In [1]:
```
import numpy as np
import pandas as pd
import os
import seaborn as sns
import matplotlib.pyplot as plt
import scipy.stats as st
%matplotlib inline
```

In [2]:
```
sns.set(style='whitegrid')
```

In [3]:
```
import warnings
warnings.filterwarnings('ignore')
```

import dataset

In [4]:
```
df=pd.read_csv(r"C:\Users\nlnar\Downloads\25th, 26th- Advanced EDA project\25th, 26
```

In [5]:
```
df.shape     # no of rows ans columns
```

Out[5]: (303, 14)

In [6]:
```
print('The shape of the dataset:',df.shape)
```

```
The shape of the dataset: (303, 14)
```

In [7]:
```
print('The top 5 rows of the dataset:',df.head())
```

```
The top 5 rows of the dataset:    age  sex  cp  trestbps  chol  fbs  restecg  thalac
h  exang  oldpeak  slope  \
0   63    1   3       145   233    1        0      150      0      2.3      0
1   37    1   2       130   250    0        1      187      0      3.5      0
2   41    0   1       130   204    0        0      172      0      1.4      2
3   56    1   1       120   236    0        1      178      0      0.8      2
4   57    0   0       120   354    0        1      163      1      0.6      2

   ca  thal  target
0   0     1       1
1   0     2       1
2   0     2       1
3   0     2       1
4   0     2       1
```

In [8]:
```
print('The summary of the dataset:',df.info)
```

```
The summary of the dataset: <bound method DataFrame.info of      age  sex  cp  trest
bps  chol  fbs  restecg  thalach  exang  oldpeak  \
0     63    1   3       145   233    1        0      150      0      2.3
1     37    1   2       130   250    0        1      187      0      3.5
2     41    0   1       130   204    0        0      172      0      1.4
3     56    1   1       120   236    0        1      178      0      0.8
4     57    0   0       120   354    0        1      163      1      0.6
..   ...  ...  ..       ...   ...  ...      ...      ...    ...      ...
298   57    0   0       140   241    0        1      123      1      0.2
299   45    1   3       110   264    0        1      132      0      1.2
300   68    1   0       144   193    1        1      141      0      3.4
301   57    1   0       130   131    0        1      115      1      1.2
302   57    0   1       130   236    0        0      174      0      0.0

     slope  ca  thal  target
0        0   0     1       1
1        0   0     2       1
2        2   0     2       1
3        2   0     2       1
4        2   0     2       1
..     ...  ..   ...     ...
298      1   0     3       0
299      1   0     3       0
300      1   2     3       0
301      1   1     3       0
302      1   1     2       0

[303 rows x 14 columns]>
```

In [9]:
```
df.info()  # info of dataset
```

```
<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 [10]:
```
df.head
```

Out[10]:
```
<bound method NDFrame.head of      age  sex  cp  trestbps  chol  fbs  restecg  tha
lach  exang  oldpeak  \
0     63    1   3       145   233    1        0      150      0      2.3
1     37    1   2       130   250    0        1      187      0      3.5
2     41    0   1       130   204    0        0      172      0      1.4
3     56    1   1       120   236    0        1      178      0      0.8
4     57    0   0       120   354    0        1      163      1      0.6
..   ...  ...  ..       ...   ...  ...      ...      ...    ...      ...
298   57    0   0       140   241    0        1      123      1      0.2
299   45    1   3       110   264    0        1      132      0      1.2
300   68    1   0       144   193    1        1      141      0      3.4
301   57    1   0       130   131    0        1      115      1      1.2
302   57    0   1       130   236    0        0      174      0      0.0

     slope  ca  thal  target
0        0   0     1       1
1        0   0     2       1
2        2   0     2       1
3        2   0     2       1
4        2   0     2       1
..     ...  ..   ...     ...
298      1   0     3       0
299      1   0     3       0
300      1   2     3       0
301      1   1     3       0
302      1   1     2       0

[303 rows x 14 columns]>
```

In [11]:
```
df.tail
```

Out[11]:
```
<bound method NDFrame.tail of      age  sex  cp  trestbps  chol  fbs  restecg  tha
lach  exang  oldpeak  \
0     63    1   3       145   233    1        0      150      0      2.3
1     37    1   2       130   250    0        1      187      0      3.5
2     41    0   1       130   204    0        0      172      0      1.4
3     56    1   1       120   236    0        1      178      0      0.8
4     57    0   0       120   354    0        1      163      1      0.6
..   ...  ...  ..       ...   ...  ...      ...      ...    ...      ...
298   57    0   0       140   241    0        1      123      1      0.2
299   45    1   3       110   264    0        1      132      0      1.2
300   68    1   0       144   193    1        1      141      0      3.4
301   57    1   0       130   131    0        1      115      1      1.2
302   57    0   1       130   236    0        0      174      0      0.0

     slope  ca  thal  target
0        0   0     1       1
1        0   0     2       1
2        2   0     2       1
3        2   0     2       1
4        2   0     2       1
..     ...  ..   ...     ...
298      1   0     3       0
299      1   0     3       0
300      1   2     3       0
301      1   1     3       0
302      1   1     2       0

[303 rows x 14 columns]>
```

In [12]:
```
df.tail()   # bottom 5 rows
```

Out[12]:

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal
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

In [13]:
```
df.head()   # top 5 rows
```

Out[13]:

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	t
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 [14]:
```
df.describe()                # description of the dataset
```

Out[14]:

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

In [15]:
```
df.describe().T       #description of the dataset in transpose
```

Out[15]:

	count	mean	std	min	25%	50%	75%	max
age	303.0	54.366337	9.082101	29.0	47.5	55.0	61.0	77.0
sex	303.0	0.683168	0.466011	0.0	0.0	1.0	1.0	1.0
cp	303.0	0.966997	1.032052	0.0	0.0	1.0	2.0	3.0
trestbps	303.0	131.623762	17.538143	94.0	120.0	130.0	140.0	200.0
chol	303.0	246.264026	51.830751	126.0	211.0	240.0	274.5	564.0
fbs	303.0	0.148515	0.356198	0.0	0.0	0.0	0.0	1.0
restecg	303.0	0.528053	0.525860	0.0	0.0	1.0	1.0	2.0
thalach	303.0	149.646865	22.905161	71.0	133.5	153.0	166.0	202.0
exang	303.0	0.326733	0.469794	0.0	0.0	0.0	1.0	1.0
oldpeak	303.0	1.039604	1.161075	0.0	0.0	0.8	1.6	6.2
slope	303.0	1.399340	0.616226	0.0	1.0	1.0	2.0	2.0
ca	303.0	0.729373	1.022606	0.0	0.0	0.0	1.0	4.0
thal	303.0	2.313531	0.612277	0.0	2.0	2.0	3.0	3.0
target	303.0	0.544554	0.498835	0.0	0.0	1.0	1.0	1.0

In [16]:
```
df.columns      # gives all the columns in the dataset
```

Out[16]:
```
Index(['age', 'sex', 'cp', 'trestbps', 'chol', 'fbs', 'restecg', 'thalach',
       'exang', 'oldpeak', 'slope', 'ca', 'thal', 'target'],
      dtype='object')
```

In [17]:
```
df.dtypes      #gives datatype of the attributes
```

Out[17]:
```
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 [18]:
```
df.describe()
```

Out[18]:

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

In [19]:
```
df.describe(include='all')
```

Out[19]:

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

In [20]:
```
df.target
```

Out[20]:
```
0      1
1      1
2      1
3      1
4      1
      ..
298    0
299    0
300    0
301    0
302    0
Name: target, Length: 303, dtype: int64
```

In [21]:
```
df['target'].nunique()
```

Out[21]: 2

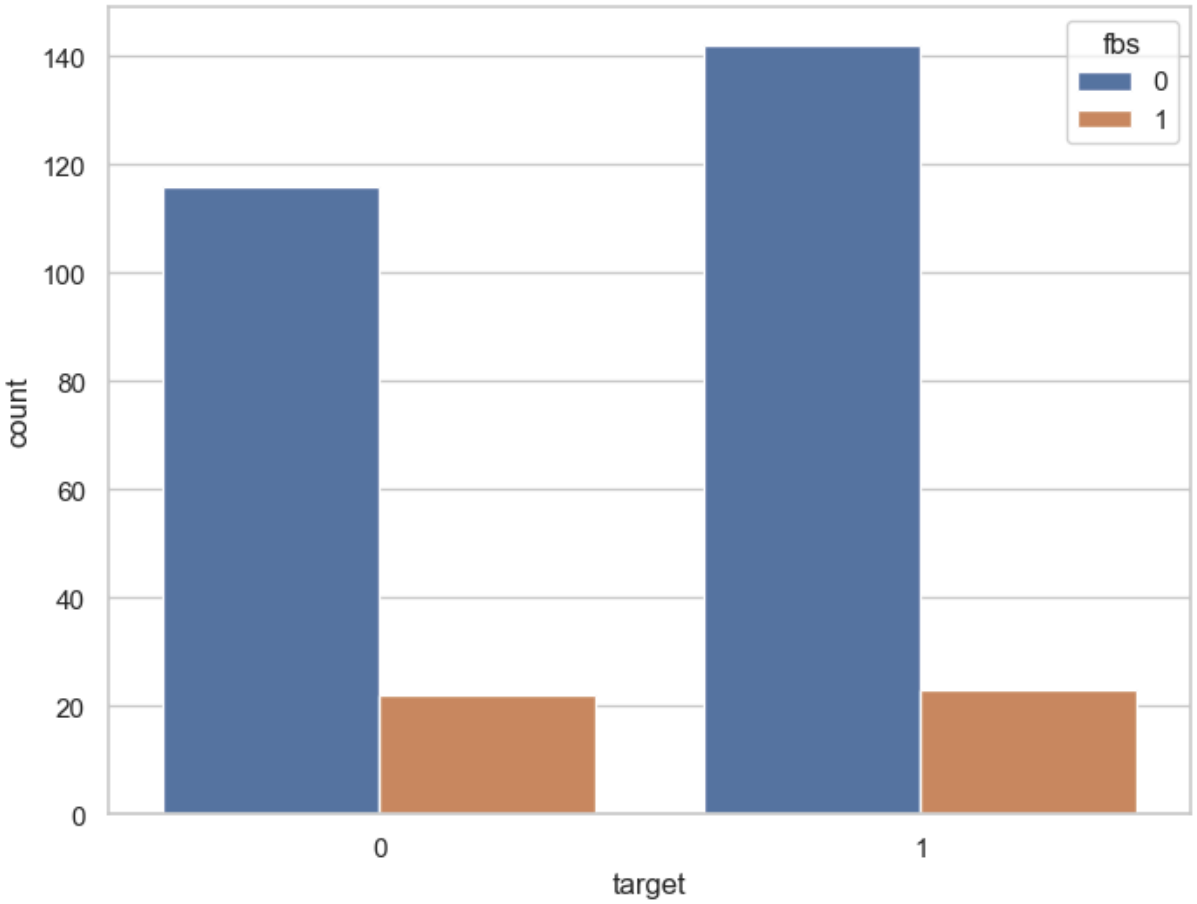
In [22]:
```
df['target'].unique()  # 1 shows presence of heart disease 0 shows no heart disease
```

Out[22]: array([1, 0])

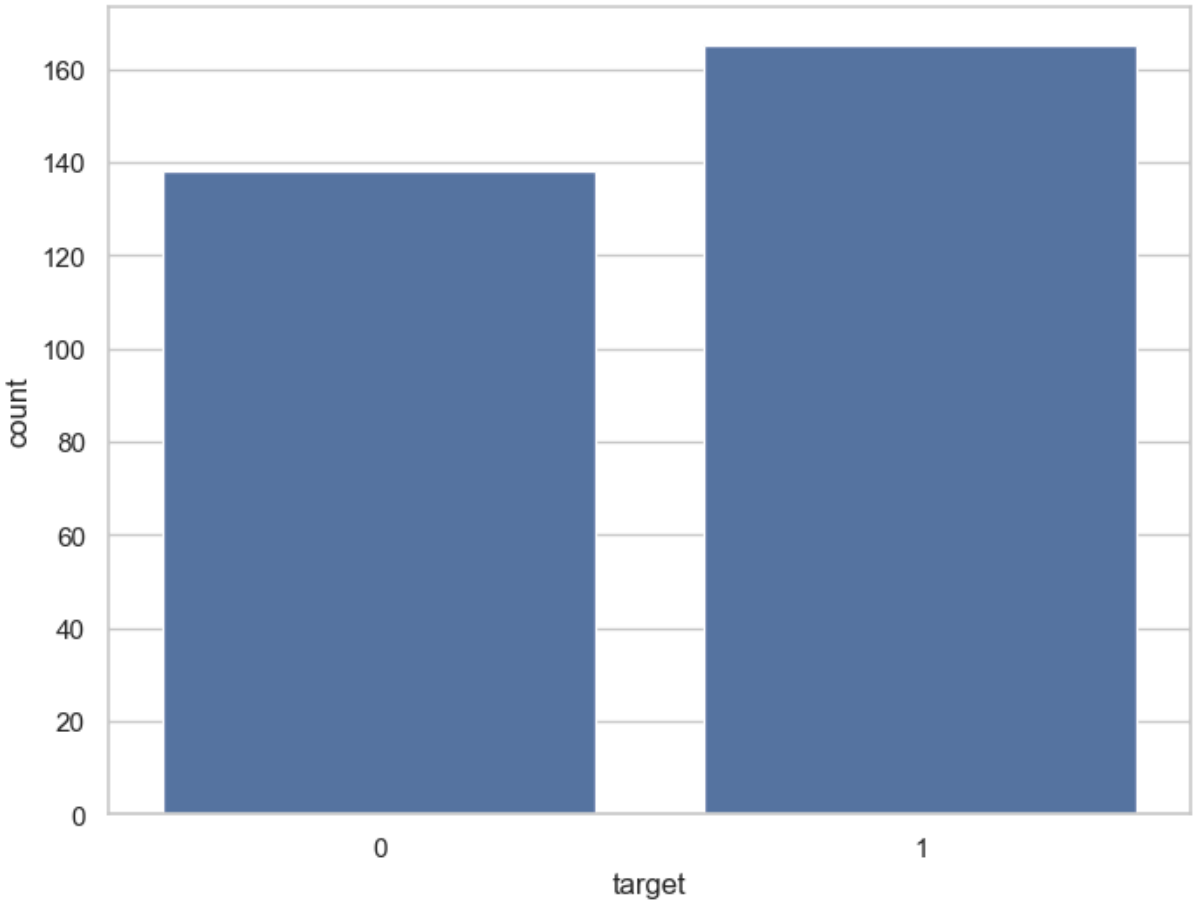
In [23]:
```
df['target'].value_counts()   # 165 ppl have heart prob 138 have no heart prob
```

Out[23]:
```
target
1    165
0    138
Name: count, dtype: int64
```

In [24]:
```
f,ax=plt.subplots(figsize=(8,6))
sns.countplot(data=df,hue='fbs',x='target')
plt.show()
```



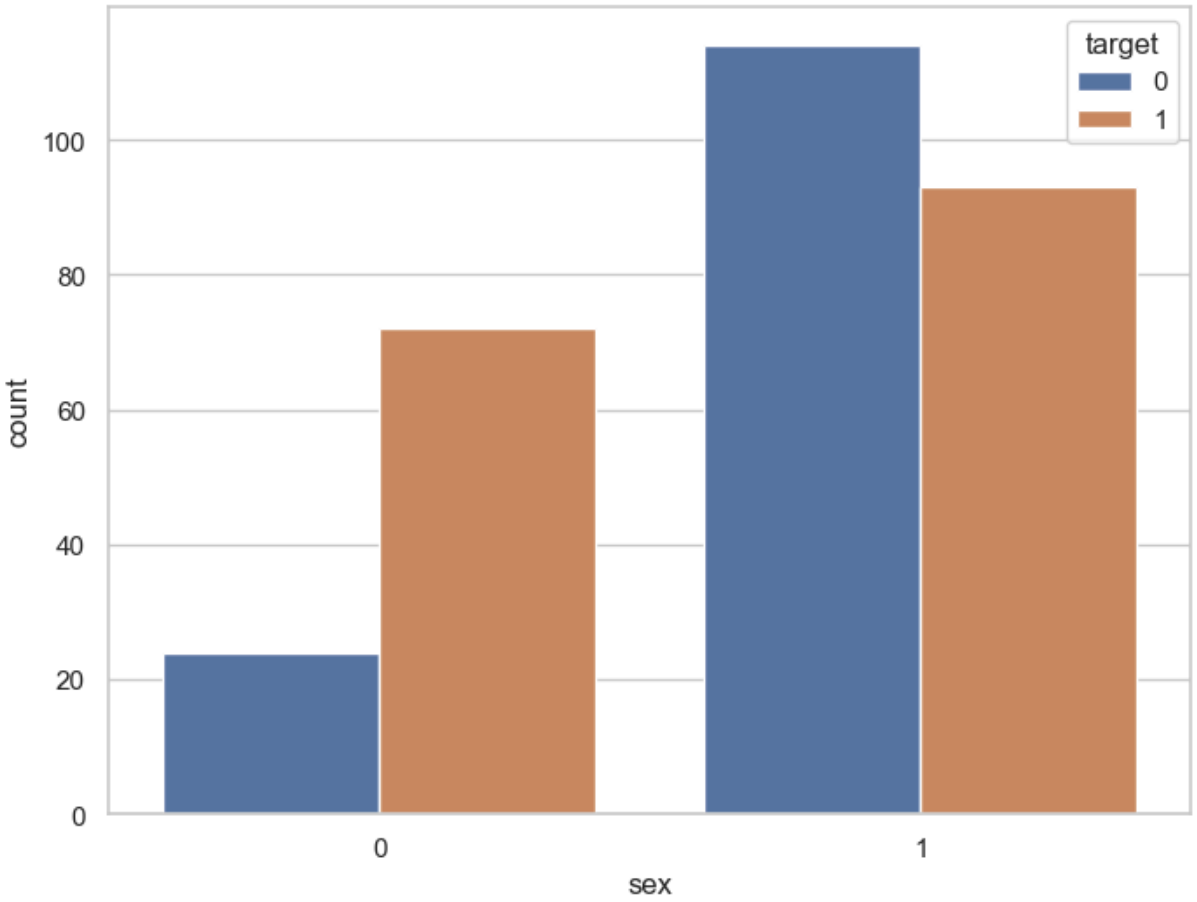
In [25]:
```
f,ax=plt.subplots(figsize=(8,6))
sns.countplot(x='target',data=df)
plt.show()                                  # this shows 138 notsuffering from heart
```



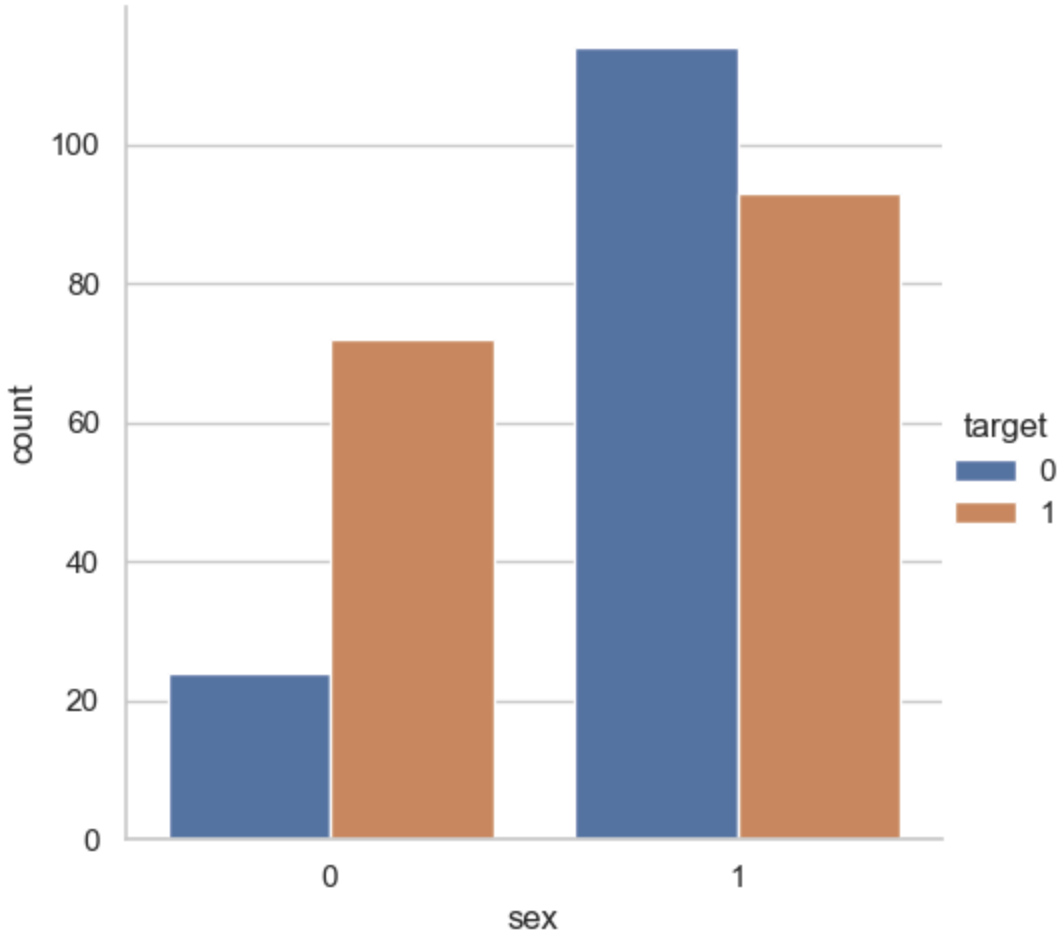
In [26]:
```
df.groupby('sex')['target'].value_counts()
```

