health

November 27, 2025

[21]:
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
import scipy.stats as st
```

[22]:
```
health = pd.read_csv(r'/Users/mahidharreddy/Downloads/Data science/Nov/26-27-␣
 ↪Nov/25th, 26th- Advanced EDA project/EDA- HEALTHCARE DOMAIN/heart.csv')
```

[23]:
```
health
```

[23]:
```
     age  sex  cp  trestbps  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]
```

[24]:

```
health.isnull().sum()
```

[24]:

```
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
```

[25]:

```
health.head()
```

[25]:

```
   age  sex  cp  trestbps  chol  fbs  restecg  thalach  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
```

[26]:

```
health.tail()
```

[26]:

```
     age  sex  cp  trestbps  chol  fbs  restecg  thalach  exang  oldpeak  \
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
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
```

[27]:
```
health.columns
```

[27]:
```
Index(['age', 'sex', 'cp', 'trestbps', 'chol', 'fbs', 'restecg', 'thalach',
       'exang', 'oldpeak', 'slope', 'ca', 'thal', 'target'],
      dtype='object')
```

[28]:
```
health.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
```

[29]:
```
health.describe()
```

[29]:
```
              age         sex          cp    trestbps        chol         fbs  \
count  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   
std      9.082101    0.466011    1.032052   17.538143   51.830751    0.356198   
min     29.000000    0.000000    0.000000   94.000000  126.000000    0.000000   
25%     47.500000    0.000000    0.000000  120.000000  211.000000    0.000000   
50%     55.000000    1.000000    1.000000  130.000000  240.000000    0.000000   
75%     61.000000    1.000000    2.000000  140.000000  274.500000    0.000000   
max     77.000000    1.000000    3.000000  200.000000  564.000000    1.000000   

          restecg     thalach       exang     oldpeak       slope          ca  \
count  303.000000  303.000000  303.000000  303.000000  303.000000  303.000000   
mean     0.528053  149.646865    0.326733    1.039604    1.399340    0.729373   
```

```
std      0.525860   22.905161    0.469794    1.161075    0.616226    1.022606
min      0.000000   71.000000    0.000000    0.000000    0.000000    0.000000
25%      0.000000  133.500000    0.000000    0.000000    1.000000    0.000000
50%      1.000000  153.000000    0.000000    0.800000    1.000000    0.000000
75%      1.000000  166.000000    1.000000    1.600000    2.000000    1.000000
max      2.000000  202.000000    1.000000    6.200000    2.000000    4.000000

             thal      target
count  303.000000  303.000000
mean     2.313531    0.544554
std      0.612277    0.498835
min      0.000000    0.000000
25%      2.000000    0.000000
50%      2.000000    1.000000
75%      3.000000    1.000000
max      3.000000    1.000000
```

[30]:

```
health['target'].nunique()
```

[30]:

```
2
```

[31]:

```
health['target'].unique()
```

[31]:

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

[32]:

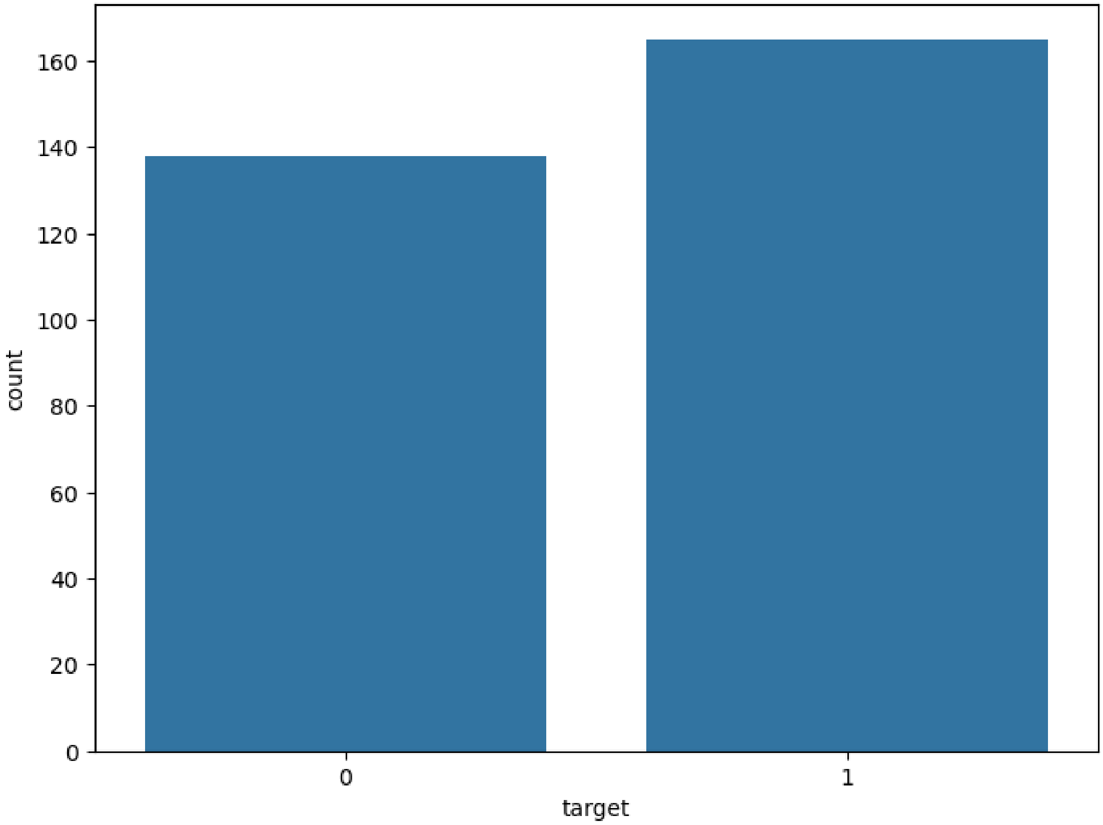
```
health['target'].value_counts()
```

[32]:

```
target
1    165
0    138
Name: count, dtype: int64
```

[33]:

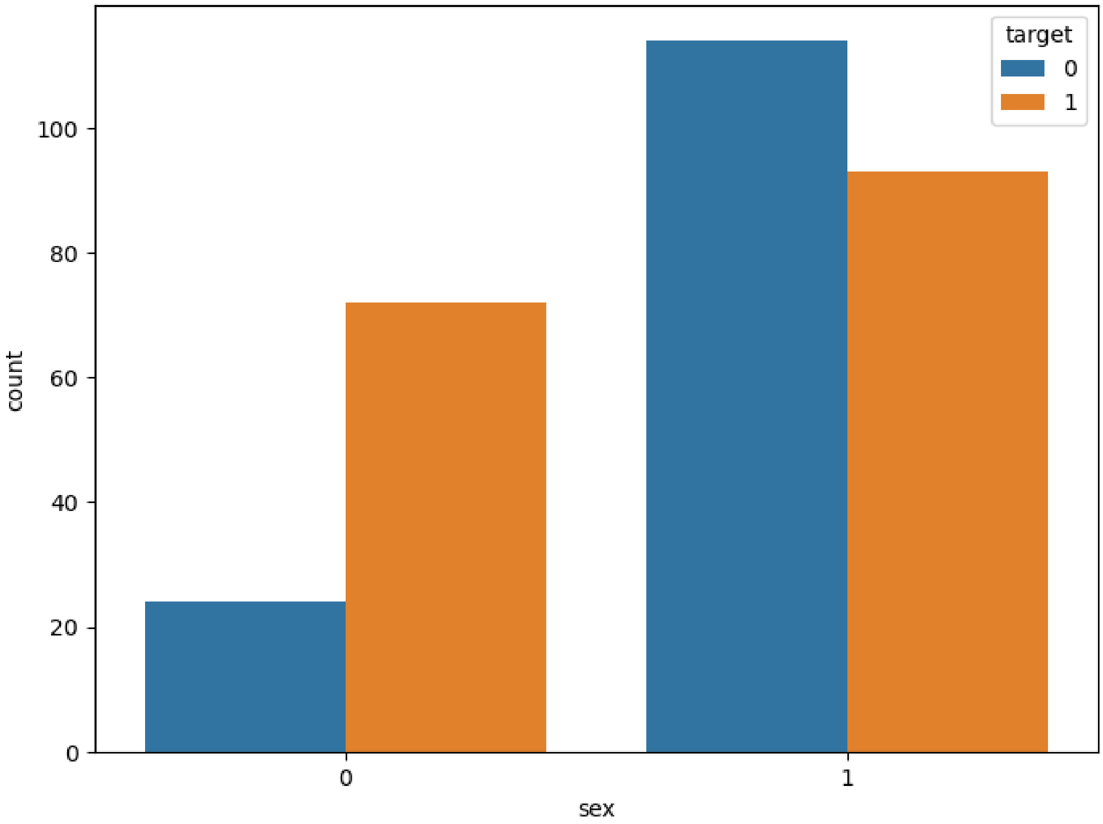
```
f, ax = plt.subplots(figsize=(8, 6))
ax = sns.countplot(x="target", data=health)
plt.show()
```



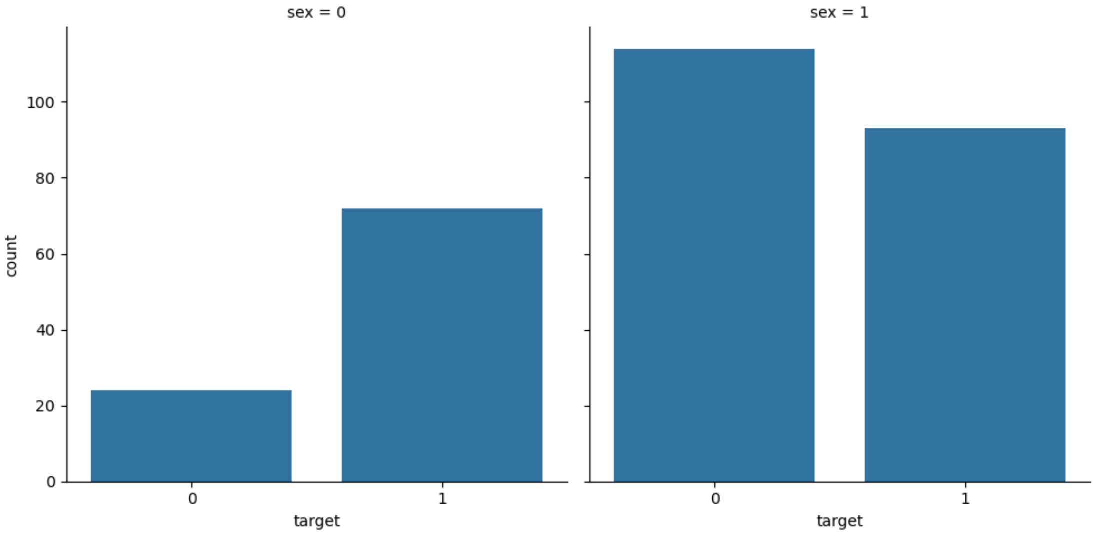
[34]:
```
health.groupby('sex')['target'].value_counts()
```

[34]:
```
sex  target
0    1          72
     0          24
1    0         114
     1          93
Name: count, dtype: int64
```

[36]:
```
f, ax = plt.subplots(figsize=(8, 6))
ax = sns.countplot(x="sex", hue="target", data=health)
plt.show()
```

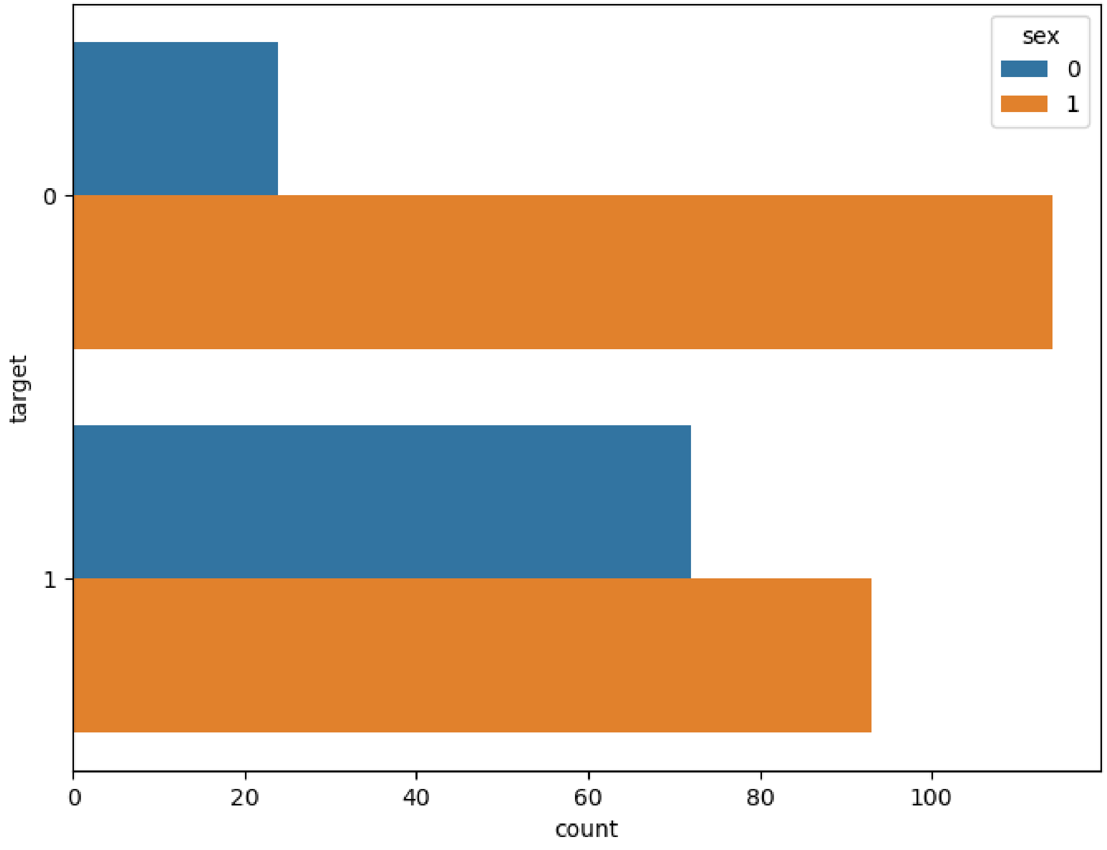


[38]:
```
ax = sns.catplot(x="target", col="sex", data=health, kind="count", height=5,␣
 ↪aspect=1)
```



[39]:

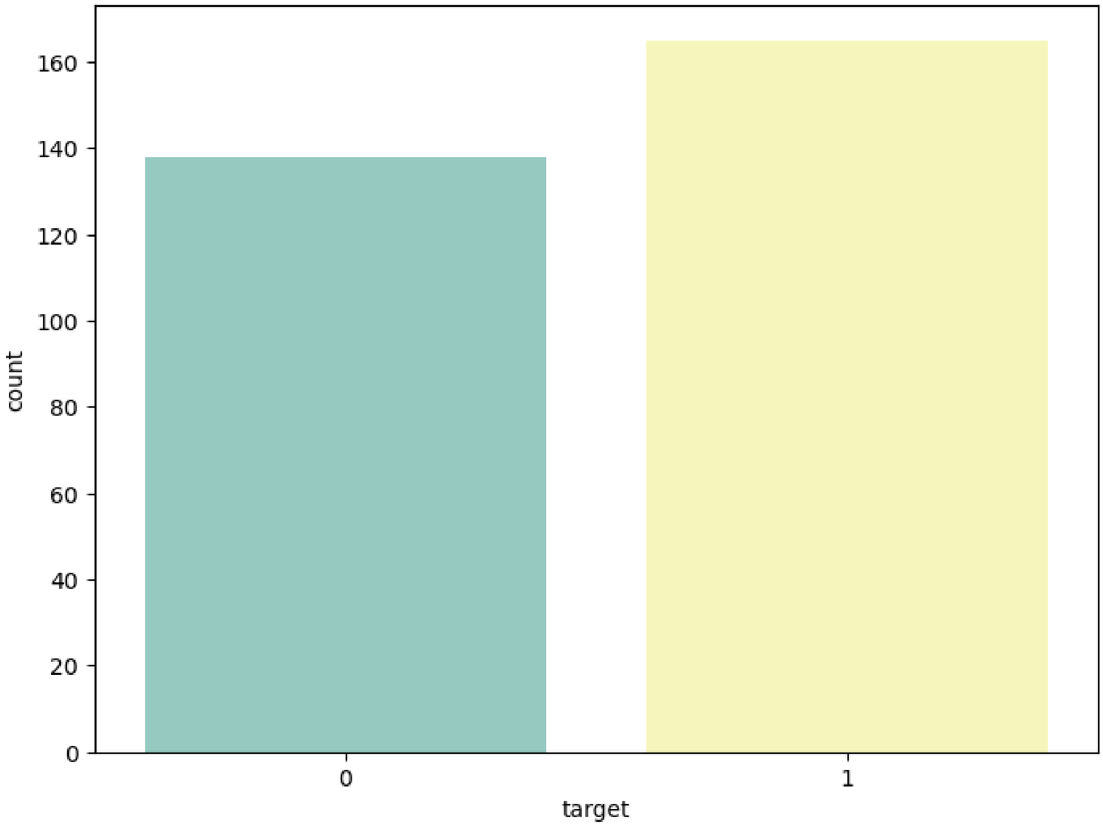
```
f, ax = plt.subplots(figsize=(8, 6))
ax = sns.countplot(y="target", hue="sex", data=health)
plt.show()
```



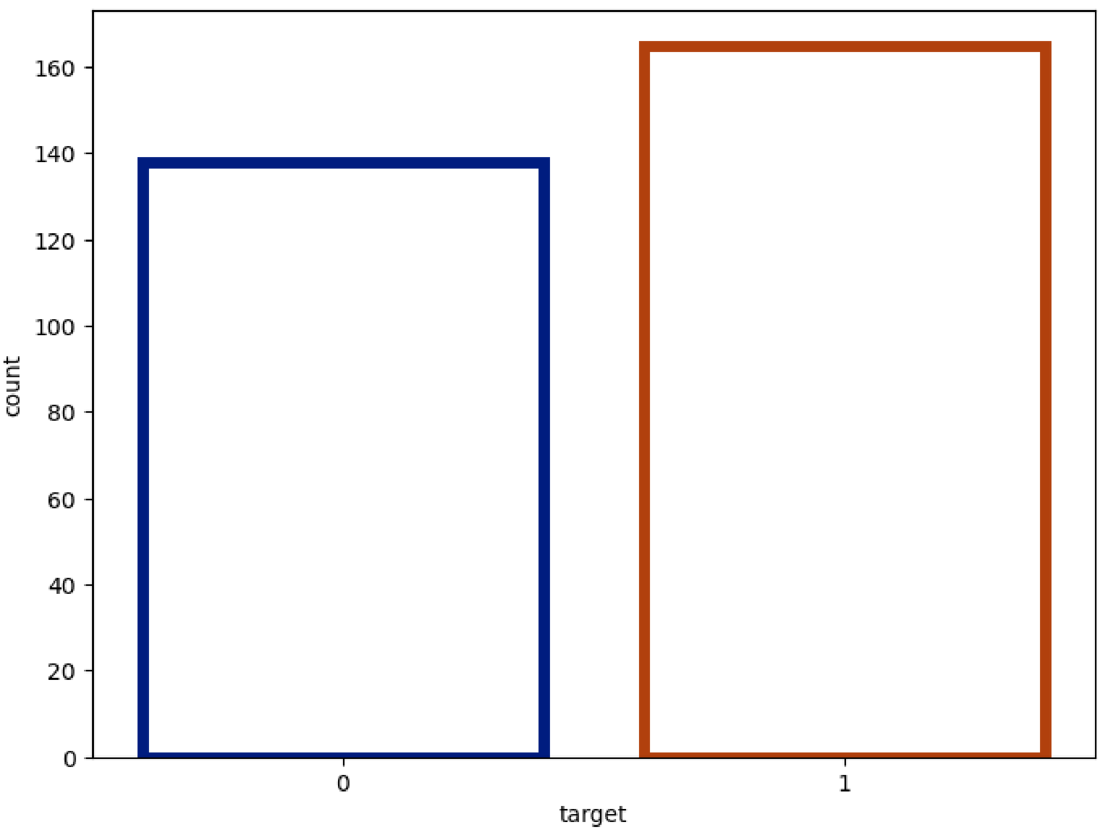
[40]:

```
f, ax = plt.subplots(figsize=(8, 6))
ax = sns.countplot(x="target", data=health, palette="Set3")
plt.show()
```

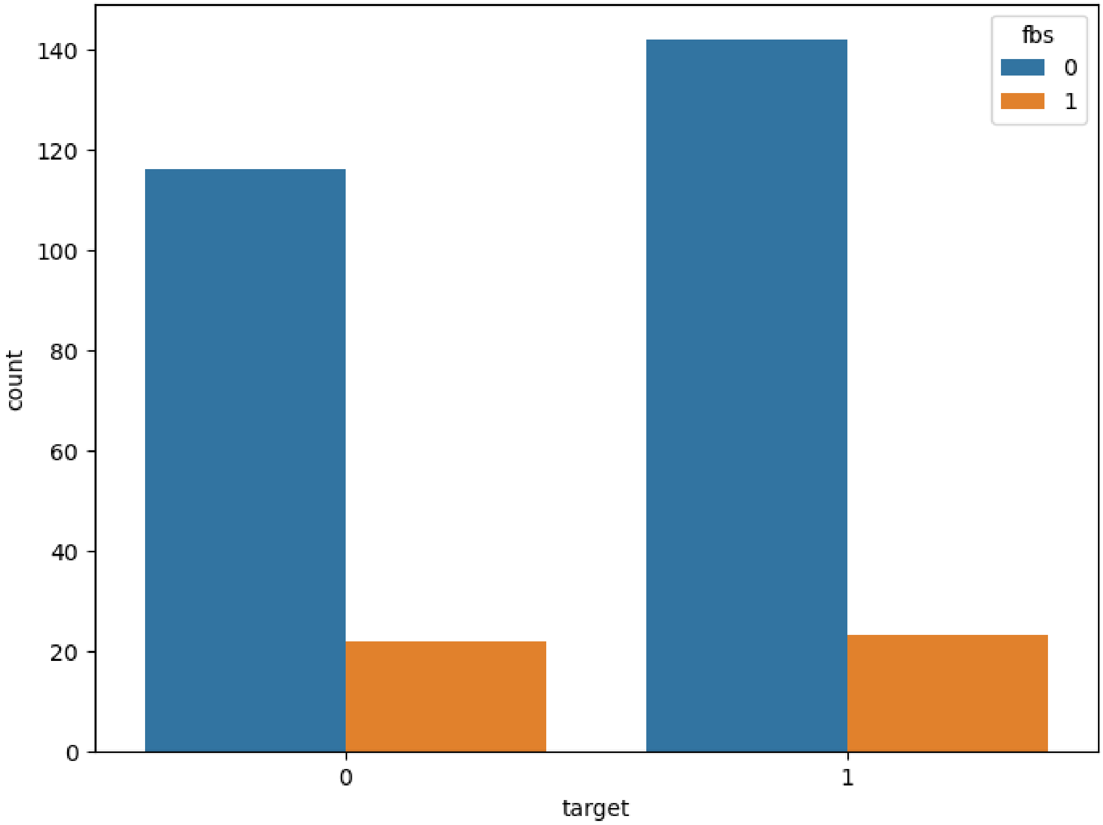
```
/var/folders/n0/q93fxsqn4kg2w2bw6zpftbth0000gn/T/ipykernel_15450/940474016.py:2:
FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in
v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same
effect.

  ax = sns.countplot(x="target", data=health, palette="Set3")
```



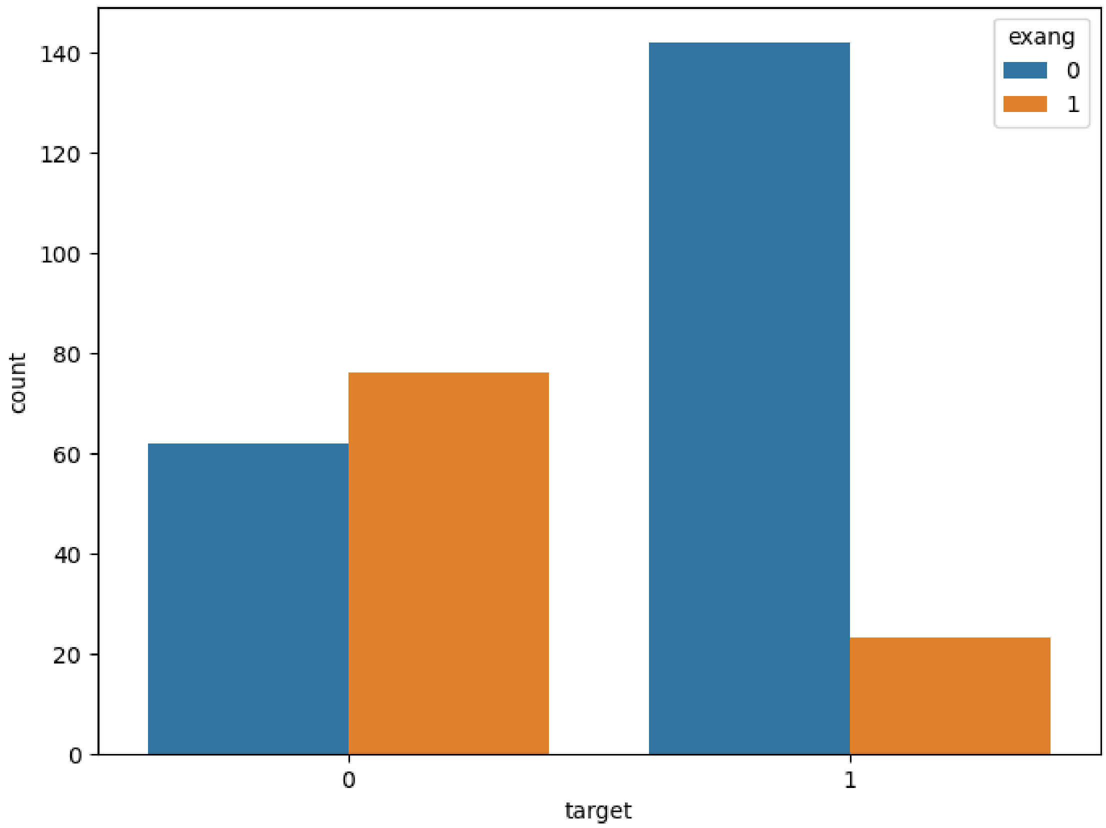
```
[41]: f, ax = plt.subplots(figsize=(8, 6))
      ax = sns.countplot(x="target", data=health, facecolor=(0, 0, 0, 0),␣
       ↪linewidth=5, edgecolor=sns.color_palette("dark", 3))
      plt.show()
```



[42]:
```
f, ax = plt.subplots(figsize=(8, 6))
ax = sns.countplot(x="target", hue="fbs", data=health)
plt.show()
```



[43]:
```
f, ax = plt.subplots(figsize=(8, 6))
ax = sns.countplot(x="target", hue="exang", data=health)
plt.show()
```



```
[46]: correlation = health.corr()     #bivariate
```

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

```
[47]: 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
```

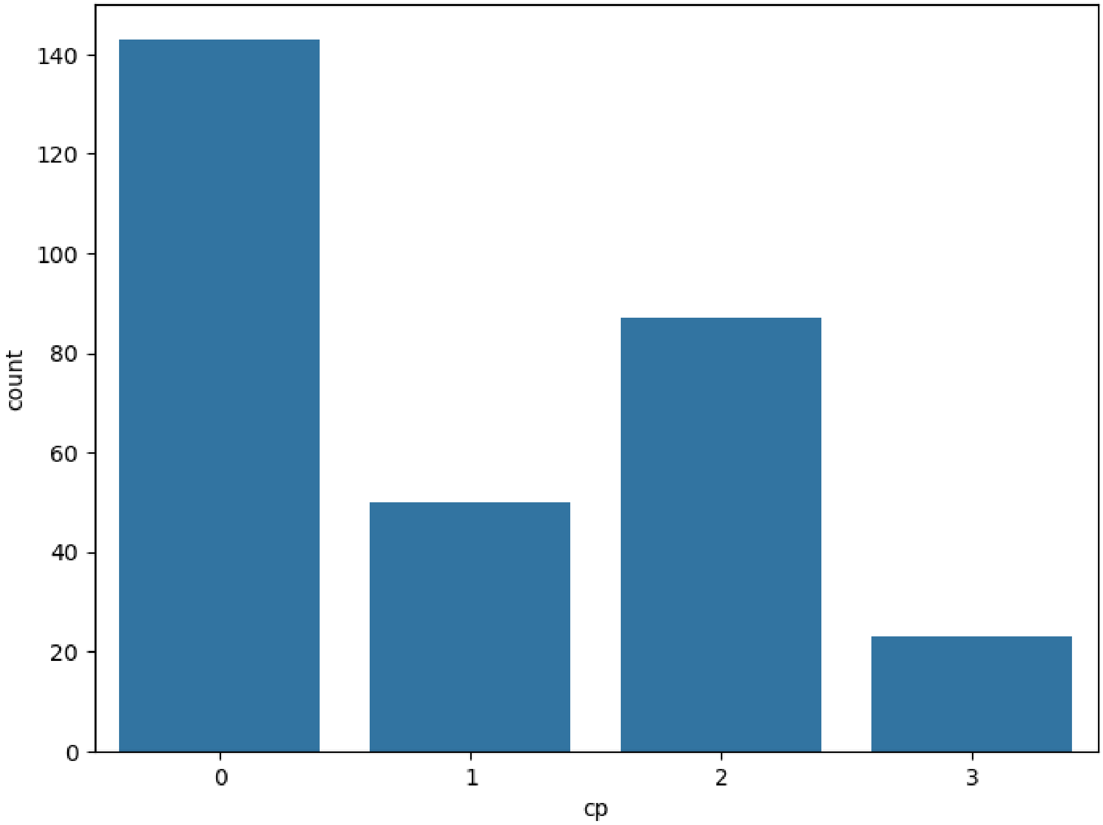
```
[49]: health['cp'].nunique()
```

```
[49]: 4
```

```
[50]: health['cp'].value_counts()
```

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

```
[51]: f, ax = plt.subplots(figsize=(8, 6))
      ax = sns.countplot(x="cp", data=health)
      plt.show()
```



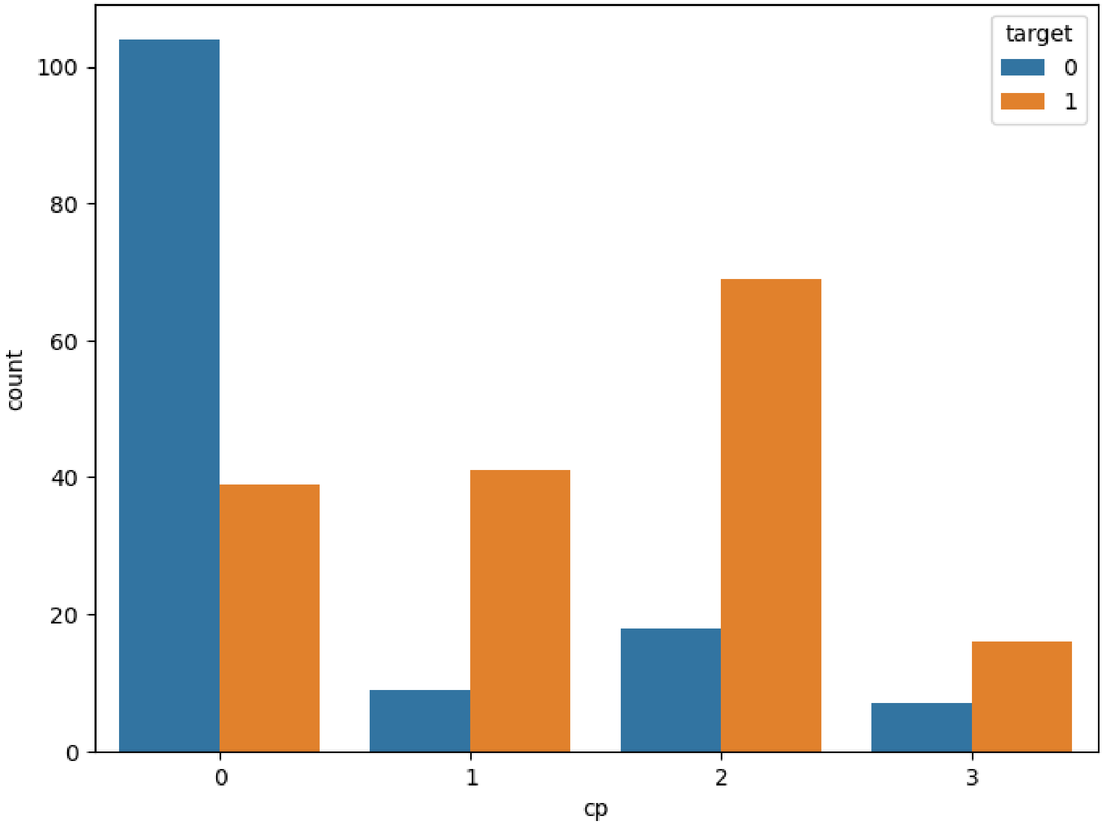
```
[52]: health.groupby('cp')['target'].value_counts()
```

[52]:

```
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
```

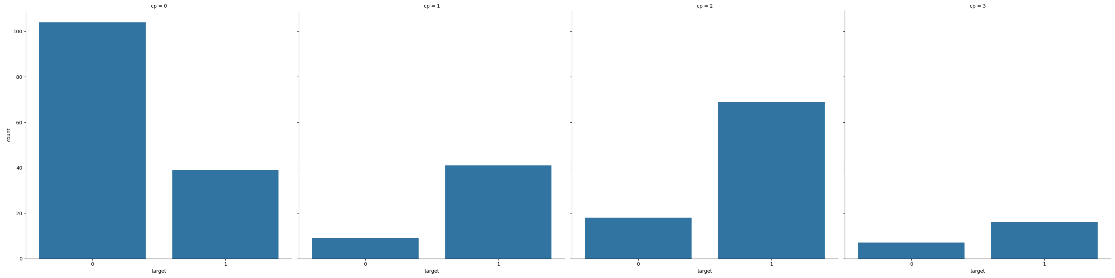
[54]:

```
f, ax = plt.subplots(figsize=(8, 6))
ax = sns.countplot(x="cp", hue="target", data=health)
plt.show()
```



[55]:

```
ax = sns.catplot(x="target", col="cp", data=health, kind="count", height=8,␣
 ↪aspect=1)
```



[56]:

```
health['thalach'].nunique()
```

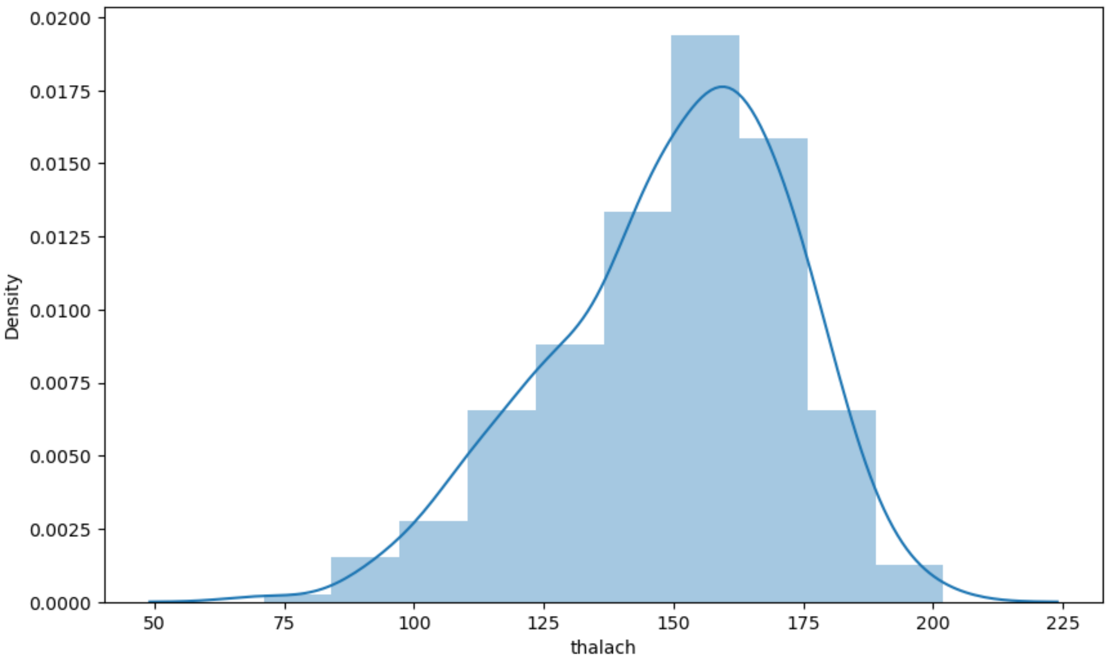
[56]:

```
91
```

[58]:

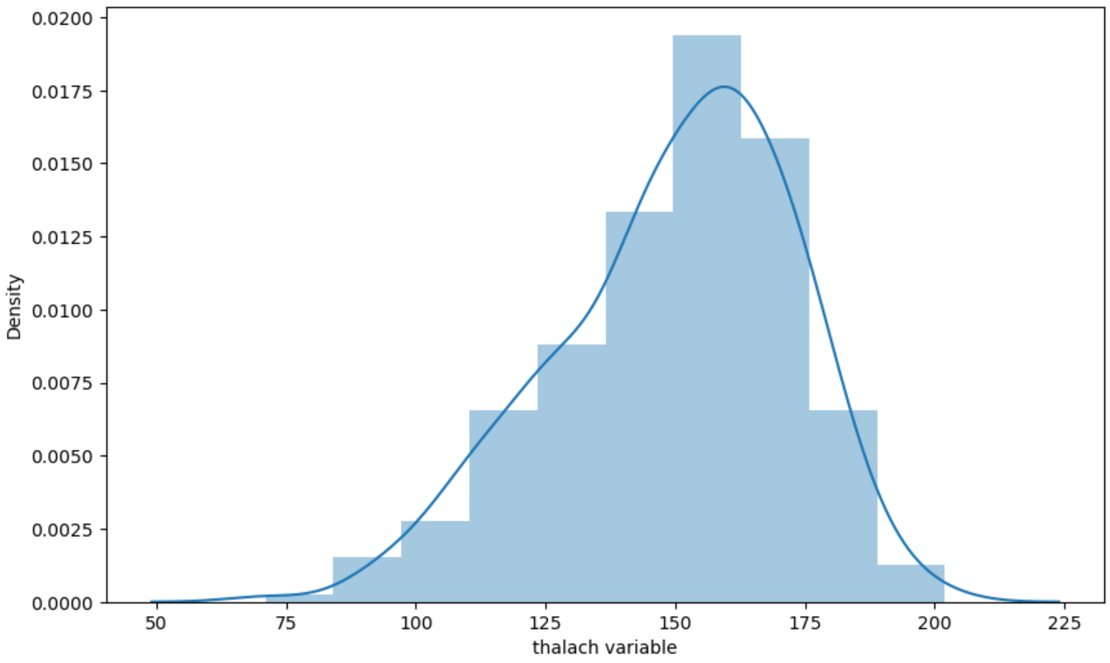
```
f, ax = plt.subplots(figsize=(10,6))
x = health['thalach']
ax = sns.distplot(x, bins=10)
plt.show()
```

```
/var/folders/n0/q93fxsqn4kg2w2bw6zpftbth0000gn/T/ipykernel_15450/1139321922.py:3
: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  ax = sns.distplot(x, bins=10)
```



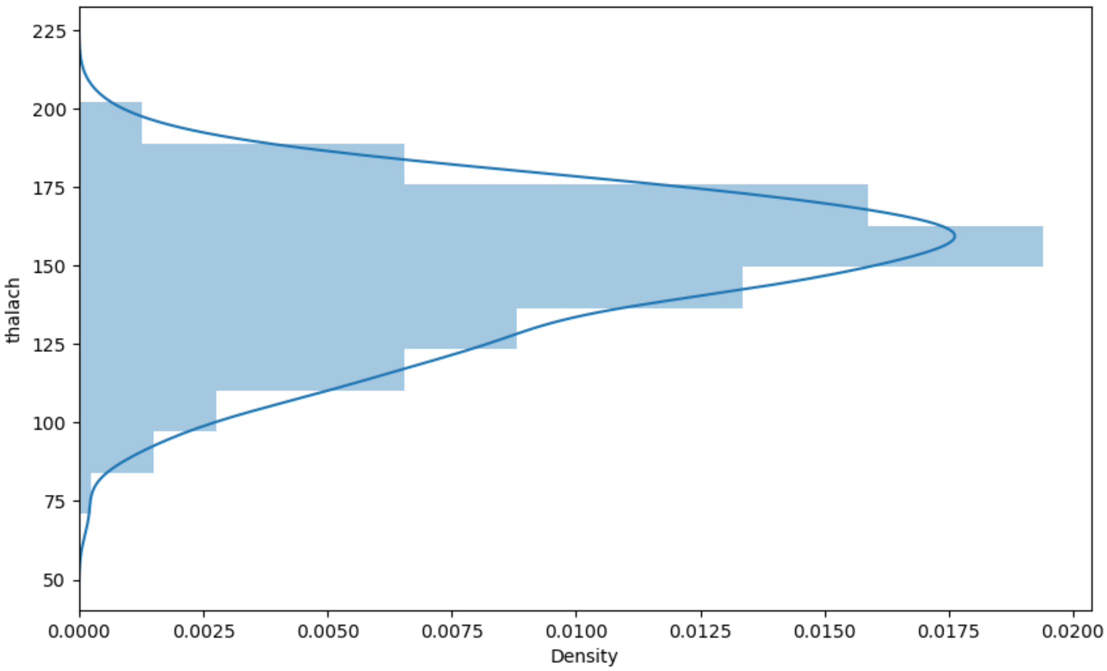
[59]:
```
f, ax = plt.subplots(figsize=(10,6))
x = health['thalach']
x = pd.Series(x, name="thalach variable")
ax = sns.distplot(x, bins=10)
plt.show()
```

```
/var/folders/n0/q93fxsqn4kg2w2bw6zpftbth0000gn/T/ipykernel_15450/2490189355.py:4
: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  ax = sns.distplot(x, bins=10)
```

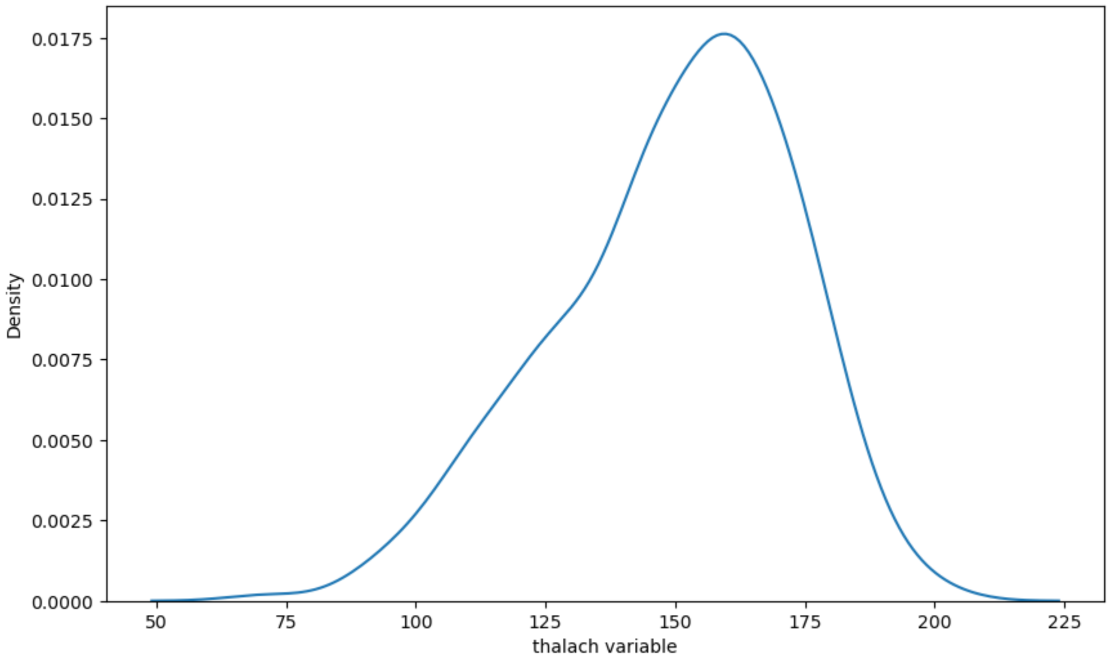


[60]:
```
f, ax = plt.subplots(figsize=(10,6))
x = health['thalach']
ax = sns.distplot(x, bins=10, vertical=True)
plt.show()
```

```
/var/folders/n0/q93fxsqn4kg2w2bw6zpftbth0000gn/T/ipykernel_15450/661047047.py:3:
UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  ax = sns.distplot(x, bins=10, vertical=True)
```

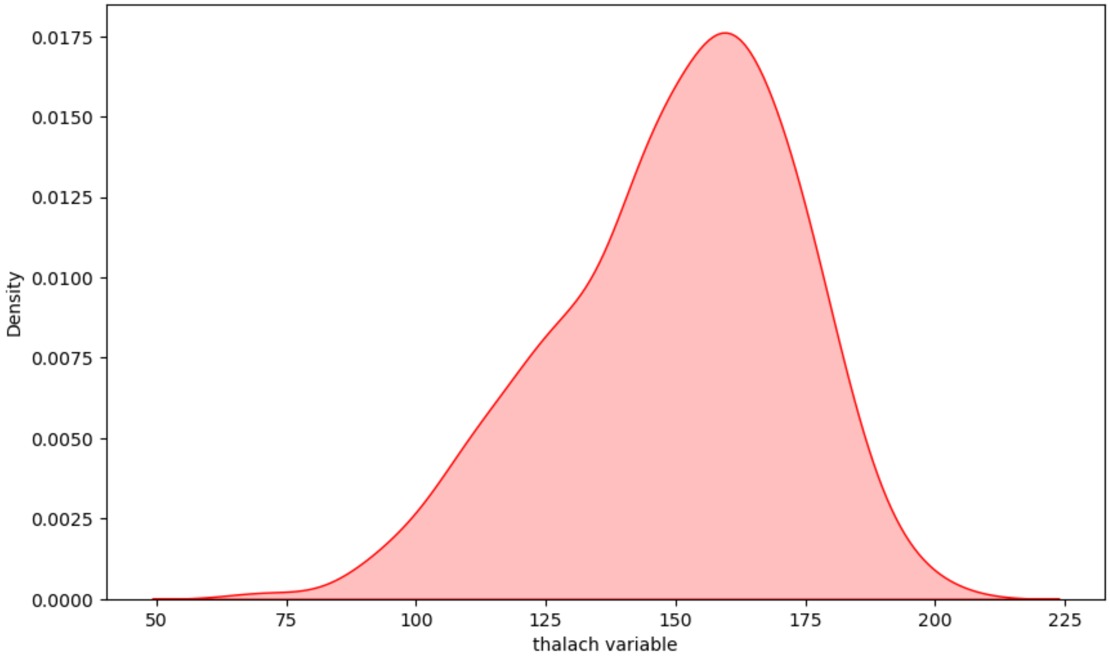


```
[61]: f, ax = plt.subplots(figsize=(10,6))
      x = health['thalach']
      x = pd.Series(x, name="thalach variable")
      ax = sns.kdeplot(x)
      plt.show()
```



[62]:
```
f, ax = plt.subplots(figsize=(10,6))
x = health['thalach']
x = pd.Series(x, name="thalach variable")
ax = sns.kdeplot(x, shade=True, color='r')
plt.show()
```

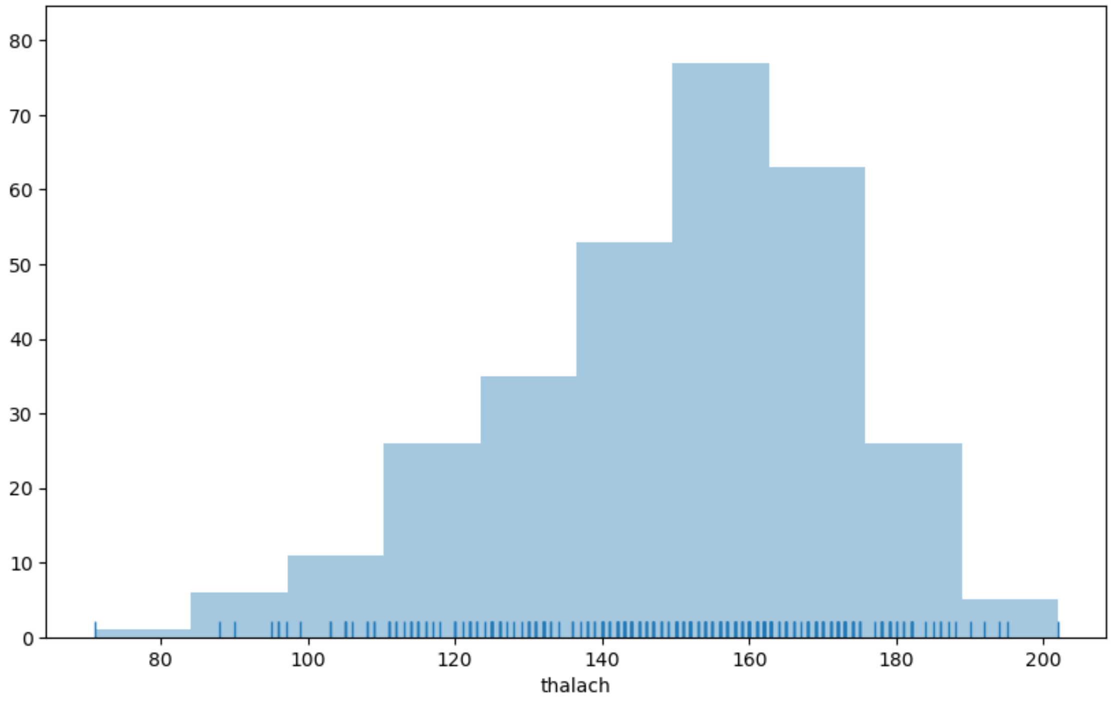
```
/var/folders/n0/q93fxsqn4kg2w2bw6zpftbth0000gn/T/ipykernel_15450/377926524.py:4:
FutureWarning:

`shade` is now deprecated in favor of `fill`; setting `fill=True`.
This will become an error in seaborn v0.14.0; please update your code.

  ax = sns.kdeplot(x, shade=True, color='r')
```



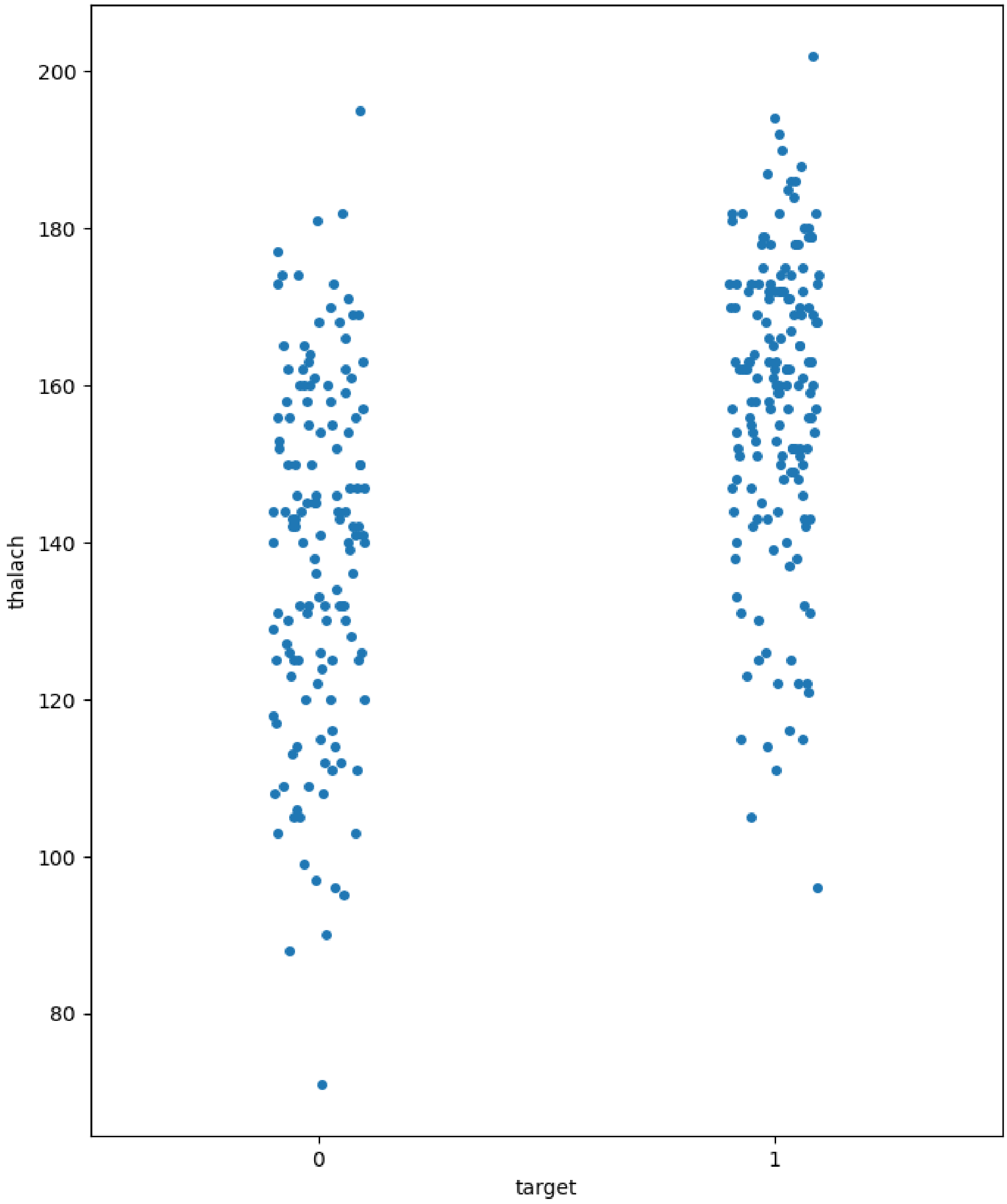
[63]:
```
f, ax = plt.subplots(figsize=(10,6))
x = health['thalach']
ax = sns.distplot(x, kde=False, rug=True, bins=10)
plt.show()
```

```
/var/folders/n0/q93fxsqn4kg2w2bw6zpftbth0000gn/T/ipykernel_15450/1175925800.py:3
: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.
```

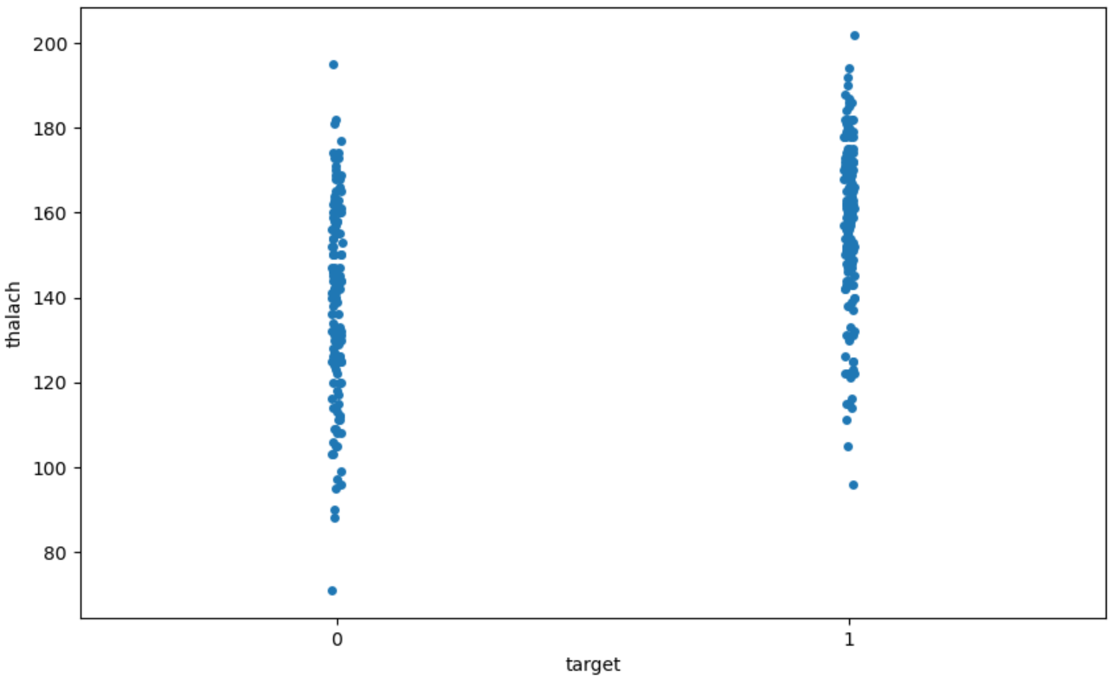
```
Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  ax = sns.distplot(x, kde=False, rug=True, bins=10)
```



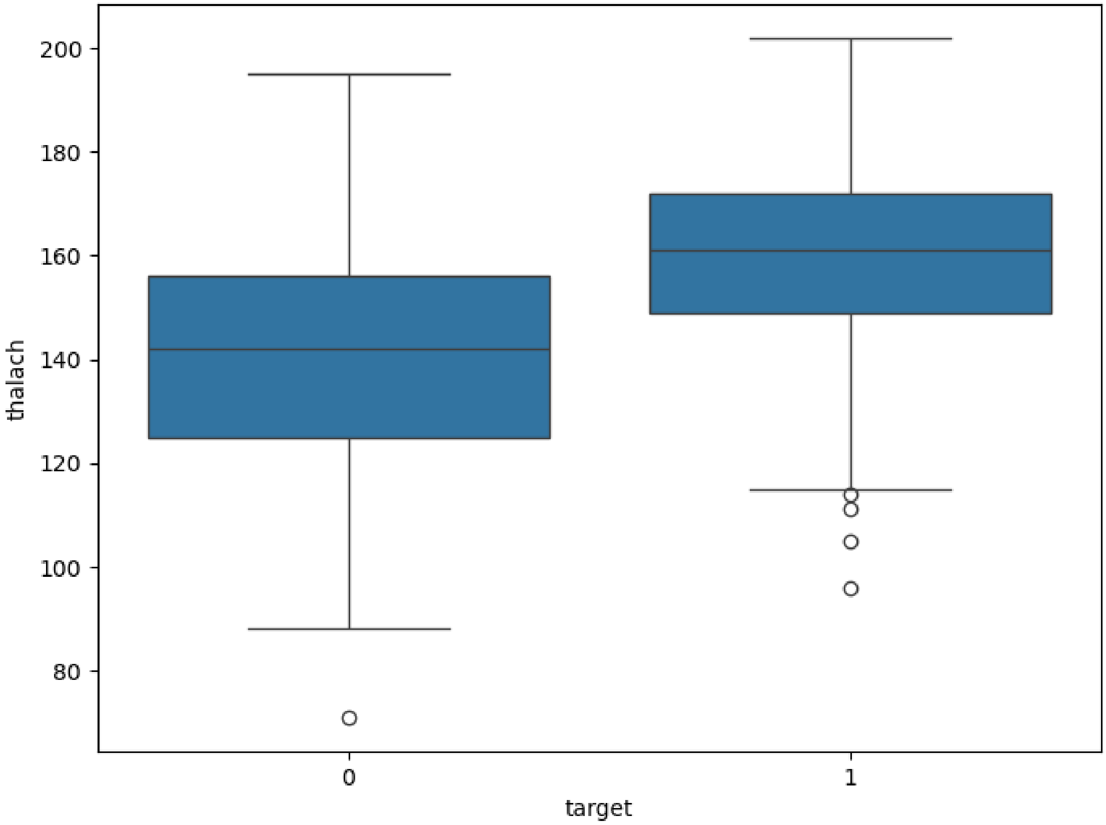
[67]:
```python
f, ax = plt.subplots(figsize=(8, 10))
sns.stripplot(x="target", y="thalach", data=health)
plt.show()
```



```
[69]: f, ax = plt.subplots(figsize=(10, 6))
      sns.stripplot(x="target", y="thalach", data=health, jitter = 0.01)
      plt.show()
```

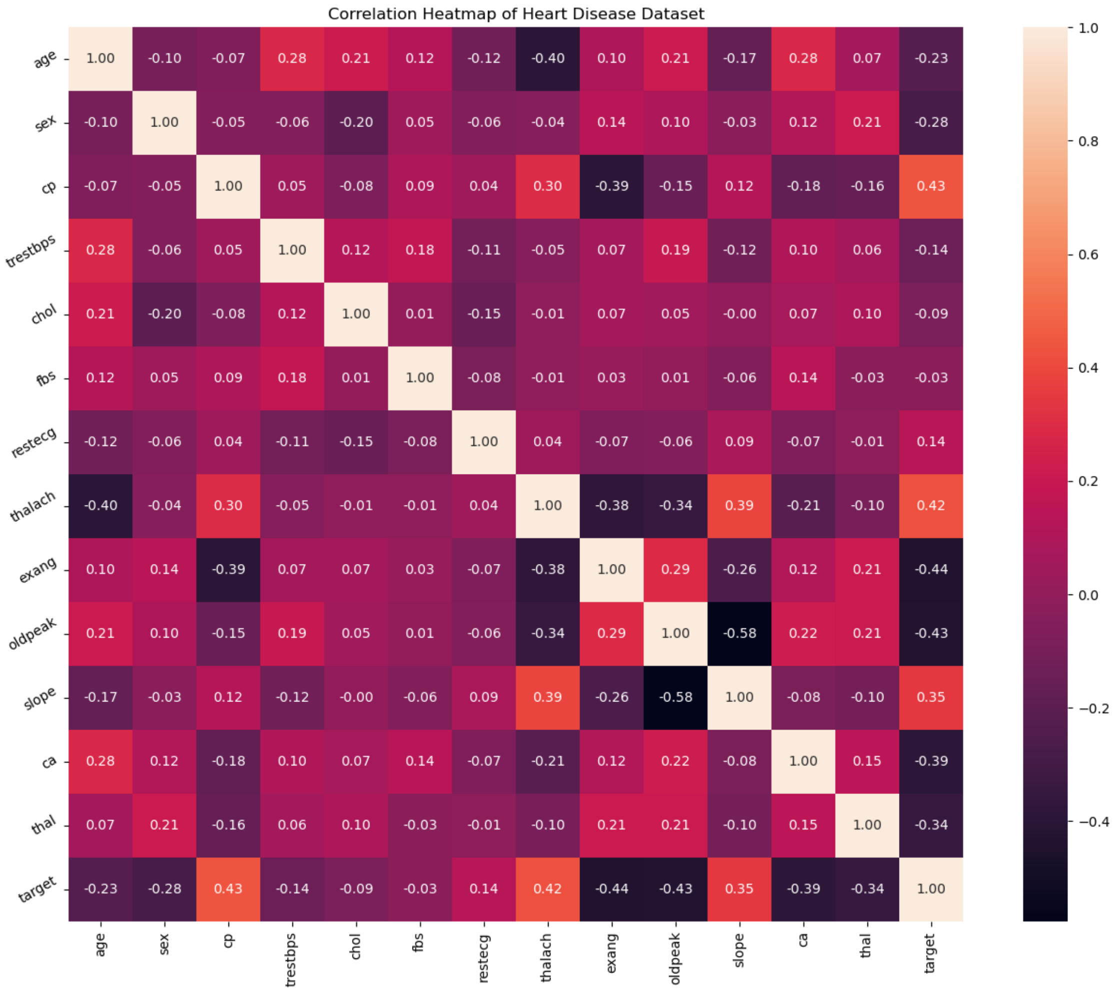


[70]:
```
f, ax = plt.subplots(figsize=(8, 6))
sns.boxplot(x="target", y="thalach", data=health)
plt.show()
```



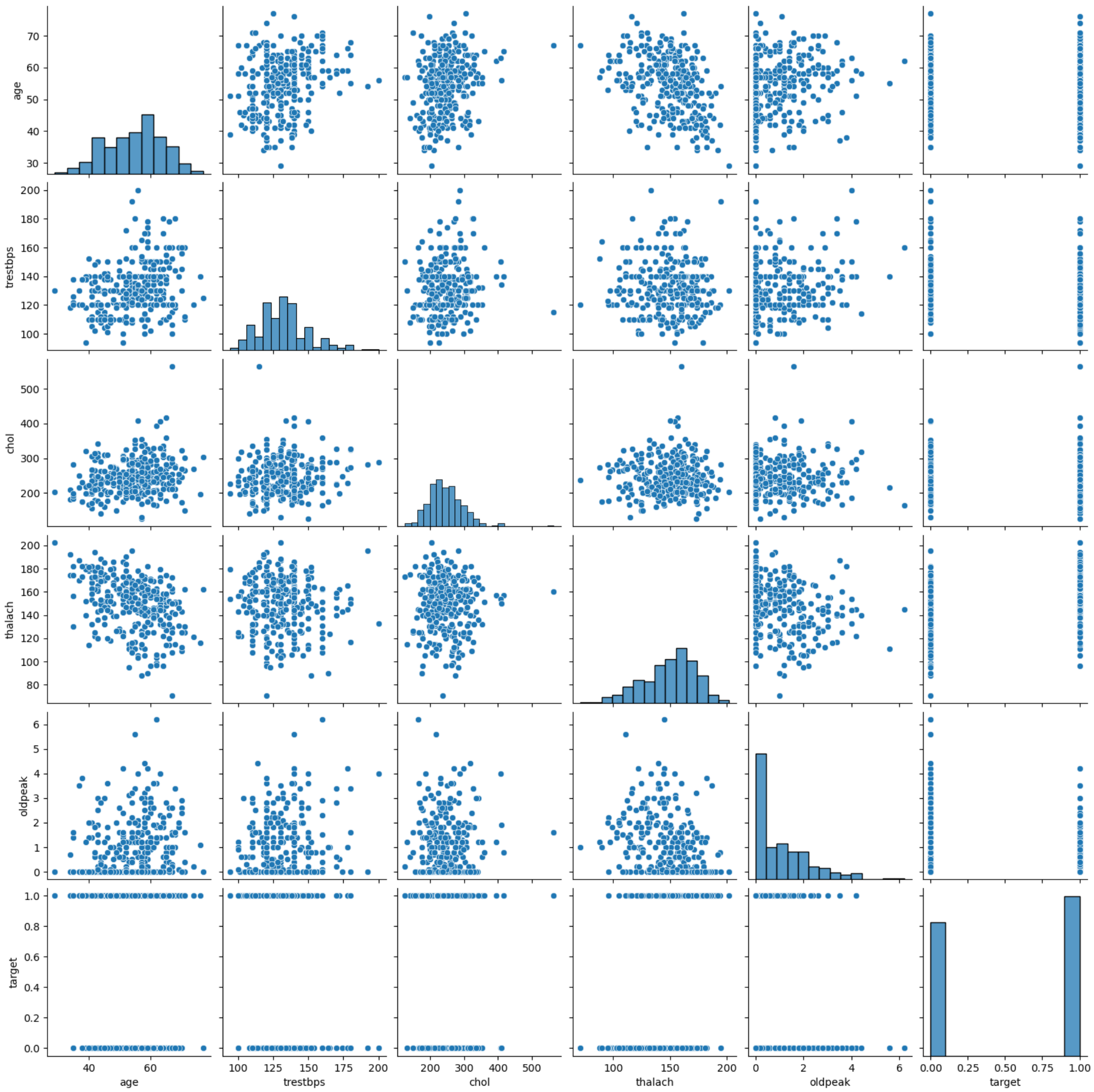
[71]:

```
#multivariate
plt.figure(figsize=(16,12))
plt.title('Correlation Heatmap of Heart Disease Dataset')
a = sns.heatmap(correlation, square=True, annot=True, fmt='.2f',␣
 ↪linecolor='white')
a.set_xticklabels(a.get_xticklabels(), rotation=90)
a.set_yticklabels(a.get_yticklabels(), rotation=30)
plt.show()
```



	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
age	1.00	-0.10	-0.07	0.28	0.21	0.12	-0.12	-0.40	0.10	0.21	-0.17	0.28	0.07	-0.23
sex	-0.10	1.00	-0.05	-0.06	-0.20	0.05	-0.06	-0.04	0.14	0.10	-0.03	0.12	0.21	-0.28
cp	-0.07	-0.05	1.00	0.05	-0.08	0.09	0.04	0.30	-0.39	-0.15	0.12	-0.18	-0.16	0.43
trestbps	0.28	-0.06	0.05	1.00	0.12	0.18	-0.11	-0.05	0.07	0.19	-0.12	0.10	0.06	-0.14
chol	0.21	-0.20	-0.08	0.12	1.00	0.01	-0.15	-0.01	0.07	0.05	-0.00	0.07	0.10	-0.09
fbs	0.12	0.05	0.09	0.18	0.01	1.00	-0.08	-0.01	0.03	0.01	-0.06	0.14	-0.03	-0.03
restecg	-0.12	-0.06	0.04	-0.11	-0.15	-0.08	1.00	0.04	-0.07	-0.06	0.09	-0.07	-0.01	0.14
thalach	-0.40	-0.04	0.30	-0.05	-0.01	-0.01	0.04	1.00	-0.38	-0.34	0.39	-0.21	-0.10	0.42
exang	0.10	0.14	-0.39	0.07	0.07	0.03	-0.07	-0.38	1.00	0.29	-0.26	0.12	0.21	-0.44
oldpeak	0.21	0.10	-0.15	0.19	0.05	0.01	-0.06	-0.34	0.29	1.00	-0.58	0.22	0.21	-0.43
slope	-0.17	-0.03	0.12	-0.12	-0.00	-0.06	0.09	0.39	-0.26	-0.58	1.00	-0.08	-0.10	0.35
ca	0.28	0.12	-0.18	0.10	0.07	0.14	-0.07	-0.21	0.12	0.22	-0.08	1.00	0.15	-0.39
thal	0.07	0.21	-0.16	0.06	0.10	-0.03	-0.01	-0.10	0.21	0.21	-0.10	0.15	1.00	-0.34
target	-0.23	-0.28	0.43	-0.14	-0.09	-0.03	0.14	0.42	-0.44	-0.43	0.35	-0.39	-0.34	1.00

[73]:
```
num_var = ['age', 'trestbps', 'chol', 'thalach', 'oldpeak', 'target' ]
sns.pairplot(health[num_var], kind='scatter', diag_kind='hist')
plt.show()
```



```
[74]: health['age'].nunique()
```

```
[74]: 41
```

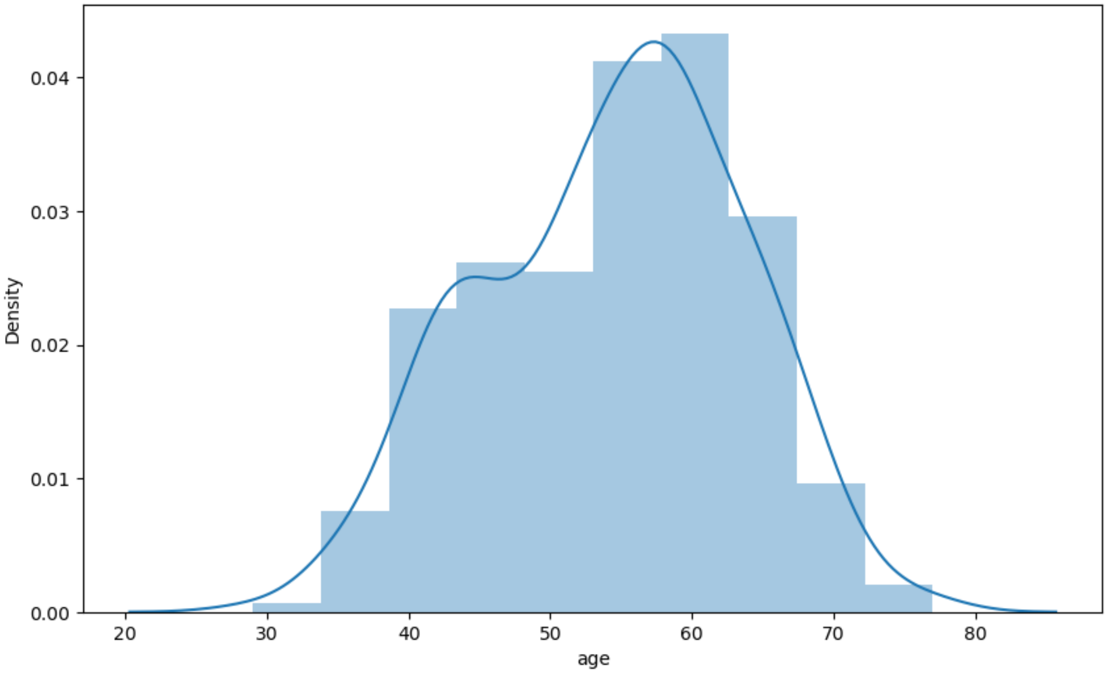
```
[75]: health['age'].describe()
```

```
[75]: count    303.000000
      mean      54.366337
      std        9.082101
      min       29.000000
      25%       47.500000
      50%       55.000000
      75%       61.000000
```

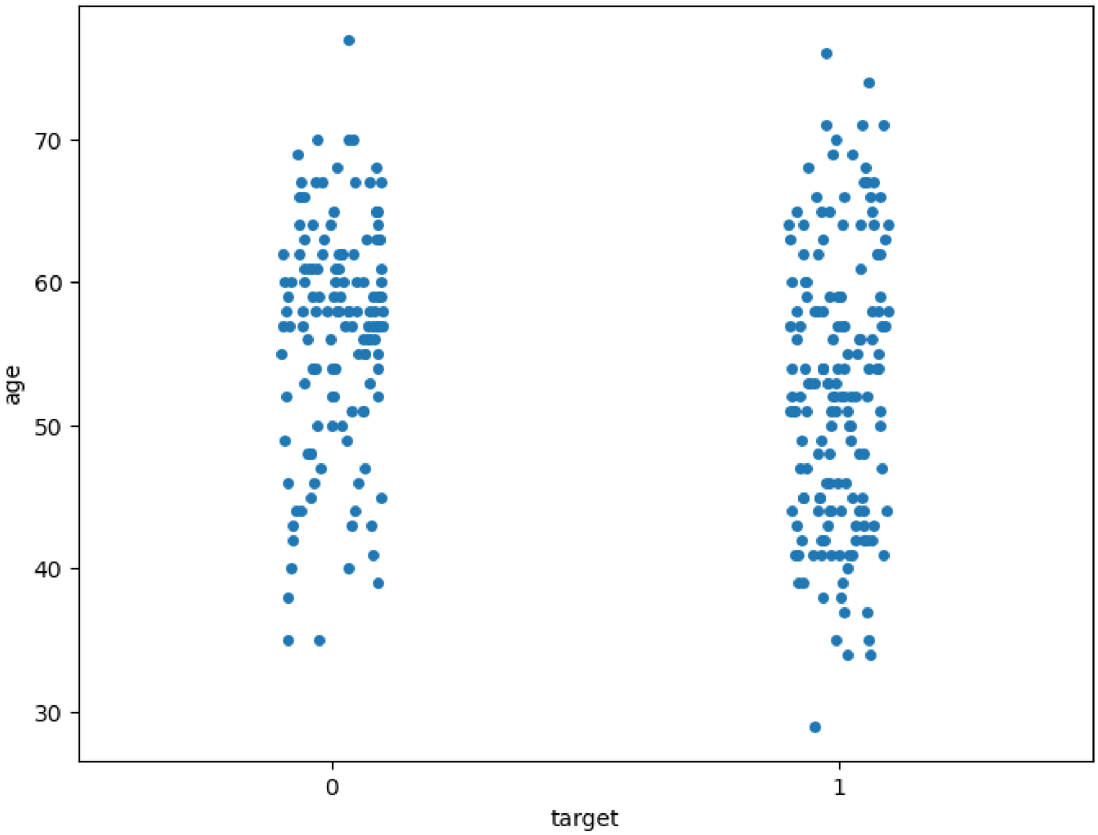
```
max        77.000000
Name: age, dtype: float64
```

[76]:
```
f, ax = plt.subplots(figsize=(10,6))
x = health['age']
ax = sns.distplot(x, bins=10)
plt.show()
```

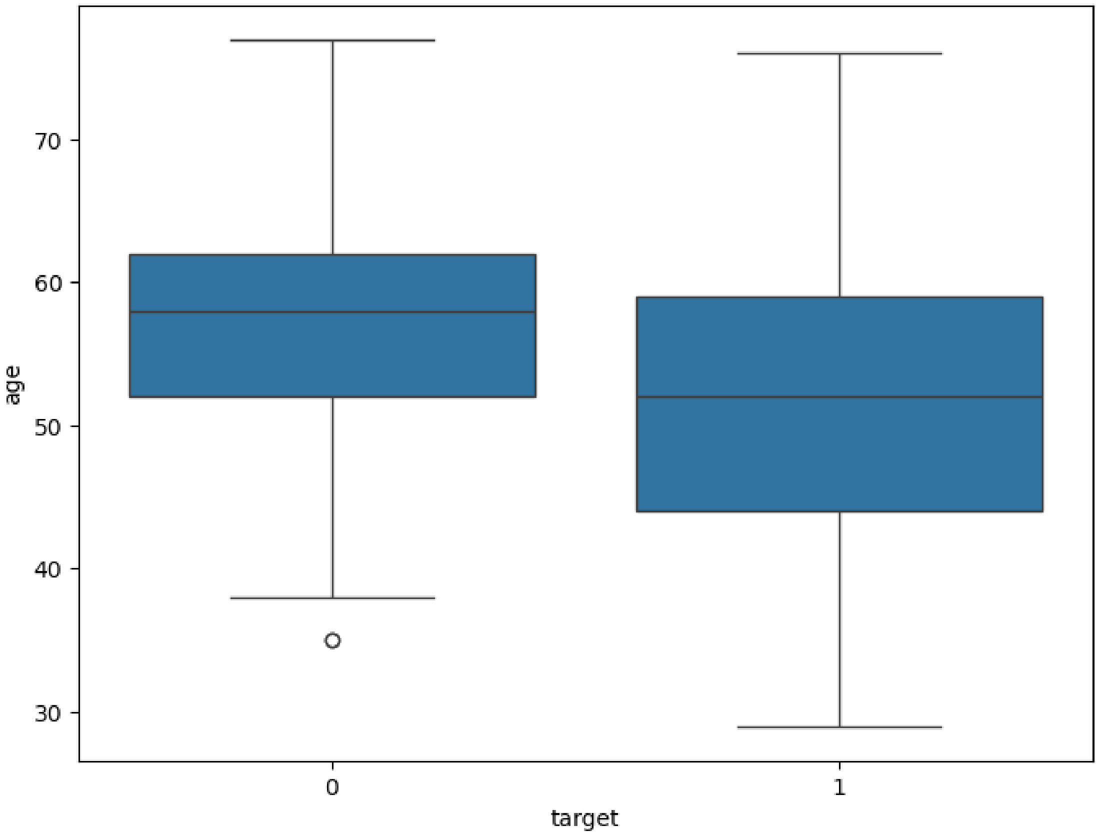
```
/var/folders/n0/q93fxsqn4kg2w2bw6zpftbth0000gn/T/ipykernel_15450/211720129.py:3:
UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  ax = sns.distplot(x, bins=10)
```



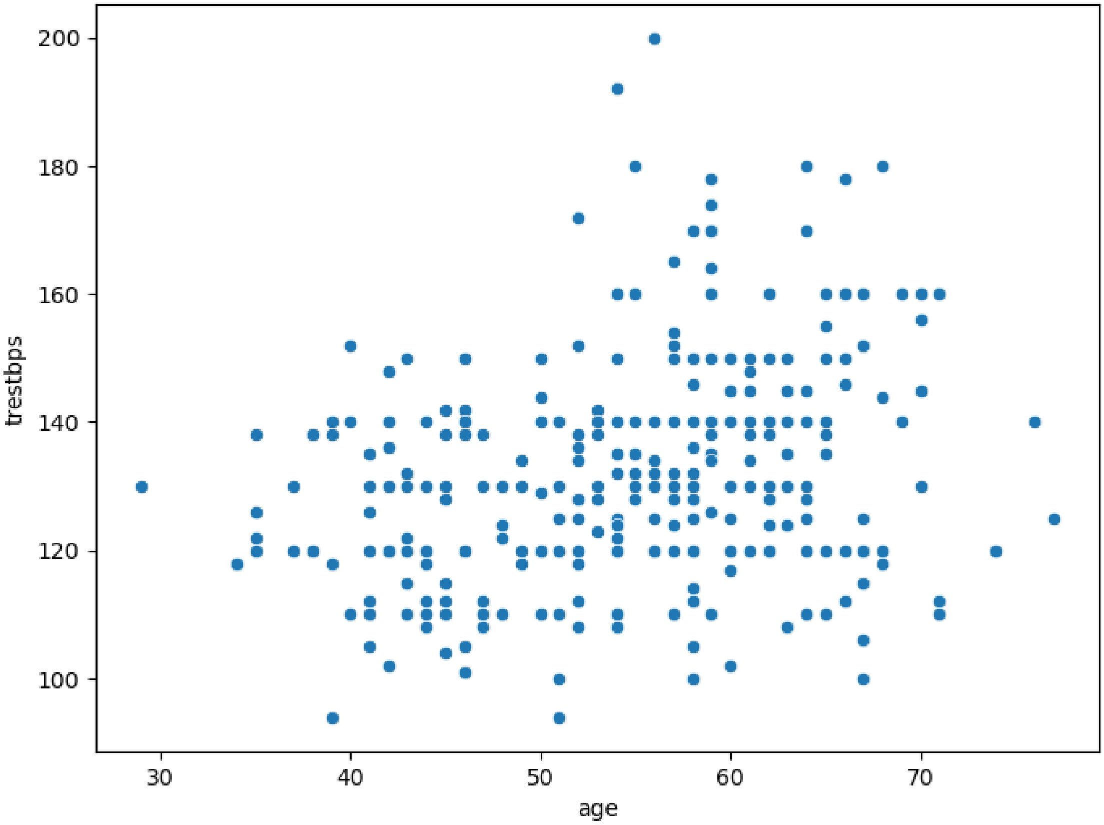
[77]:
```
f, ax = plt.subplots(figsize=(8, 6))
sns.stripplot(x="target", y="age", data=health)
plt.show()
```



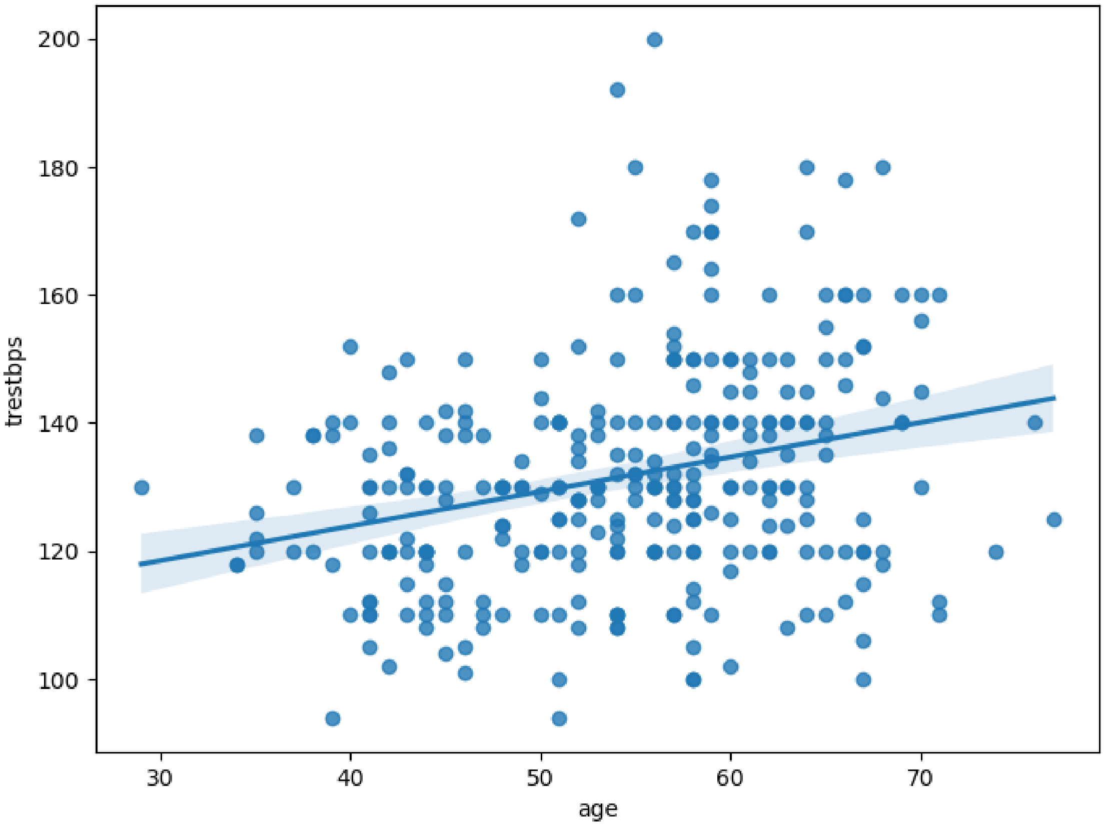
[78]:
```
f, ax = plt.subplots(figsize=(8, 6))
sns.boxplot(x="target", y="age", data=health)
plt.show()
```



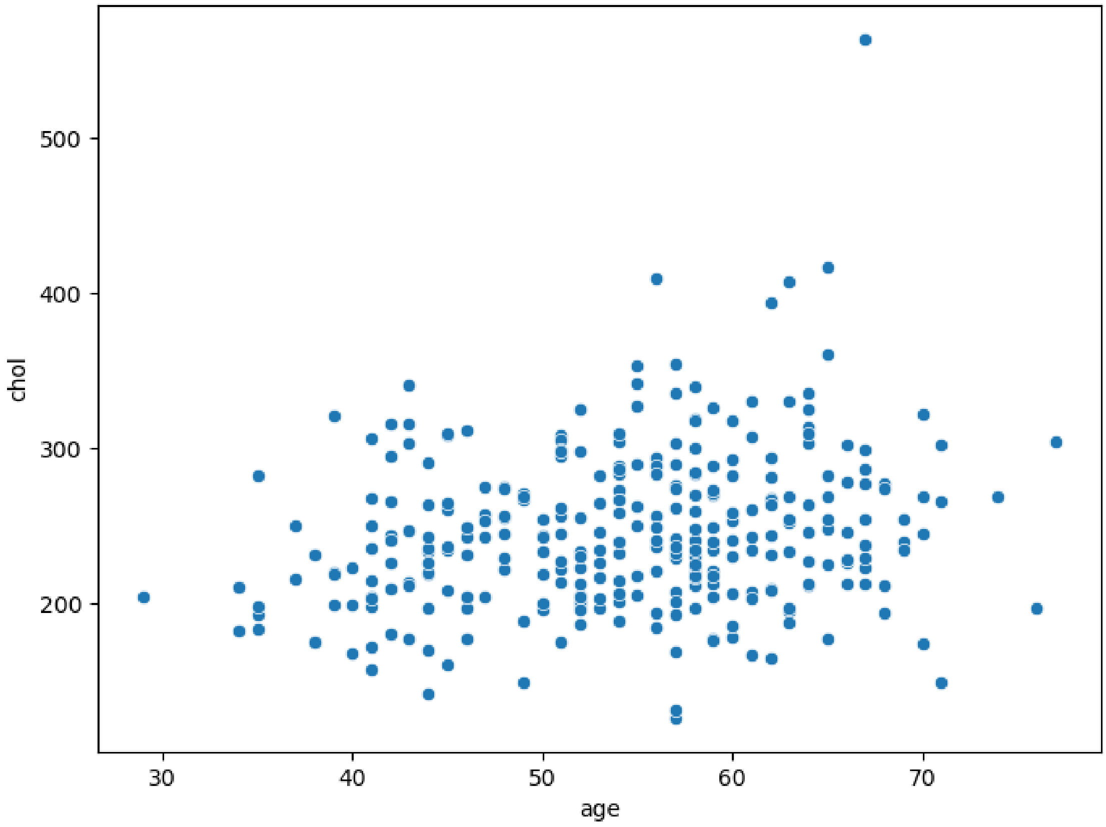
[79]:
```
f, ax = plt.subplots(figsize=(8, 6))
ax = sns.scatterplot(x="age", y="trestbps", data=health)
plt.show()
```



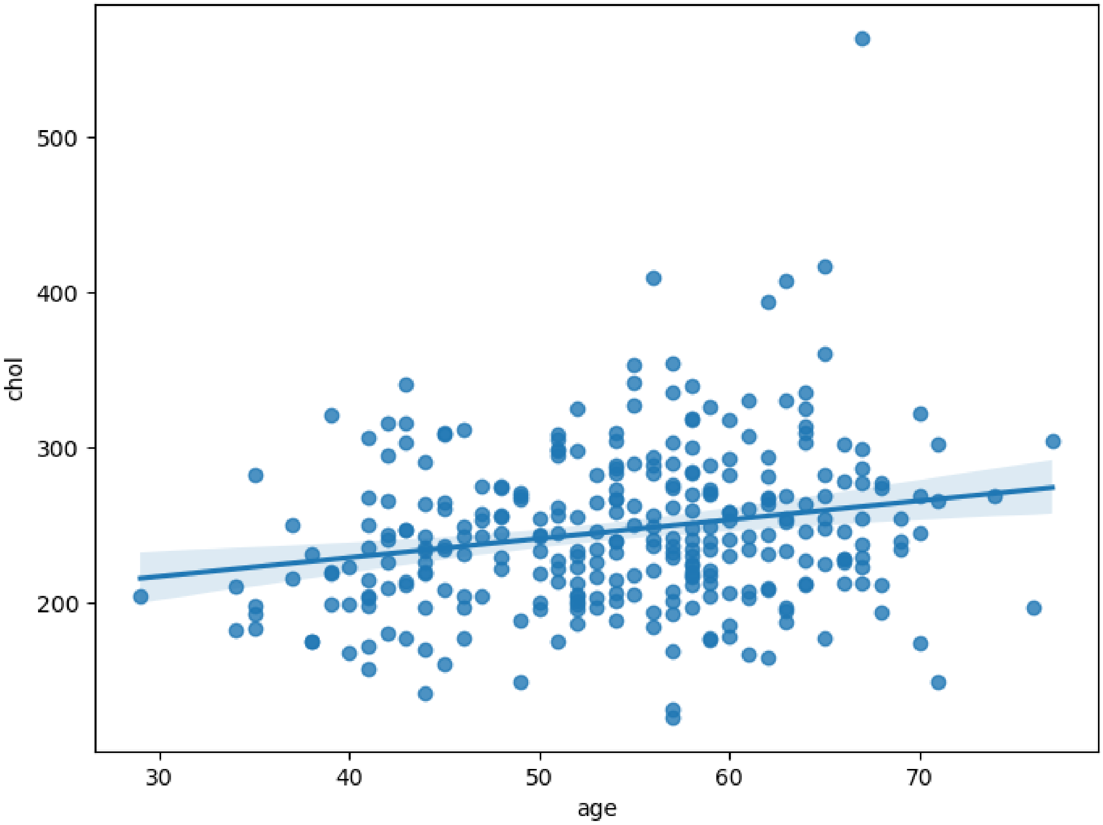
```
[80]: f, ax = plt.subplots(figsize=(8, 6))
      ax = sns.regplot(x="age", y="trestbps", data=health)
      plt.show()
```



```
[81]: f, ax = plt.subplots(figsize=(8, 6))
      ax = sns.scatterplot(x="age", y="chol", data=health)
      plt.show()
```

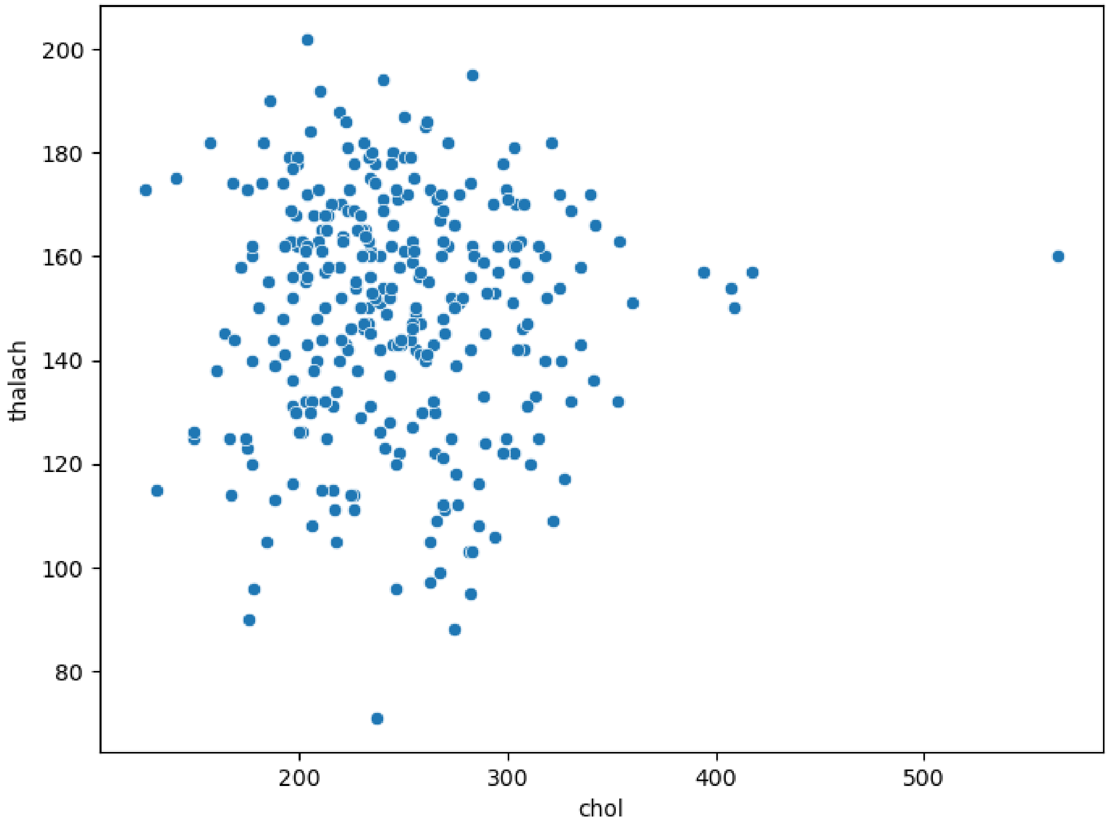


```
[82]: f, ax = plt.subplots(figsize=(8, 6))
      ax = sns.regplot(x="age", y="chol", data=health)
      plt.show()
```

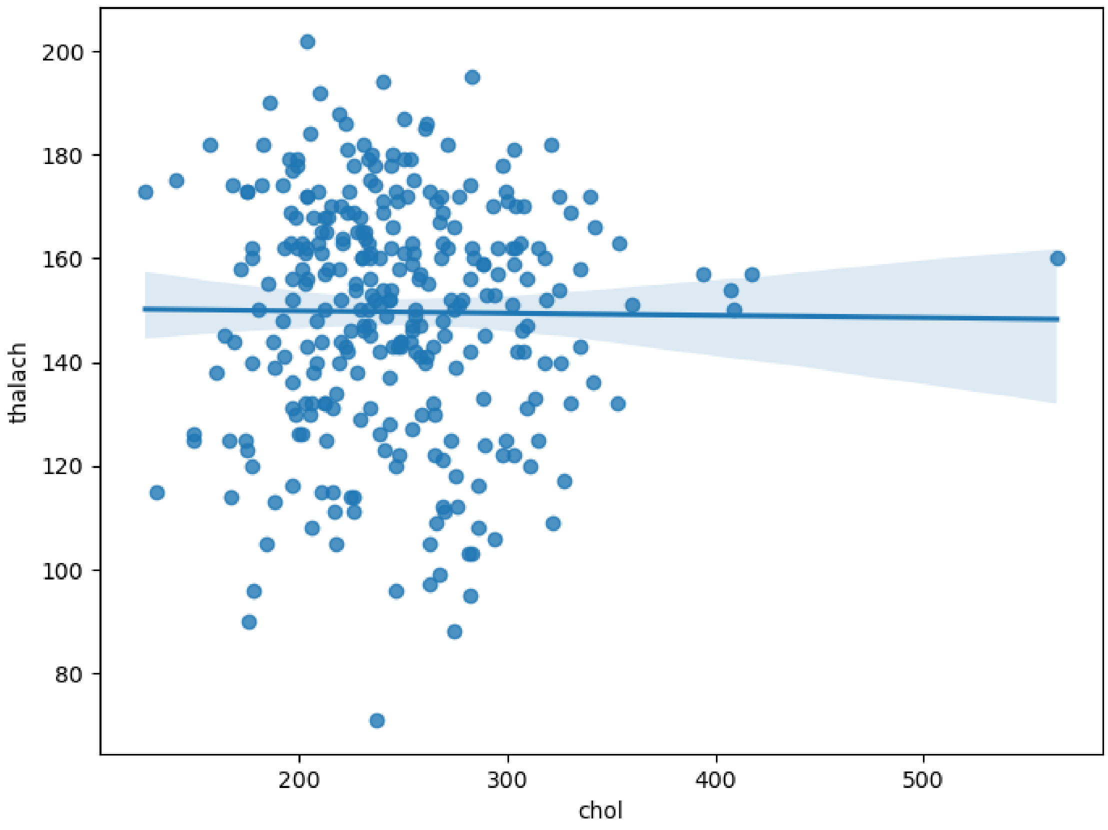


[83]:

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



```
[84]: f, ax = plt.subplots(figsize=(8, 6))
      ax = sns.regplot(x="chol", y="thalach", data=health)
      plt.show()
```



```
[85]: health.isnull().sum()
```

```
[85]: 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
```

```
[86]: health.isnull().sum().sum()
```

[86]:

```
np.int64(0)
```

[87]:

```
health.isnull().mean()
```

[87]:

```
age         0.0
sex         0.0
cp          0.0
trestbps    0.0
chol        0.0
fbs         0.0
restecg     0.0
thalach     0.0
exang       0.0
oldpeak     0.0
slope       0.0
ca          0.0
thal        0.0
target      0.0
dtype: float64
```

[88]:

```
assert pd.notnull(health).all().all()
```

[89]:

```
assert (health >= 0).all().all()
```

[90]:

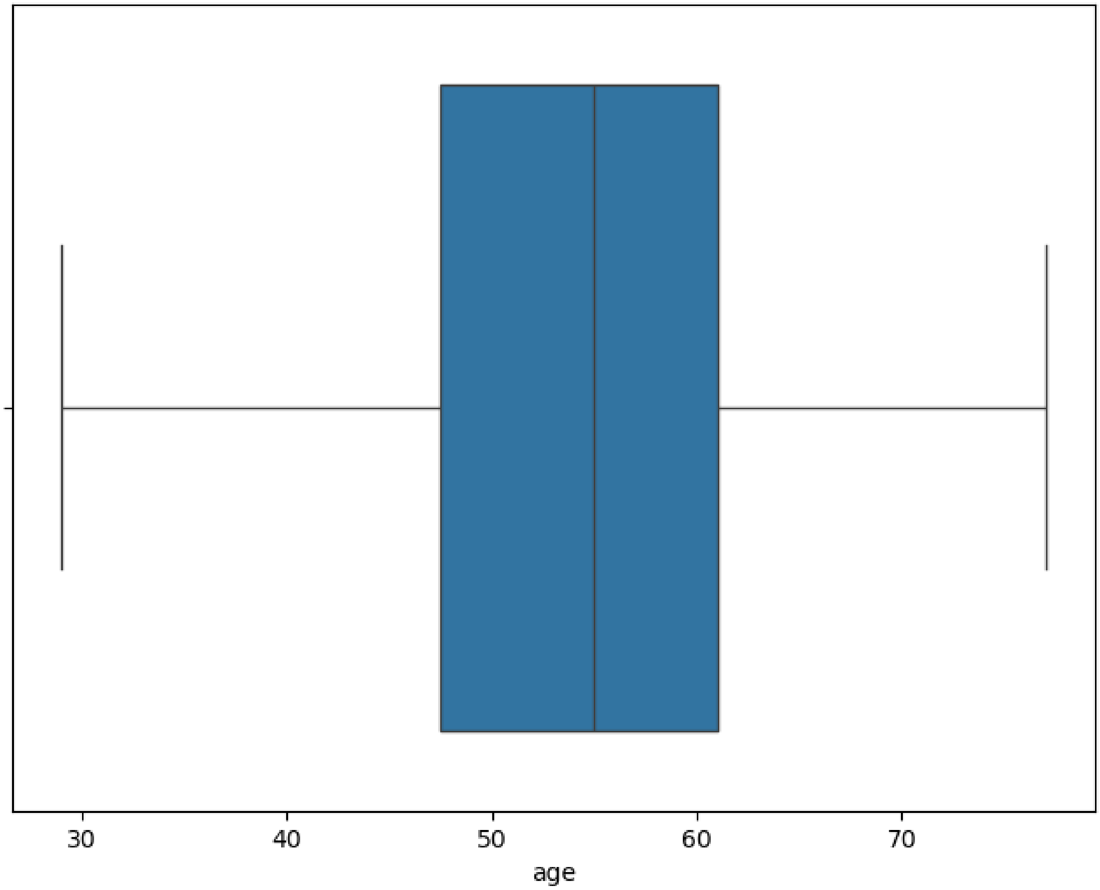
```
health['age'].describe()
```

[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
```

[91]:

```
f, ax = plt.subplots(figsize=(8, 6))
sns.boxplot(x=health["age"])
plt.show()
```



[92]:
```
health['trestbps'].describe()
```

[92]:
```
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
```

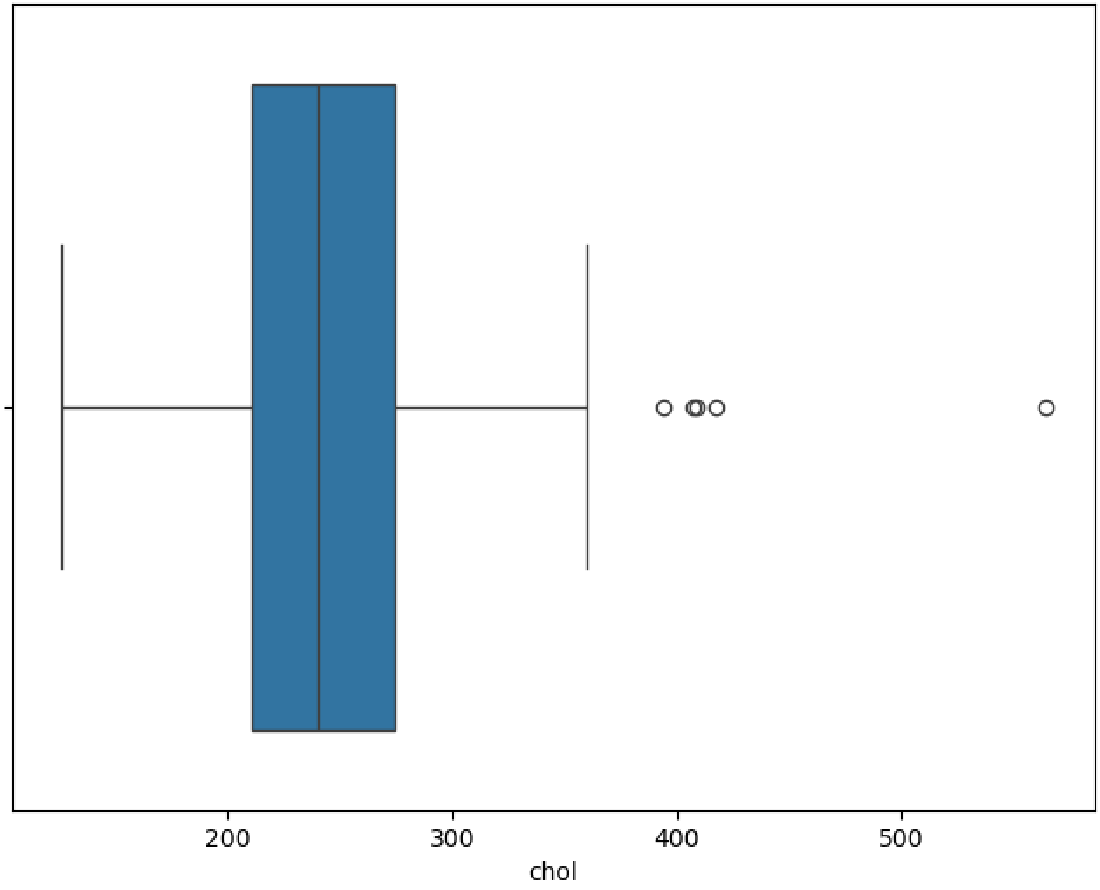
[93]:
```
health['chol'].describe()
```

[93]:
```
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
```

[94]:

```
f, ax = plt.subplots(figsize=(8, 6))
sns.boxplot(x=health["chol"])
plt.show()
```



[95]:

```
health['thalach'].describe()
```

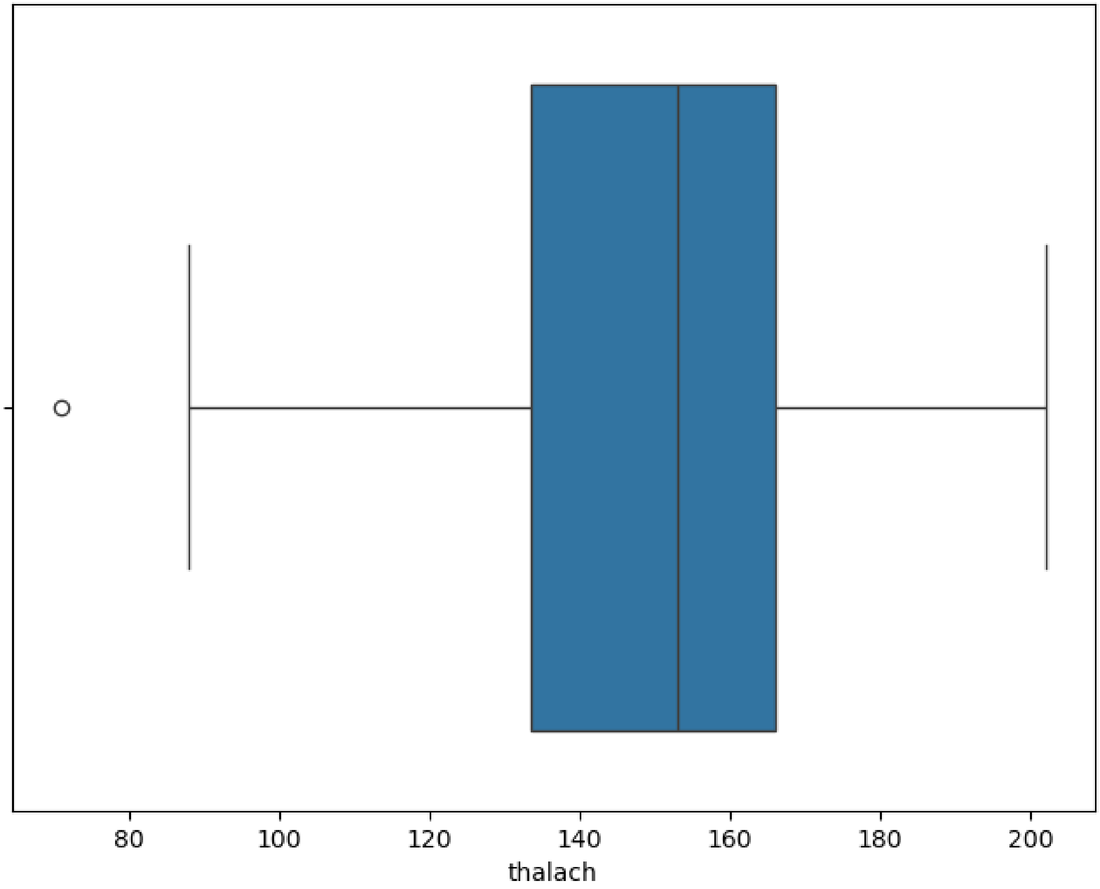
[95]:

```
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
```

[96]:

```
f, ax = plt.subplots(figsize=(8, 6))
sns.boxplot(x=health["thalach"])
plt.show()
```



[97]:

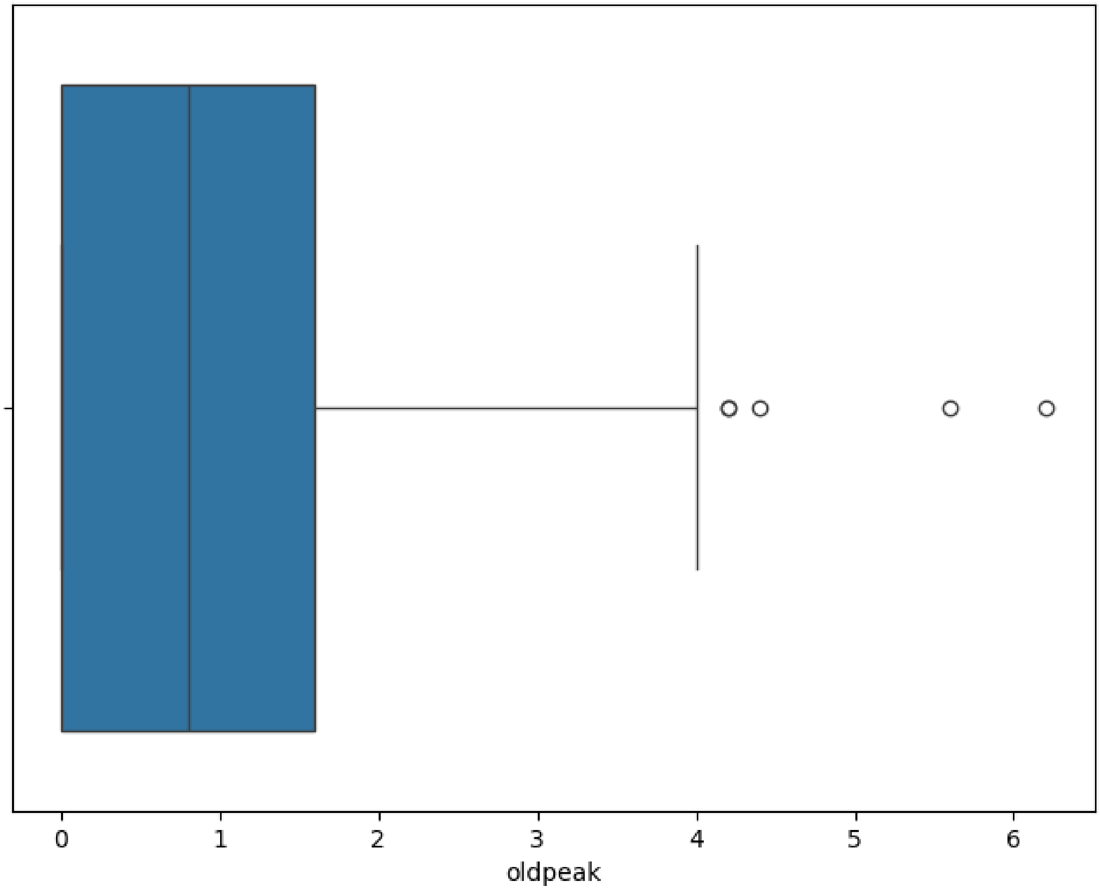
```
health['oldpeak'].describe()
```

[97]:

```
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
```

[98]:

```
f, ax = plt.subplots(figsize=(8, 6))
sns.boxplot(x=health["oldpeak"])
plt.show()
```



[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

[ ]: