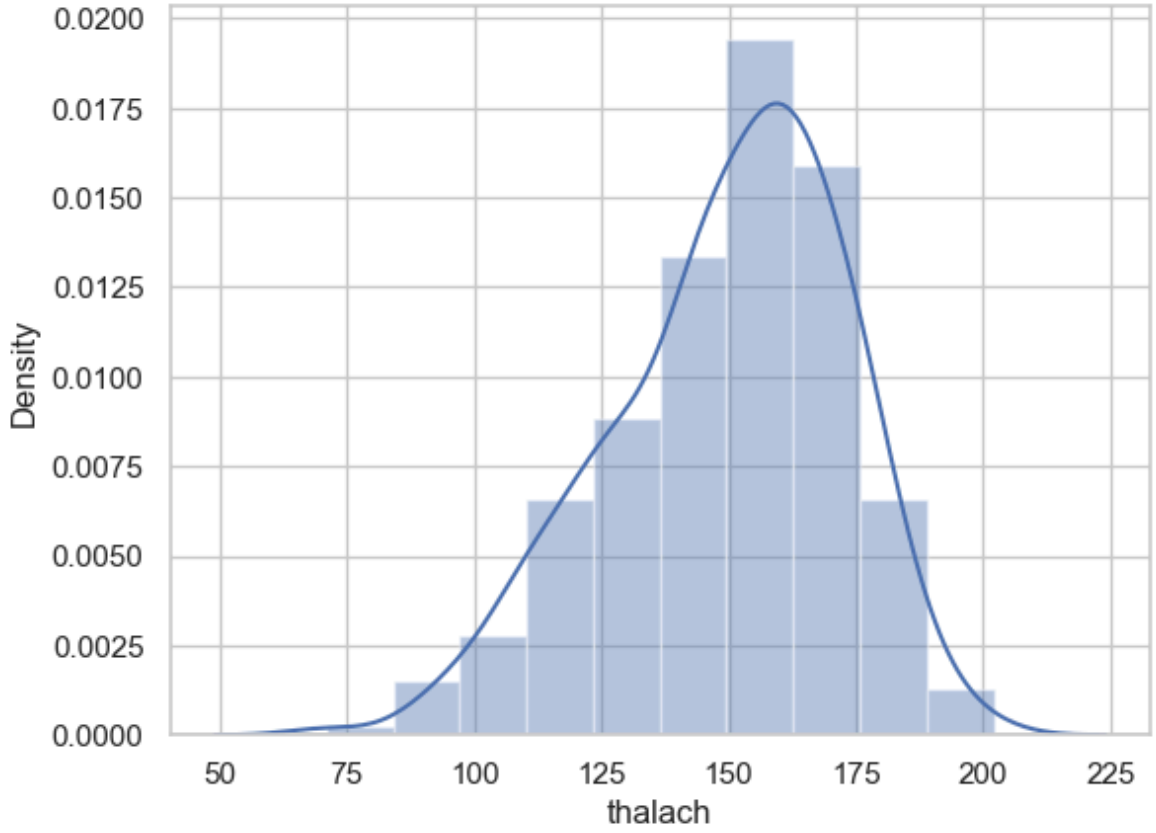
in 0 severity of chest pain the heart disease presence is low

whereas in chest pain severity of 3 the heart disease presence is comparatively high

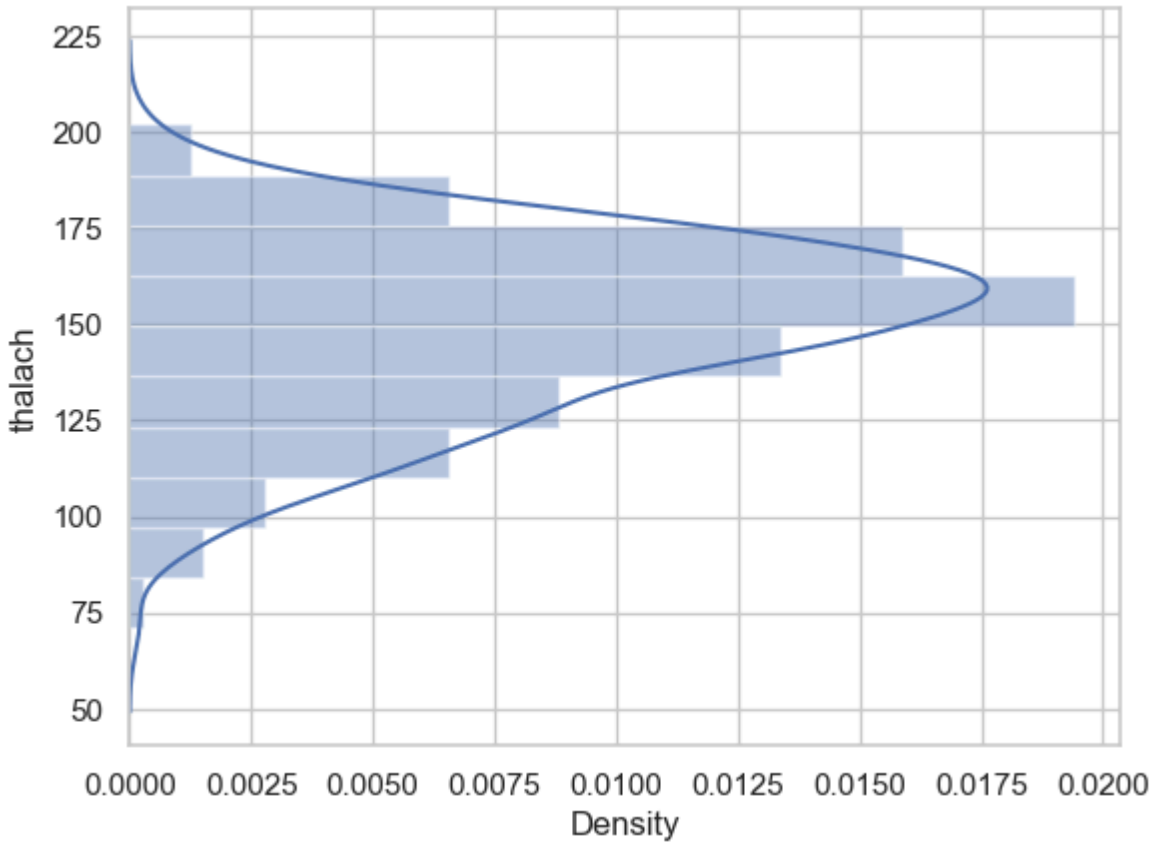
In [31]:
```
df['thalach'].nunique() # thalach refers to maximum heart rate achieved
```

Out[31]: 91

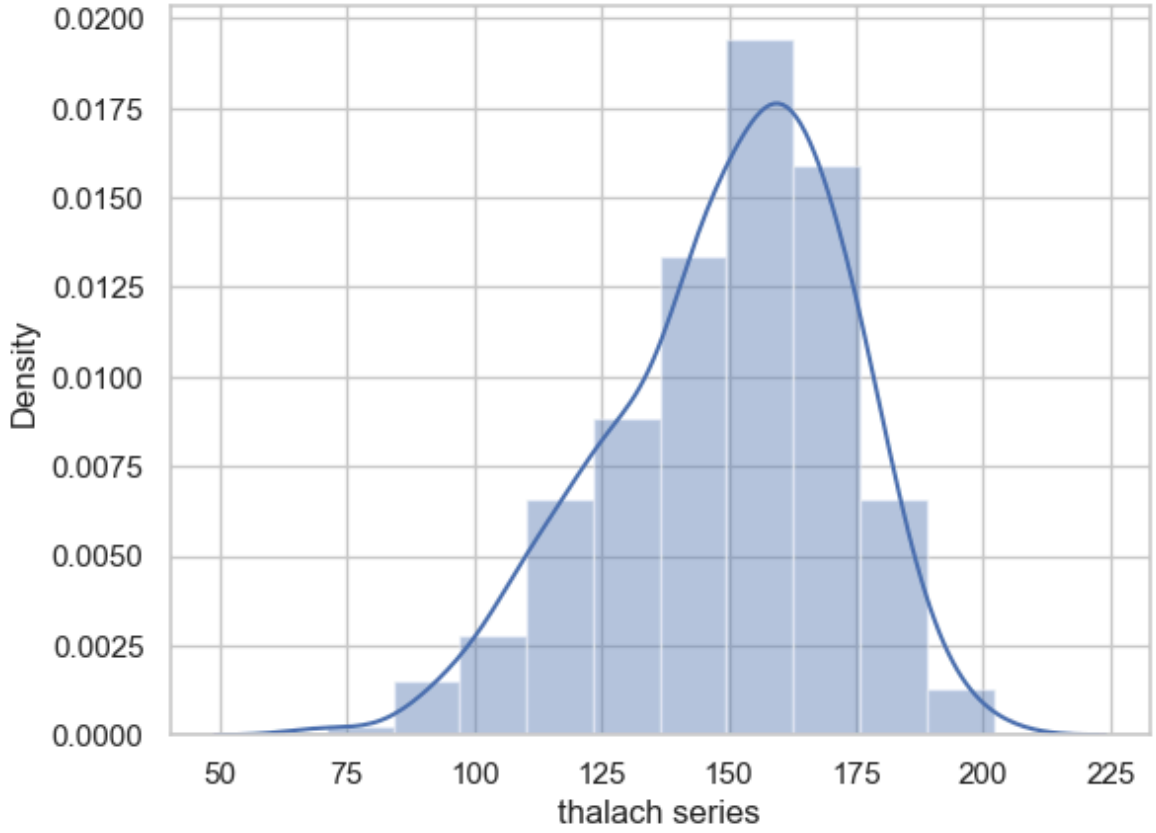
In [32]:
```
x=df['thalach'] #visualiztion of thalach is done using various plots is done
ax=sns.distplot(x,bins=10)
```



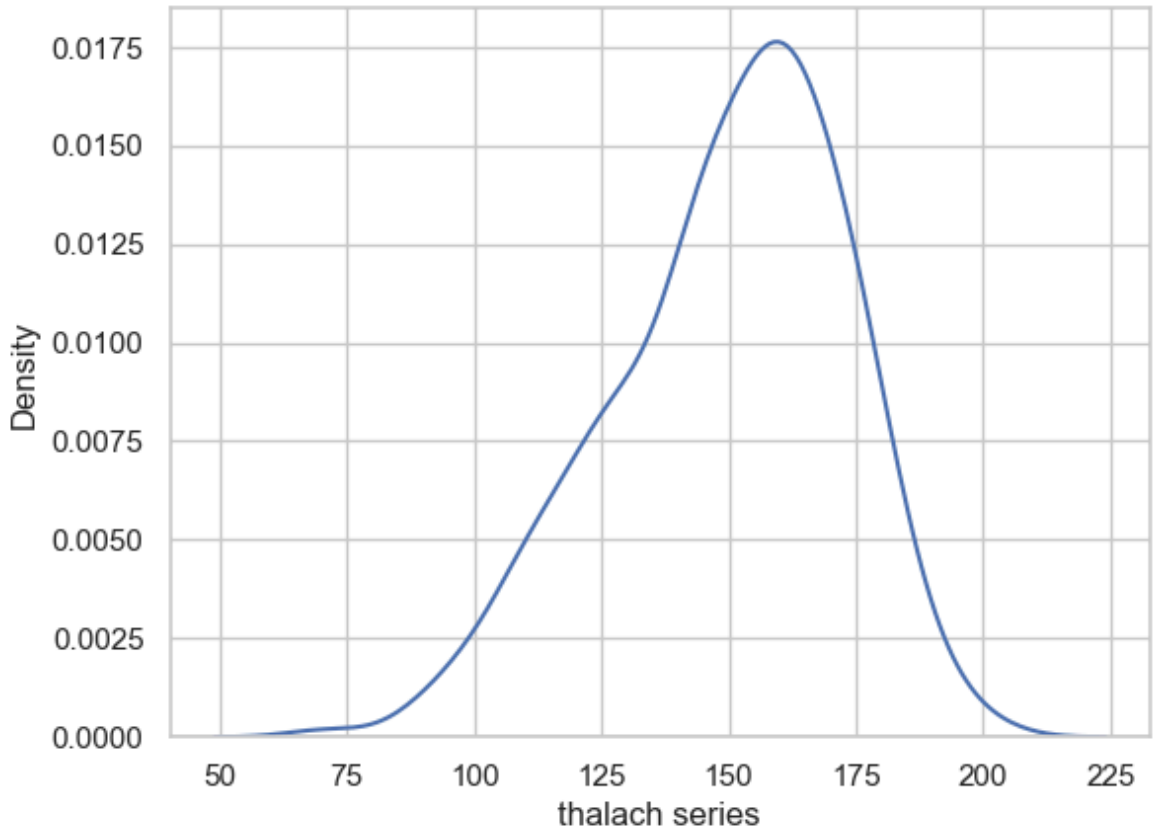
In [33]:
```
x=df['thalach']
ax=sns.distplot(x,bins=10,vertical=True)
```



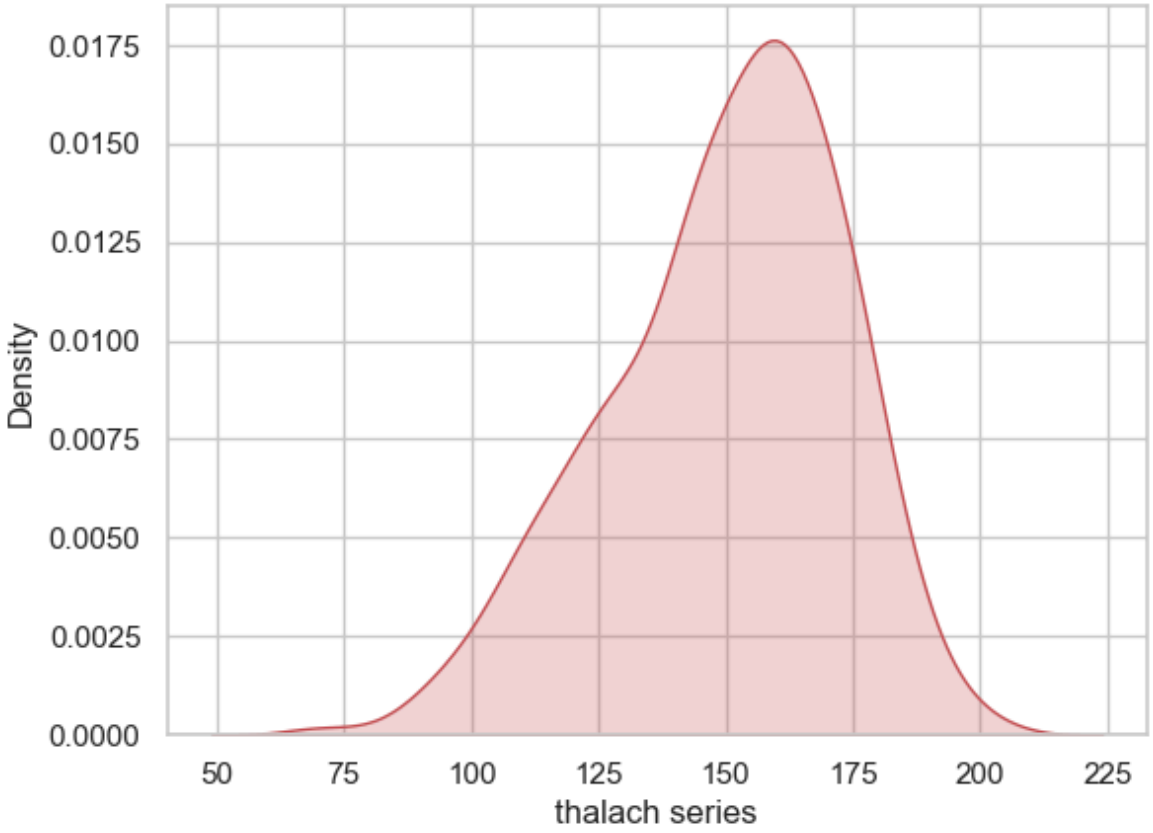
In [34]:
```
x=df['thalach']
x=pd.Series(x,name='thalach series')
ax=sns.distplot(x,bins=10)
```



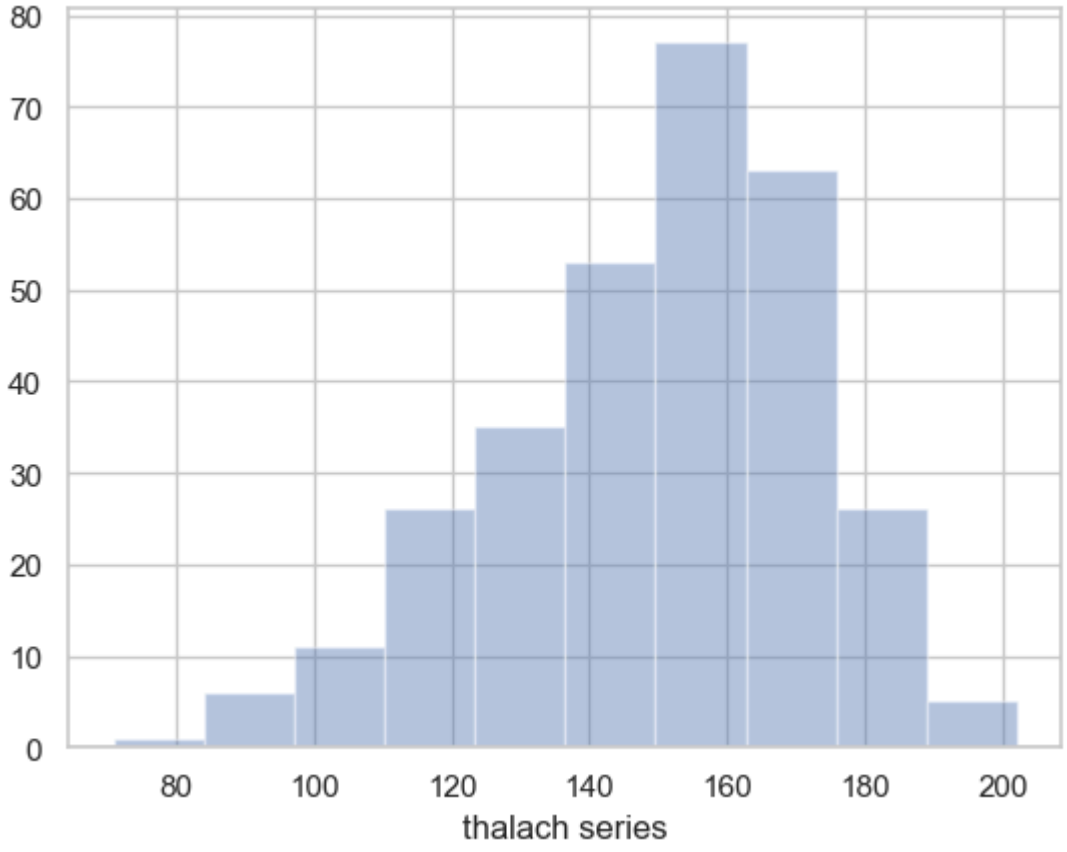
In [35]:
```
x=df['thalach']
x=pd.Series(x,name='thalach series')
ax=sns.kdeplot(x)
```



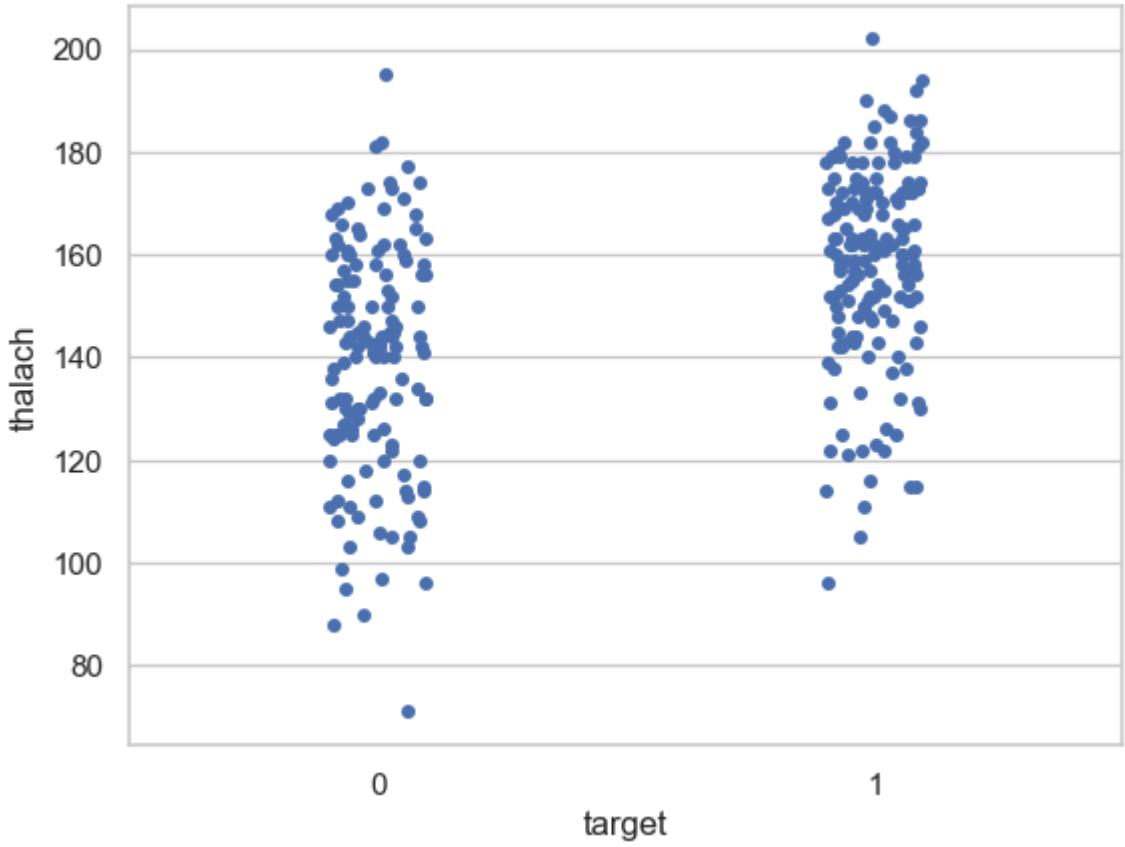
In [36]:
```
x=df['thalach']
x=pd.Series(x,name='thalach series')
ax=sns.kdeplot(x,color='r',shade=True)
```



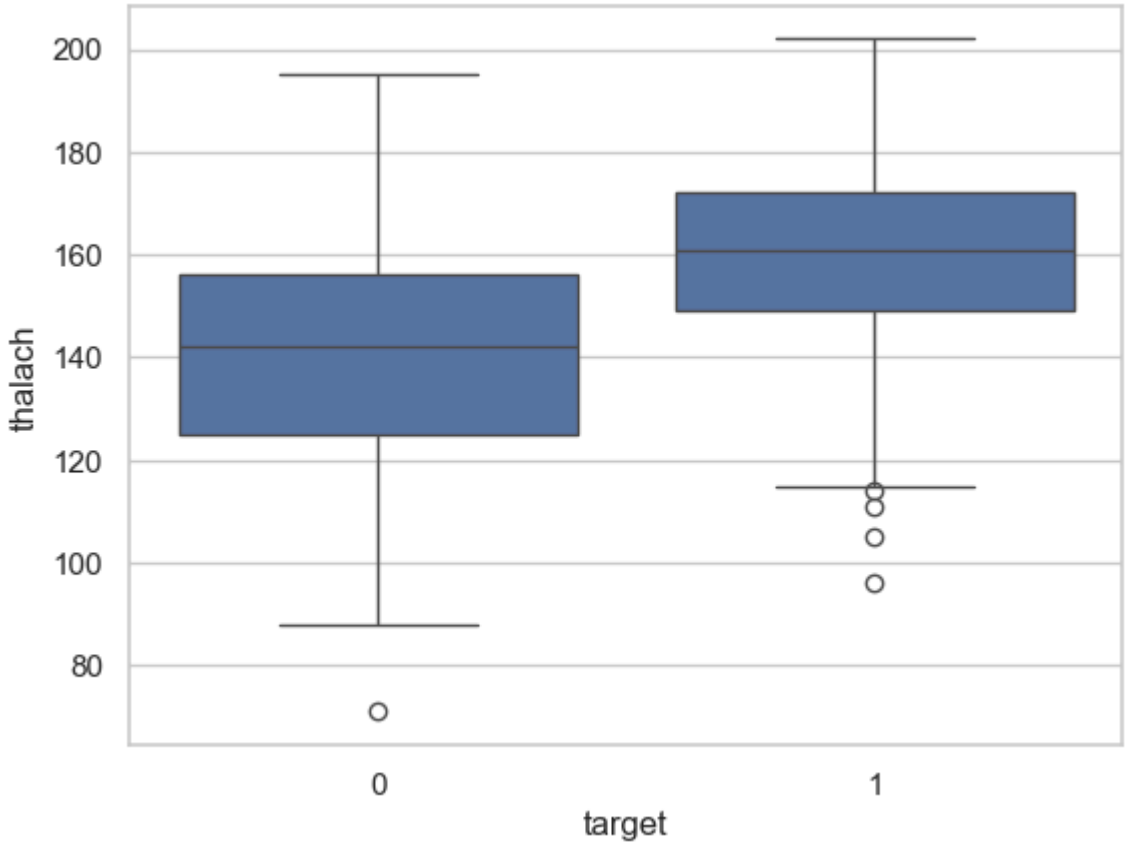
In [37]:
```
x=df['thalach']
x=pd.Series(x,name='thalach series')
ax=sns.distplot(x,kde=False,bins=10)
```



In [38]:
```
ax=sns.stripplot(x='target',y='thalach',data=df)
```



In [39]:
```
ax=sns.boxplot(x='target',y='thalach',data=df)
```



comparation of thalach (maximum heart rate achieved) and target (presence of heart disease) is done in the above comparion

MULTIVARIANTE ANALYSIS

In [40]:
```
plt.subplots(figsize=(14,10))
plt.title('correlation heatmap of heart disease')
ax=sns.heatmap(correlation,square=True,annot=True,fmt='.2f',linecolor='white')
```

the above heat map tells about the correlation between target with thal,ca,slope,oldpeak,exang,restecg,fbs,chol,trestbps, cp,sex and age

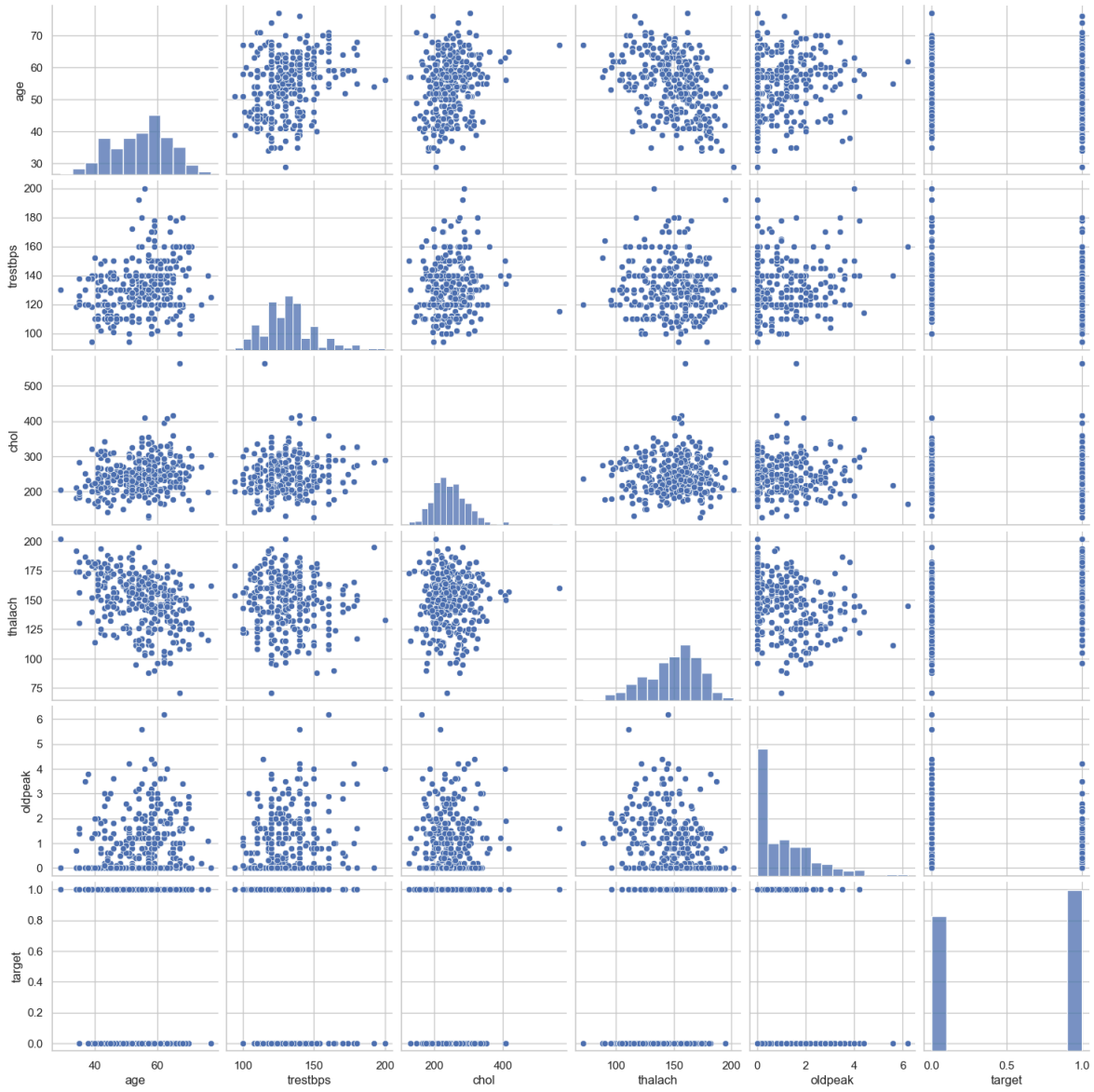
In [41]:
```
df
```

Out[41]:

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	th
0	63	1	3	145	233	1	0	150	0	2.3	0	0	
1	37	1	2	130	250	0	1	187	0	3.5	0	0	
2	41	0	1	130	204	0	0	172	0	1.4	2	0	
3	56	1	1	120	236	0	1	178	0	0.8	2	0	
4	57	0	0	120	354	0	1	163	1	0.6	2	0	
...	...	...	...	...	...	...	...	...	...	...	...	...	
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	
302	57	0	1	130	236	0	0	174	0	0.0	1	1	

303 rows × 14 columns

In [42]:
```
num_var=['age', 'trestbps','chol','thalach','oldpeak','target']
ax=sns.pairplot(df[num_var],kind='scatter',diag_kind='hist')
```



the above visualization is an pair plot of age, trestbps,chol,thalach,oldpeak,target

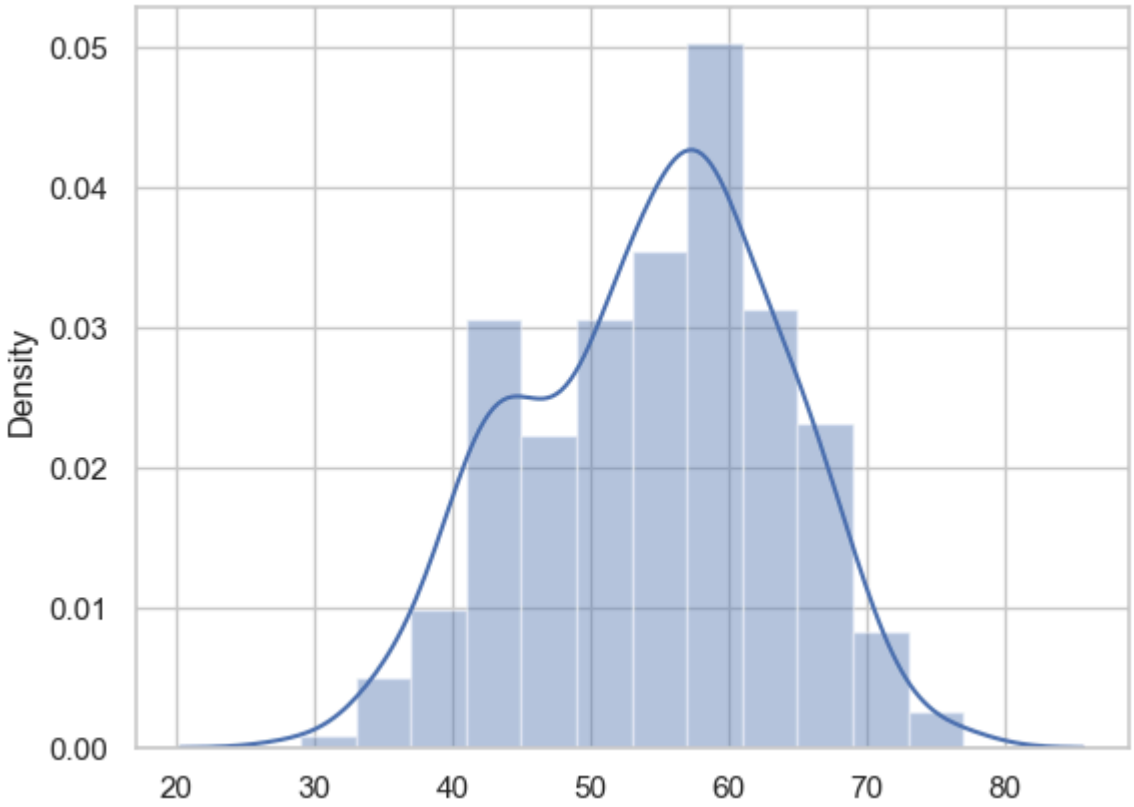
In [43]:
```
df['age'].nunique()
```

Out[43]:
```
41
```

In [44]:
```
df['age'].describe()
```

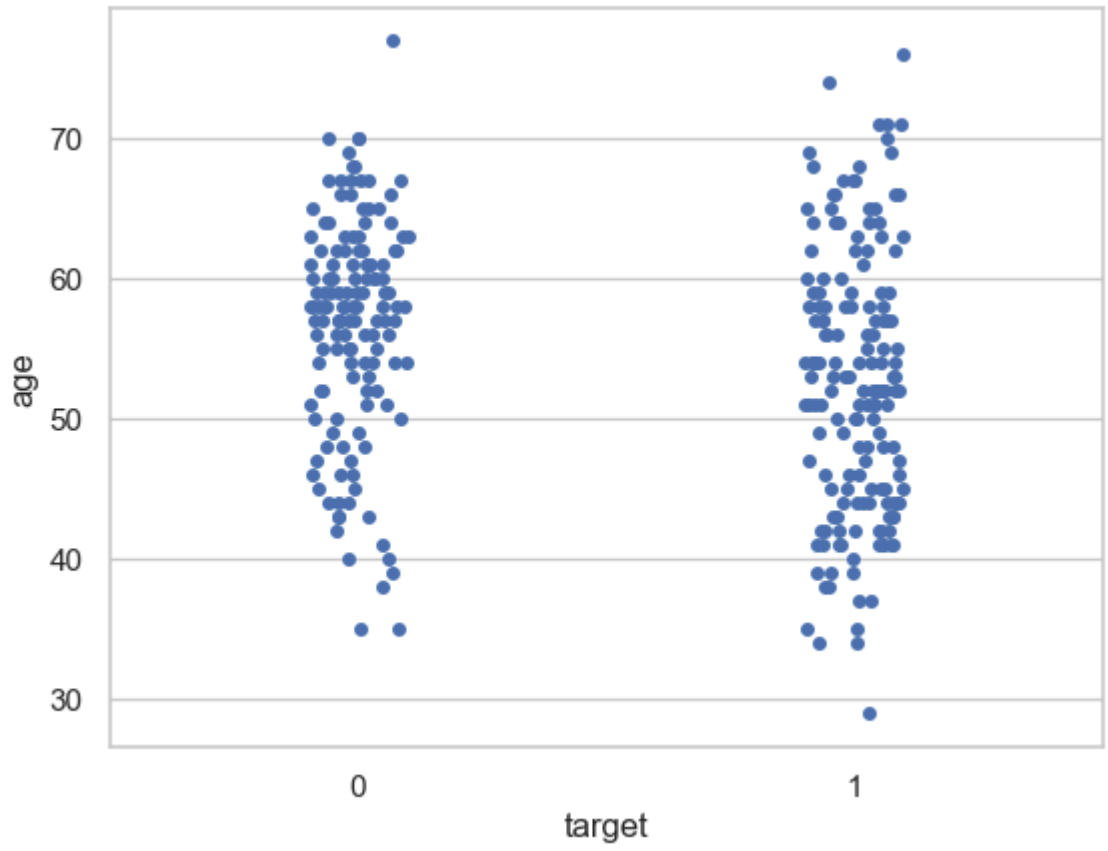
Out[44]:
```
count    303.000000
mean      54.366337
std        9.082101
min       29.000000
25%       47.500000
50%       55.000000
75%       61.000000
max       77.000000
Name: age, dtype: float64
```

In [45]:
```
ax=sns.distplot(x=df['age'])
```

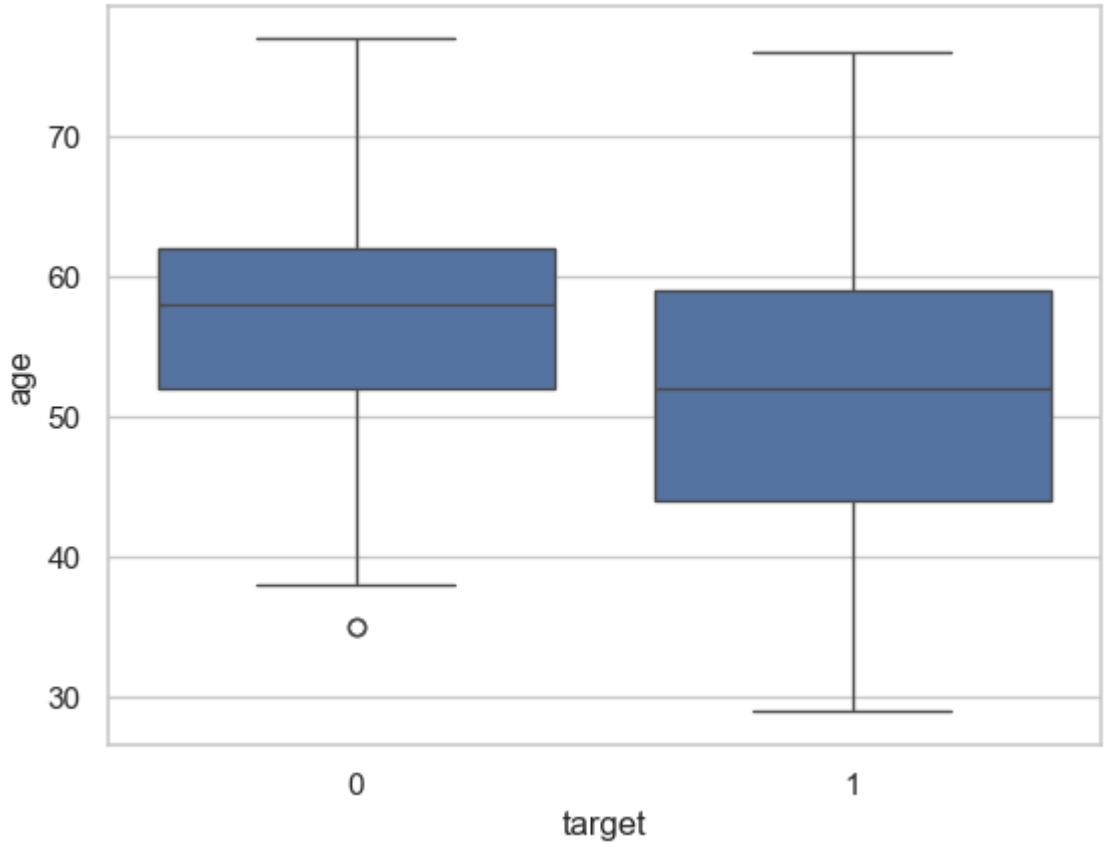


univariante analysis of age

In [46]:
```
ax=sns.stripplot(x='target',y='age',data=df)
```

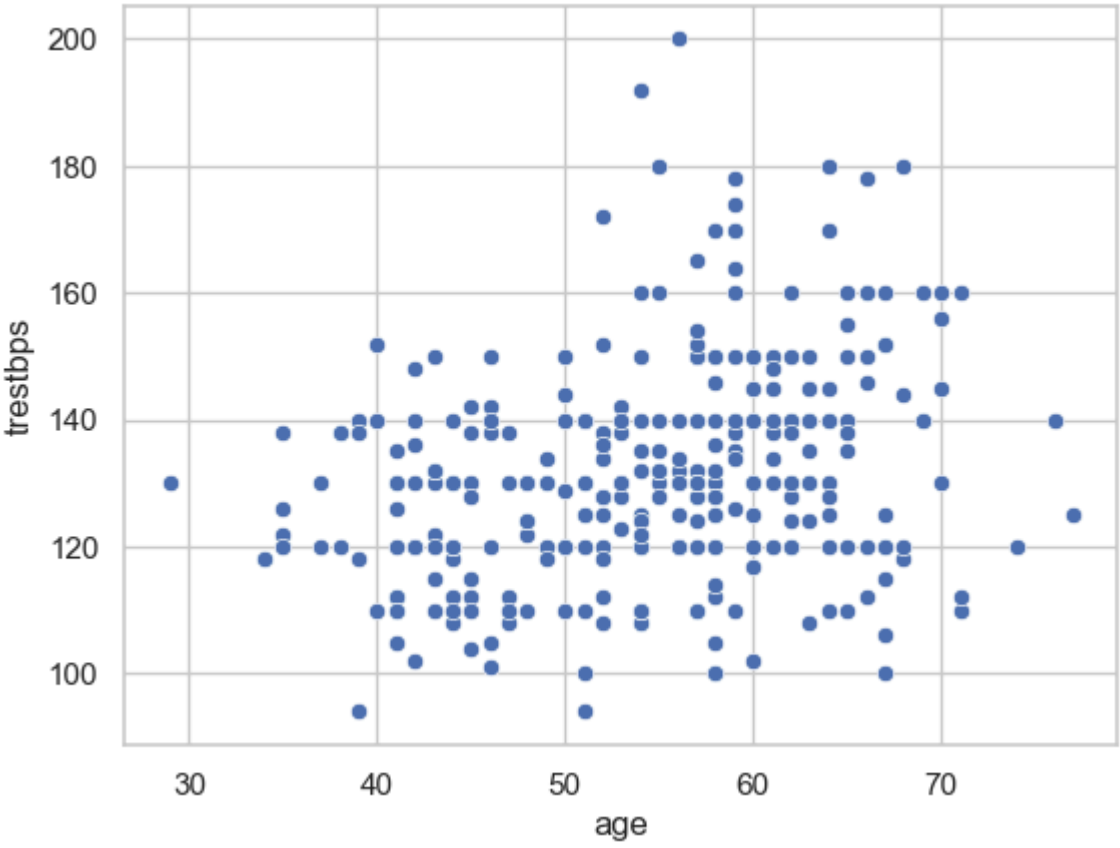


In [47]:
```
ax=sns.boxplot(x='target',y='age',data=df)
```

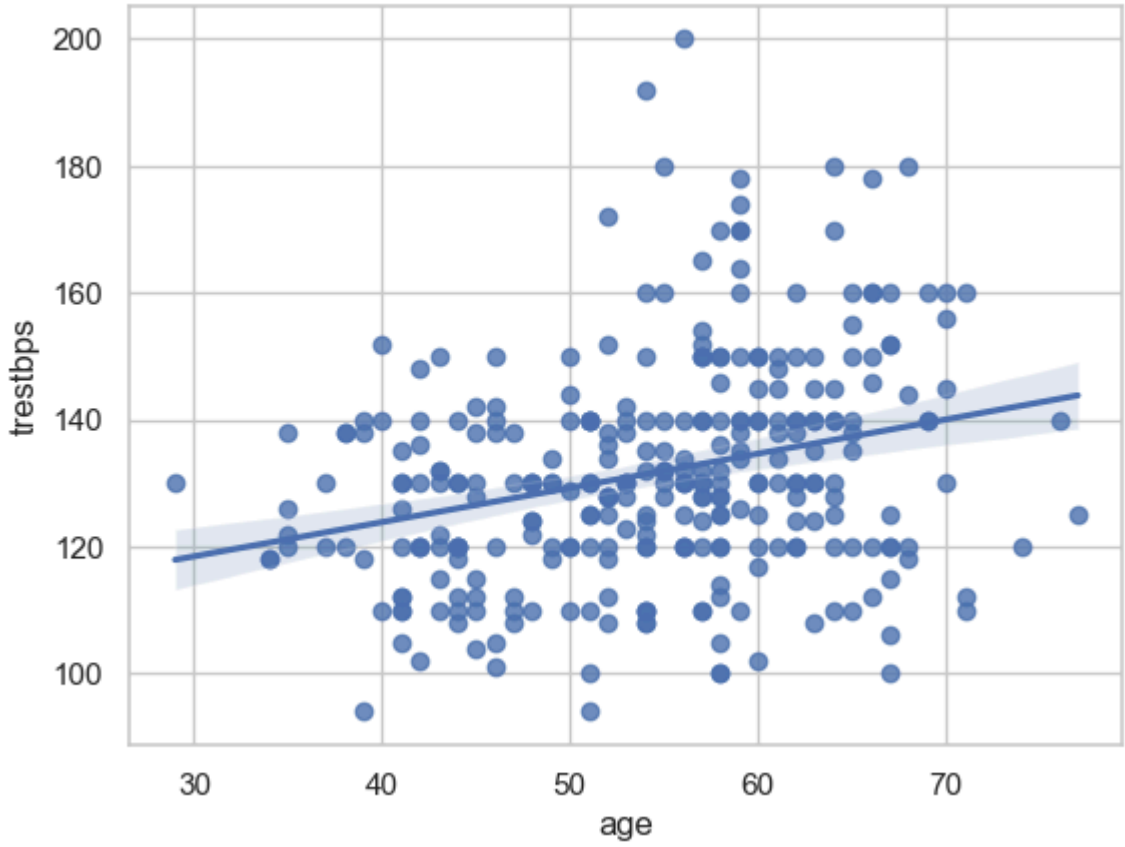


comparsion of age to the presence of heart disease

In [48]:
```
ax=sns.scatterplot(x='age',y='trestbps',data=df)
```

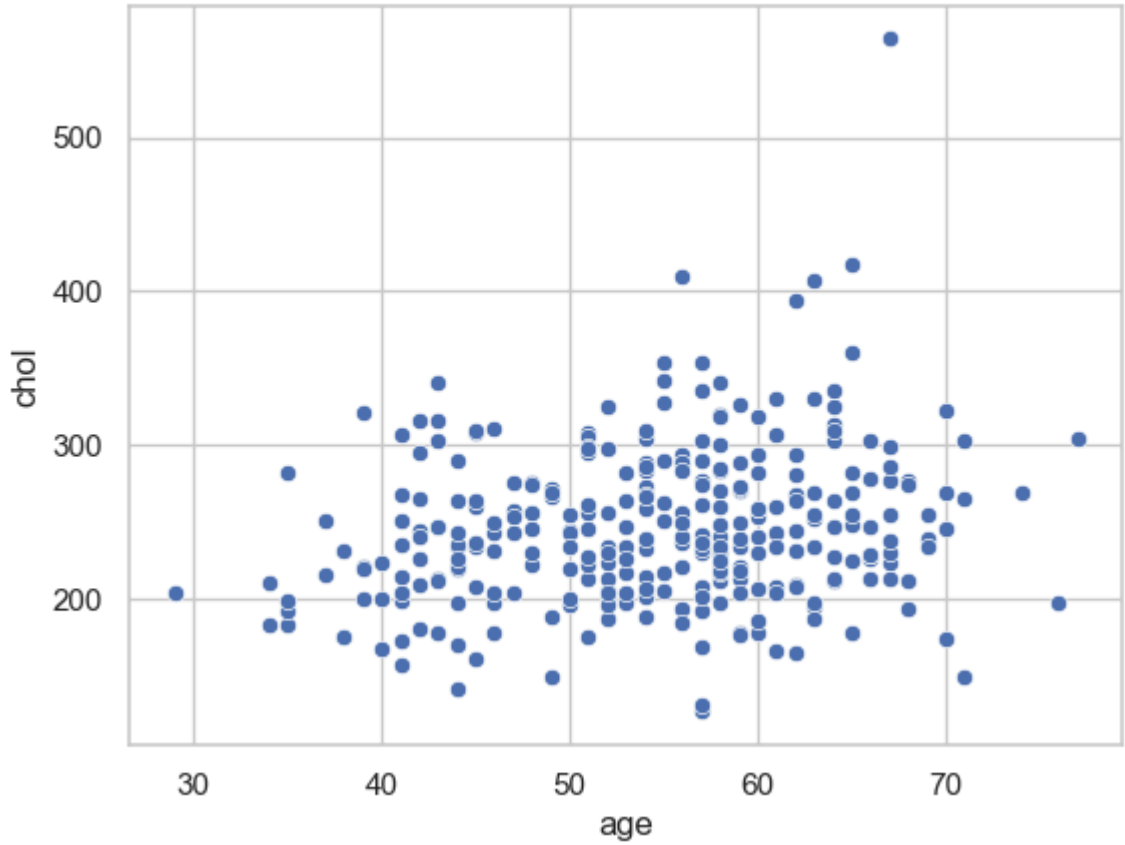


In [49]:
```
ax=sns.regplot(x='age',y='trestbps',data=df)
```

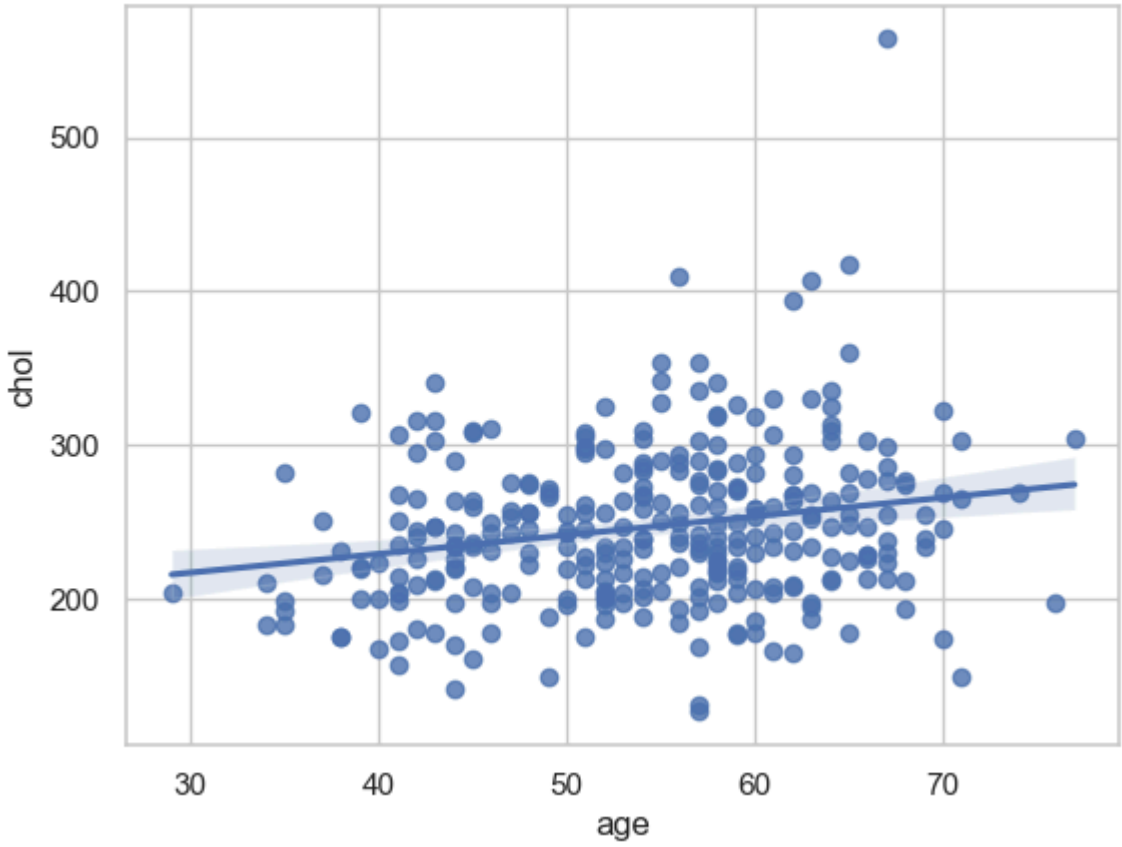


comparsion of age with the trestbps

In [50]:
```
ax=sns.scatterplot(x='age',y='chol',data=df)
```

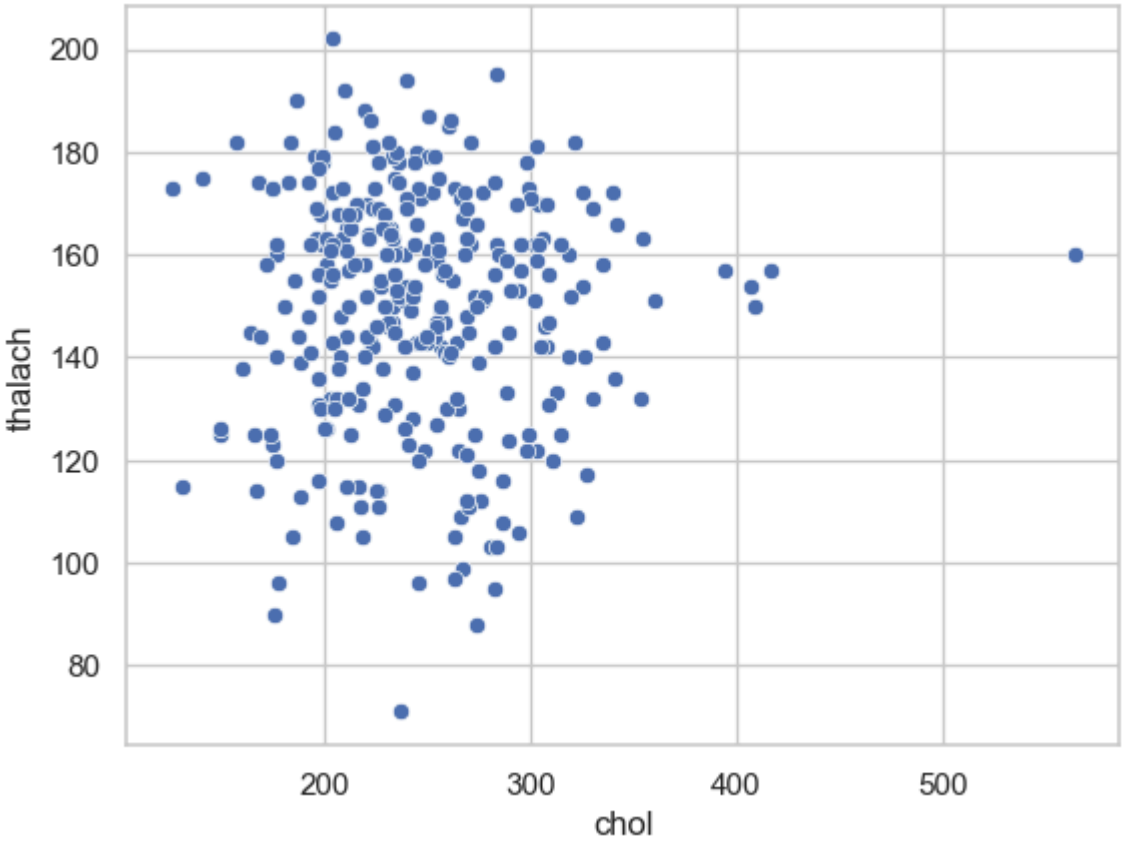


In [51]:
```
ax=sns.regplot(x='age',y='chol',data=df)
```

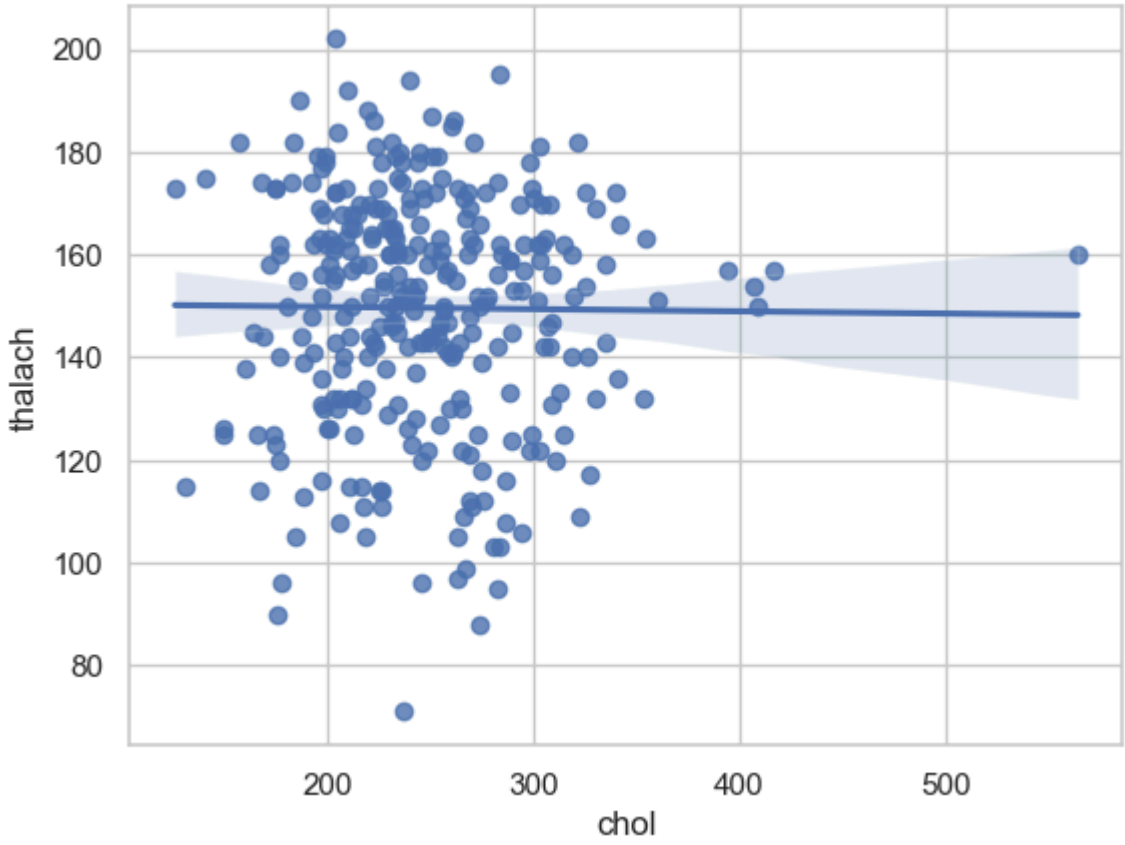


comparsion of age and chol

In [52]:
```
ax=sns.scatterplot(x='chol',y='thalach',data=df)
```



In [53]:
```
ax=sns.regplot(x='chol',y='thalach',data=df)
```



comparsion of chol to thalach

In [54]:
```
df.isnull().sum()
```

Out[54]:
```
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 [55]:
```
assert pd.notnull(df).all().all()
```

In [56]:
```
assert (df>=0).all().all()
```

In [57]:
```
df
```

Out[57]:

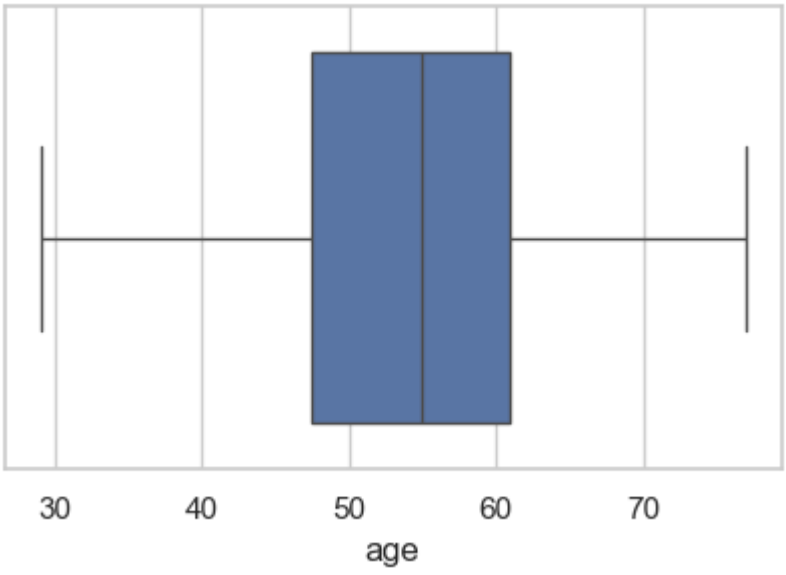
	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	th
0	63	1	3	145	233	1	0	150	0	2.3	0	0	
1	37	1	2	130	250	0	1	187	0	3.5	0	0	
2	41	0	1	130	204	0	0	172	0	1.4	2	0	
3	56	1	1	120	236	0	1	178	0	0.8	2	0	
4	57	0	0	120	354	0	1	163	1	0.6	2	0	
...	...	...	...	...	...	...	...	...	...	...	...	...	...
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	
302	57	0	1	130	236	0	0	174	0	0.0	1	1	

303 rows × 14 columns

In [58]:
```
df['age'].describe()
```

Out[58]:
```
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
```

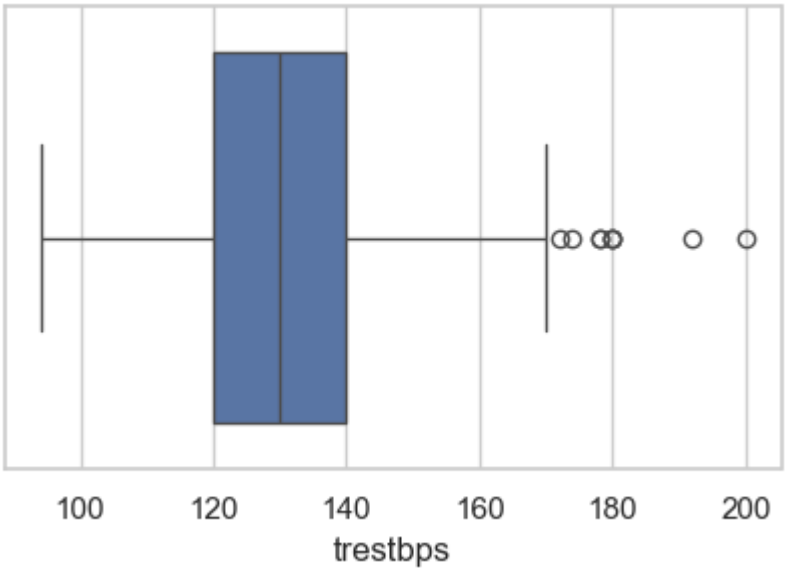
In [59]:
```
plt.subplots(figsize=(5,3))
ax=sns.boxplot(data=df,x='age')
```



In [60]:
```
df['trestbps'].describe()
```

Out[60]:
```
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
```

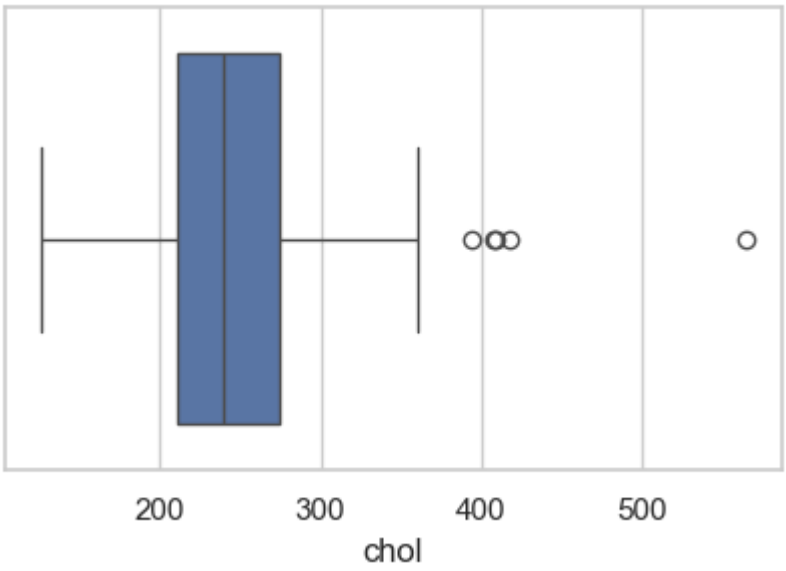
In [61]:
```
plt.subplots(figsize=(5,3))
ax=sns.boxplot(data=df,x='trestbps')
```



In [62]:
```
df['chol'].describe()
```

Out[62]:
```
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
```

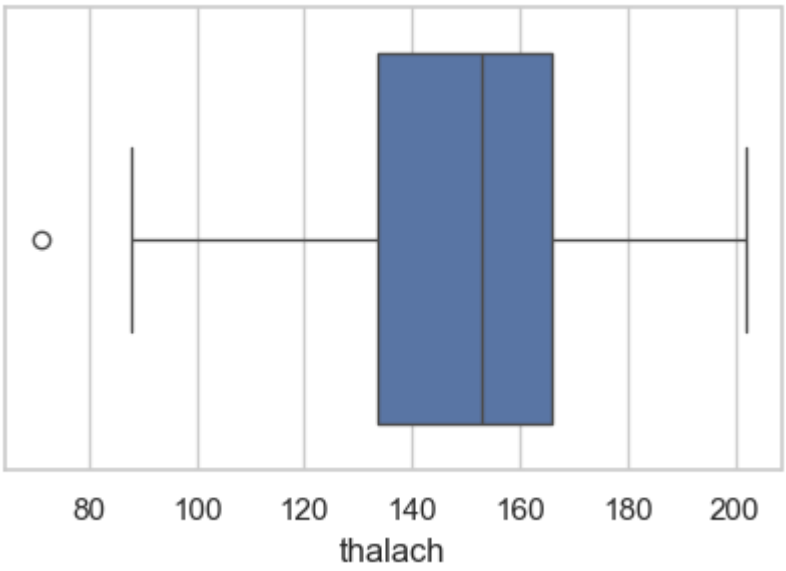
In [63]:
```
plt.subplots(figsize=(5,3))
ax=sns.boxplot(data=df,x='chol')
```



In [64]:
```
df['thalach'].describe()
```

Out[64]:
```
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
```

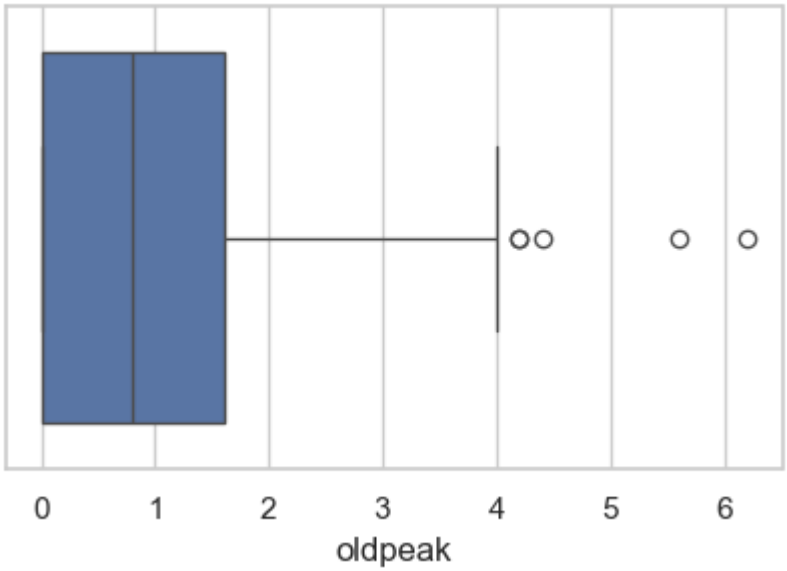
In [65]:
```
plt.subplots(figsize=(5,3))
ax=sns.boxplot(data=df,x='thalach')
```



In [66]:
```
df['oldpeak'].describe()
```

Out[66]:
```
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 [67]:
```
plt.subplots(figsize=(5,3))
ax=sns.boxplot(data=df,x='oldpeak')
```



here visualization of age,trestbps,chol,thalach,oldpeak by box plot is done inorder to find the outliers in the data