Out[26]:
```
sex  target
0    1          72
     0          24
1    0         114
     1          93
Name: count, dtype: int64
```

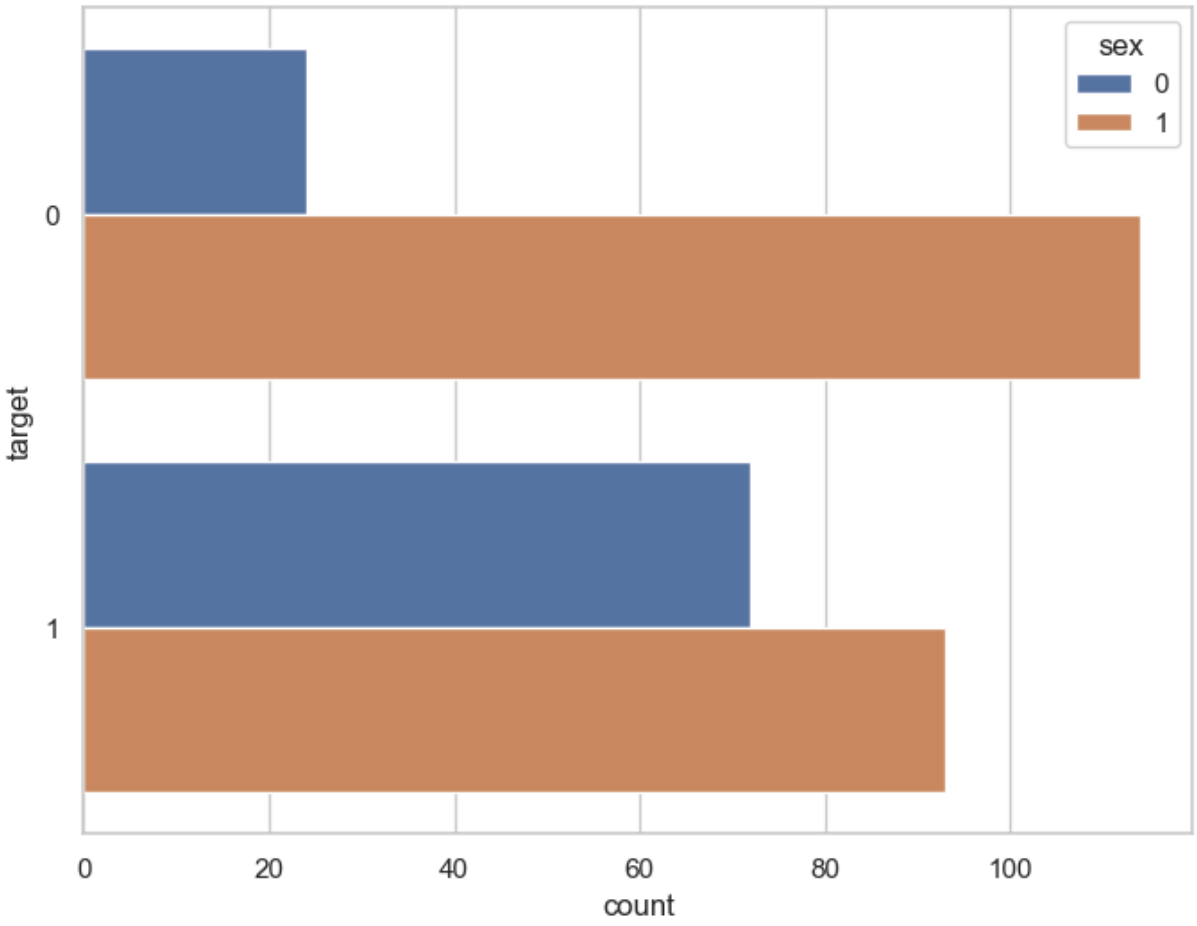
In [27]:
```
f,ax=plt.subplots(figsize=(8,6))
ax=sns.countplot(data=df,x='sex',hue='target',)
plt.show()                                          #  sex 0 for female and 1 for male
```



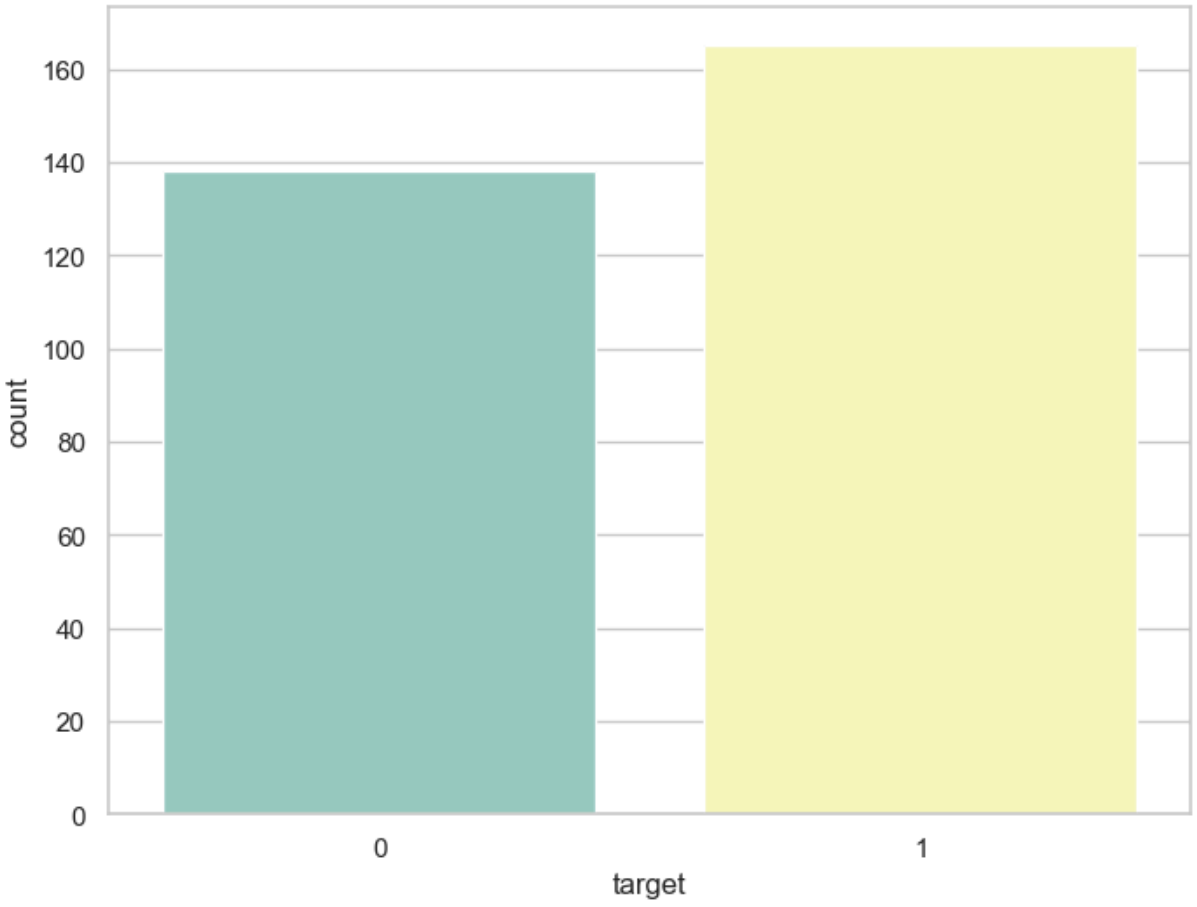
In [28]:
```
ax=sns.catplot(data=df,x='sex',hue='target',kind='count',height=5,aspect=1)
plt.show()
```



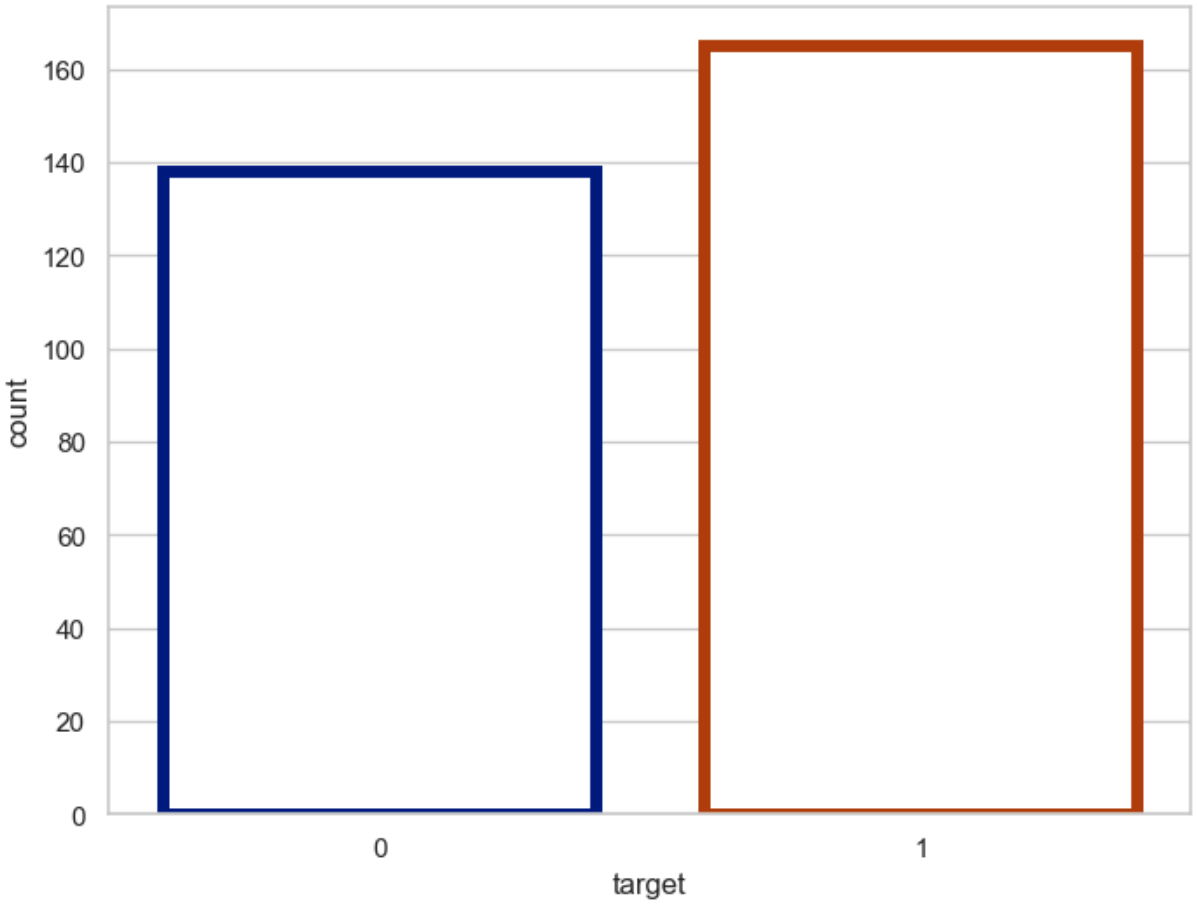
In [29]:
```
f,ax=plt.subplots(figsize=(8,6))
ax=sns.countplot(y='target',hue='sex',data=df,)
plt.show()
```



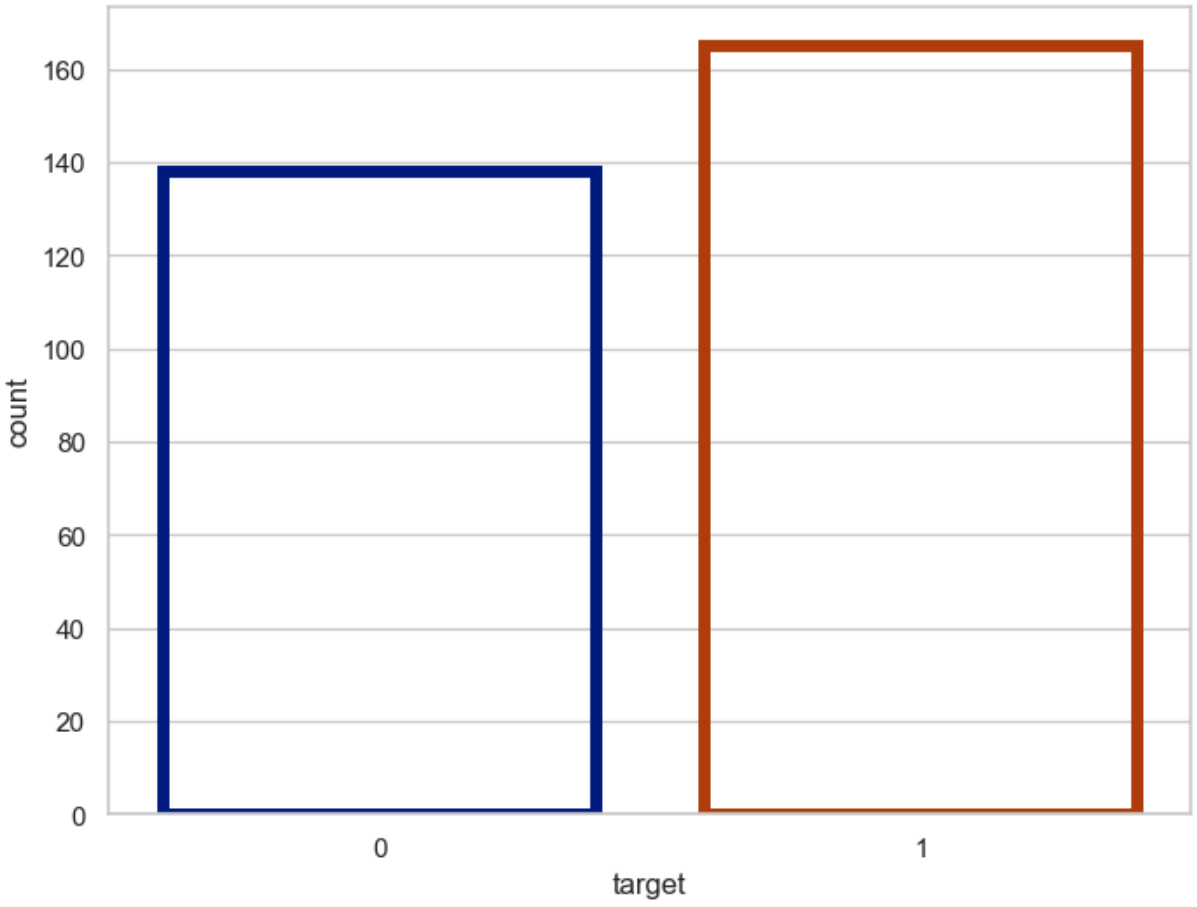
In [30]:
```
f,ax=plt.subplots(figsize=(8,6))
ax=sns.countplot(x='target',palette="Set3",data=df)
plt.show()
```



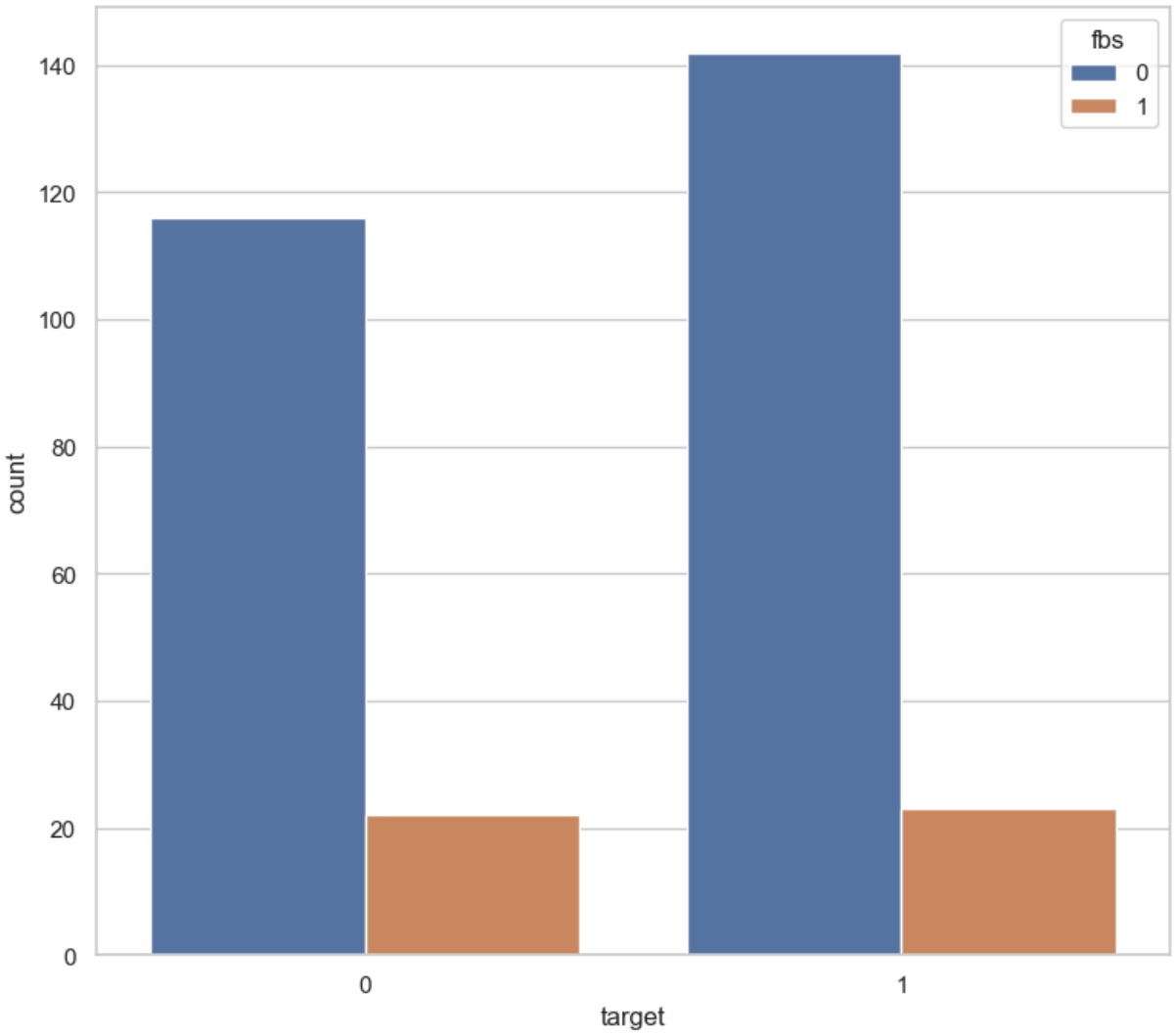
In [31]:
```
f,ax=plt.subplots(figsize=(8,6))
ax=sns.countplot(x='target',data=df,facecolor=(1,1,1,1),linewidth=5,edgecolor=sns.c
plt.show()
```



In [32]:
```
f,ax=plt.subplots(figsize=(8,6))
ax=sns.countplot(x='target',data=df,facecolor=(0,0,0,0),linewidth=5,edgecolor=sns.c
plt.show()
```



In [33]:
```
f,ax=plt.subplots(figsize=(9,8))
ax=sns.countplot(x='target',hue='fbs',data=df)
plt.show()
```



In [34]:
```
correlation=df.corr()
```

In [35]:
```
correlation['target'].sort_values(ascending=False)
```

Out[35]:
```
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 [36]:
```
df['cp'].nunique()
```

Out[36]: 4

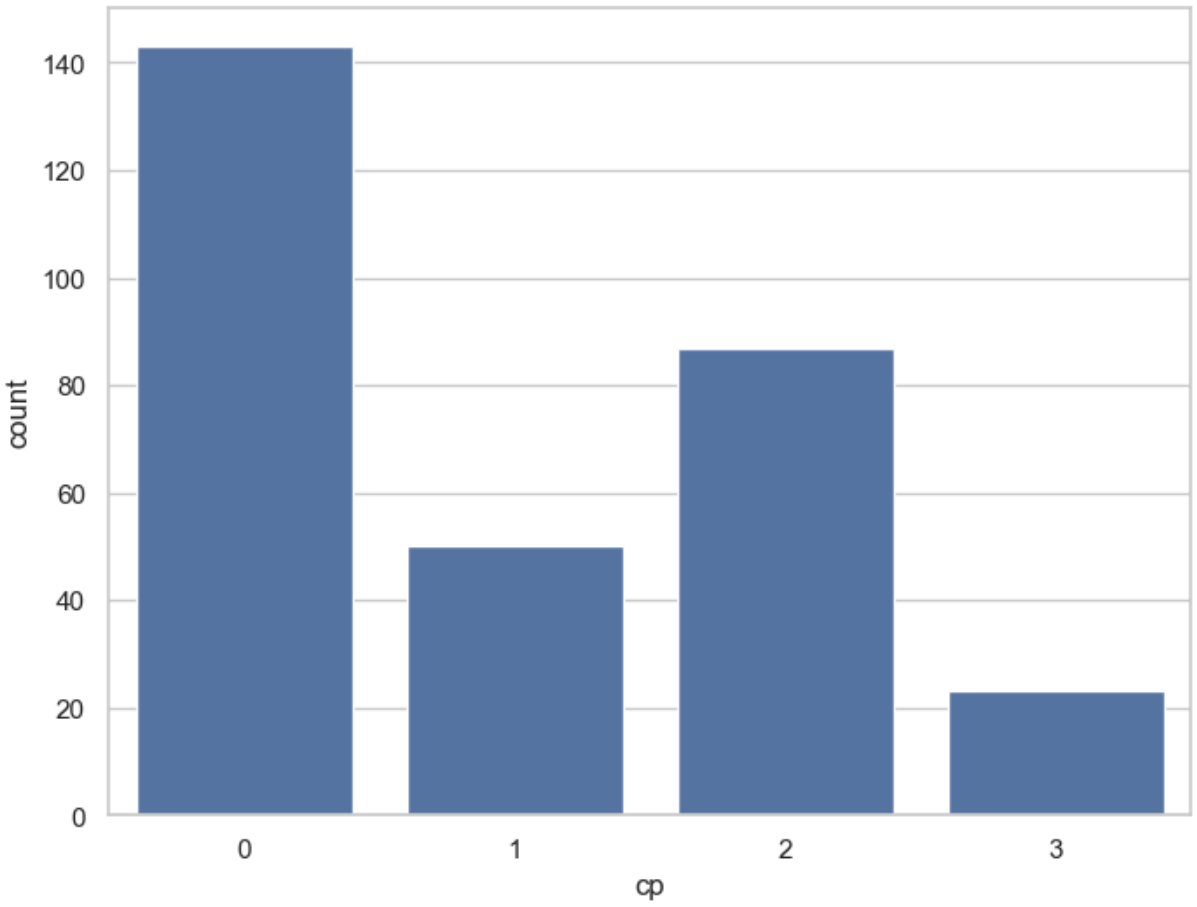
In [37]:
```
df['cp'].value_counts()
```

Out[37]:
```
cp
0    143
2     87
1     50
3     23
Name: count, dtype: int64
```

In [38]:
```
df['cp'].unique()
```

Out[38]:
```
array([3, 2, 1, 0])
```

In [39]:
```
f,ax=plt.subplots(figsize=(8,6))
ax=sns.countplot(x='cp',data=df)
plt.show()
```

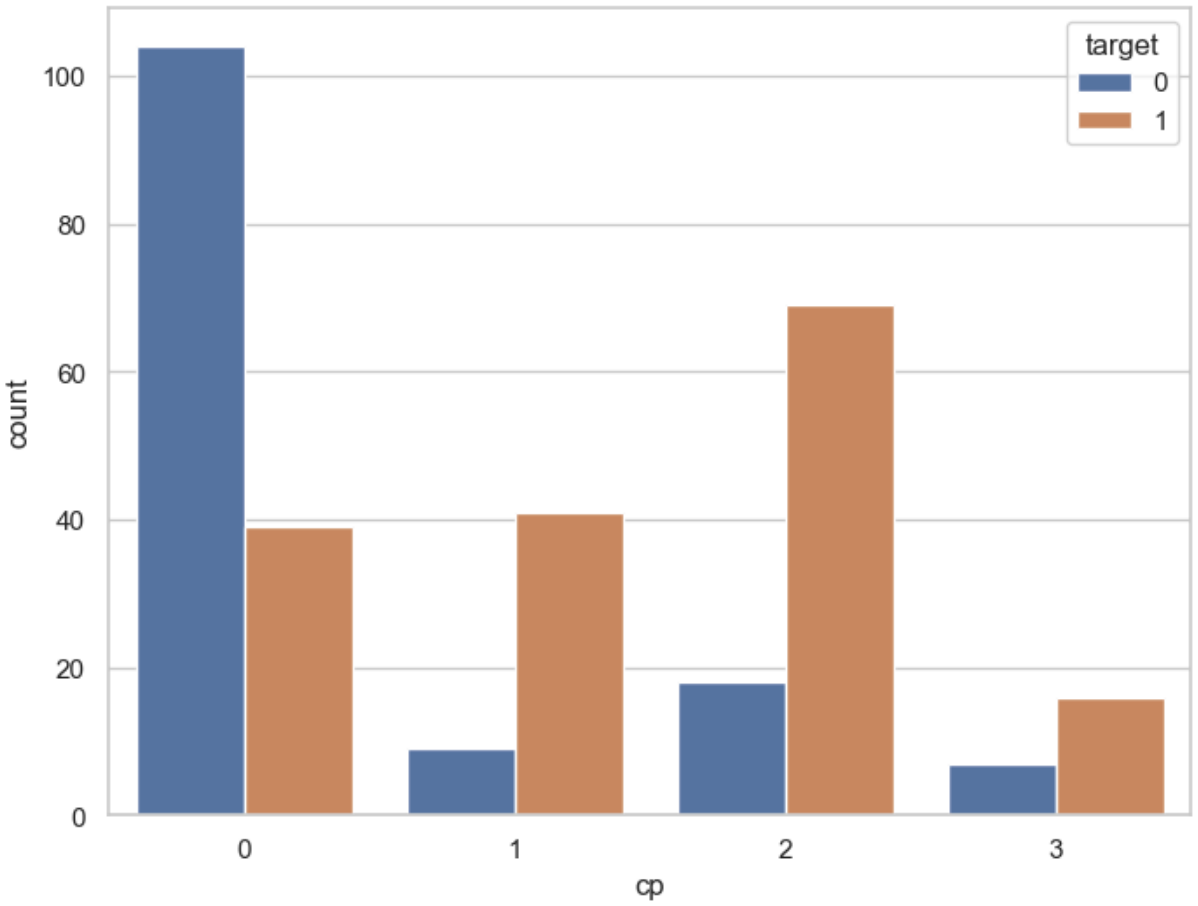


In [40]:
```
df.groupby('cp')['target'].value_counts().unstack()
```

Out[40]:

target	0	1
cp		
0	104	39
1	9	41
2	18	69
3	7	16

In [41]:
```
f,ax=plt.subplots(figsize=(8,6))
ax=sns.countplot(x='cp',hue='target',data=df)
plt.show()
```



In [42]:
```
df['thalach'].nunique()
```

Out[42]: 91

In [43]:
```
df['thalach'].unique()
```

Out[43]:
```
array([150, 187, 172, 178, 163, 148, 153, 173, 162, 174, 160, 139, 171,
       144, 158, 114, 151, 161, 179, 137, 157, 123, 152, 168, 140, 188,
       125, 170, 165, 142, 180, 143, 182, 156, 115, 149, 146, 175, 186,
       185, 159, 130, 190, 132, 147, 154, 202, 166, 164, 184, 122, 169,
       138, 111, 145, 194, 131, 133, 155, 167, 192, 121,  96, 126, 105,
       181, 116, 108, 129, 120, 112, 128, 109, 113,  99, 177, 141, 136,
        97, 127, 103, 124,  88, 195, 106,  95, 117,  71, 118, 134,  90])
```

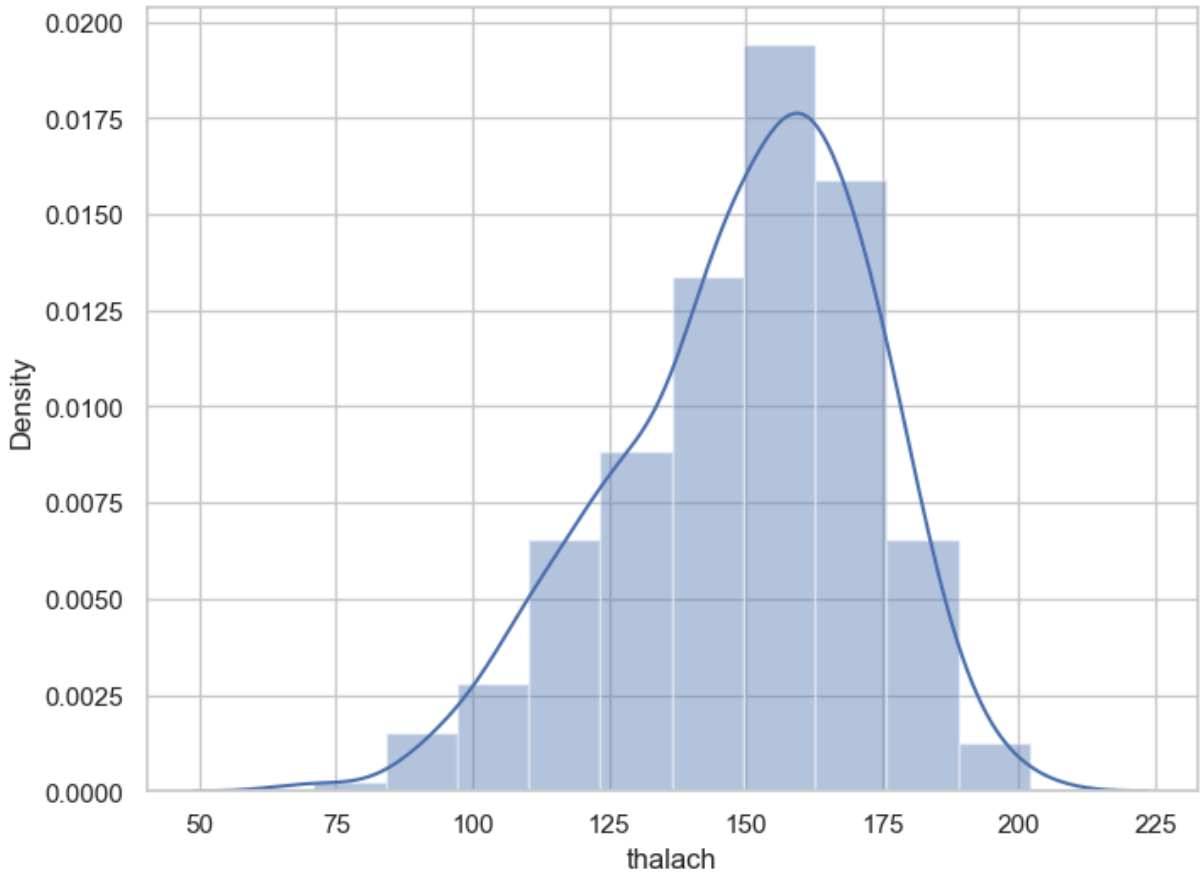
In [44]:
```
df['thalach'].value_counts()     # 91 unique values in thalach variable
```

Out[44]:
```
thalach
162    11
163     9
160     9
173     8
152     8
       ..
117     1
71      1
118     1
134     1
90      1
Name: count, Length: 91, dtype: int64
```

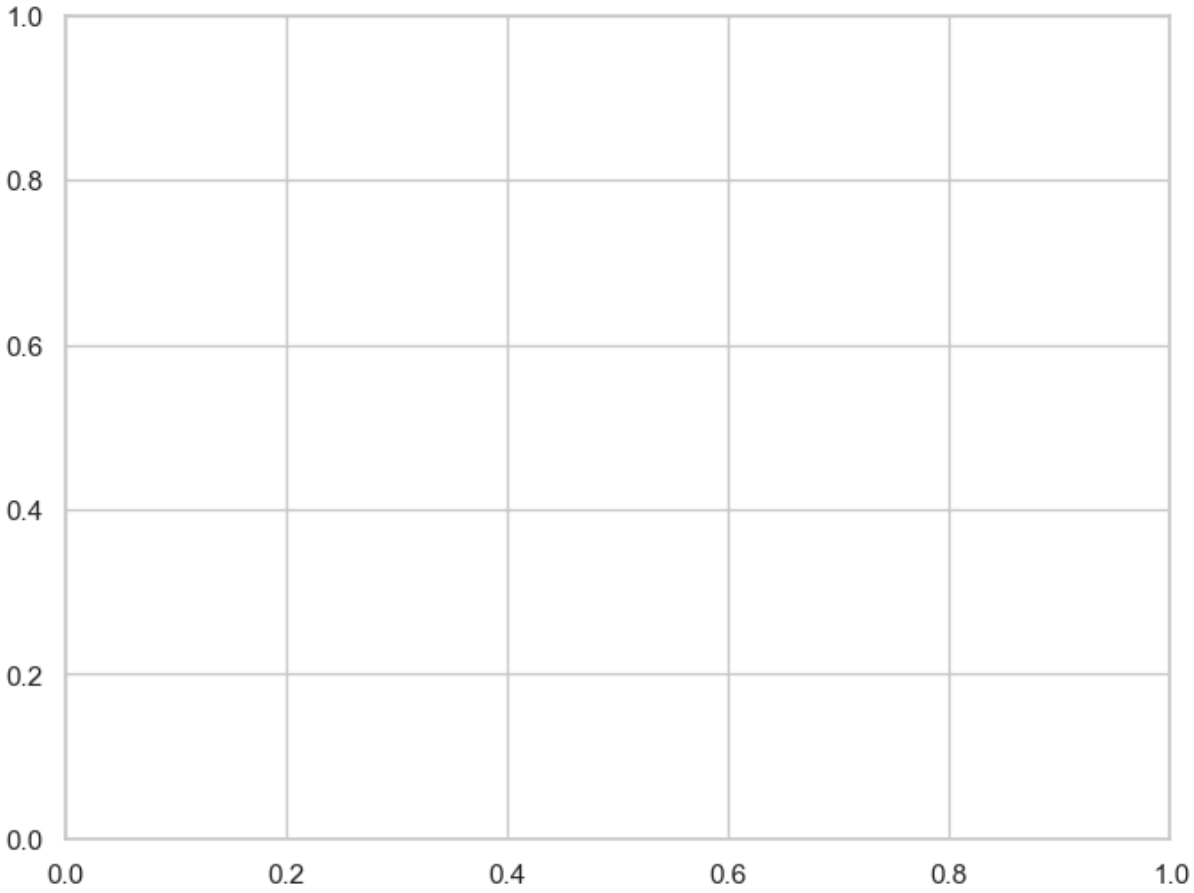
visualisation of frequency distribution of thalach variable

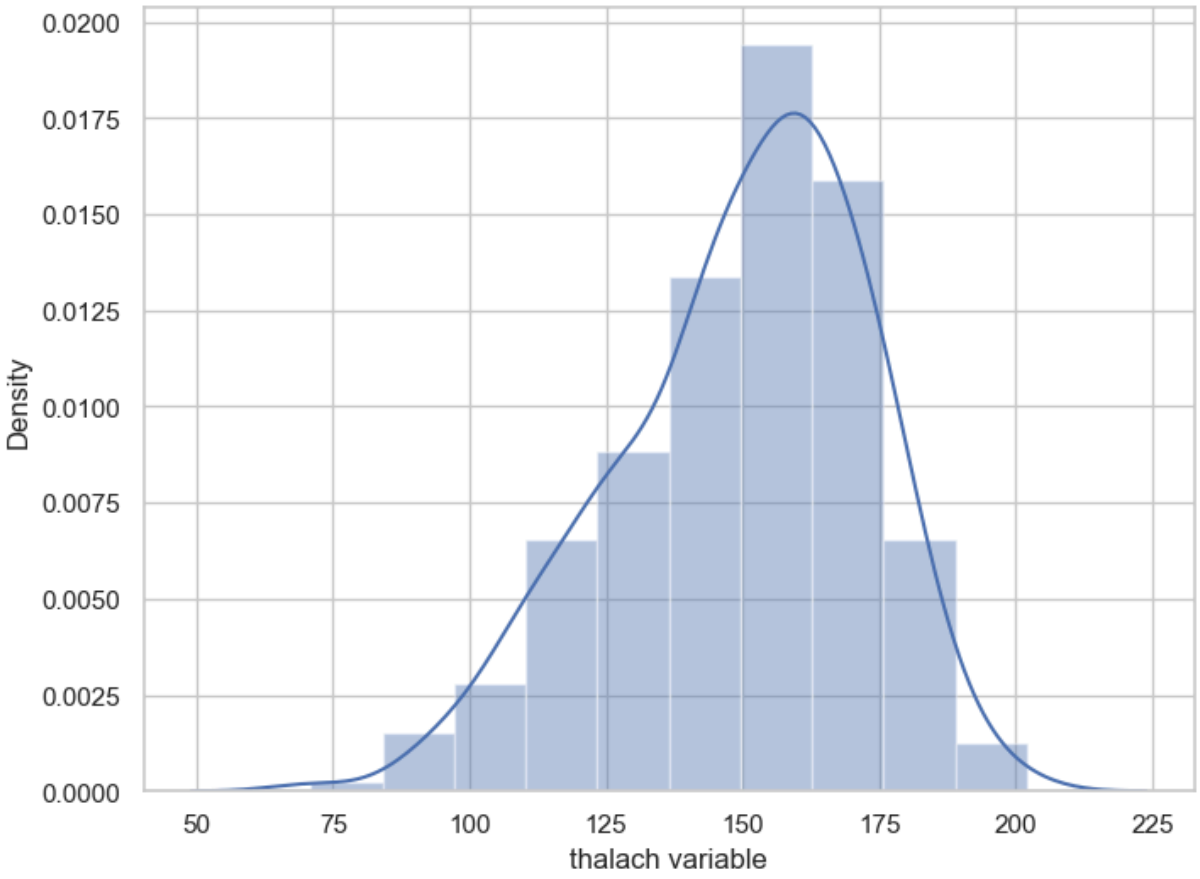
In [52]:
```
f,ax=plt.subplots(figsize=(8,6))                              # negative skew is observe
x=df['thalach']
ax=sns.distplot(x,bins=10)
plt.show()
```


1.0
0.8
0.6
0.4
0.2
0.0
71

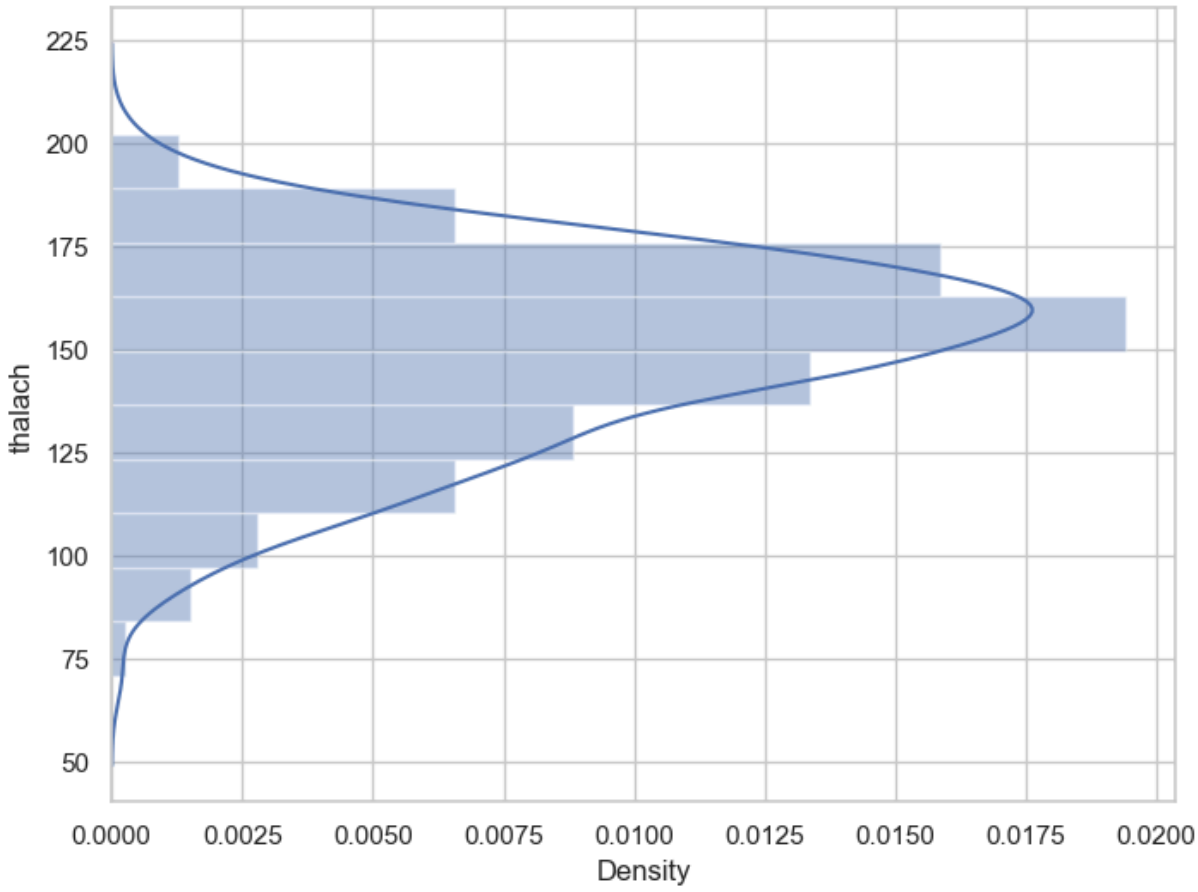

0.0200
0.0175
0.0150
0.0125
0.0100
0.0075
0.0050
0.0025
0.0000
Density
50
75
100
125
150
175
200
225
thalach

In [54]:
```
f,ax=plt.subplots(figsize=(8,6))
x=df['thalach']
x=pd.Series(x,name='thalach variable')
ax=sns.distplot(x,bins=10)
plt.show()
```

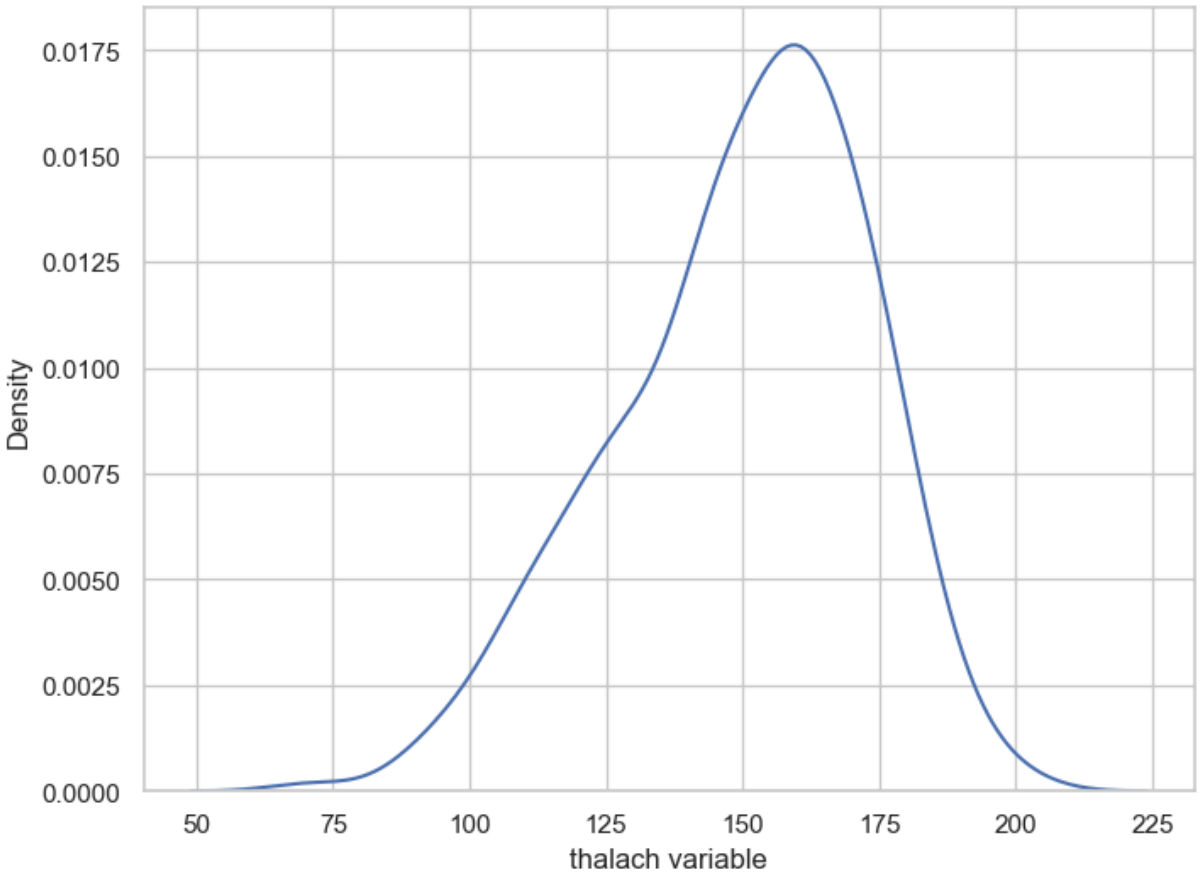




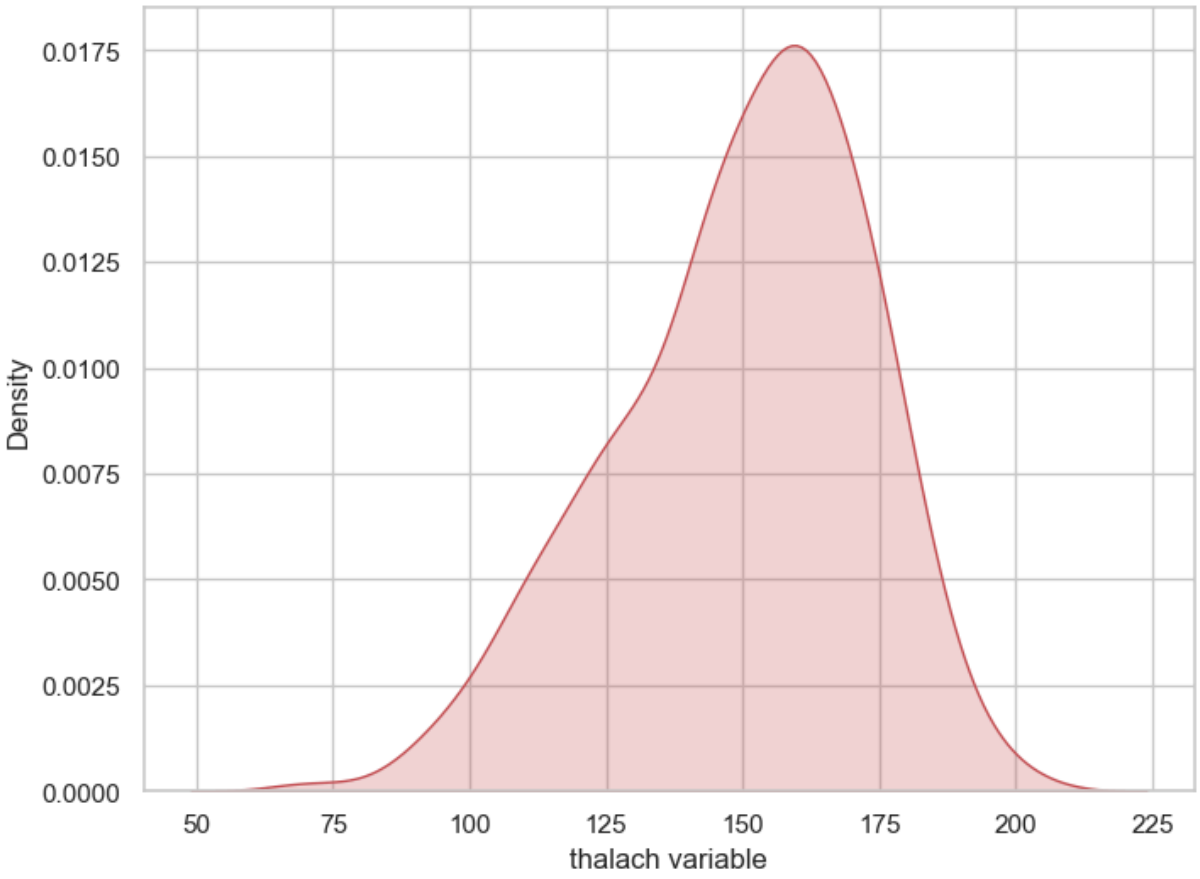
In [55]:
```
f,ax=plt.subplots(figsize=(8,6))
x=df['thalach']
ax=sns.distplot(x,bins=10,vertical=True)              # distribution on the vertic
plt.show()
```



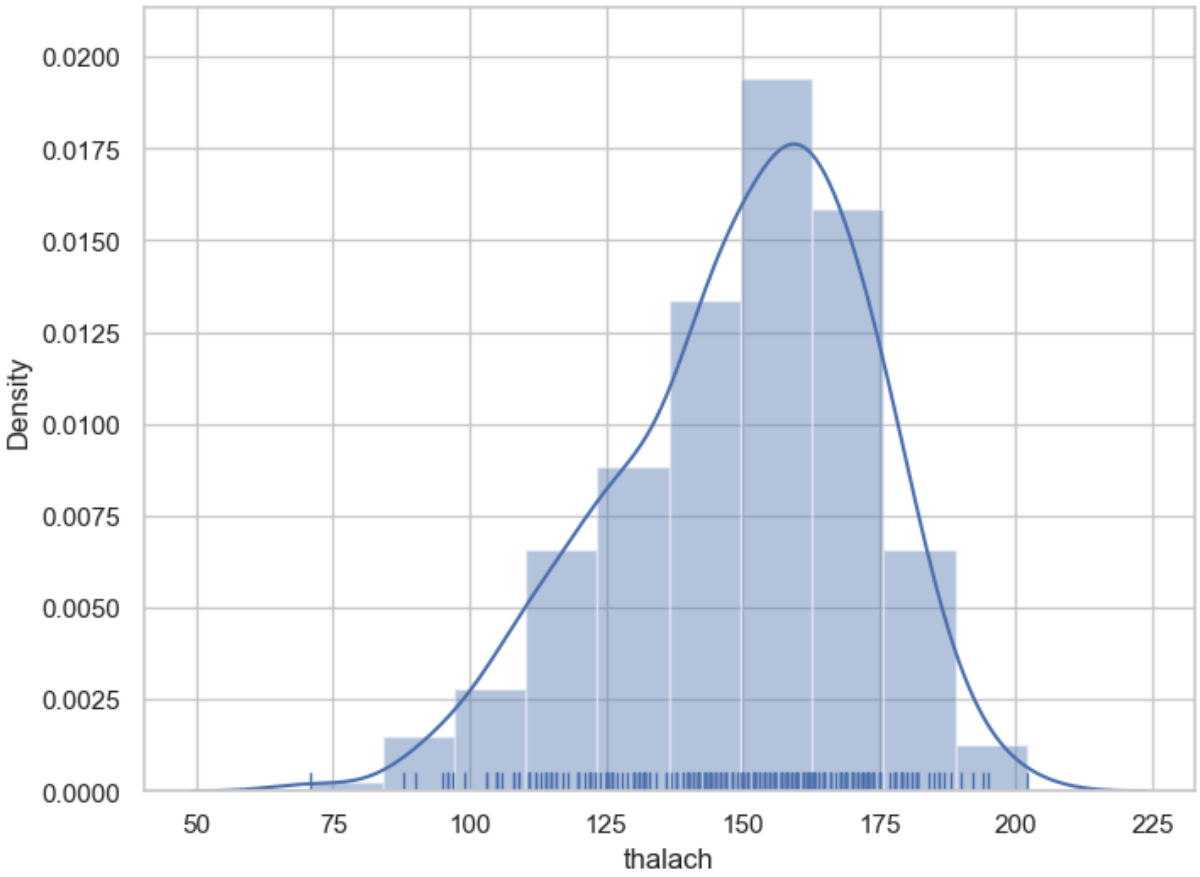
In [56]:
```
f,ax=plt.subplots(figsize=(8,6))
x=df['thalach']
x=pd.Series(x,name='thalach variable')
ax=sns.kdeplot(x)
plt.show()
```



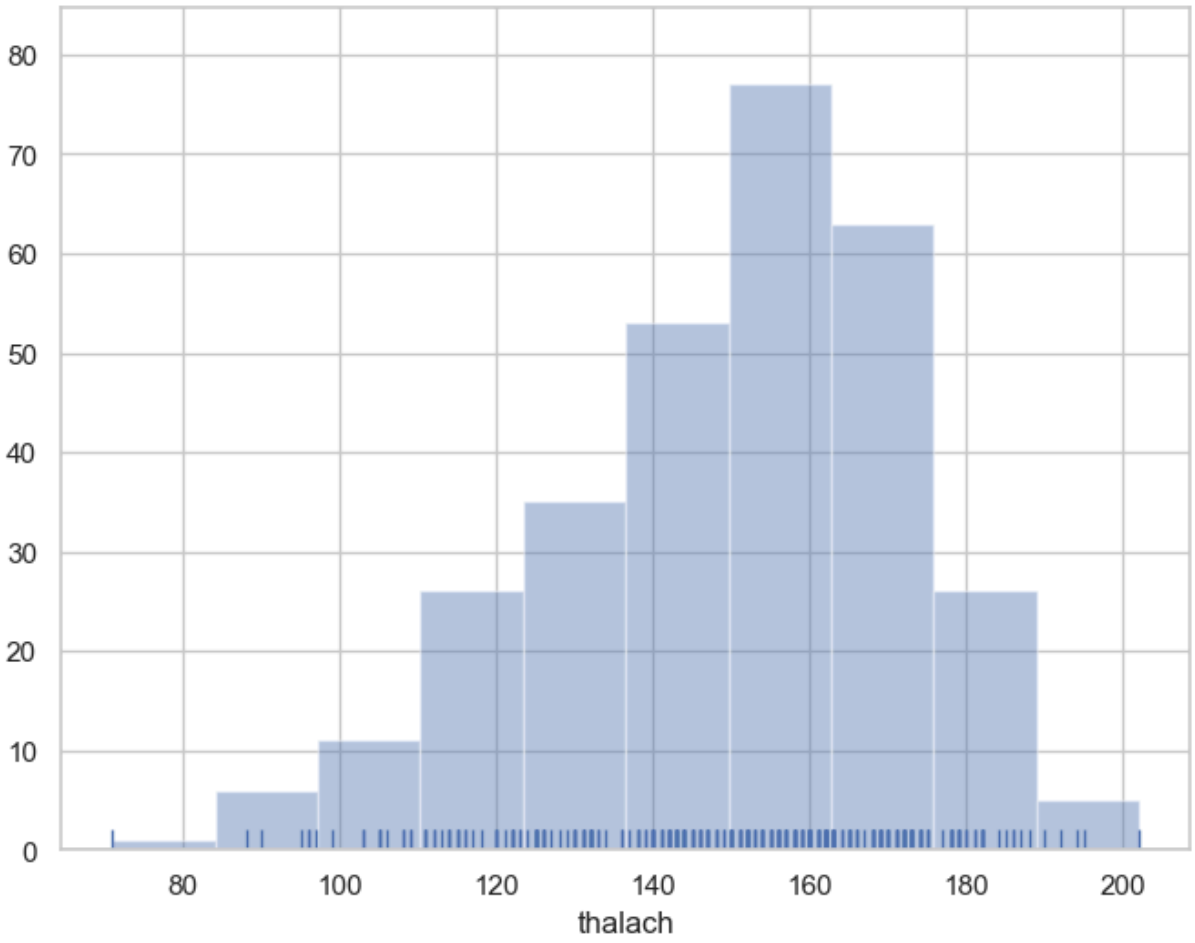
In [57]:
```
f,ax=plt.subplots(figsize=(8,6))
x=df['thalach']
x=pd.Series(x,name='thalach variable')
ax=sns.kdeplot(x,shade=True,color='r')
plt.show()
```



In [58]:
```
f,ax=plt.subplots(figsize=(8,6))
x=df['thalach']
ax=sns.distplot(x,bins=10,rug=True)
plt.show()
```

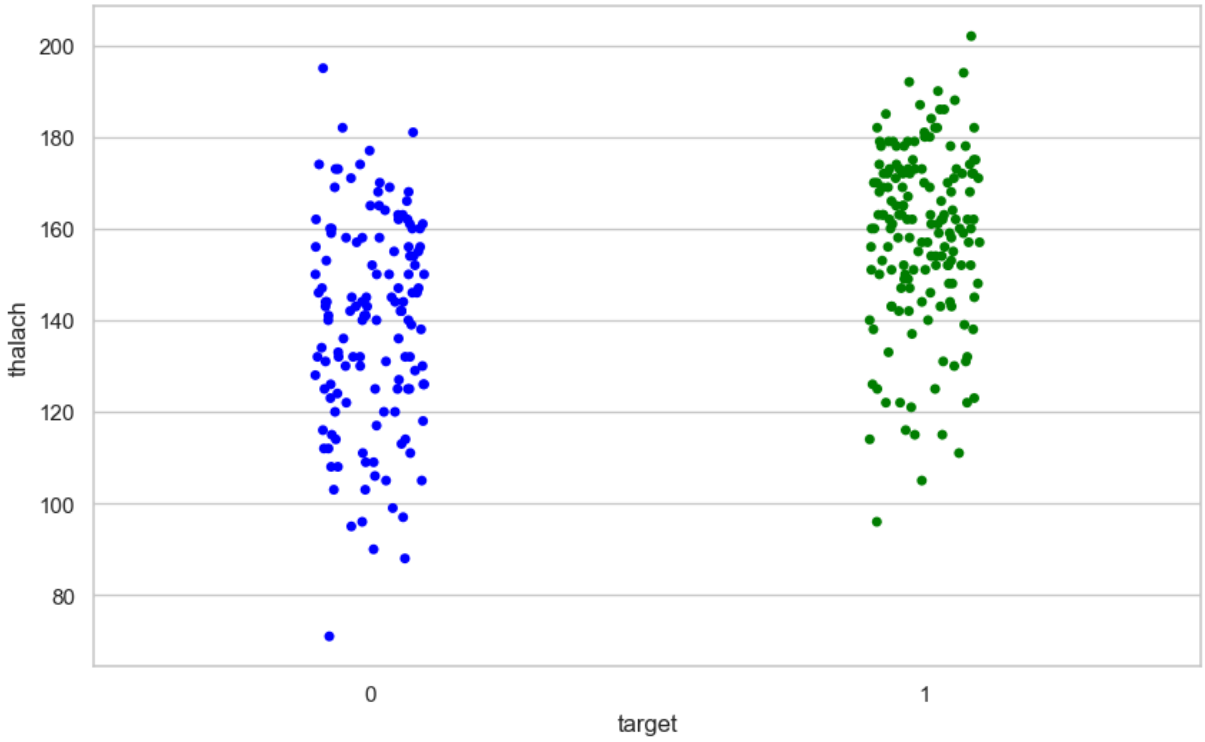


In [59]:
```
f,ax=plt.subplots(figsize=(8,6))
x=df['thalach']
ax=sns.distplot(x,bins=10,rug=True,kde=False)
plt.show()
```



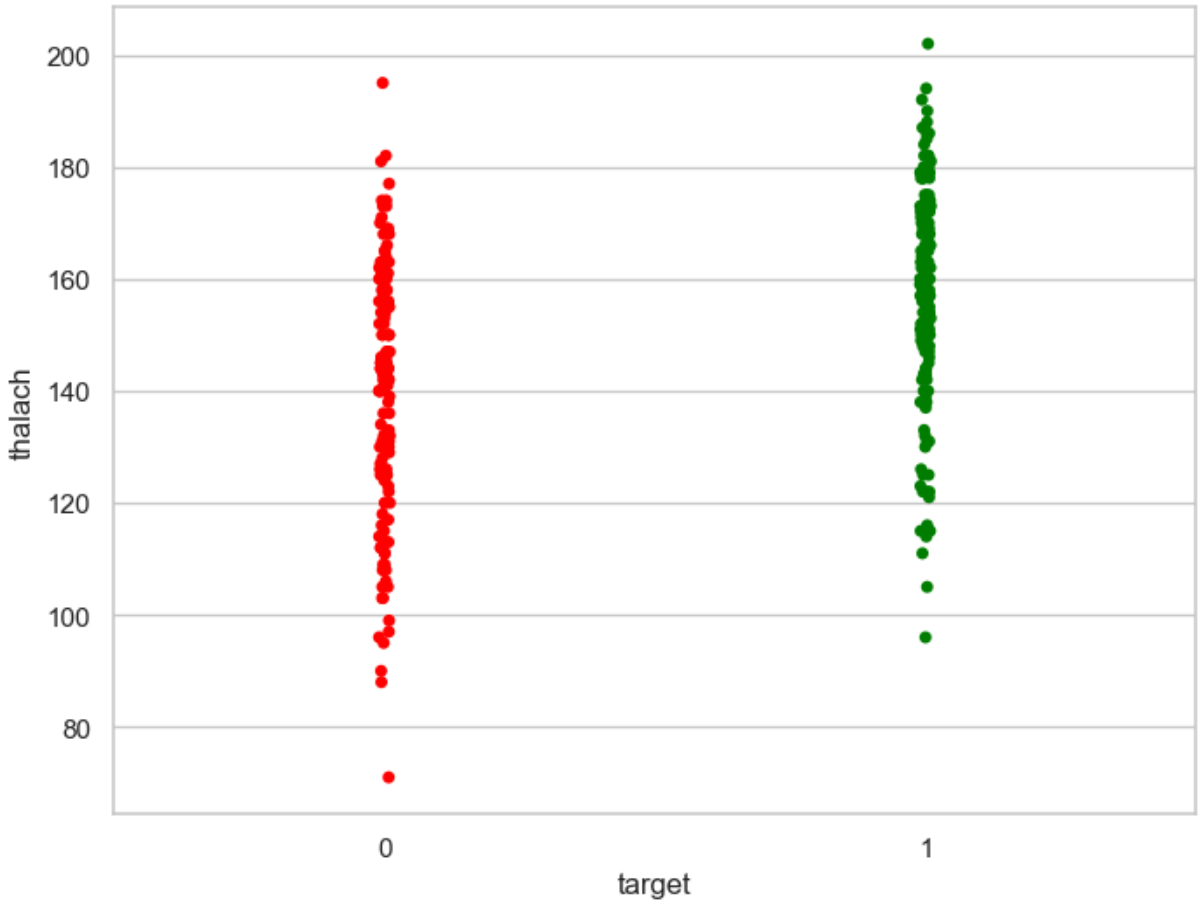
visualising frequency distribution of thalach variable with respect to target

In [66]:
```
f,ax=plt.subplots(figsize=(10,6))
sns.stripplot(x="target",y="thalach",data=df,palette=['blue','green'])    # palette
plt.show()
```

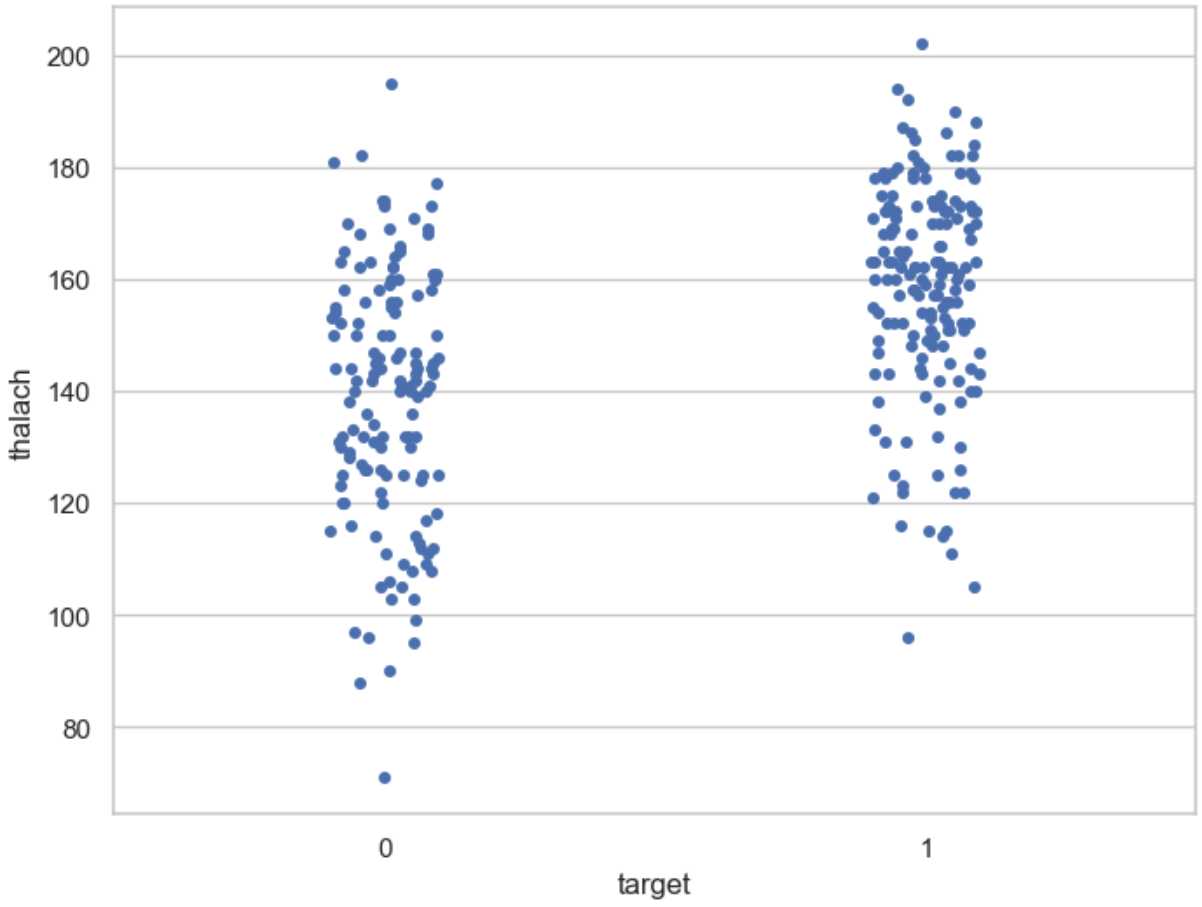


In [67]:
```
f,ax=plt.subplots(figsize=(8,6))
ax=sns.stripplot(x='thalach',y='target',jitter=0.01,data=df)
plt.show()
```

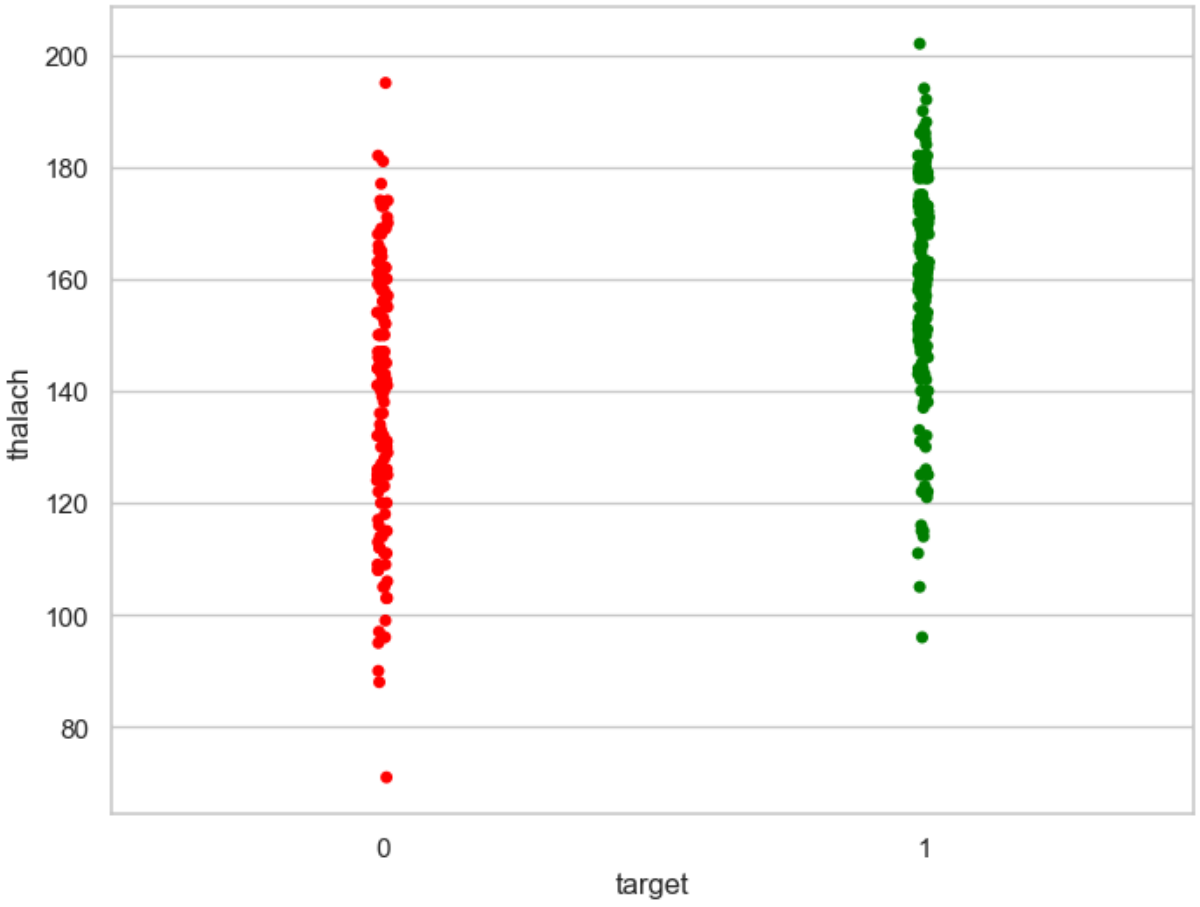
In [69]:
```
f,ax=plt.subplots(figsize=(8,6))
ax=sns.stripplot(y='thalach',x='target',jitter=0.01,data=df,palette=['red','green']
plt.show()
```



In [71]:
```
f,ax=plt.subplots(figsize=(8,6))
ax=sns.stripplot(y='thalach',x='target',jitter=True,data=df)
plt.show()
```

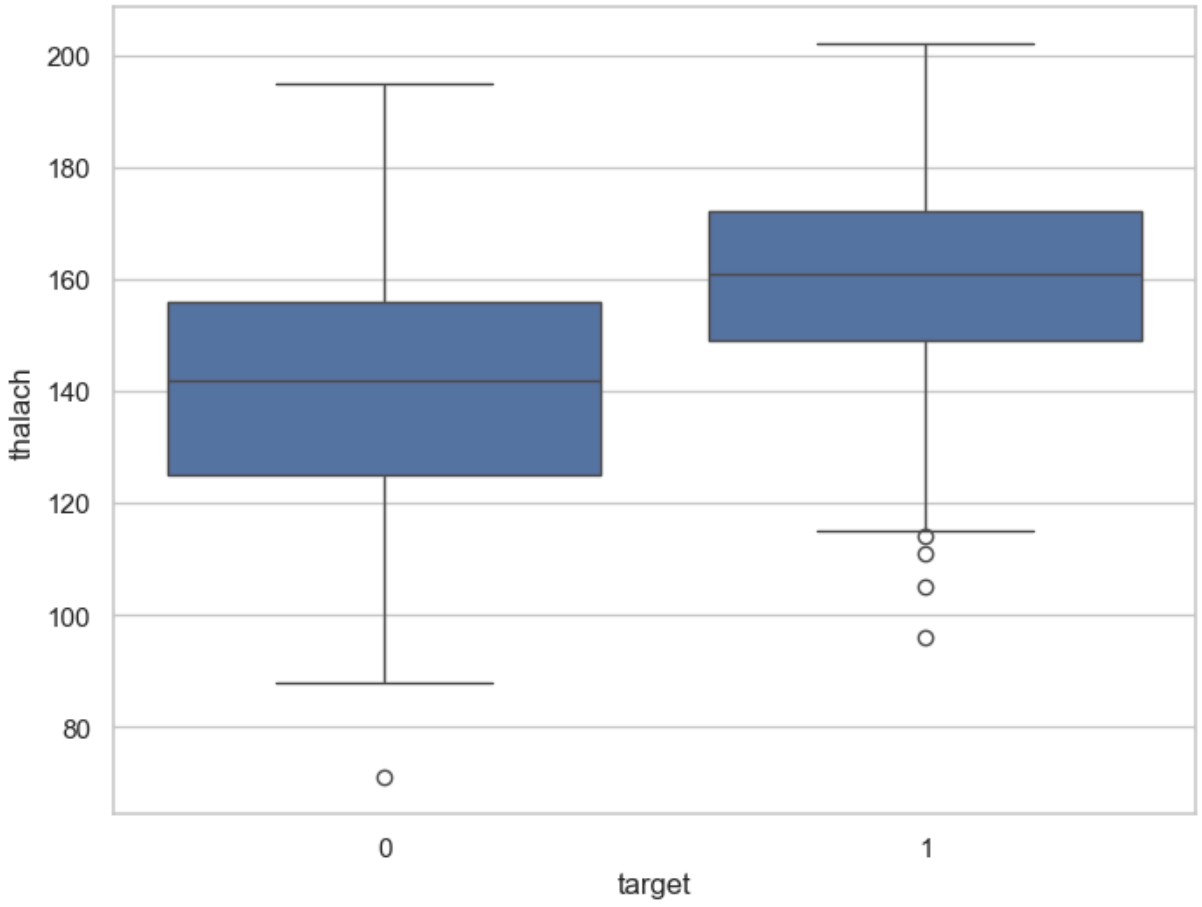


In [73]:
```
f,ax=plt.subplots(figsize=(8,6))
ax=sns.stripplot(y='thalach',x='target',jitter=0.01,data=df,palette=['red','green']
plt.show()
```

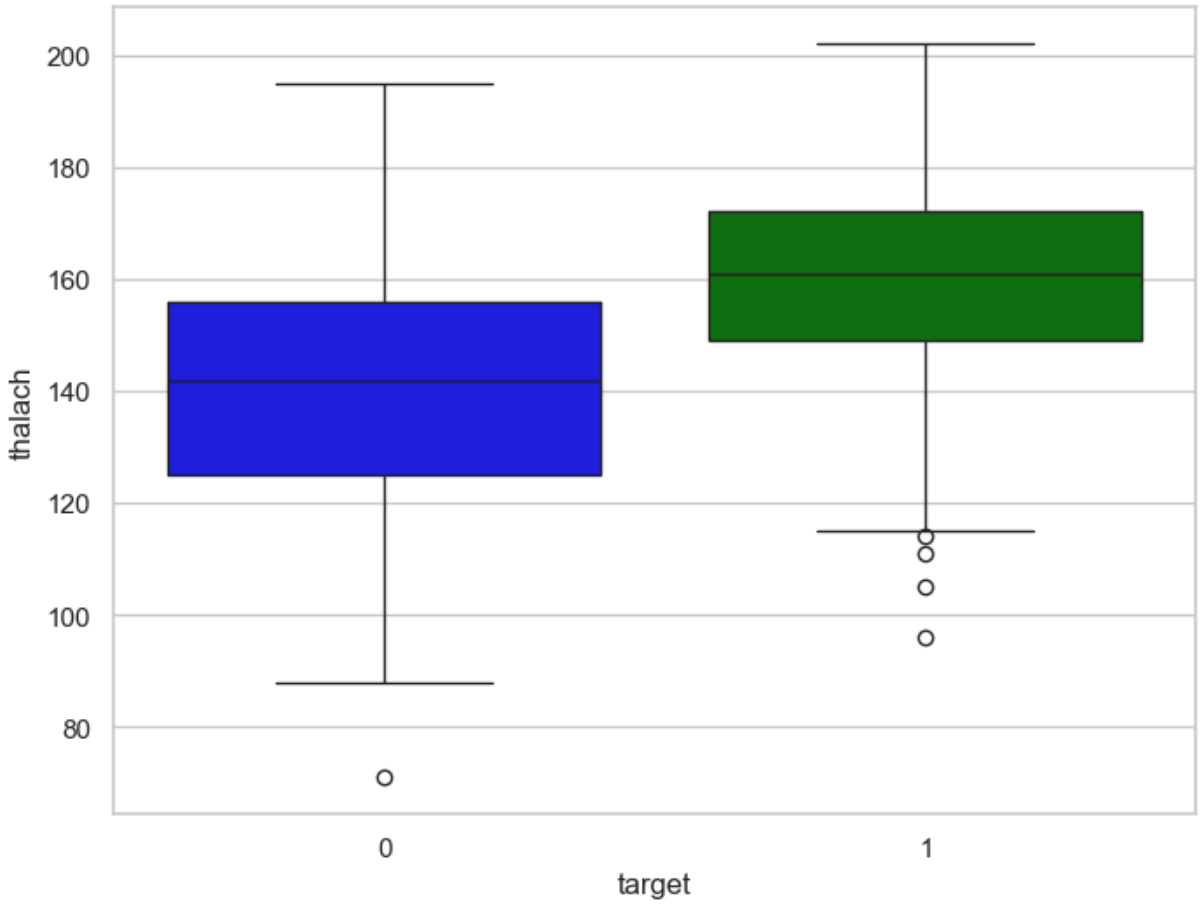


visualize the frequency distribution of thalach variable using boxplot

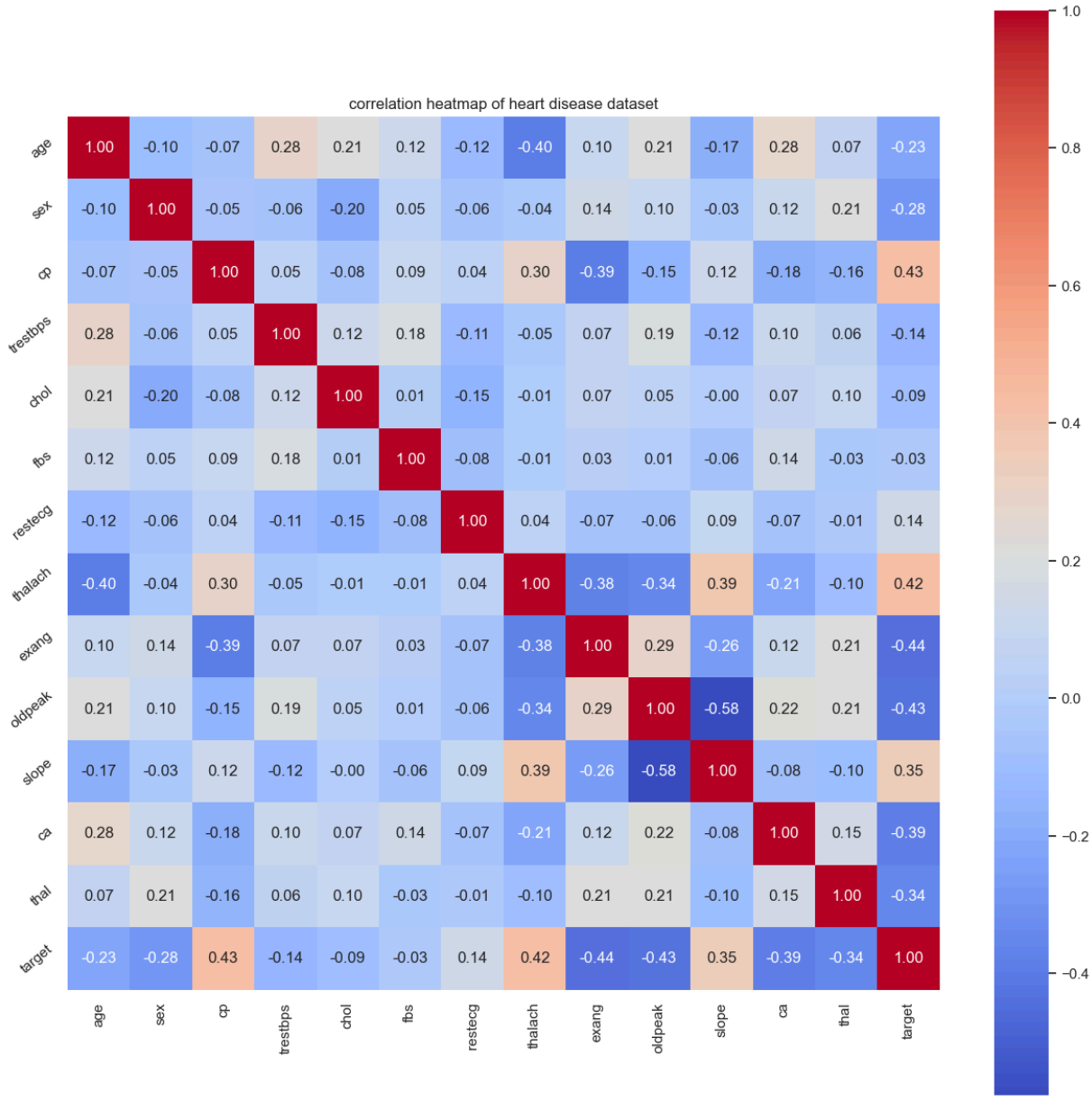
In [75]:
```
f,ax=plt.subplots(figsize=(8,6))
ax=sns.boxplot(x='target',y='thalach',data=df)
plt.show()
```



In [76]:
```
f,ax=plt.subplots(figsize=(8,6))
ax=sns.boxplot(x='target',y='thalach',data=df,palette=['blue','green'])
plt.show()          # here we infer that ppl who suffering from heart disease have
```



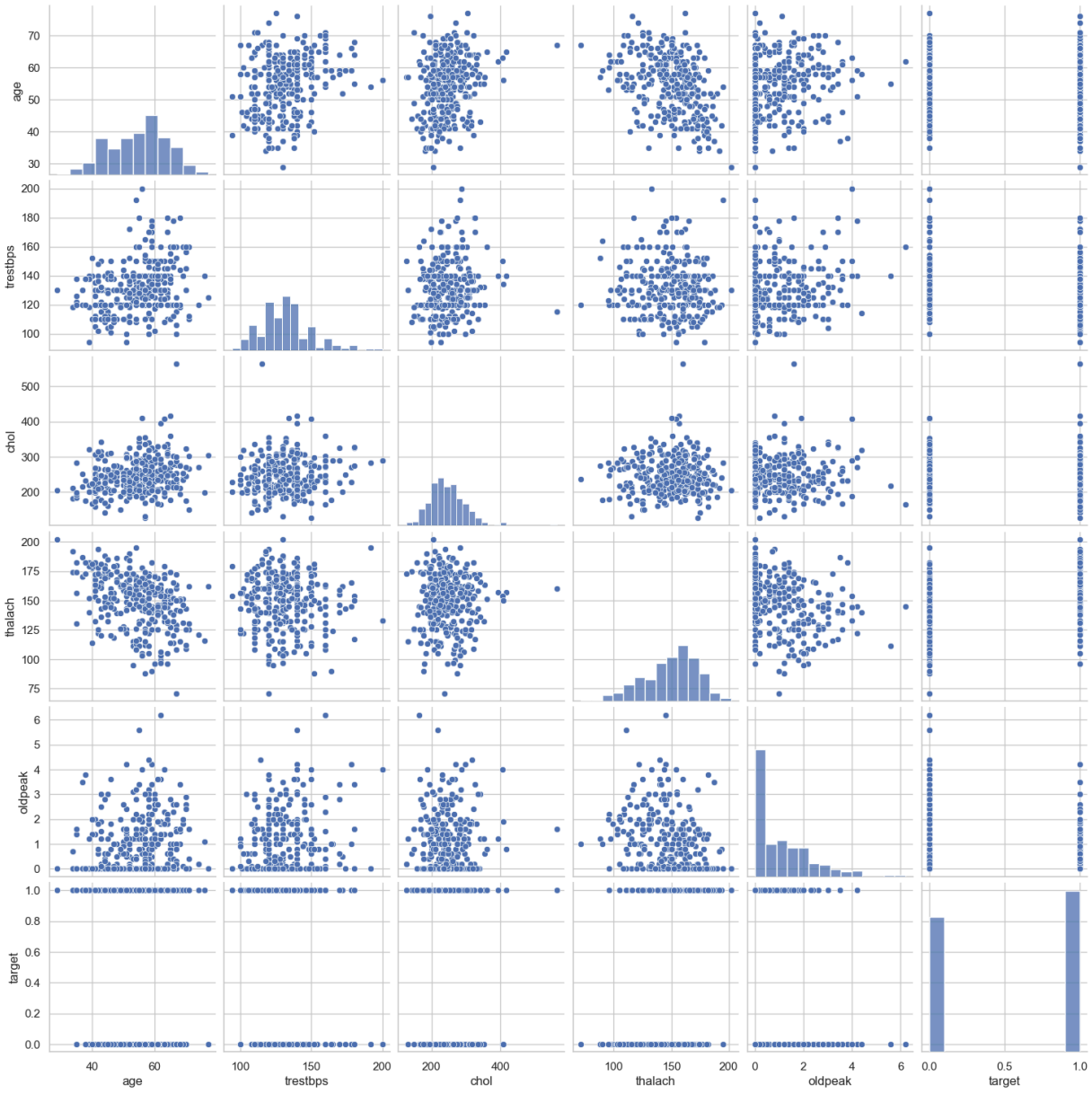
In [79]:
```
plt.figure(figsize=(15,15))
plt.title('correlation heatmap of heart disease dataset')
a=sns.heatmap(correlation,square=True,annot=True,fmt='.2f',linecolor='white',cmap='
a.set_xticklabels(a.get_xticklabels(),rotation=90)
a.set_yticklabels(a.get_yticklabels(),rotation=40)
plt.show()
```



In [80]:
```
df.columns
```

Out[80]:
```
Index(['age', 'sex', 'cp', 'trestbps', 'chol', 'fbs', 'restecg', 'thalach',
       'exang', 'oldpeak', 'slope', 'ca', 'thal', 'target'],
      dtype='object')
```

In [81]:
```
num_var=['age','trestbps','chol','thalach','oldpeak','target']
sns.pairplot(df[num_var],kind='scatter',diag_kind='hist')
plt.show()
```



Analysis of age and other variables

In [82]:
```
df['age'].nunique()
```

Out[82]: 41

In [83]:
```
df['age'].unique
```

Out[83]:
```
<bound method Series.unique of 0      63
1      37
2      41
3      56
4      57
       ..
298    57
299    45
300    68
301    57
302    57
Name: age, Length: 303, dtype: int64>
```

In [87]:
```
df['age'].value_counts
```

Out[87]:
```
<bound method IndexOpsMixin.value_counts of 0      63
1      37
2      41
3      56
4      57
       ..
298    57
299    45
300    68
301    57
302    57
Name: age, Length: 303, dtype: int64>
```

statistical summary of age variable

In [89]:
```
df['age'].describe()
```

Out[89]:
```
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 [90]:
```
df['age'].describe().T
```

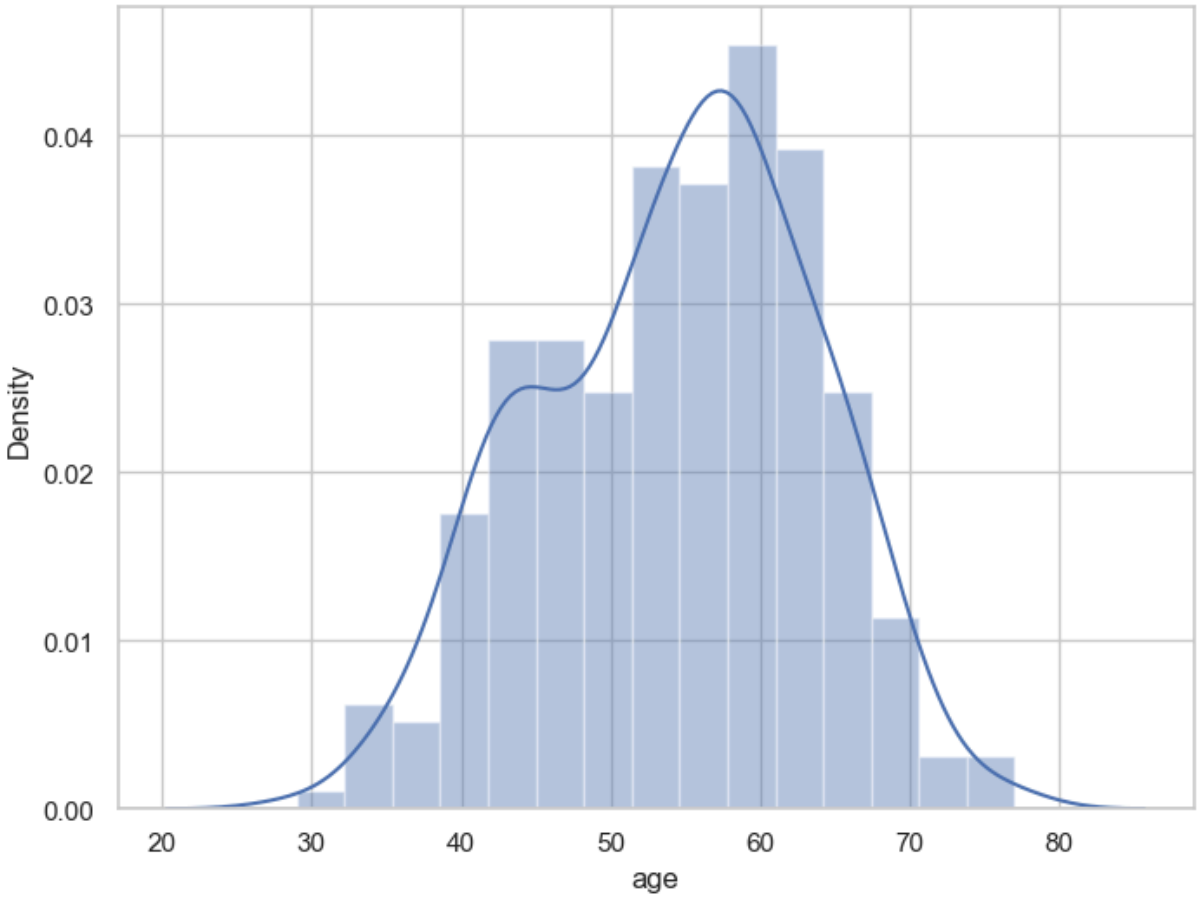
Out[90]:
```
count    303.000000
mean      54.366337
std        9.082101
min       29.000000
25%       47.500000
50%       55.000000
75%       61.000000
max       77.000000
Name: age, dtype: float64
```

In [91]:
```
df['age'].info()
```

```
<class 'pandas.core.series.Series'>
RangeIndex: 303 entries, 0 to 302
Series name: age
Non-Null Count  Dtype
--------------  -----
303 non-null    int64
dtypes: int64(1)
memory usage: 2.5 KB
```

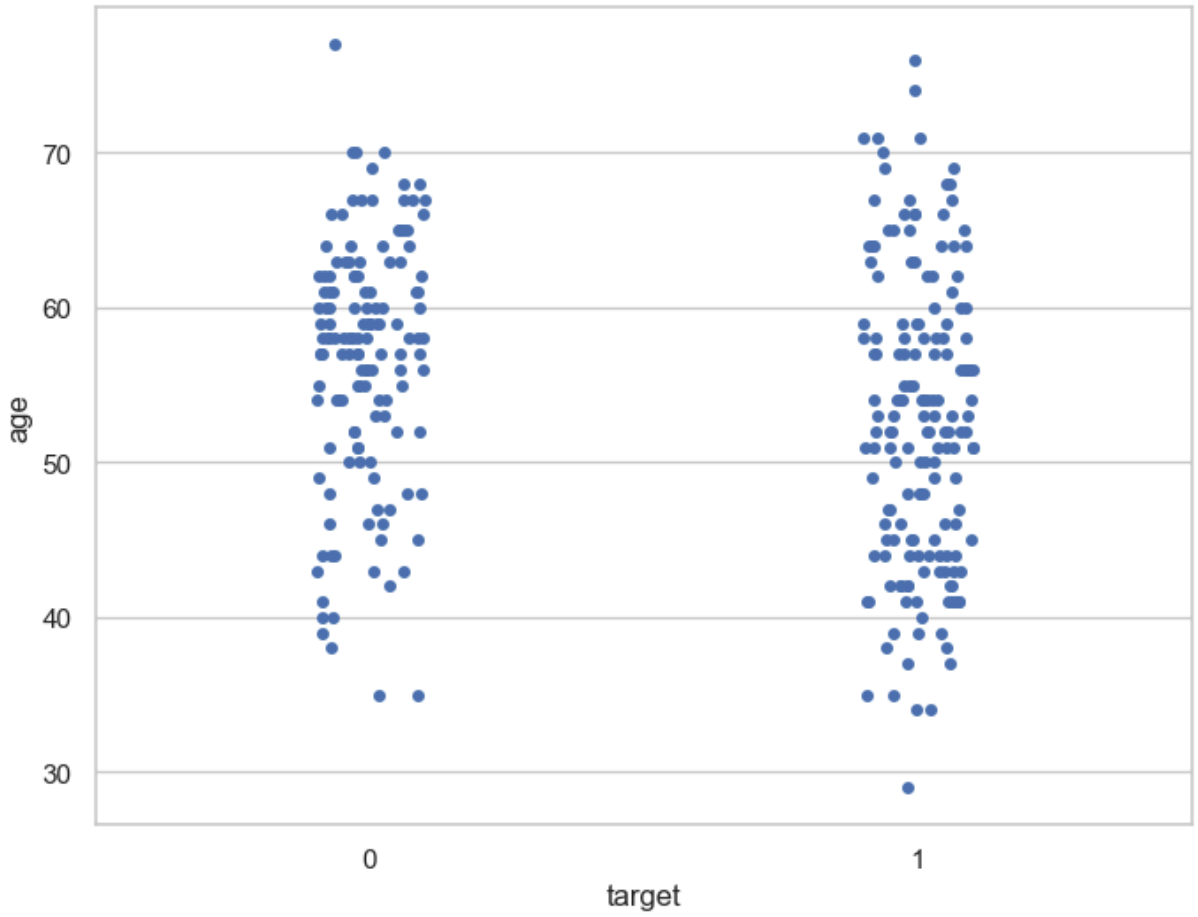
plotting the distribution of age variable

In [92]:
```
f,ax=plt.subplots(figsize=(8,6))
x=df['age']
ax=sns.distplot(x,bins=15)
plt.show()
```

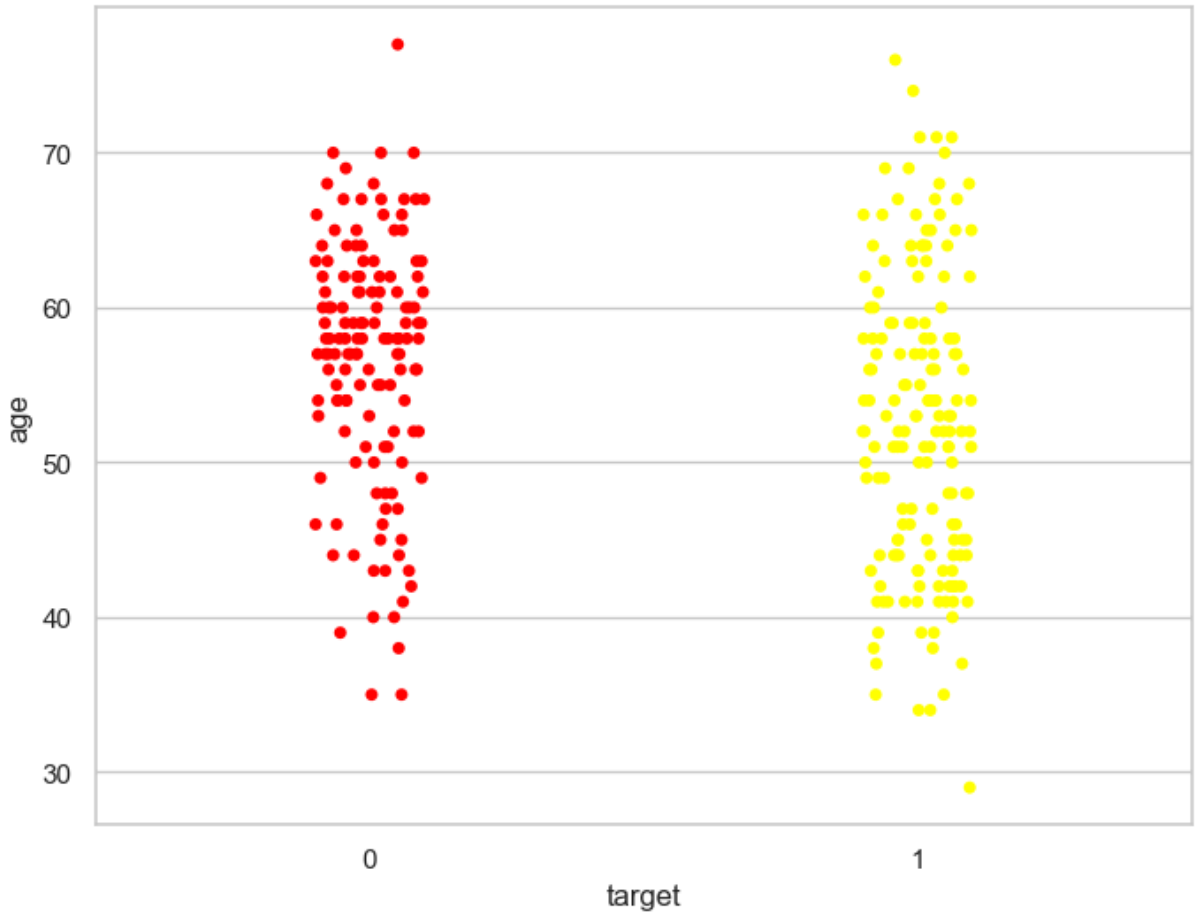


analyse the age and target variable

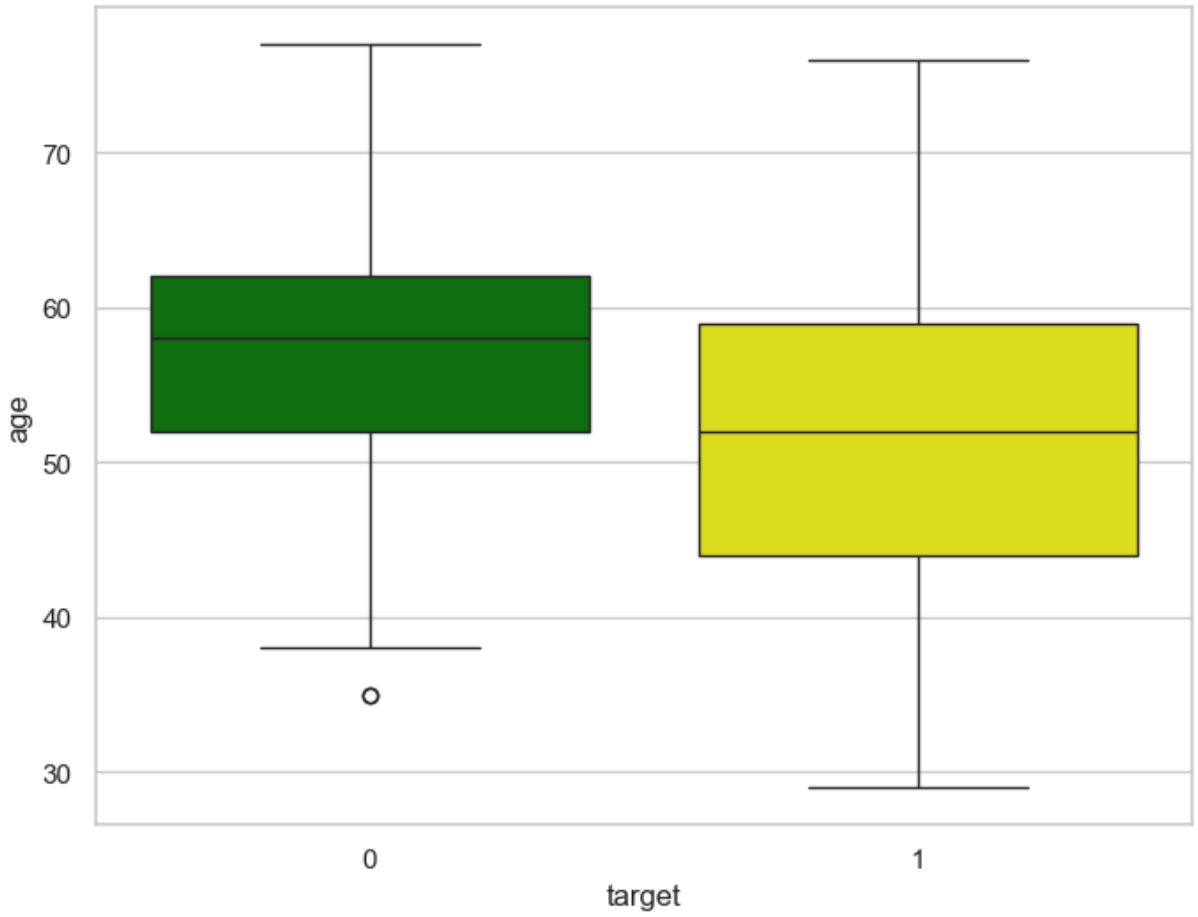
In [93]:
```
f,ax=plt.subplots(figsize=(8,6))
ax=sns.stripplot(x='target',y='age',data=df)
plt.show()
```

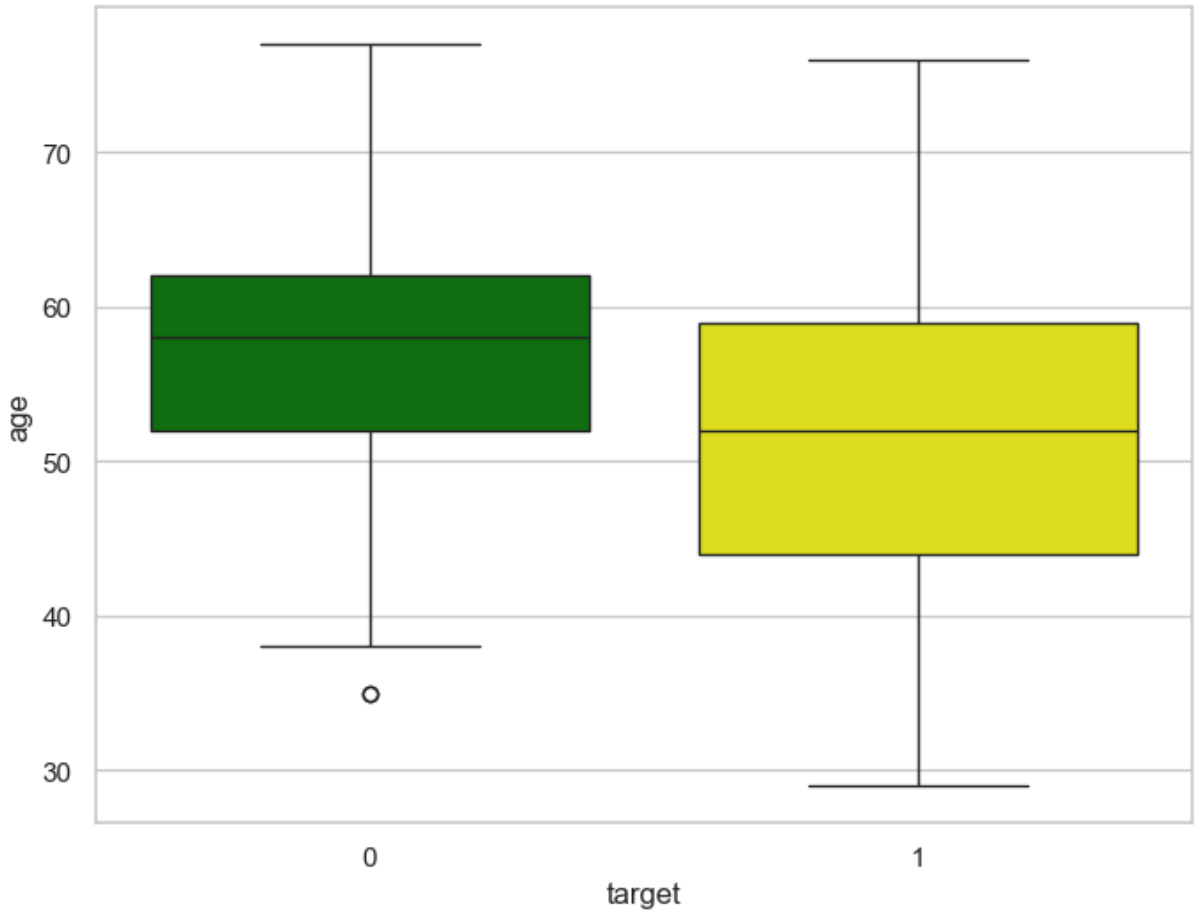


In [94]:
```
f,ax=plt.subplots(figsize=(8,6))
ax=sns.stripplot(x='target',y='age',data=df,palette=['red','yellow'])     # using s
plt.show()
```

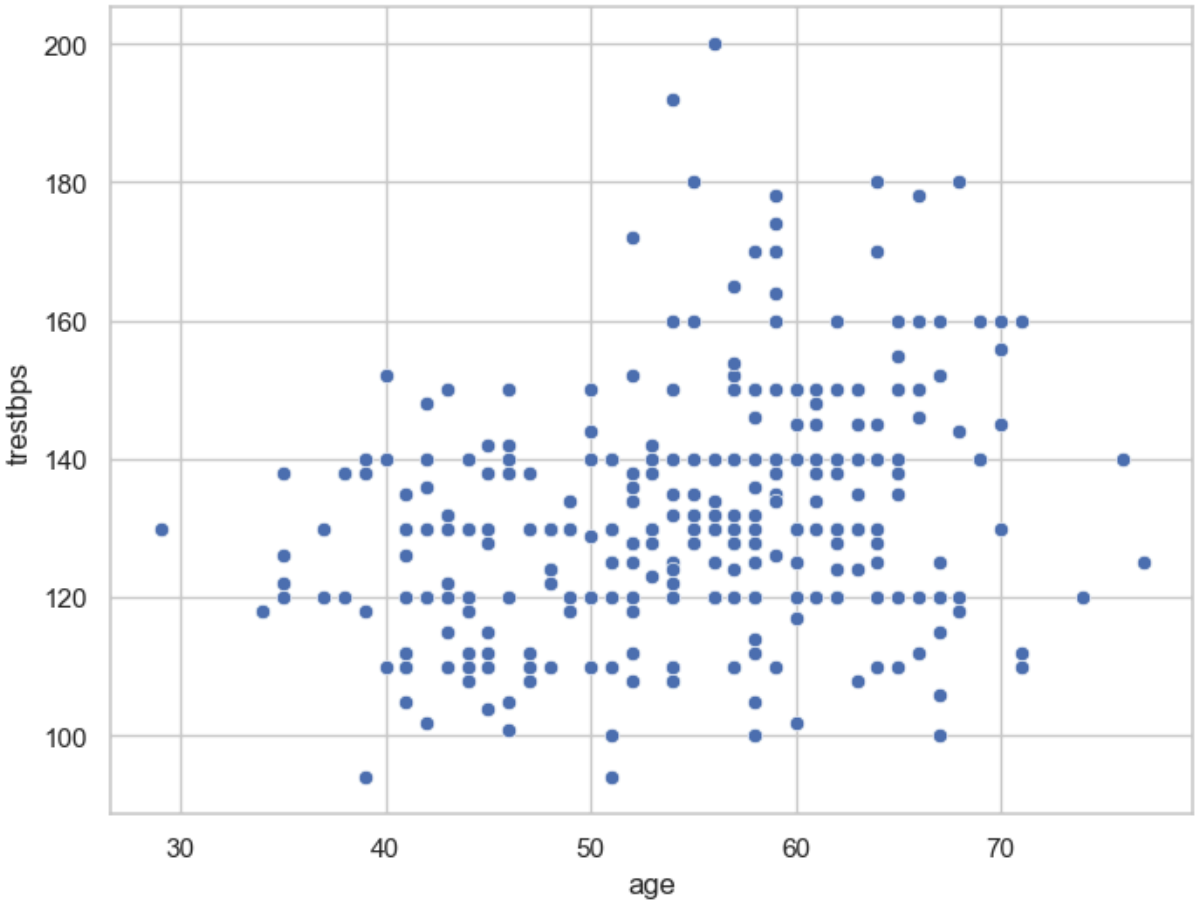


In [96]:
```
f,ax=plt.subplots(figsize=(8,6))
ax=sns.boxplot(x='target',y='age',data=df,palette=['green','yellow'])
plt.show()
```

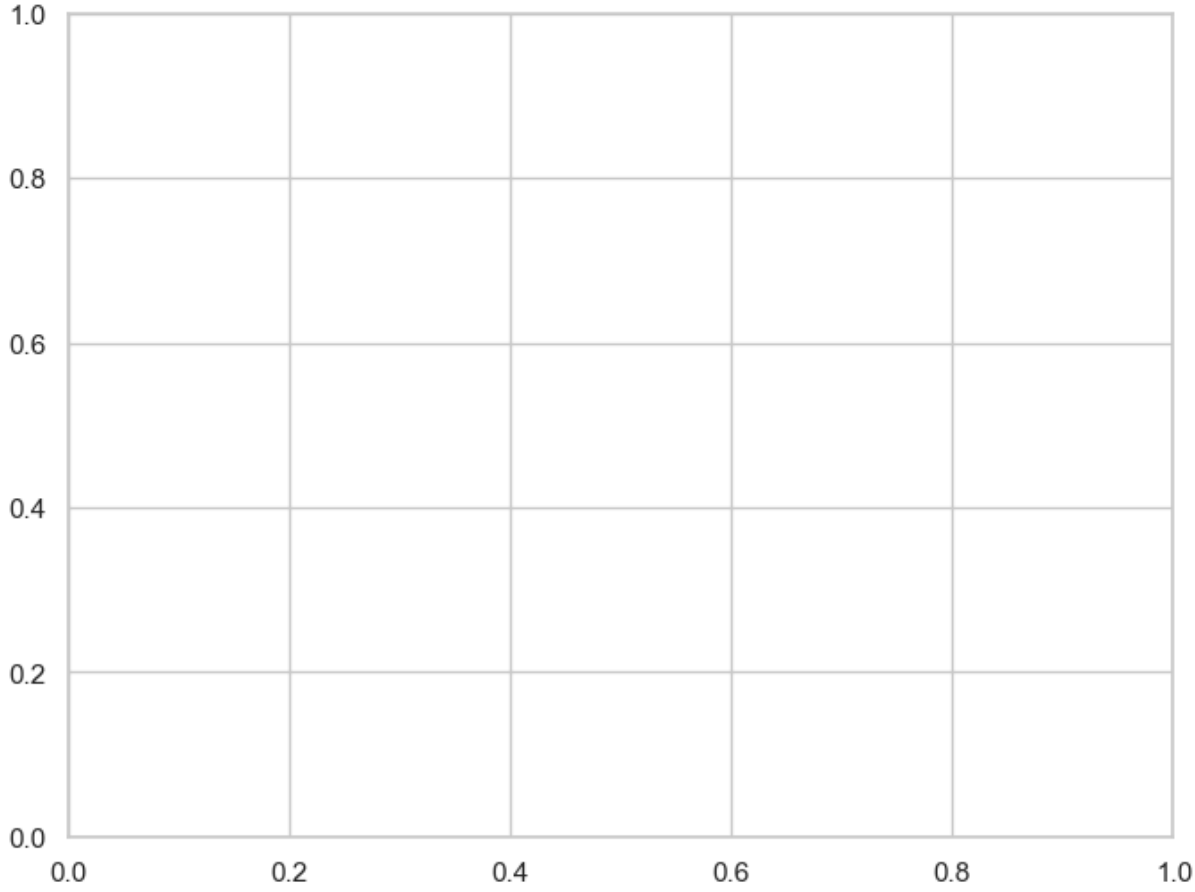

70
60
50
40
30
age
0
1
target

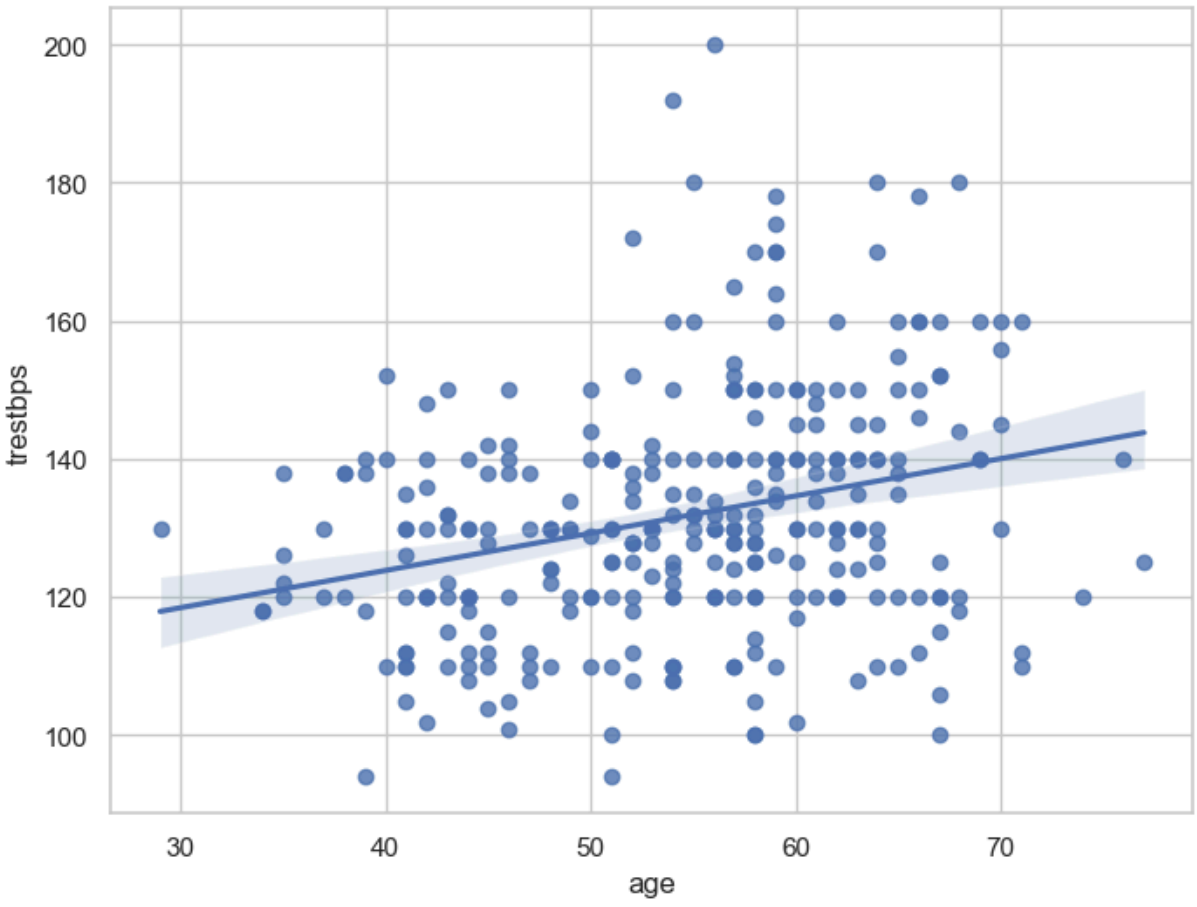

70
60
50
40
30
age
0
1
target

In [99]:
```
f,ax=plt.subplots(figsize=(8,6))
ax=sns.scatterplot(x='age',y='trestbps',data=df,palette=['red','green'])     # this
plt.show()
```



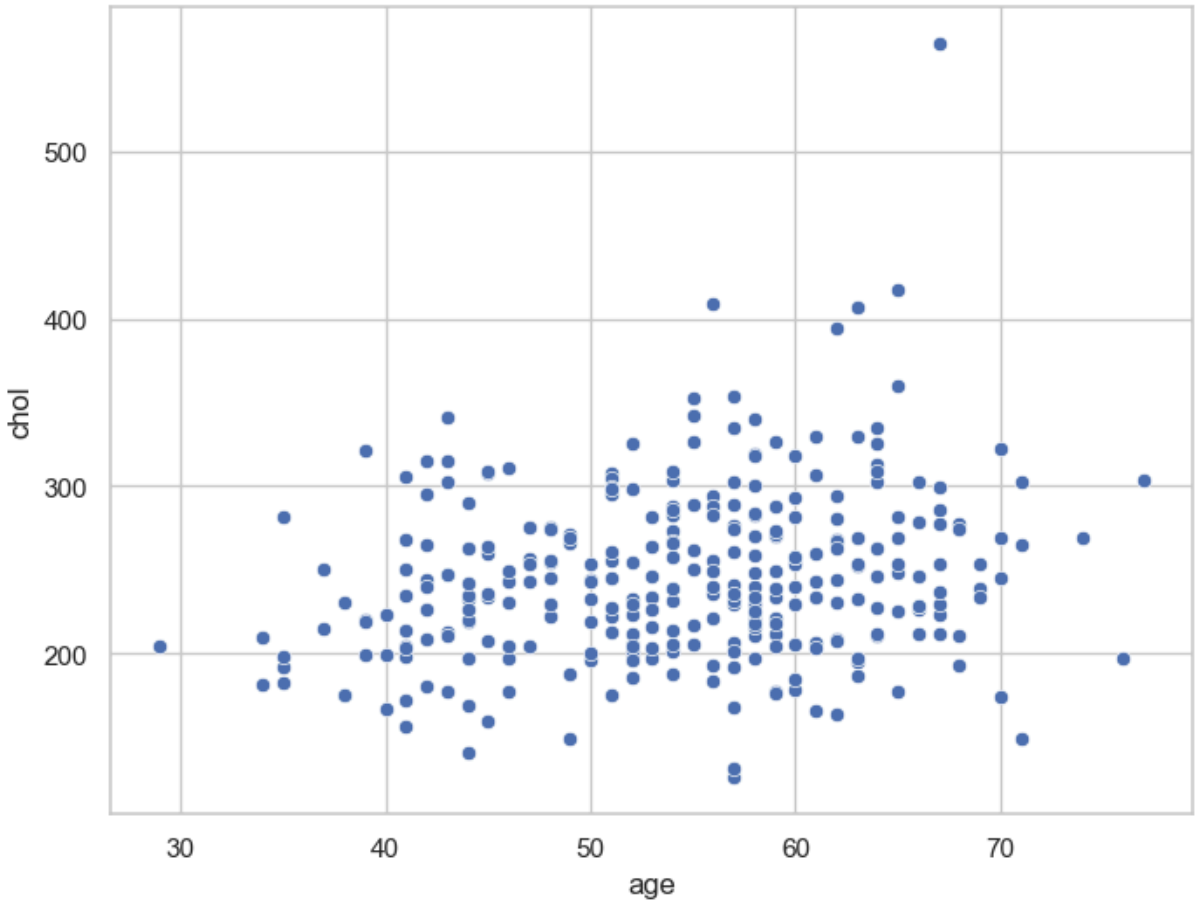
In [101…
```
f,ax=plt.subplots(figsize=(8,6))
ax=sns.regplot(x='age',y='trestbps',data=df)
plt.show()
```


1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0


200
180
160
140
120
100
trestbps
30
40
50
60
70
age

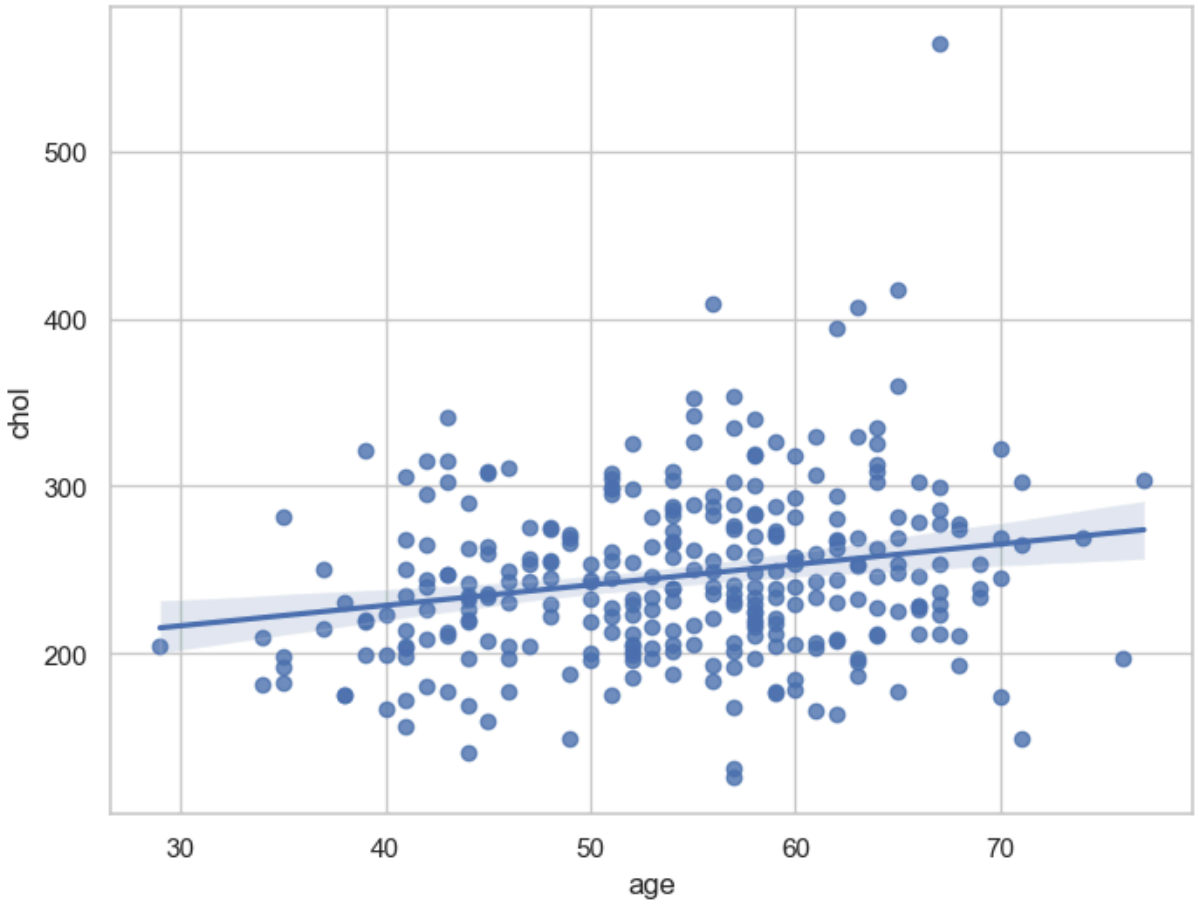
In [102…

```
f,ax=plt.subplots(figsize=(8,6))
ax=sns.scatterplot(x='age',y='chol',data=df)
plt.show()
```



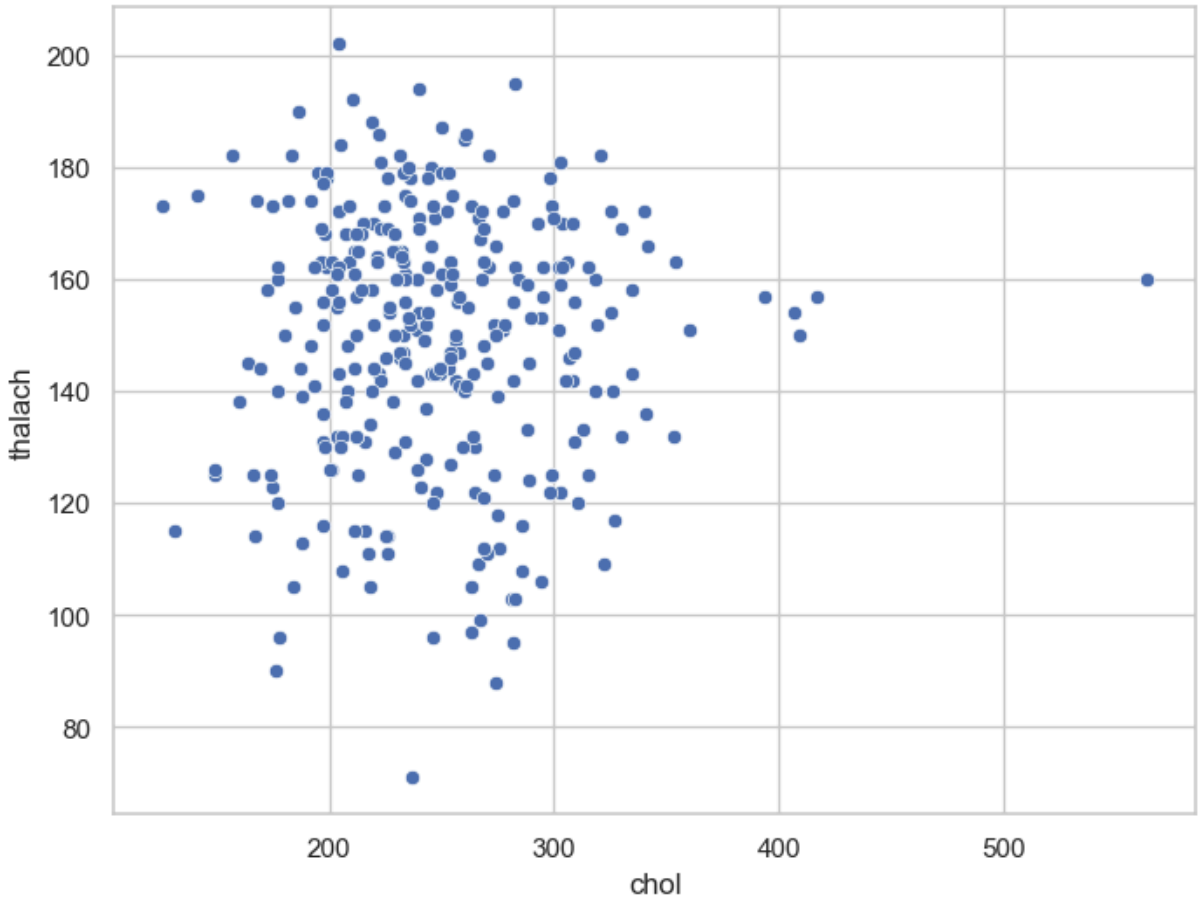
In [103…

```
f,ax=plt.subplots(figsize=(8,6))
ax=sns.regplot(x='age',y='chol',data=df)
plt.show()
                                                    # slightly positive correlation bet
```



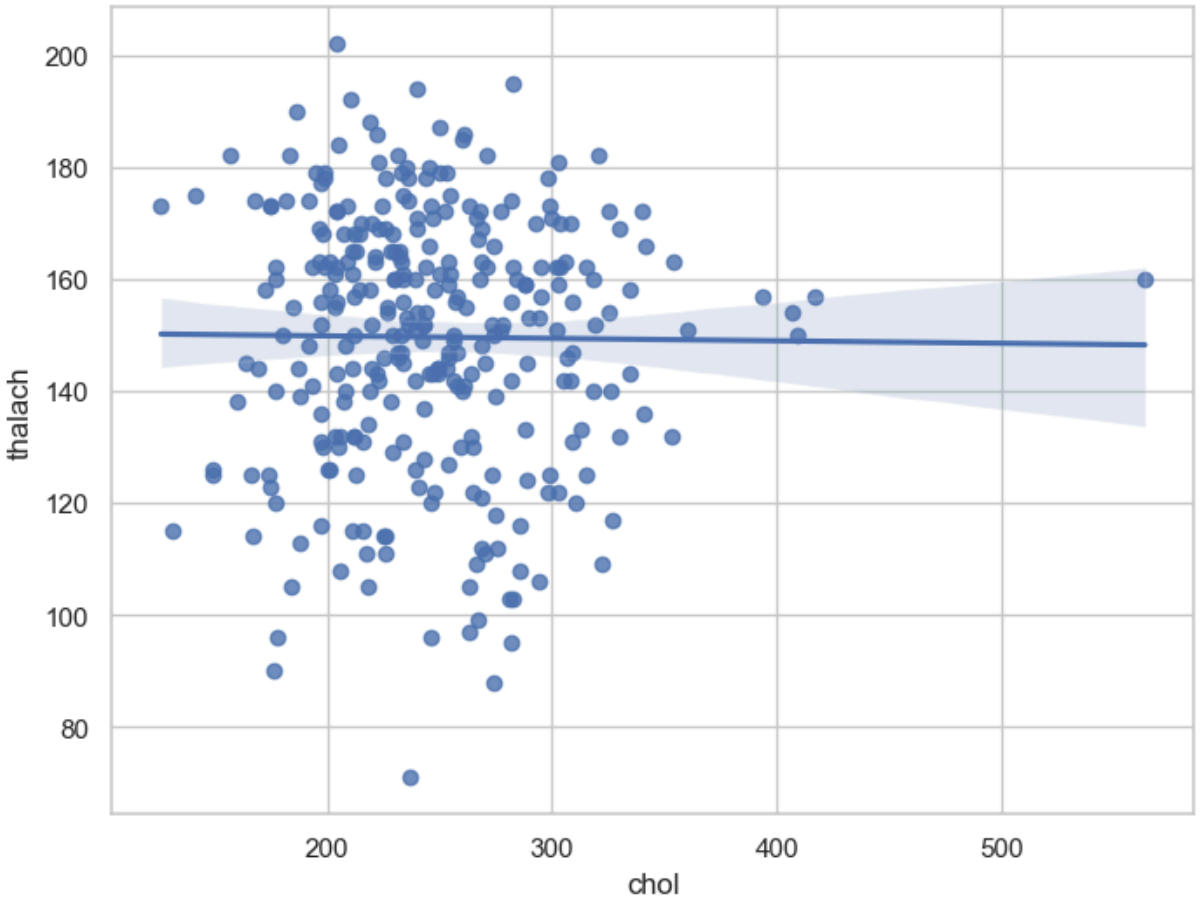
In [104…

```
f,ax=plt.subplots(figsize=(8,6))
ax=sns.scatterplot(x='chol',y='thalach',data=df)
plt.show()
```



In [106…

```
f,ax=plt.subplots(figsize=(8,6))
ax=sns.regplot(x='chol',y='thalach',data=df)
plt.show()
```



In [107…
```
df.isnull().sum()
```

Out[107…
```
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 [111…
```
assert pd.notnull(df).any().any()    # no missing values in the data frame
```

In [112…
```
assert pd.notnull(df).all().all()      # since no error is thrown it shows that the
```

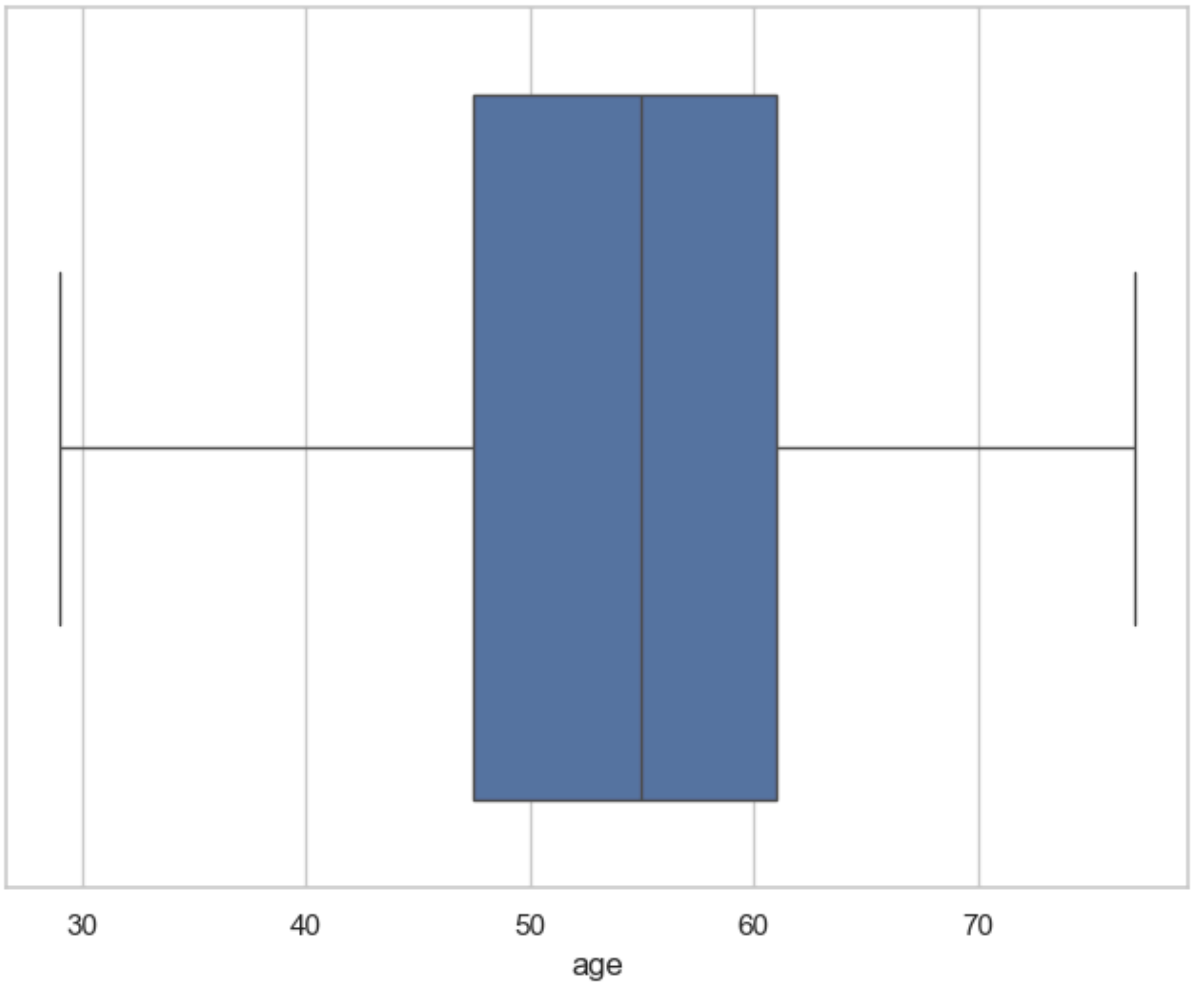
In [113…
```
assert ( df>=0).all().all()    # it asserts that all values are greater than and eq
```

Outlier detection

In [114…
```
df['age'].describe()
```

Out[114…
```
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
```

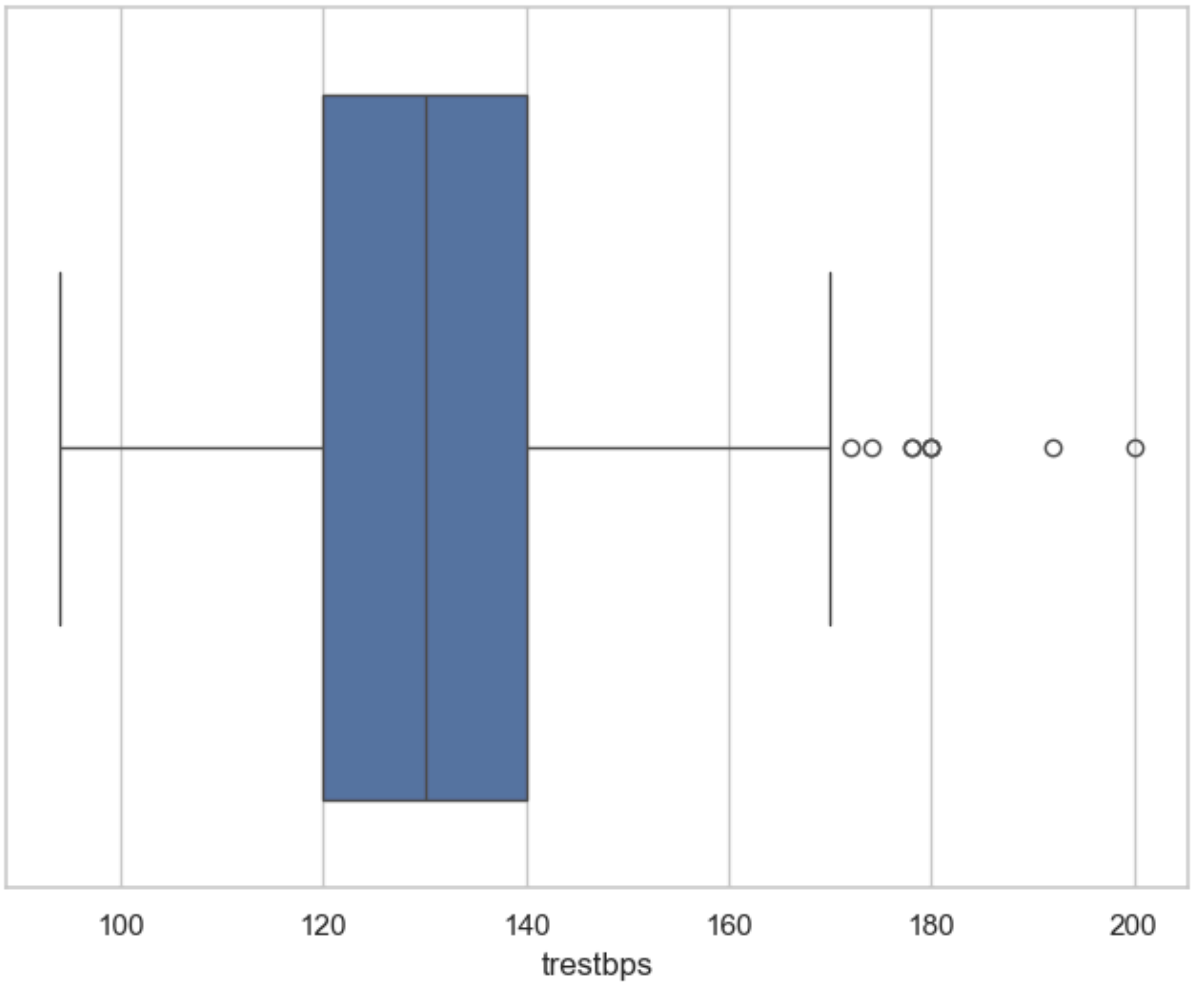
In [115…
```
f,ax=plt.subplots(figsize=(8,6))
sns.boxplot(x=df['age'])
plt.show()
```



In [116…
```
df['trestbps'].describe()
```

Out[116…
```
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
```

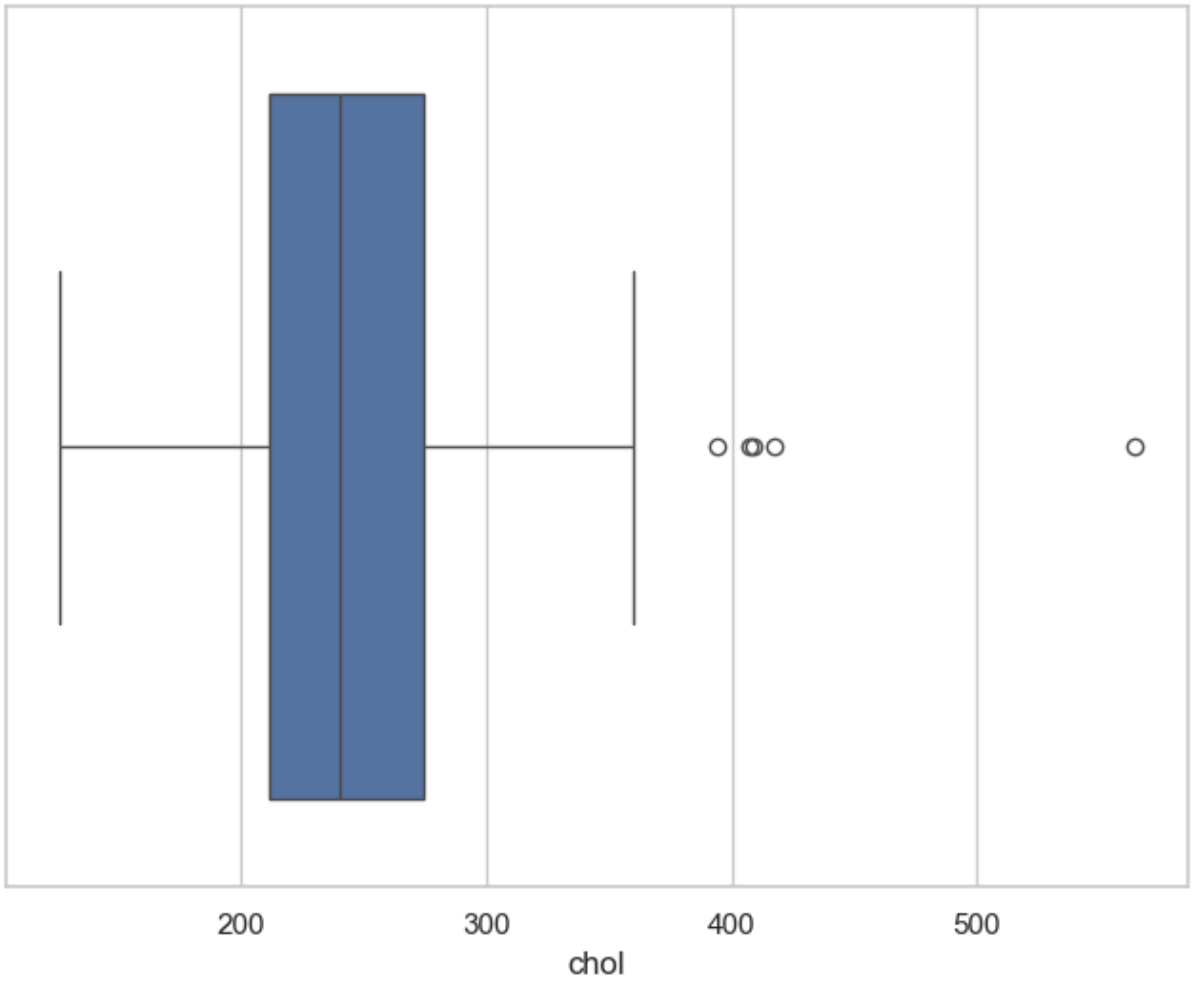
In [117…
```
f,ax=plt.subplots(figsize=(8,6))
sns.boxplot(x=df['trestbps'])
plt.show()
```



In [118…
```
df['chol'].describe()
```

Out[118…
```
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
```

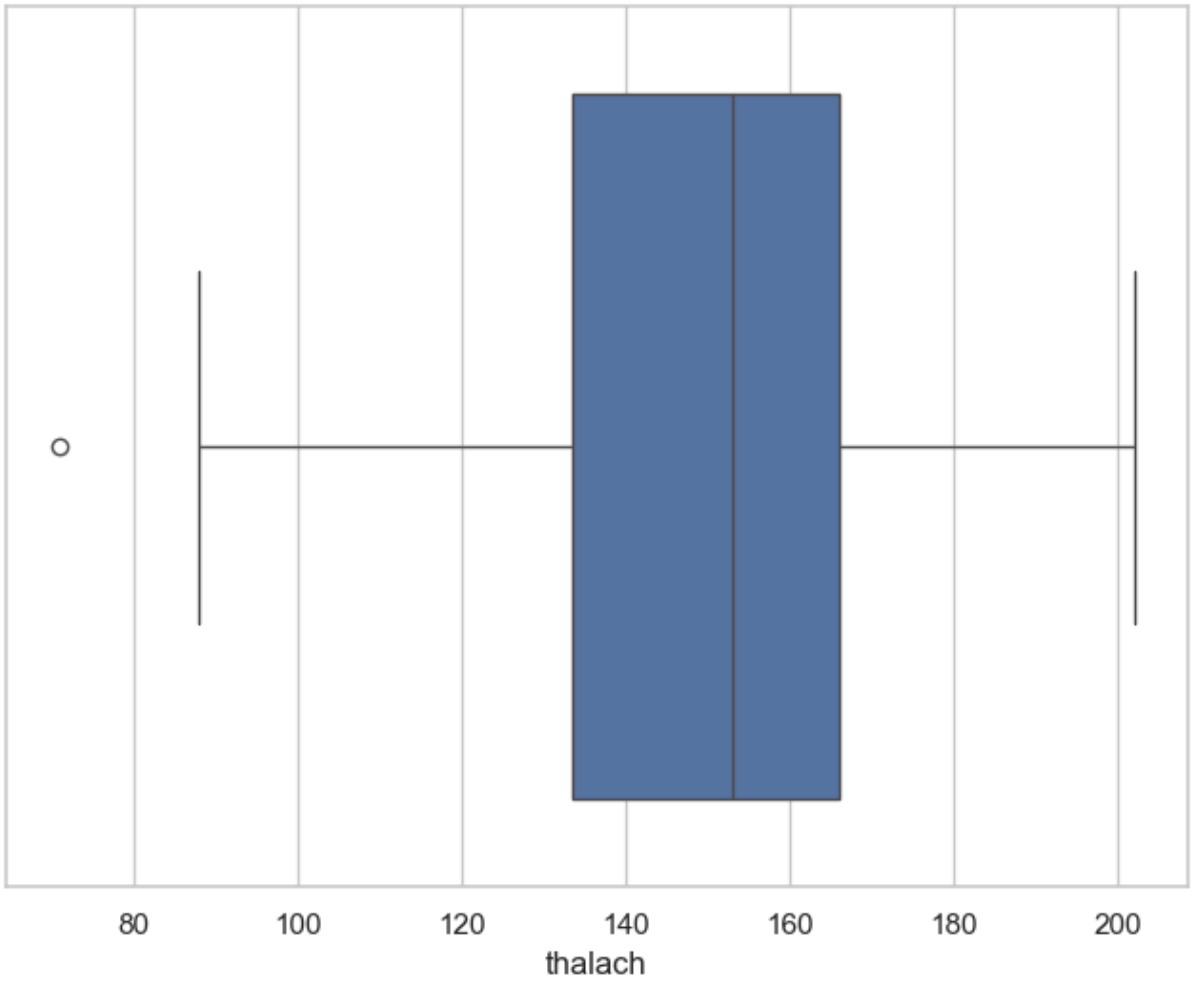
In [119…
```
f,ax=plt.subplots(figsize=(8,6))
sns.boxplot(x=df['chol'])
plt.show()
```



In [120…
```
df['thalach'].describe()
```

Out[120…
```
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
```

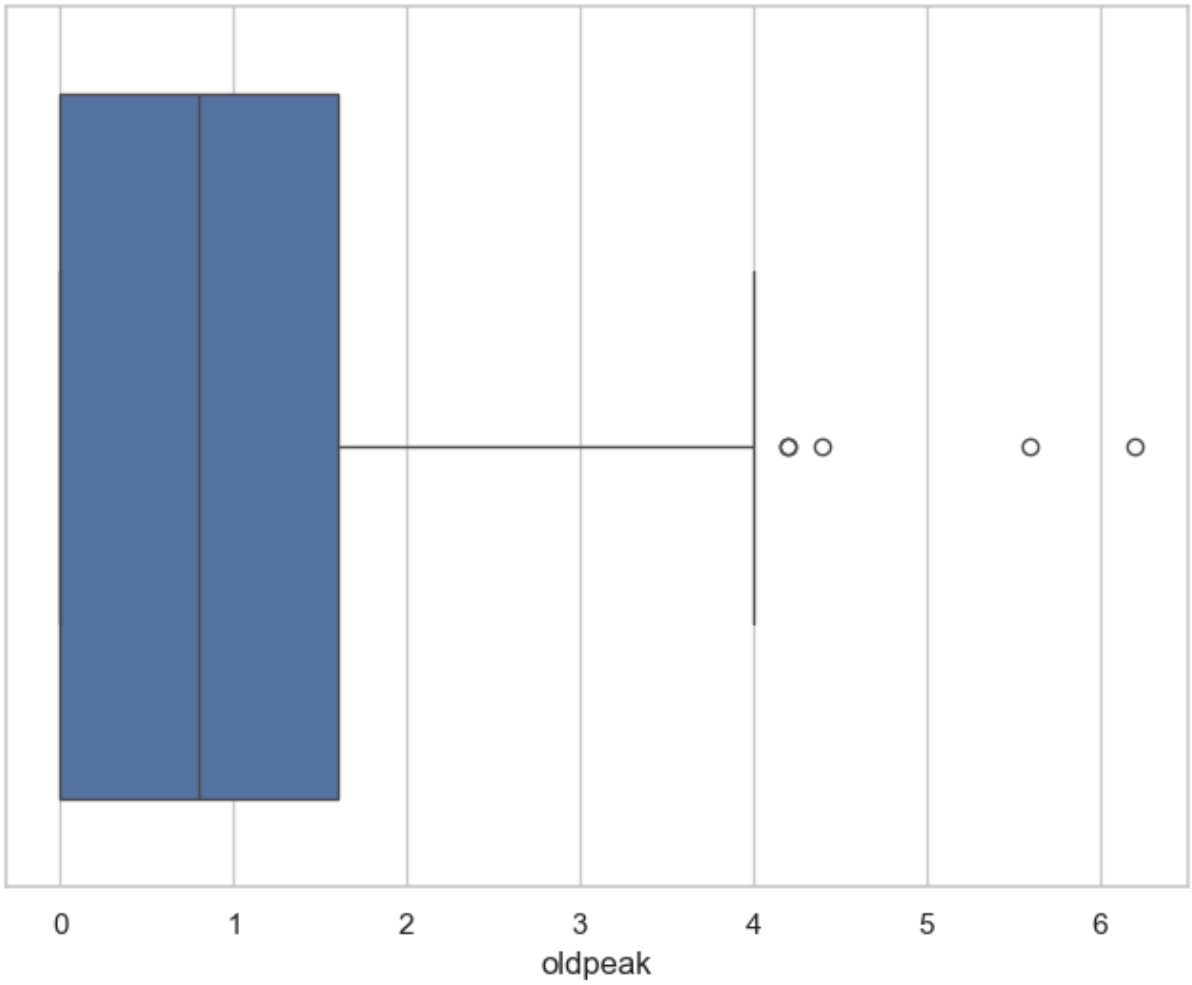
In [121…
```
f,ax=plt.subplots(figsize=(8,6))
sns.boxplot(x=df['thalach'])
plt.show()
```



In [122…
```
df['oldpeak'].describe()
```

Out[122…
```
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 [123…
```
f,ax=plt.subplots(figsize=(8,6))
sns.boxplot(x=df['oldpeak'])
plt.show()
```



In [ ]: