In [6]:
```
df.shape
```

Out[6]:
```
(303, 14)
```

In [7]:
```
#We can also try someother way too i.e..
print('The shape of the dataset : ', df.shape)
```

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

In [8]:
```
df.head() #print top 5 rows
```

Out[8]:

	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 [9]:
```
df.info() # gives summary 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.dtypes
```

Out[10]:
```
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 [11]:
```
df.describe() # gives statistical properties of a dataset
```

Out[11]:

	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 [12]:
```
df.columns
```

Out[12]:
```
Index(['age', 'sex', 'cp', 'trestbps', 'chol', 'fbs', 'restecg', 'thalach',
       'exang', 'oldpeak', 'slope', 'ca', 'thal', 'target'],
      dtype='object')
```

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

Out[13]:
```
2
```

In [14]:
```
df['target'].unique()
```

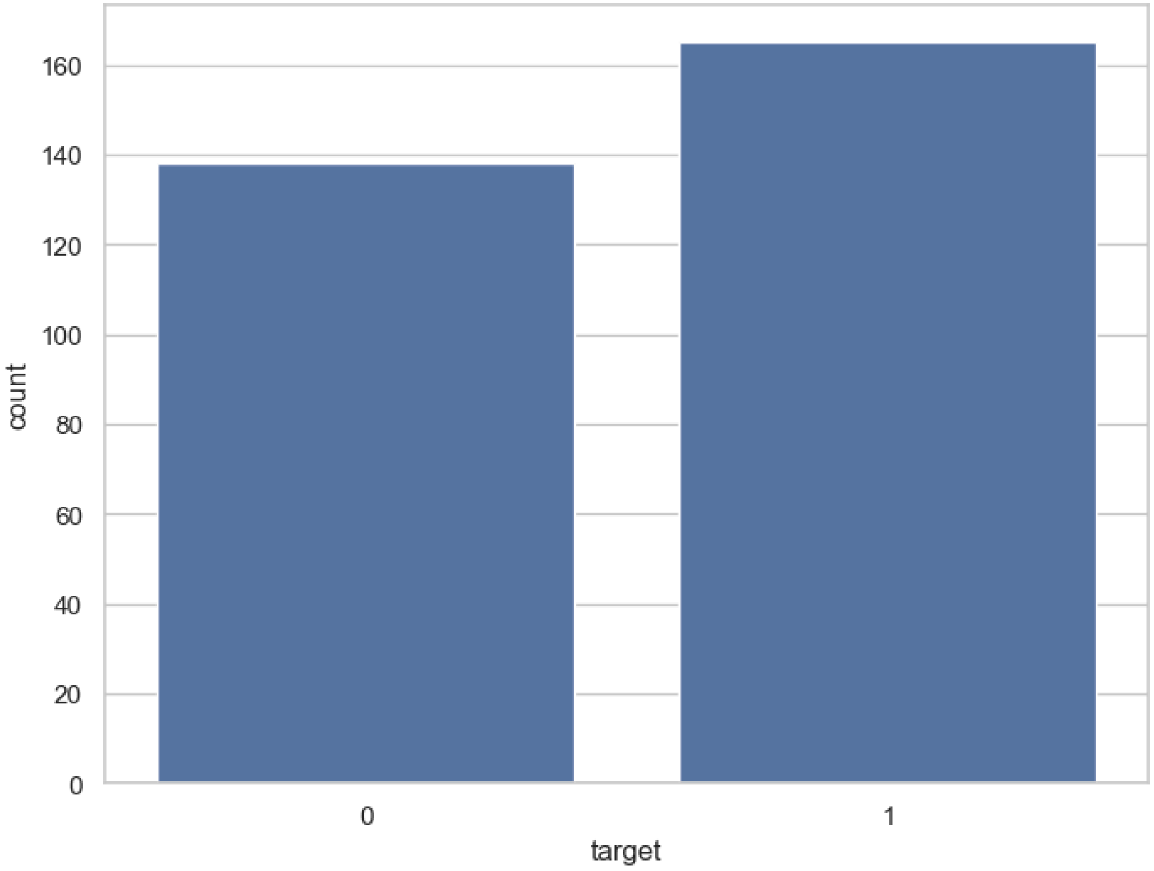
Out[14]:
```
array([1, 0], dtype=int64)
```

In [15]:
```
df['target'].value_counts()
```

Out[15]:
```
target
1    165
0    138
Name: count, dtype: int64
```

In [16]:
```
# value_counts()-counts the occurrences of each unique values.
```

In [21]:
```
f, ax = plt.subplots(figsize=(8, 6))
ax = sns.countplot(x="target", data=df)
plt.show()
```

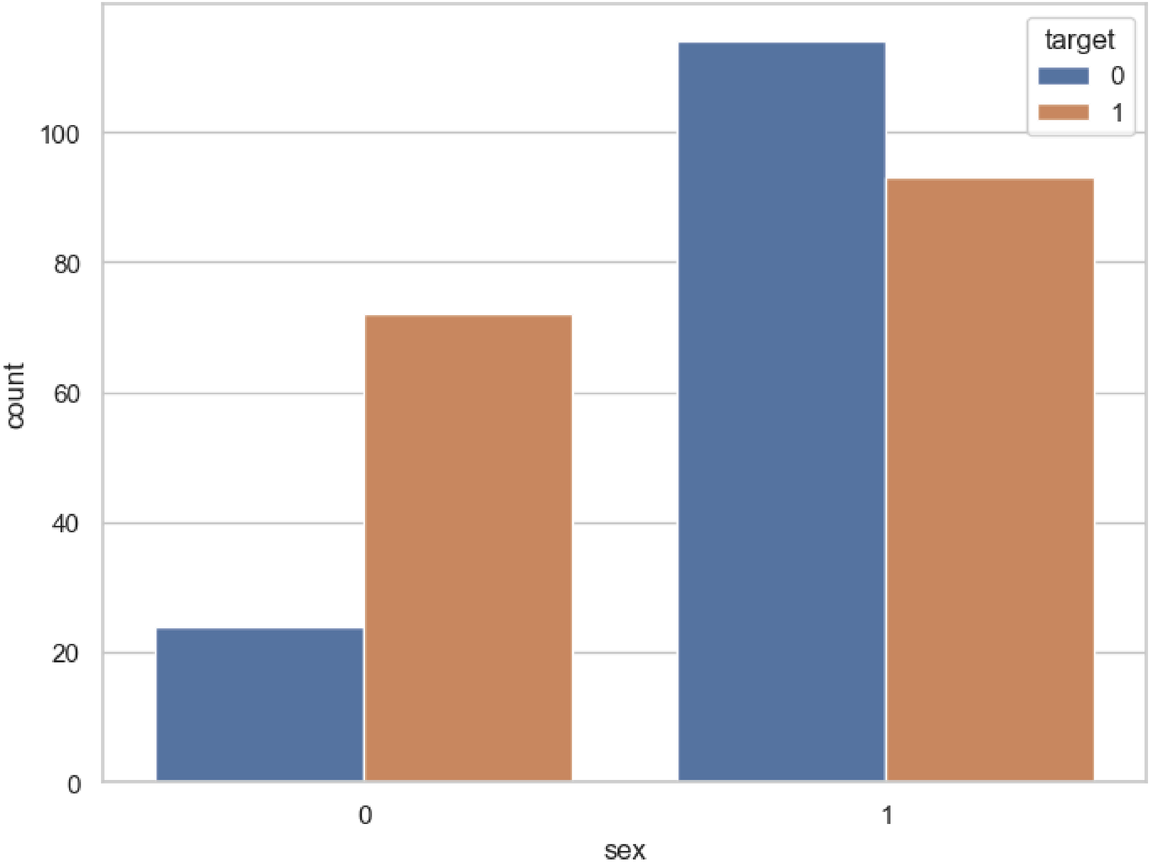


In [18]:
```
df.groupby('sex')['target'].value_counts()
```

Out[18]:
```
sex  target
0    1          72
     0          24
1    0         114
     1          93
Name: count, dtype: int64
```

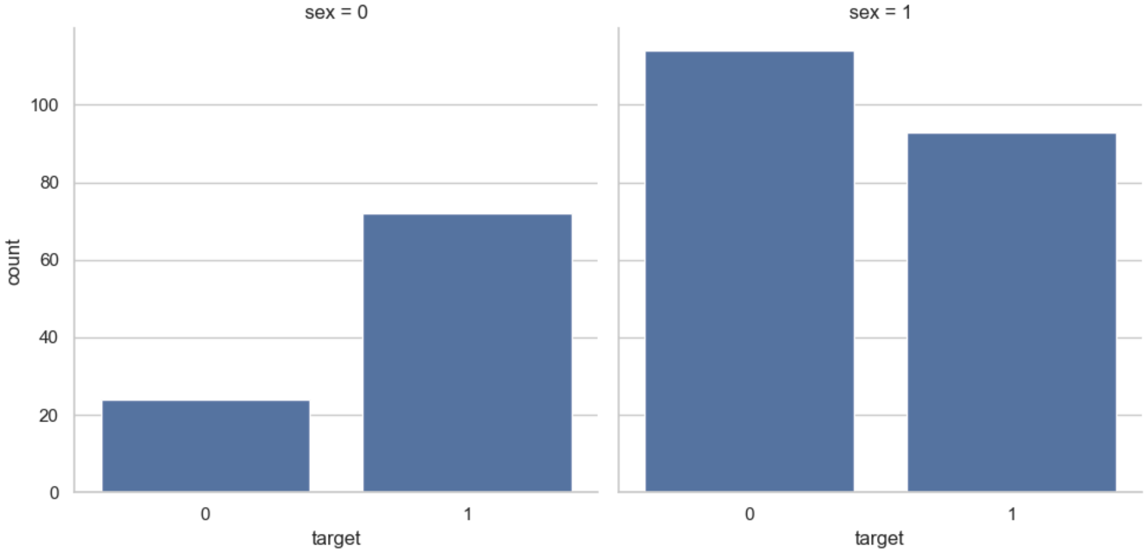
In [19]:
```
# groupby()- Splits the data into groups, applies a function to each group indep
```

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



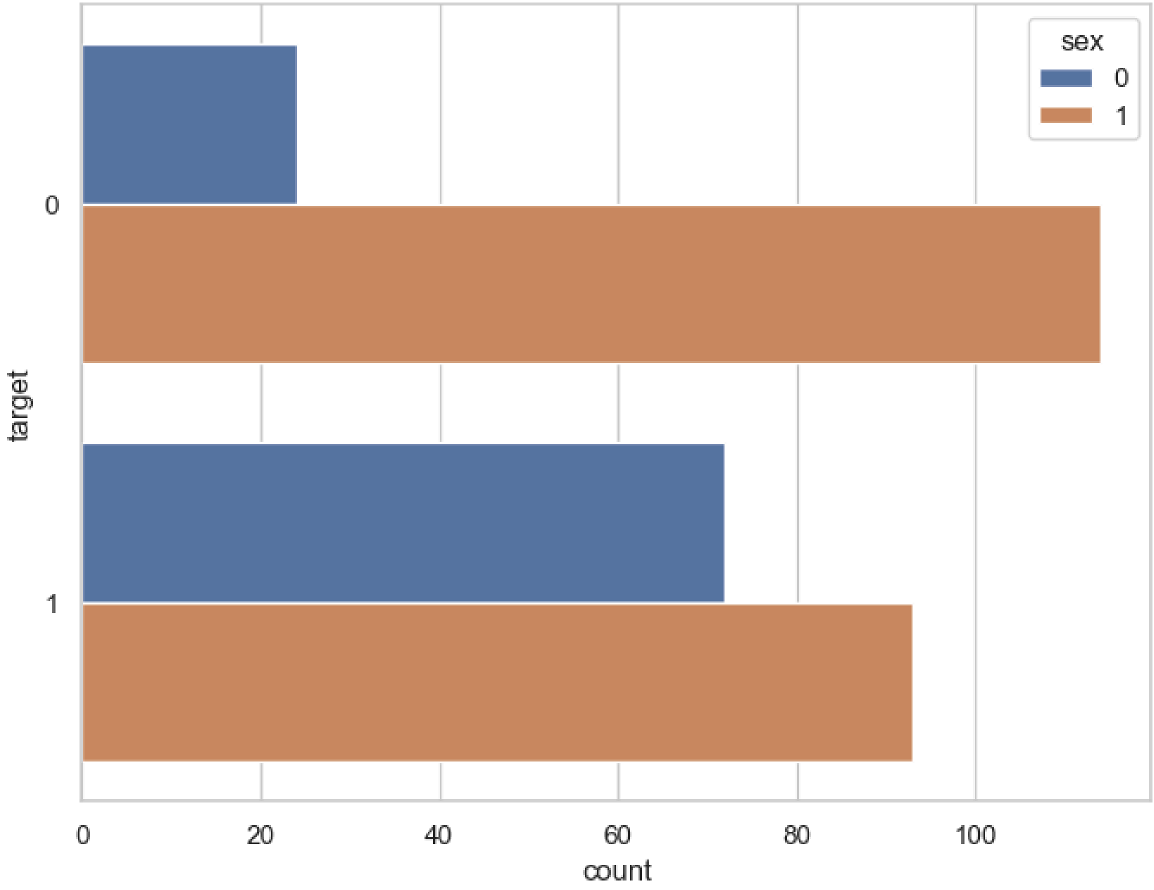
In [25]:
```
# subplots- subplots can display multiple plots side by side.
# countplot- countplot can display the count of observations in each categorical
```

In [26]:
```
ax = sns.catplot(x="target", col="sex", data=df, kind="count", height=5, aspect=
```

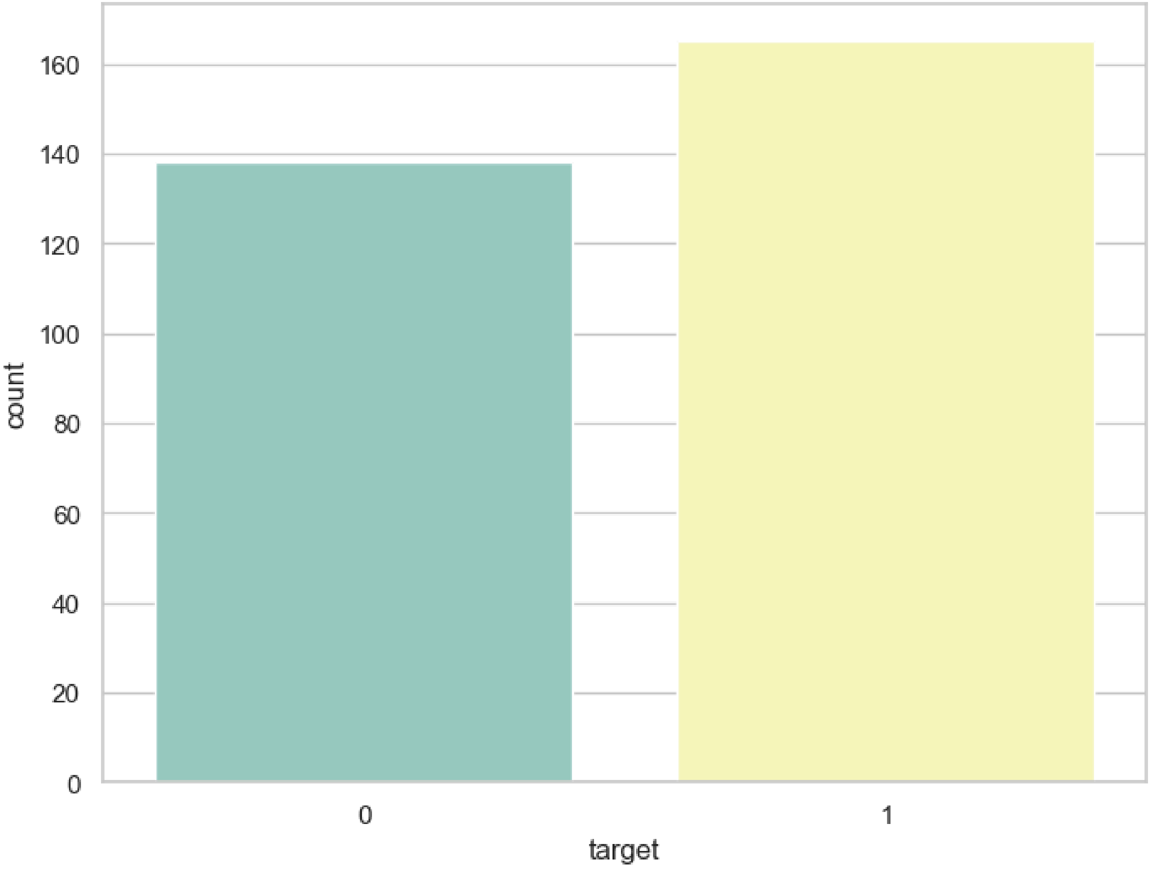


In [27]:
```
# catplot- A categorical plot (catplot) is a multi-plot function in Seaborn that
```

In [28]:
```
f, ax = plt.subplots(figsize=(8, 6))
ax = sns.countplot(y="target", hue="sex", data=df)
plt.show()
```

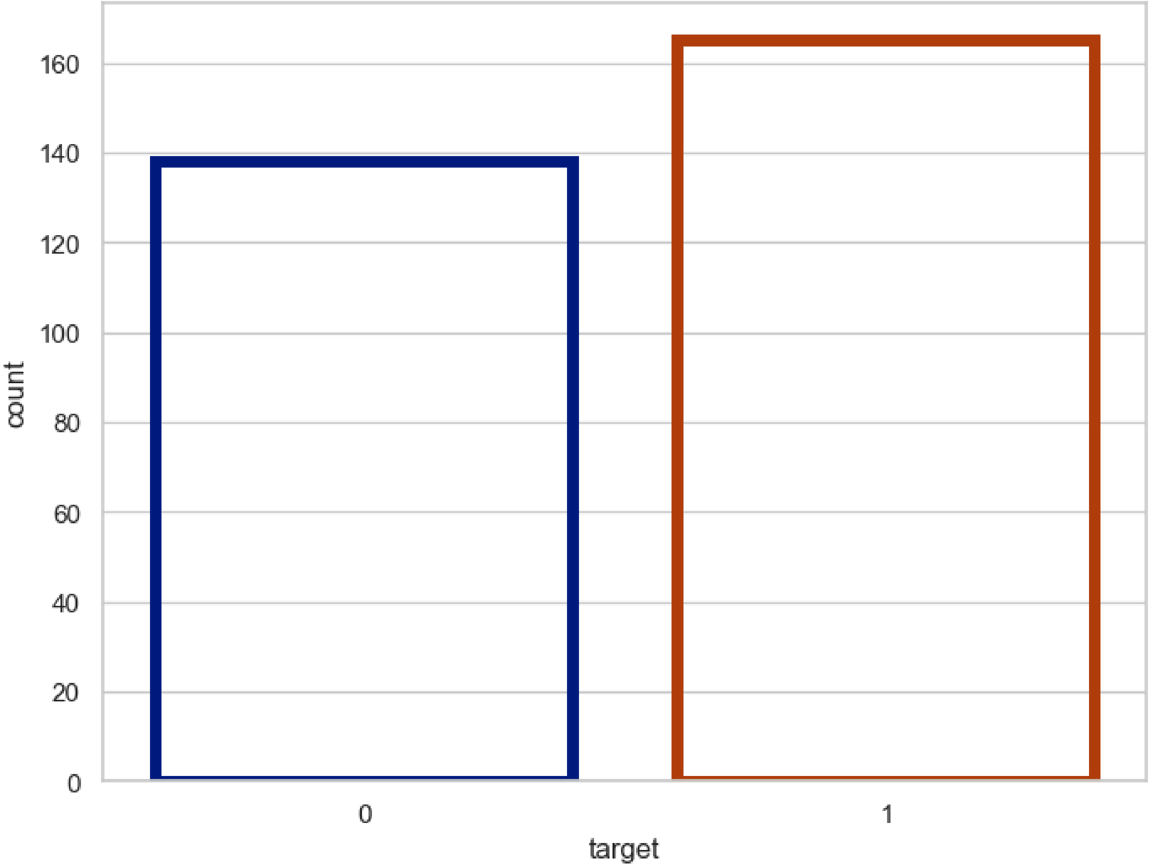


In [29]:
```
f, ax = plt.subplots(figsize=(8, 6))
ax = sns.countplot(x="target", data=df, palette="Set3")
plt.show()
```



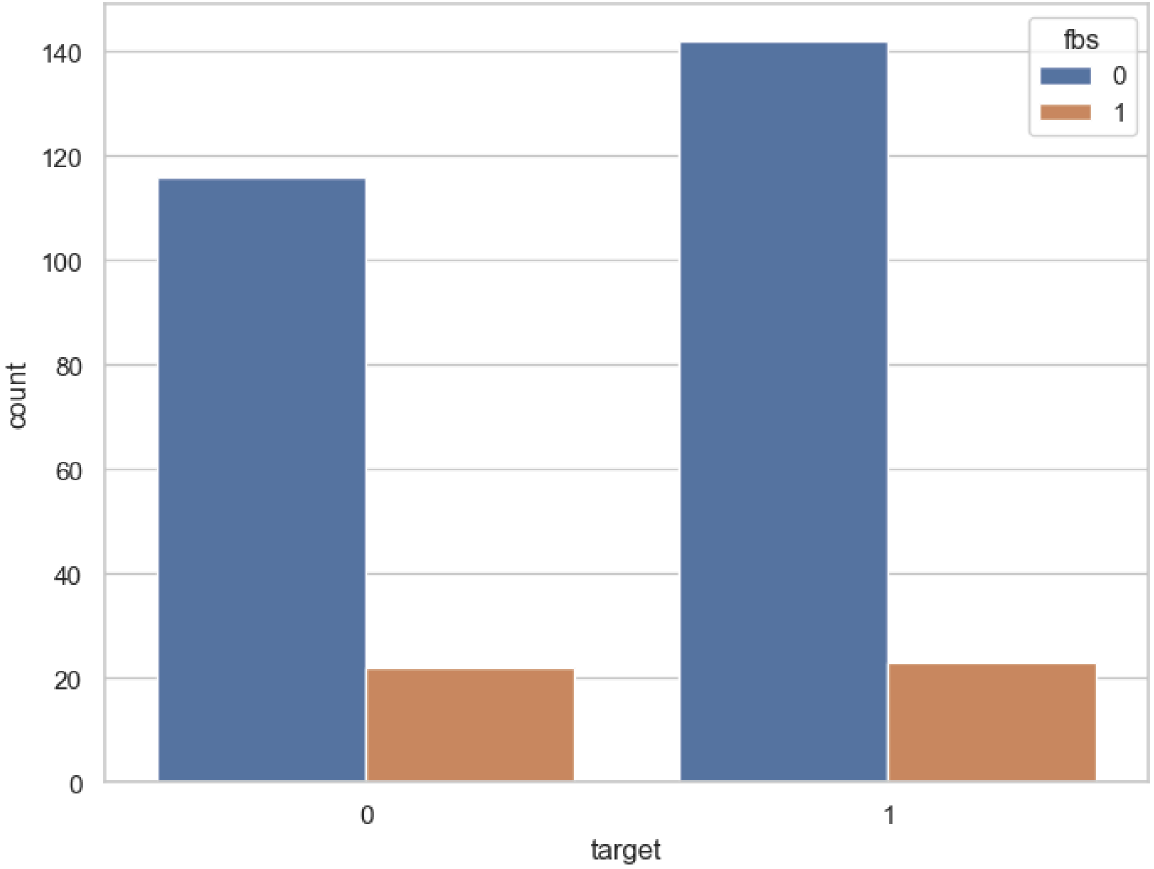
In [30]:

```
f, ax = plt.subplots(figsize=(8, 6))
ax = sns.countplot(x="target", data=df, facecolor=(0, 0, 0, 0), linewidth=5, edg
plt.show()
```

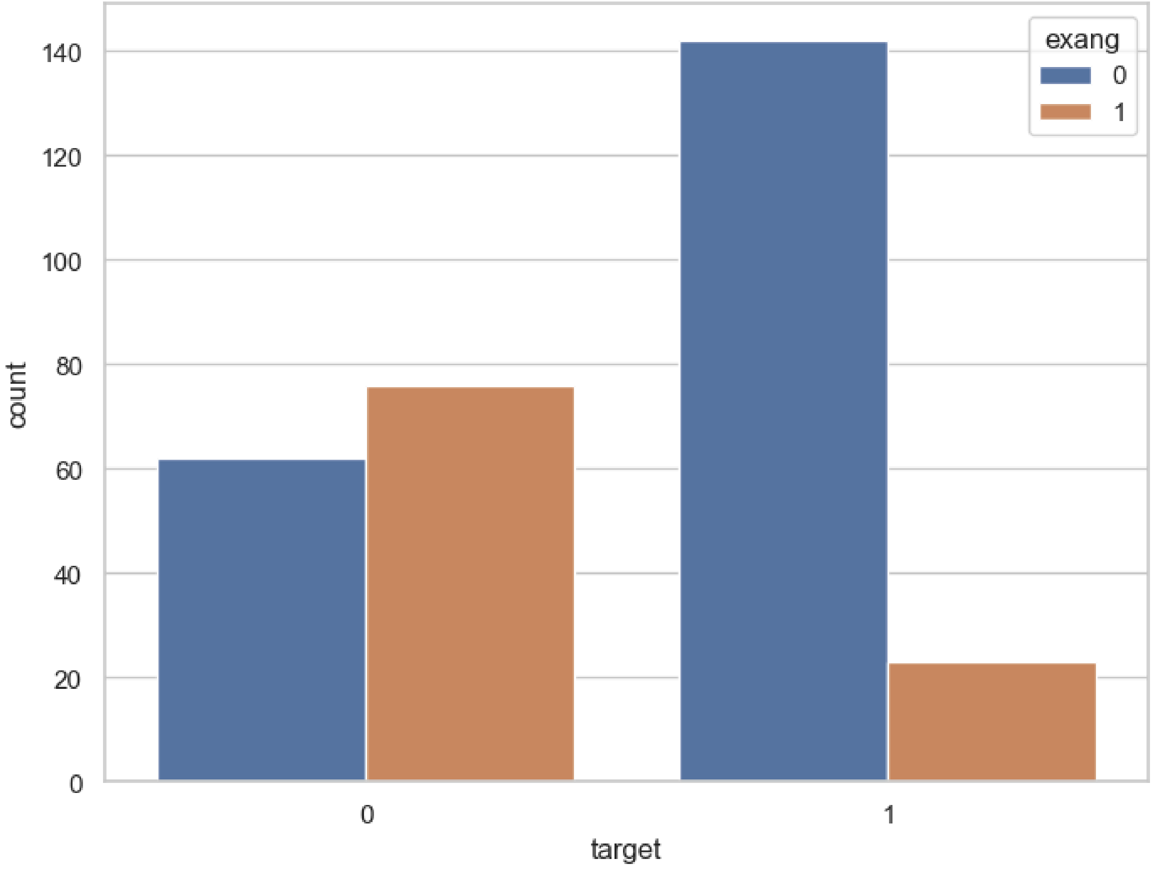


In [31]:

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



```
In [32]: f, ax = plt.subplots(figsize=(8, 6))
         ax = sns.countplot(x="target", hue="exang", data=df)
         plt.show()
```



```
In [33]: correlation = df.corr()
```

In [34]:

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

Out[34]:

```
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 [35]:

```
df['cp'].nunique()
```

Out[35]:

```
4
```

In [36]:

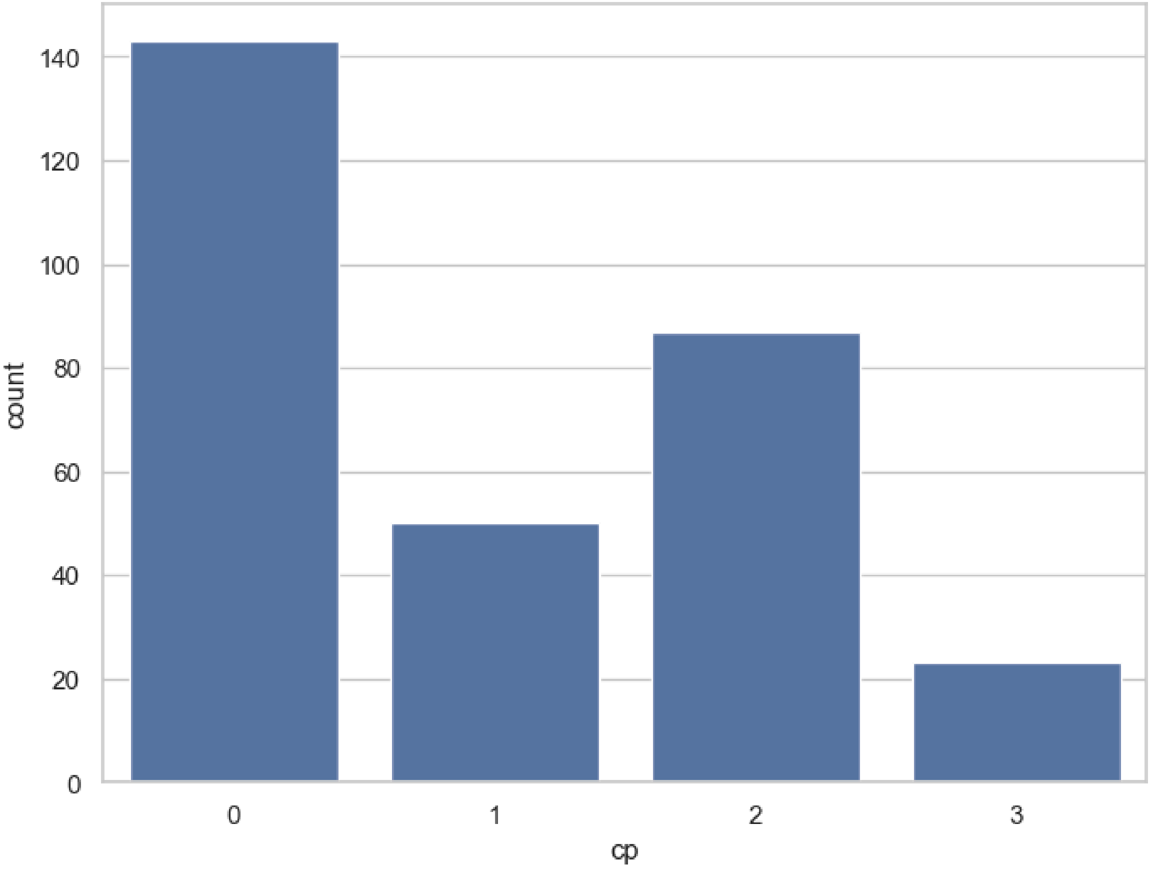
```
df['cp'].value_counts()
```

Out[36]:

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

In [37]:

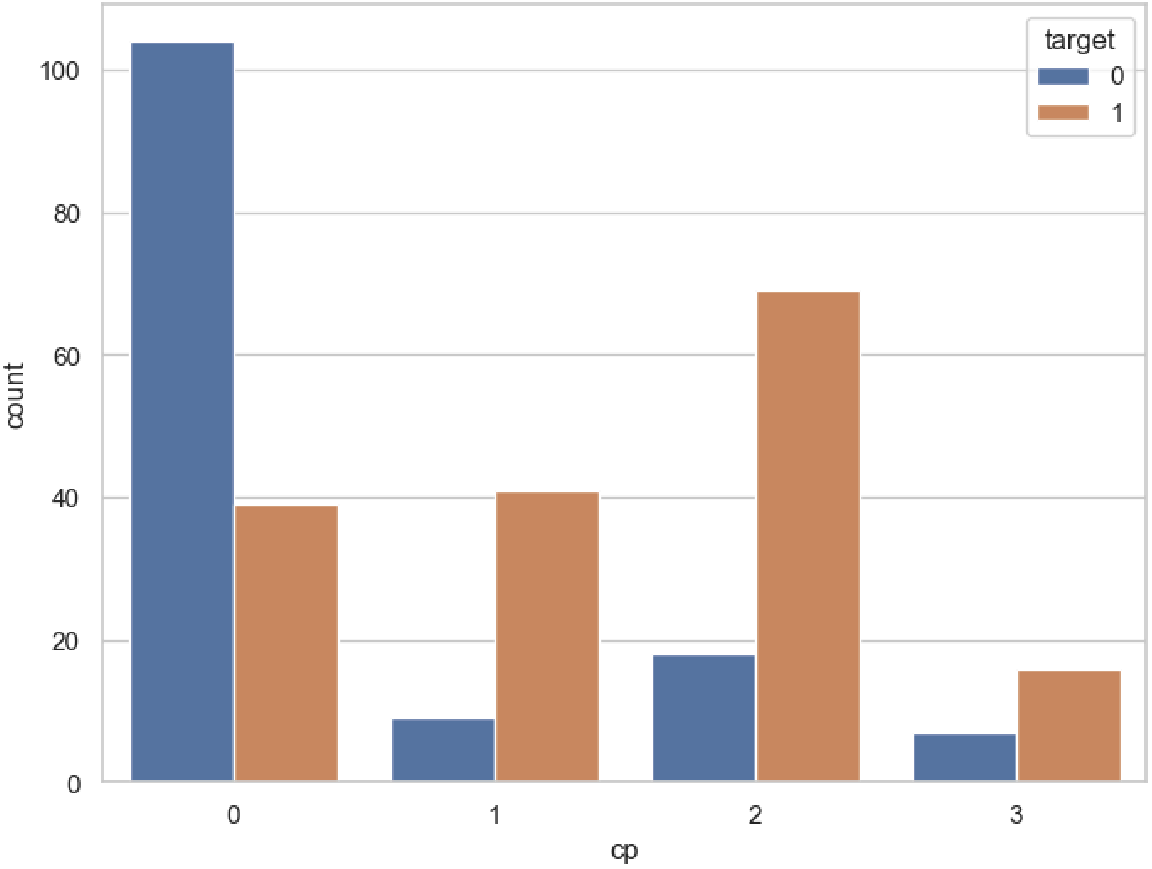
```
f, ax = plt.subplots(figsize=(8, 6))
ax = sns.countplot(x="cp", data=df)
plt.show()
```



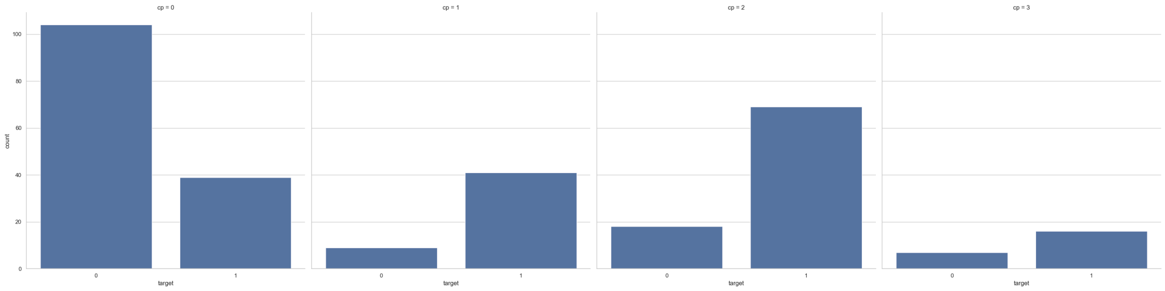
```
In [38]: df.groupby('cp')['target'].value_counts()
```

```
Out[38]: 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 [39]: f, ax = plt.subplots(figsize=(8, 6))
         ax = sns.countplot(x="cp", hue="target", data=df)
         plt.show()
```



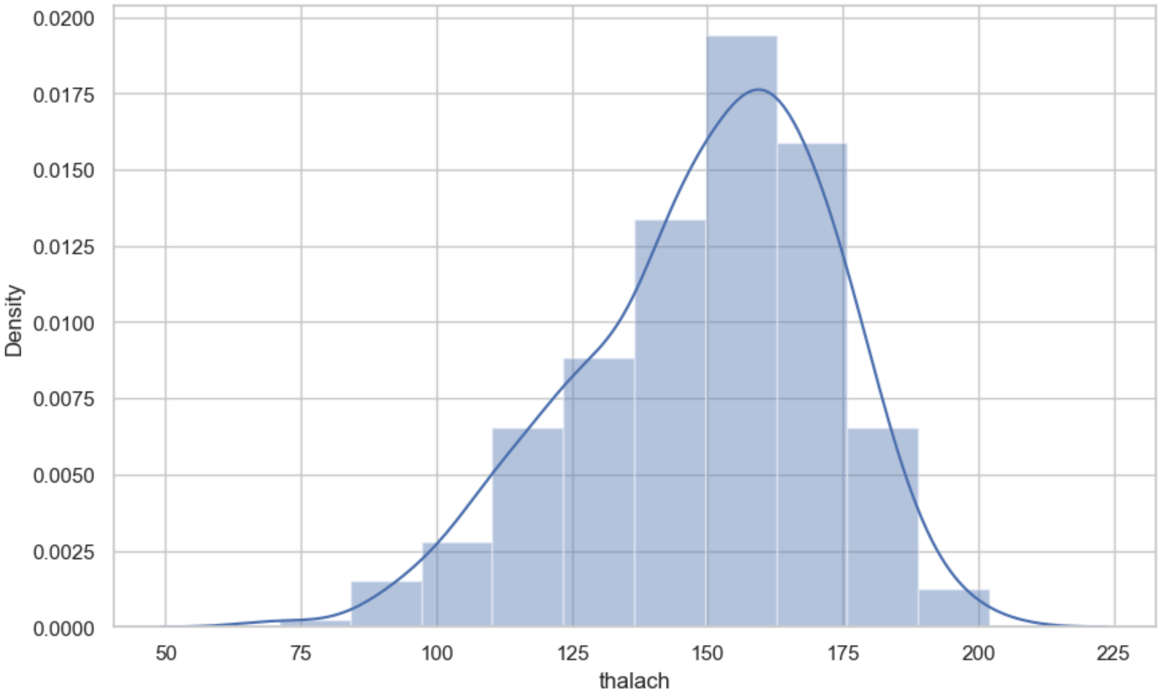
In [40]:
```
ax = sns.catplot(x="target", col="cp", data=df, kind="count", height=8, aspect=1
```



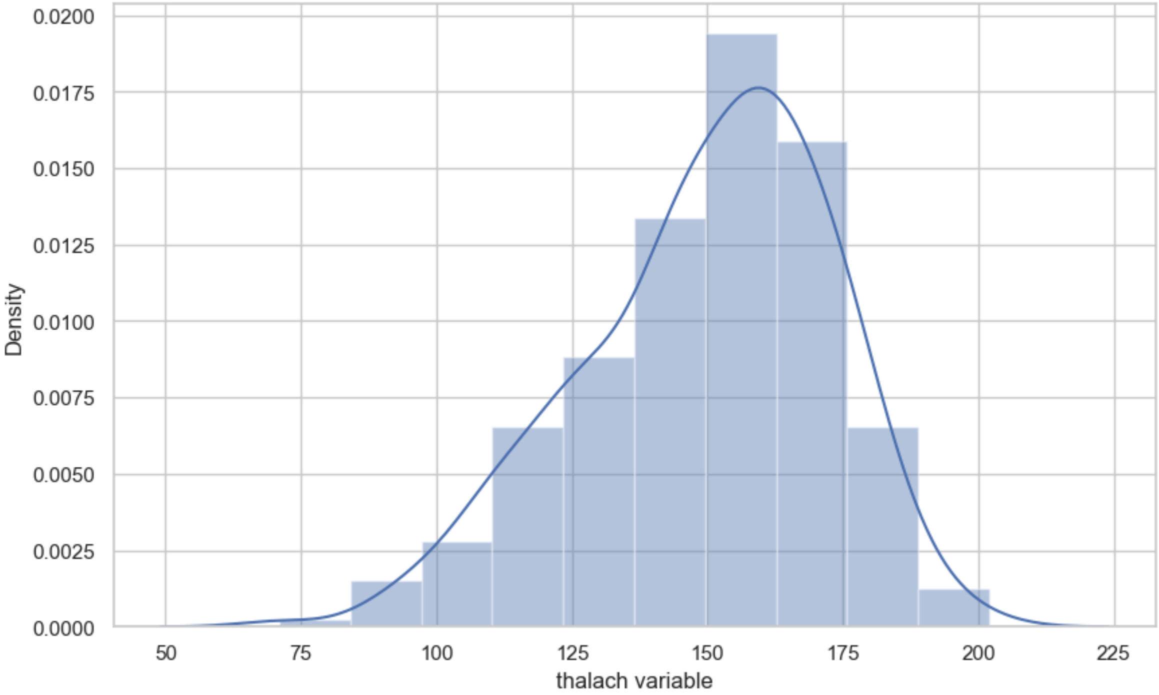
In [41]:
```
df['thalach'].nunique()
```

Out[41]:
```
91
```

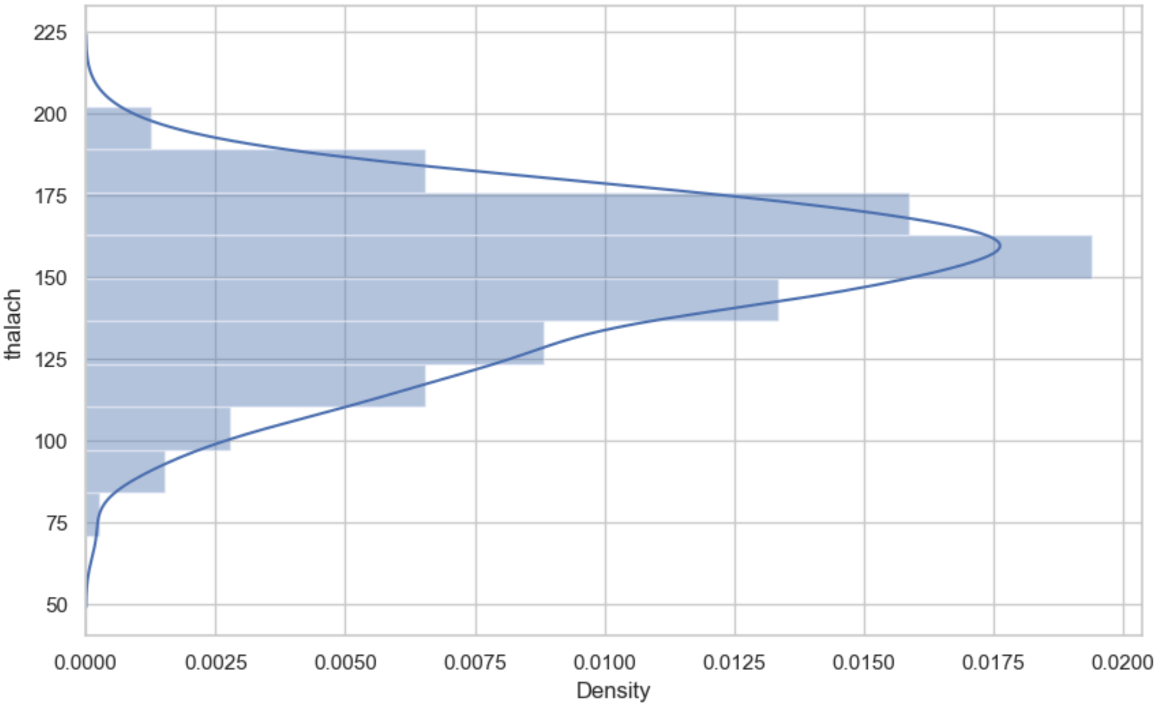
In [42]:
```
f, ax = plt.subplots(figsize=(10,6))
x = df['thalach']
ax = sns.distplot(x, bins=10)
plt.show()
```



In [43]:
```
f, ax = plt.subplots(figsize=(10,6))
x = df['thalach']
x = pd.Series(x, name="thalach variable")
ax = sns.distplot(x, bins=10)
plt.show()
```

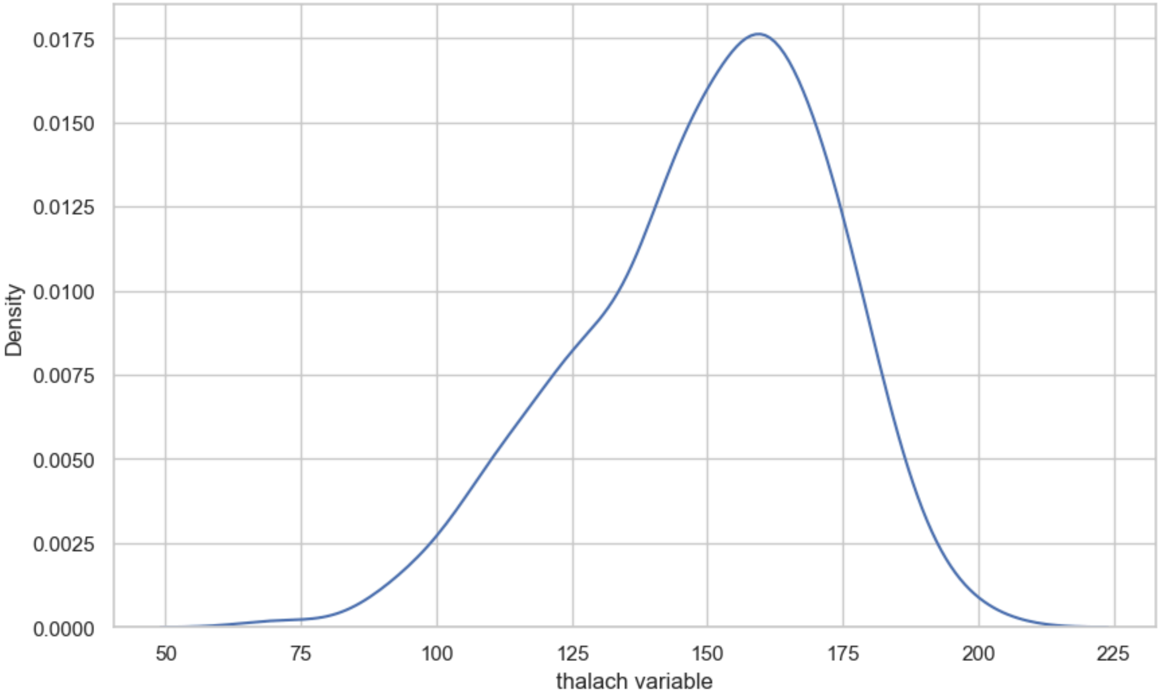


In [44]:
```
f, ax = plt.subplots(figsize=(10,6))
x = df['thalach']
ax = sns.distplot(x, bins=10, vertical=True)
plt.show()
```



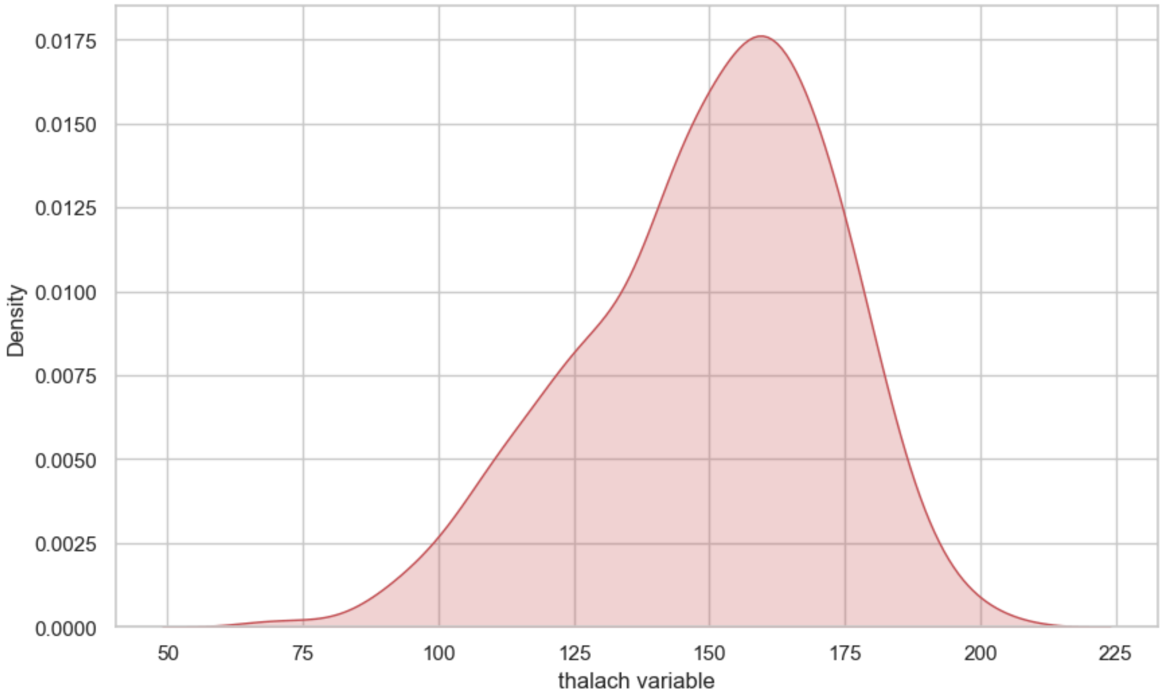
In [45]:

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

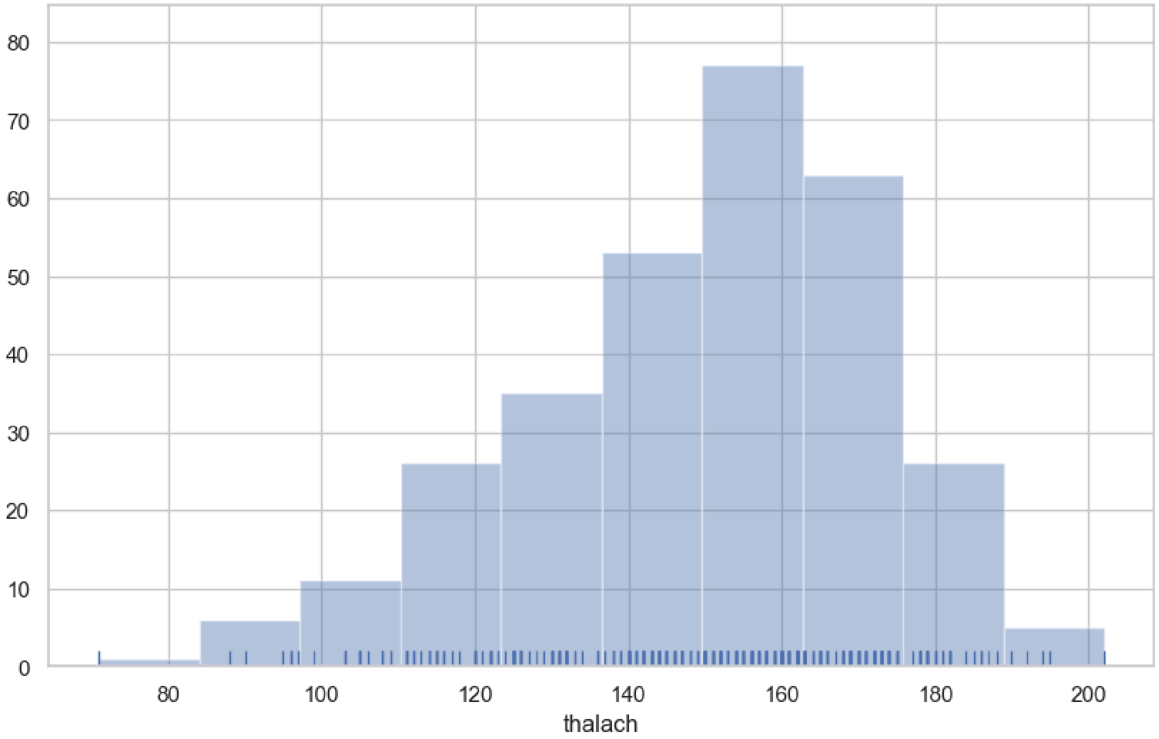


In [46]:

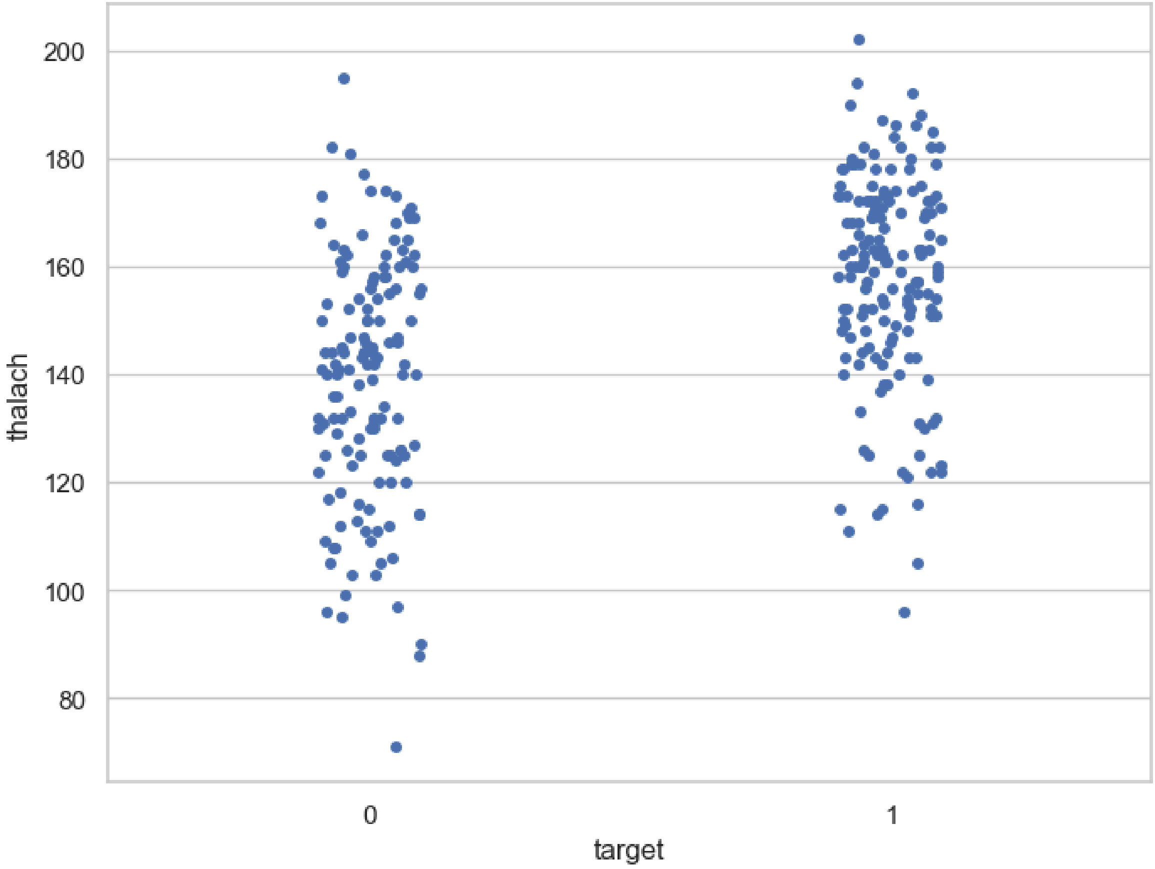
```
f, ax = plt.subplots(figsize=(10,6))
x = df['thalach']
x = pd.Series(x, name="thalach variable")
ax = sns.kdeplot(x, shade=True, color='r')
plt.show()
```



In [47]:
```
f, ax = plt.subplots(figsize=(10,6))
x = df['thalach']
ax = sns.distplot(x, kde=False, rug=True, bins=10)
plt.show()
```

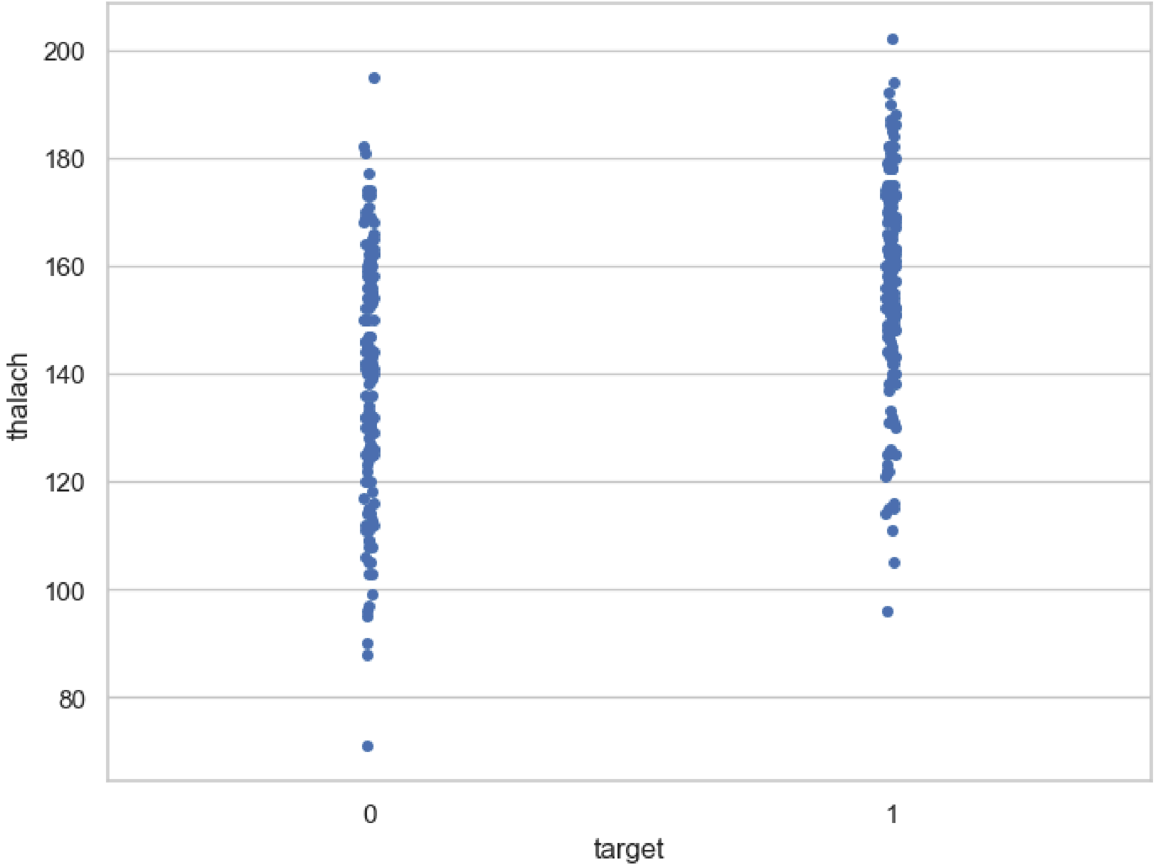


In [48]:
```
f, ax = plt.subplots(figsize=(8, 6))
sns.stripplot(x="target", y="thalach", data=df)
plt.show()
```



In [49]:

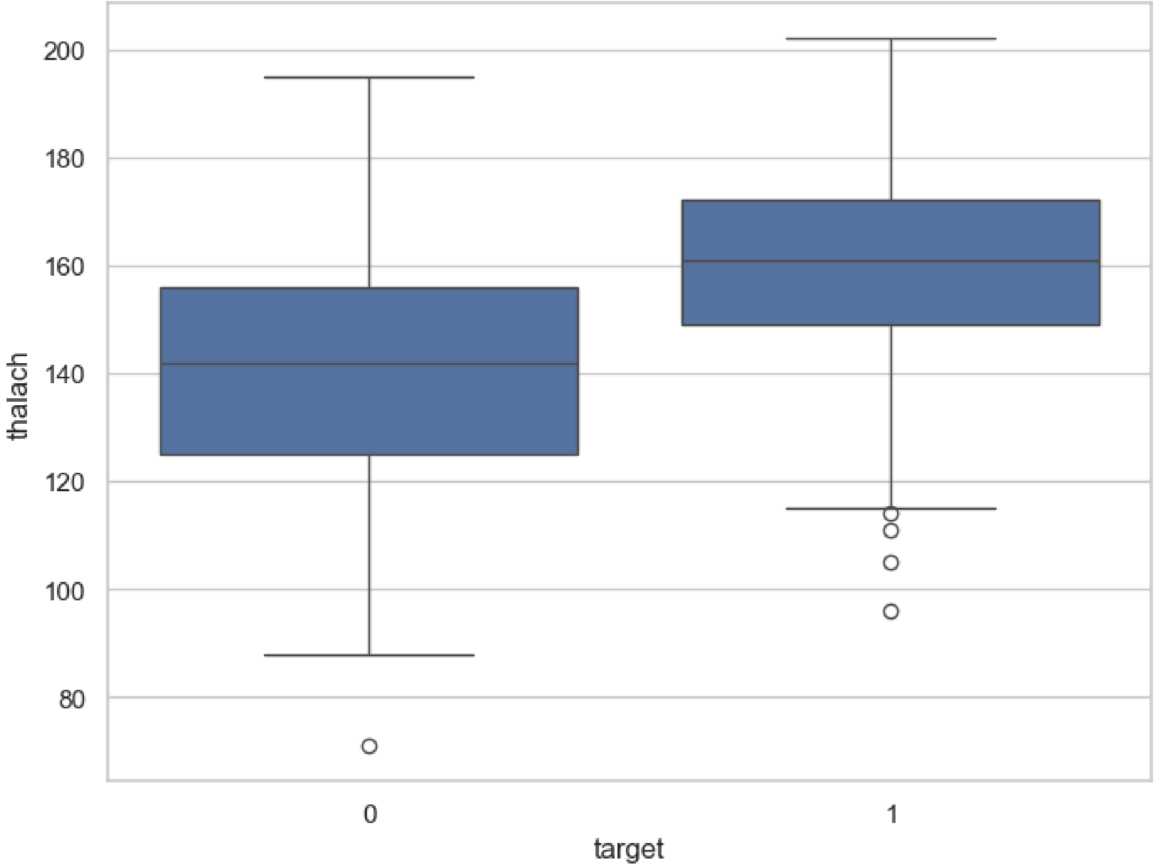
```
f, ax = plt.subplots(figsize=(8, 6))
sns.stripplot(x="target", y="thalach", data=df, jitter = 0.01)
plt.show()
```



In [50]:

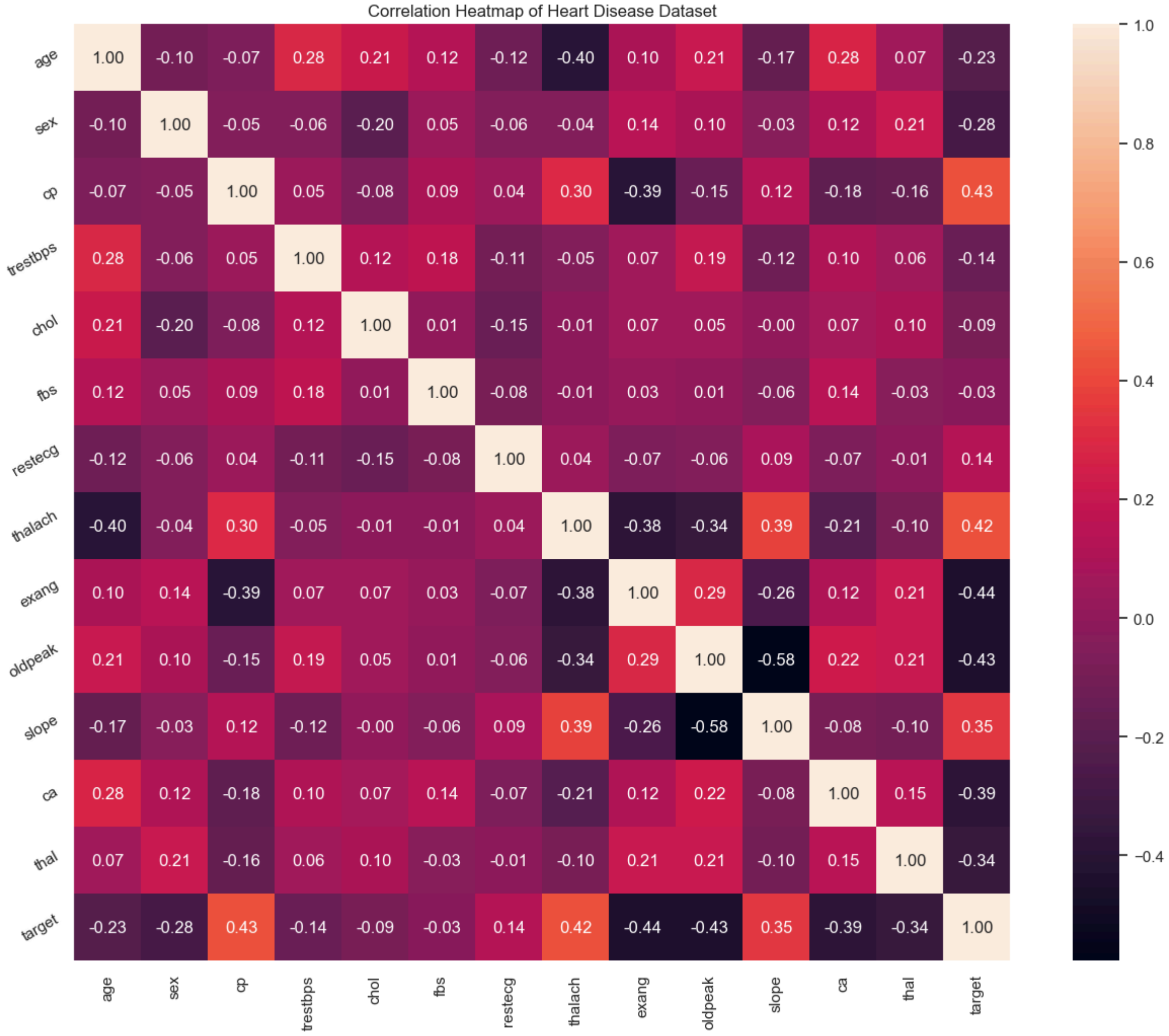
```
f, ax = plt.subplots(figsize=(8, 6))
sns.boxplot(x="target", y="thalach", data=df)
```

```
plt.show()
```



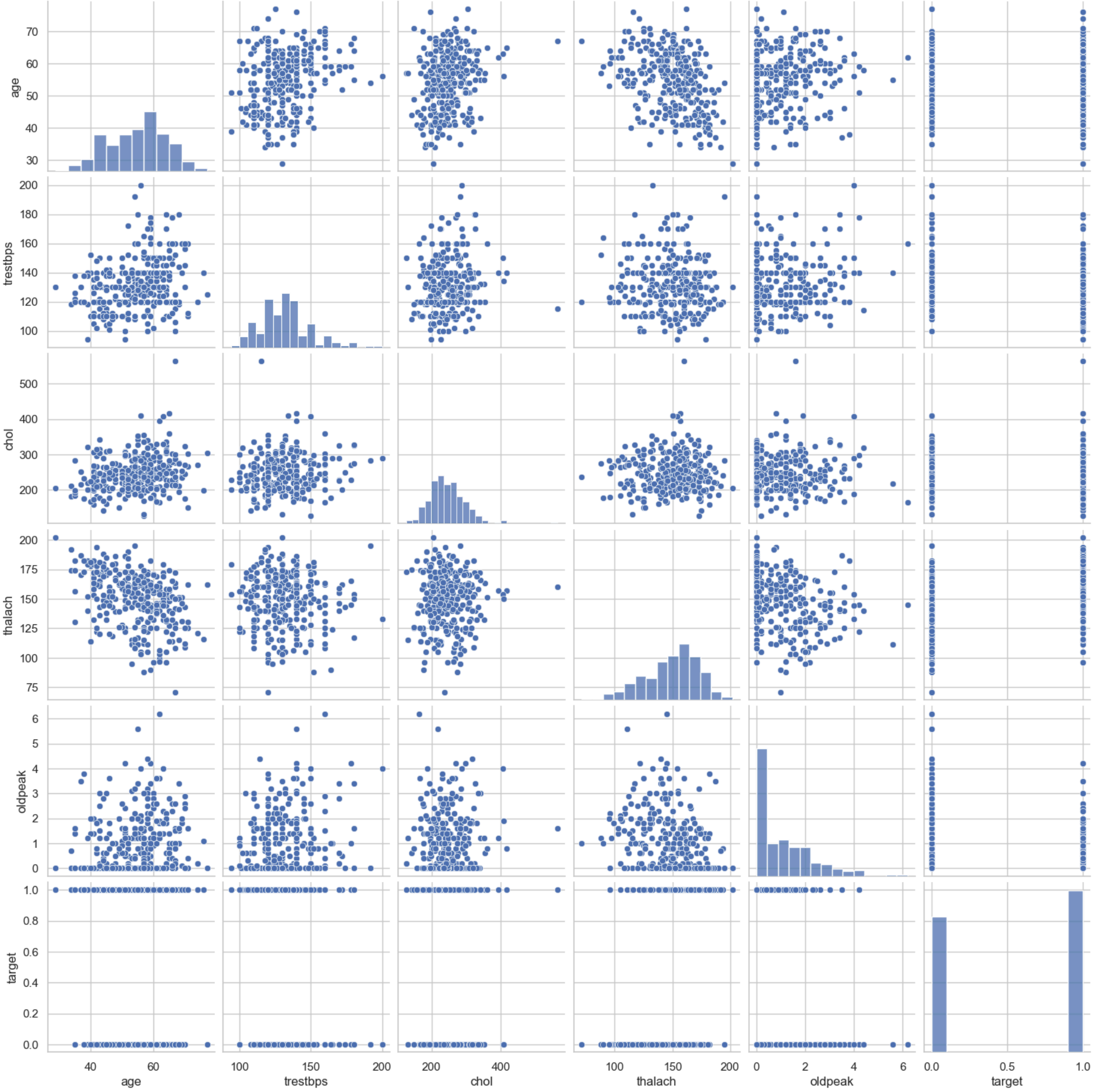
In [51]:
```
plt.figure(figsize=(16,12))
plt.title('Correlation Heatmap of Heart Disease Dataset')
a = sns.heatmap(correlation, square=True, annot=True, fmt='.2f', linecolor='whit
a.set_xticklabels(a.get_xticklabels(), rotation=90)
a.set_yticklabels(a.get_yticklabels(), rotation=30)
plt.show()
```

Correlation Heatmap of Heart Disease Dataset



	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

```
In [52]: num_var = ['age', 'trestbps', 'chol', 'thalach', 'oldpeak', 'target' ]
         sns.pairplot(df[num_var], kind='scatter', diag_kind='hist')
         plt.show()
```



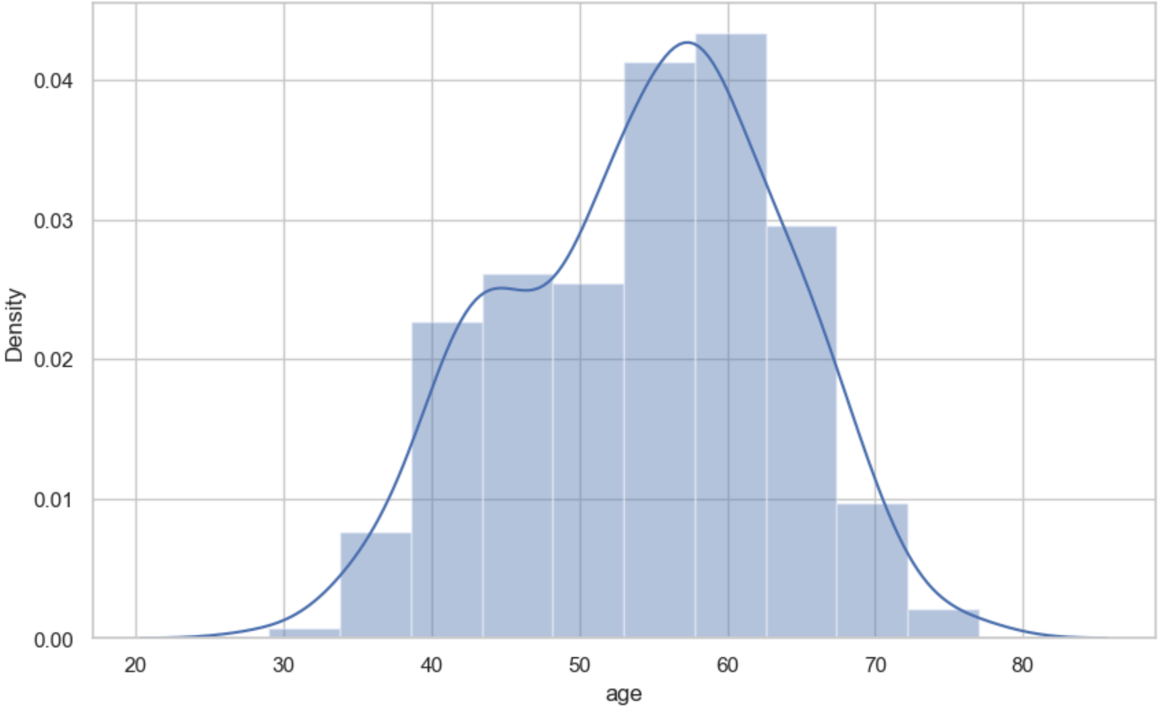
In [53]:
```
df['age'].nunique()
```

Out[53]:
```
41
```

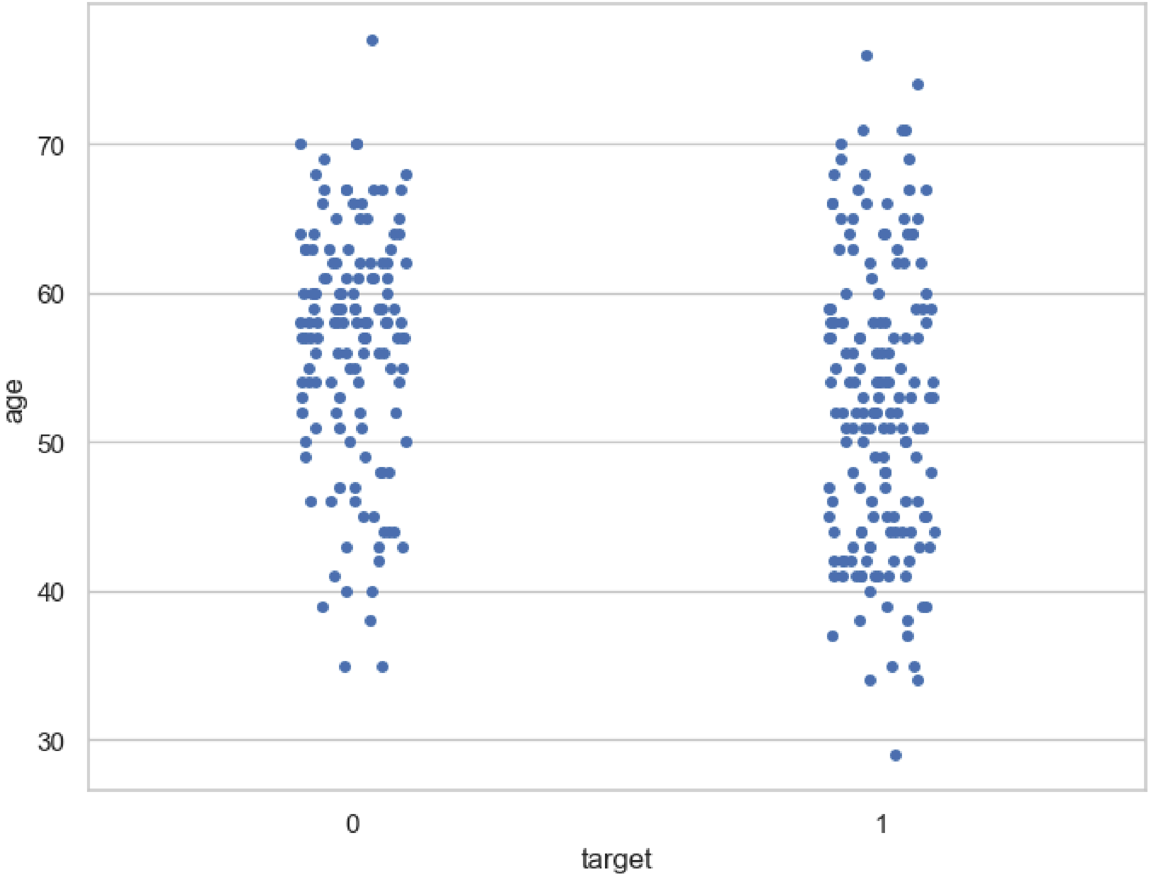
In [54]:
```
df['age'].describe()
```

Out[54]:
```
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
```

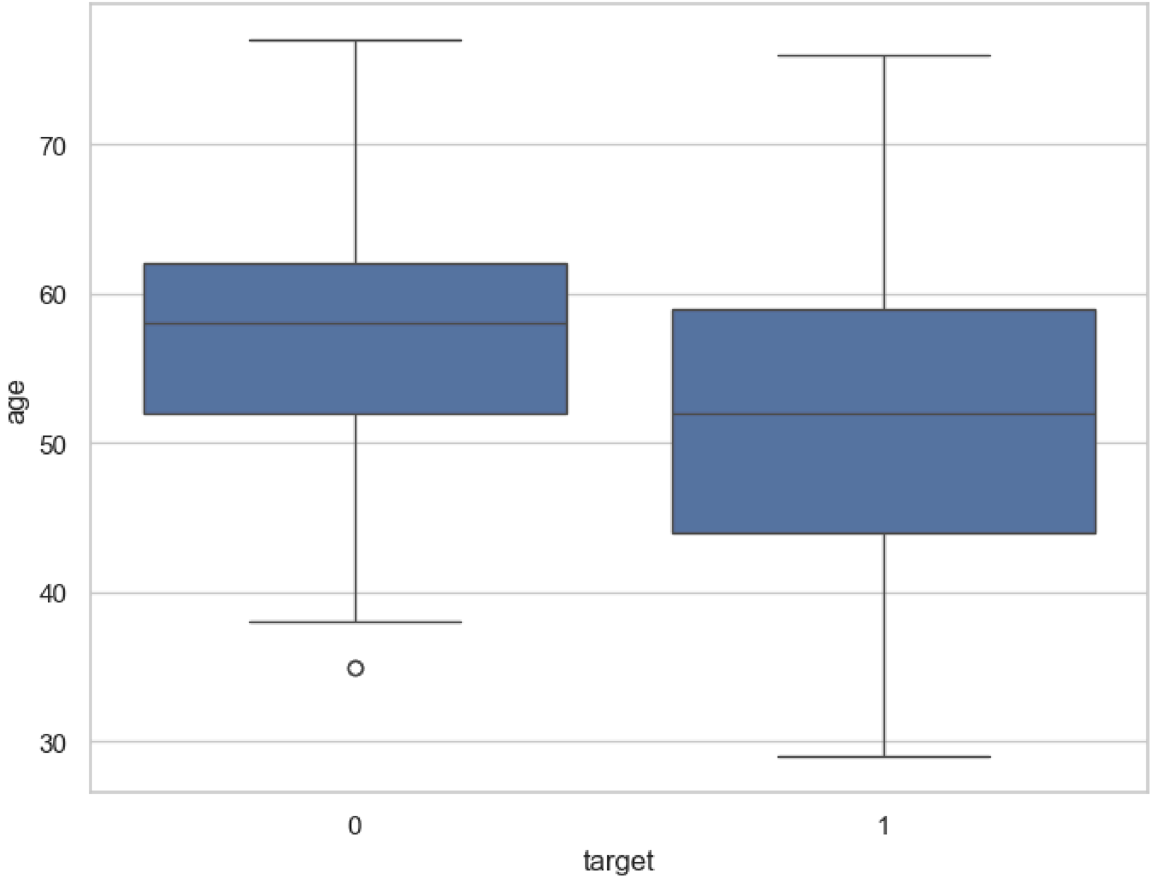
In [55]:
```
f, ax = plt.subplots(figsize=(10,6))
x = df['age']
ax = sns.distplot(x, bins=10)
plt.show()
```



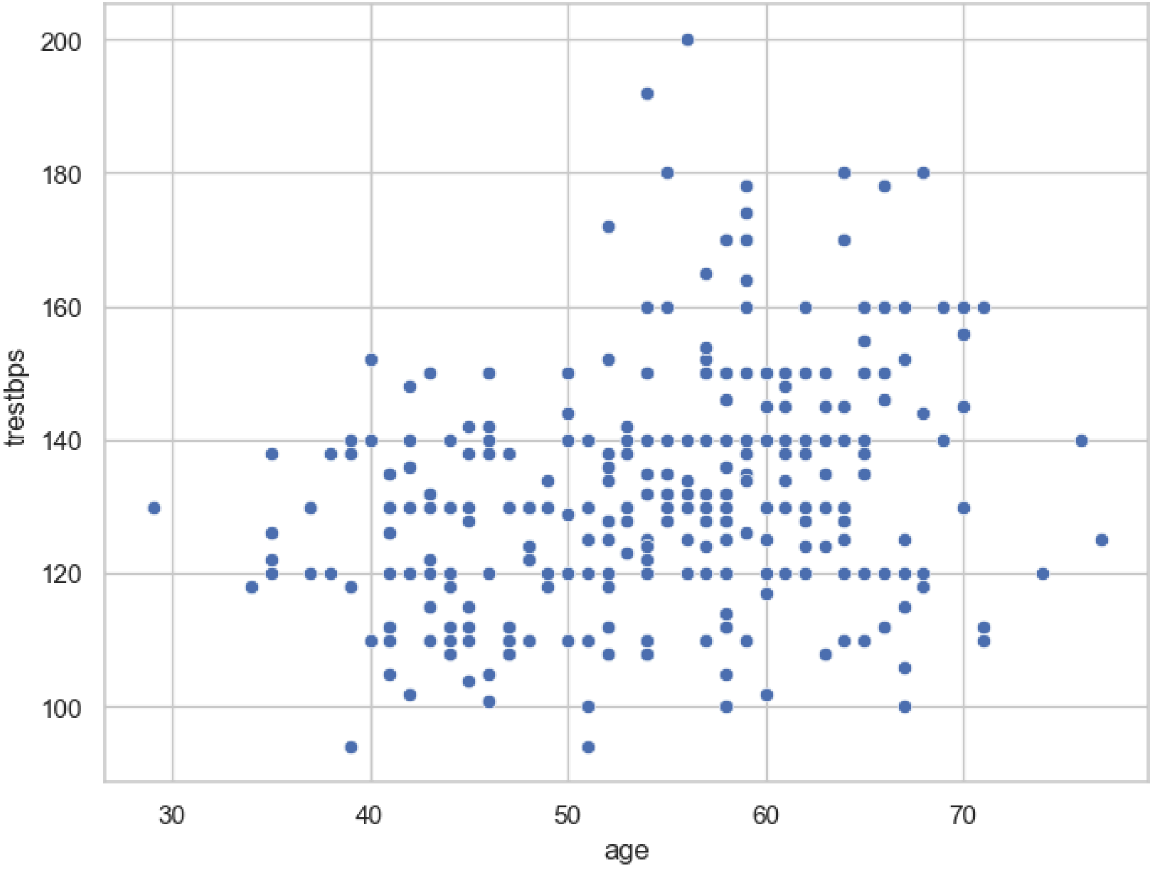
In [56]:
```
f, ax = plt.subplots(figsize=(8, 6))
sns.stripplot(x="target", y="age", data=df)
plt.show()
```



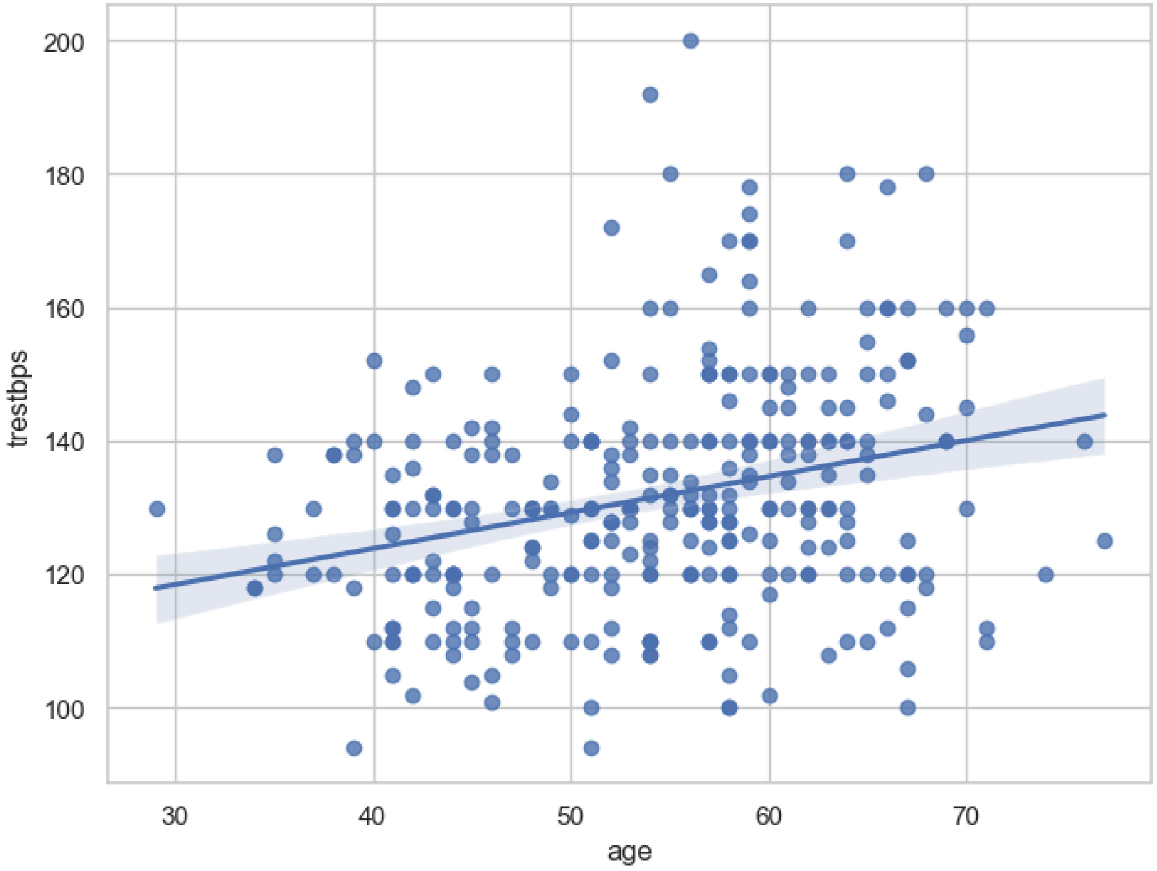
In [57]:
```
f, ax = plt.subplots(figsize=(8, 6))
sns.boxplot(x="target", y="age", data=df)
plt.show()
```



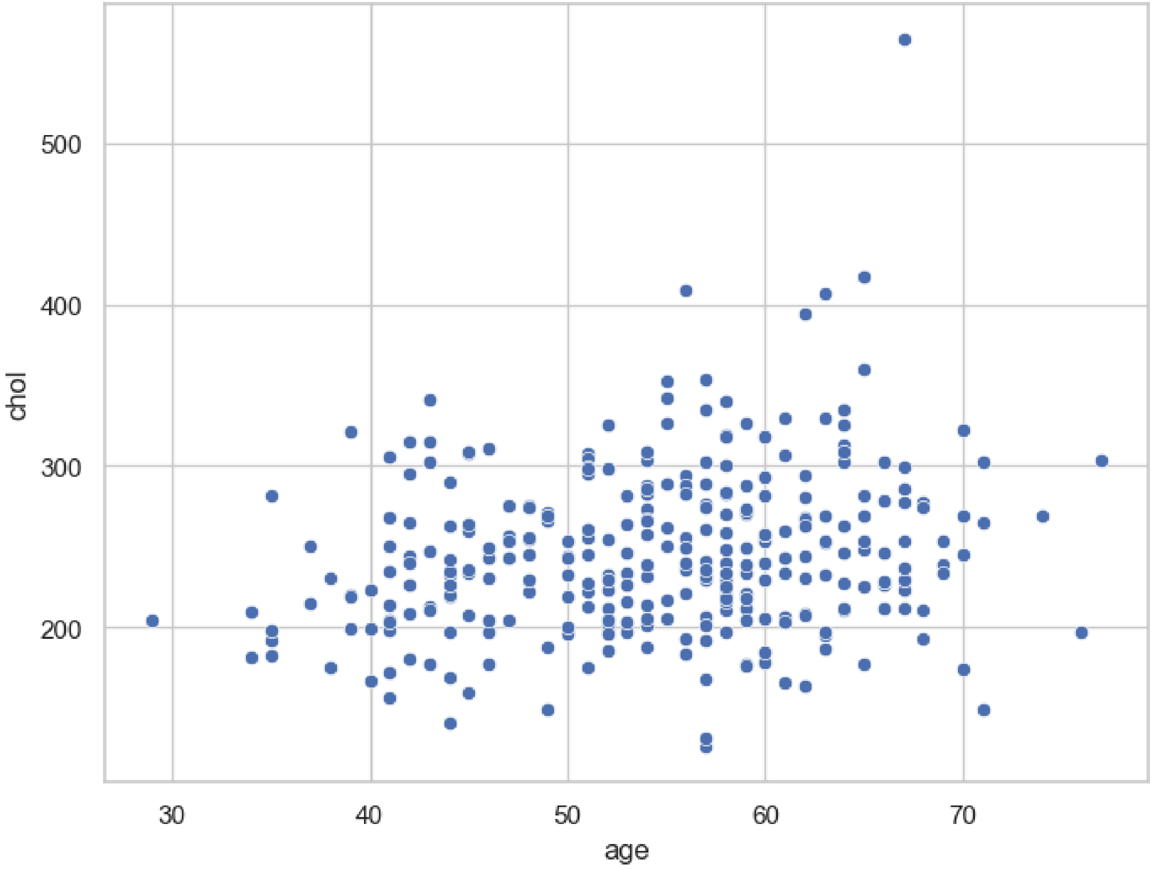
```
In [58]: f, ax = plt.subplots(figsize=(8, 6))
         ax = sns.scatterplot(x="age", y="trestbps", data=df)
         plt.show()
```



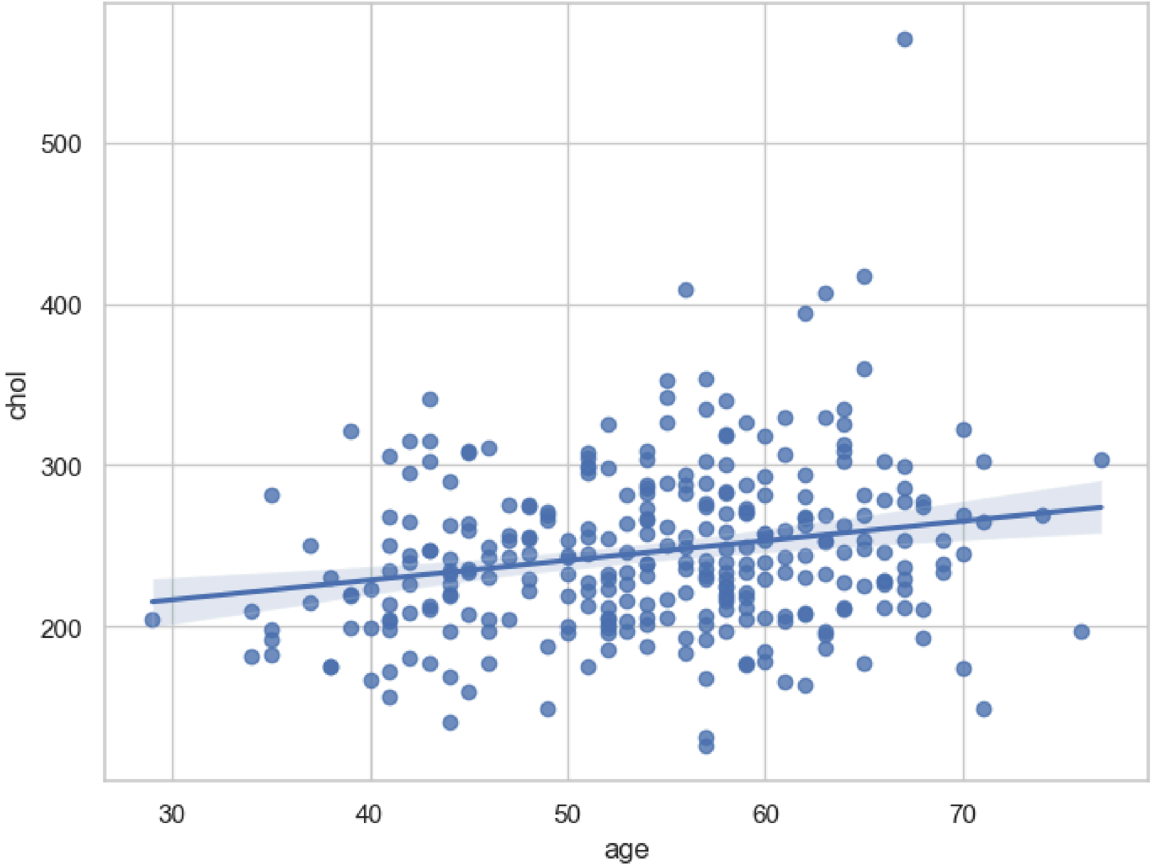
In [59]:
```
f, ax = plt.subplots(figsize=(8, 6))
ax = sns.regplot(x="age", y="trestbps", data=df)
plt.show()
```



In [60]:
```
f, ax = plt.subplots(figsize=(8, 6))
ax = sns.scatterplot(x="age", y="chol", data=df)
plt.show()
```

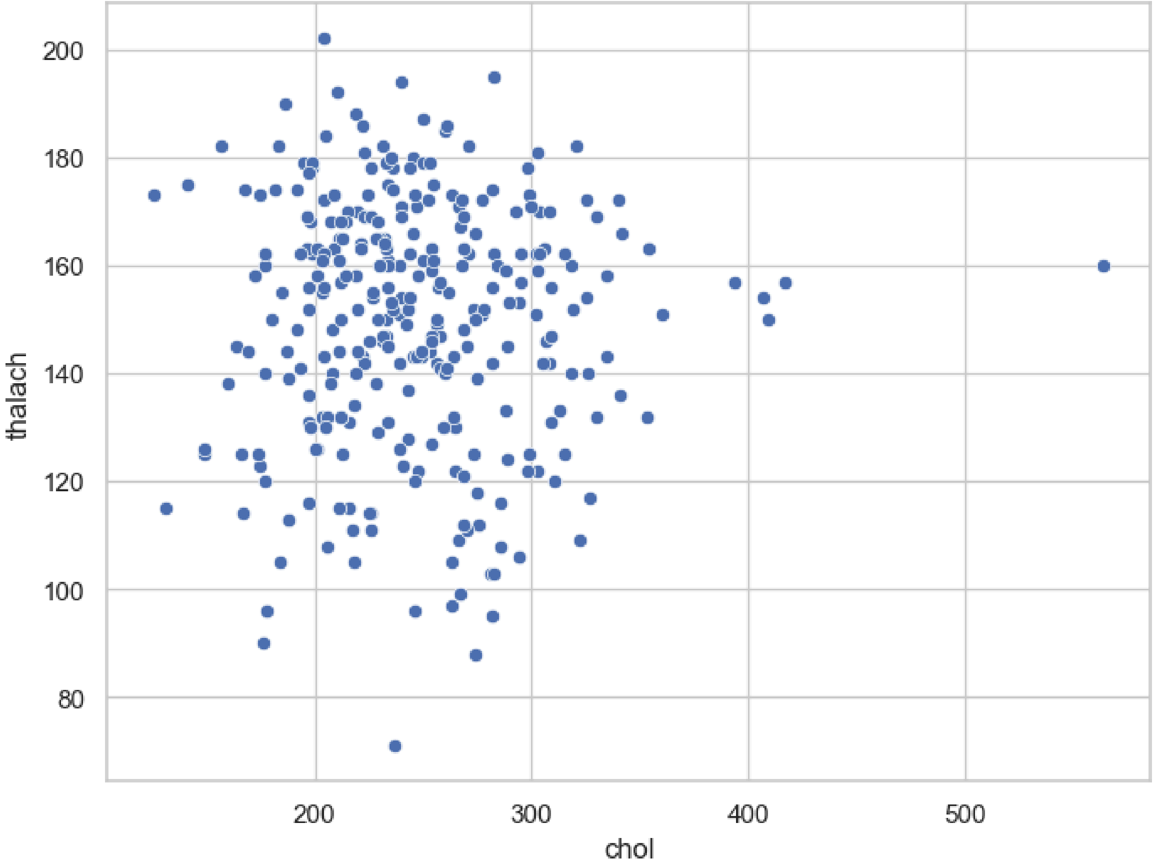


```
In [61]: f, ax = plt.subplots(figsize=(8, 6))
         ax = sns.regplot(x="age", y="chol", data=df)
         plt.show()
```

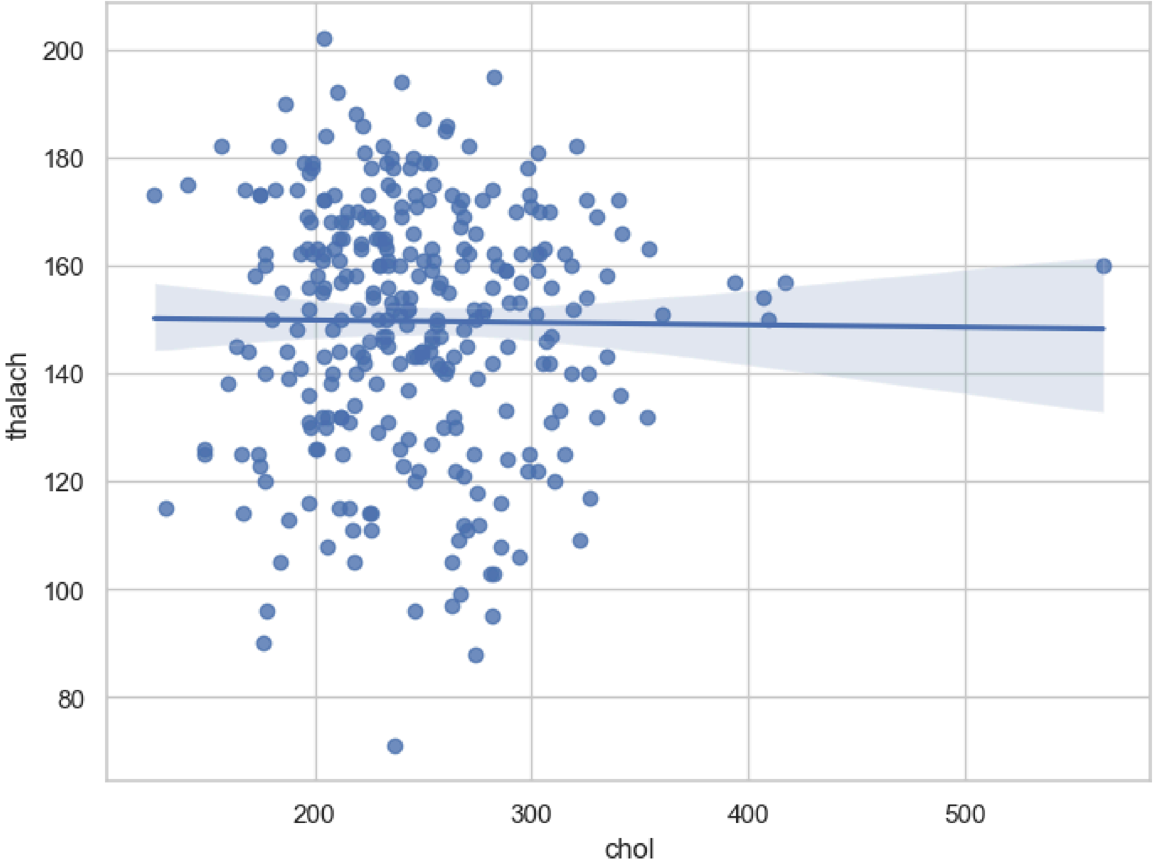


```
In [62]: f, ax = plt.subplots(figsize=(8, 6))
         ax = sns.scatterplot(x="chol", y = "thalach", data=df)
```

```
plt.show()
```



In [63]:
```
f, ax = plt.subplots(figsize=(8, 6))
ax = sns.regplot(x="chol", y="thalach", data=df)
plt.show()
```



In [64]:
```
df.isnull().sum() #This checks for any missing values
```

Out[64]:
```
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
```

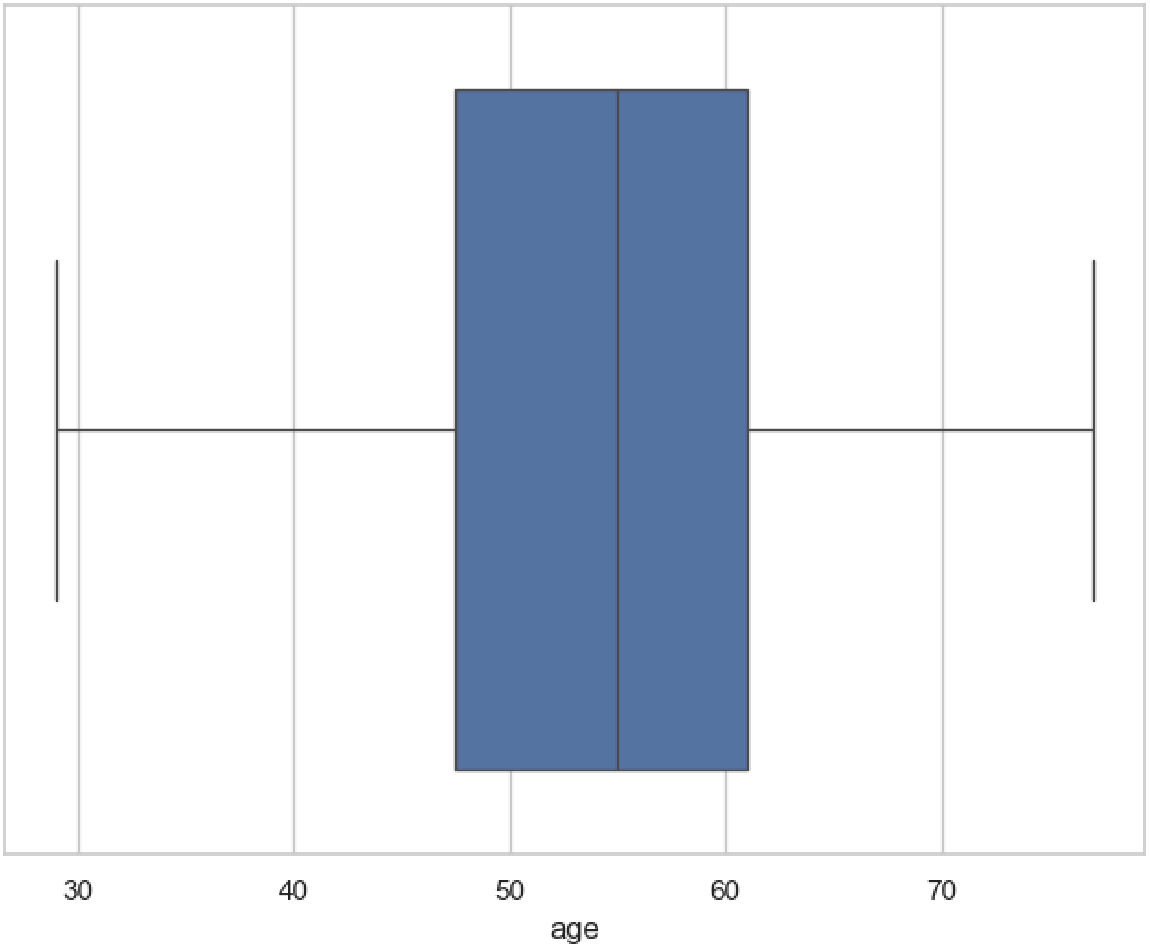
In [65]:
```
assert pd.notnull(df).all().all() # assert says there are no missing values in t
```

In [66]:
```
assert (df >= 0).all().all() # assert says all values are greater than or equal
```

In [67]:
```
df['age'].describe()
```

Out[67]:
```
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 [68]:
```
f, ax = plt.subplots(figsize=(8, 6))
sns.boxplot(x=df["age"])
plt.show()
```



In [69]:

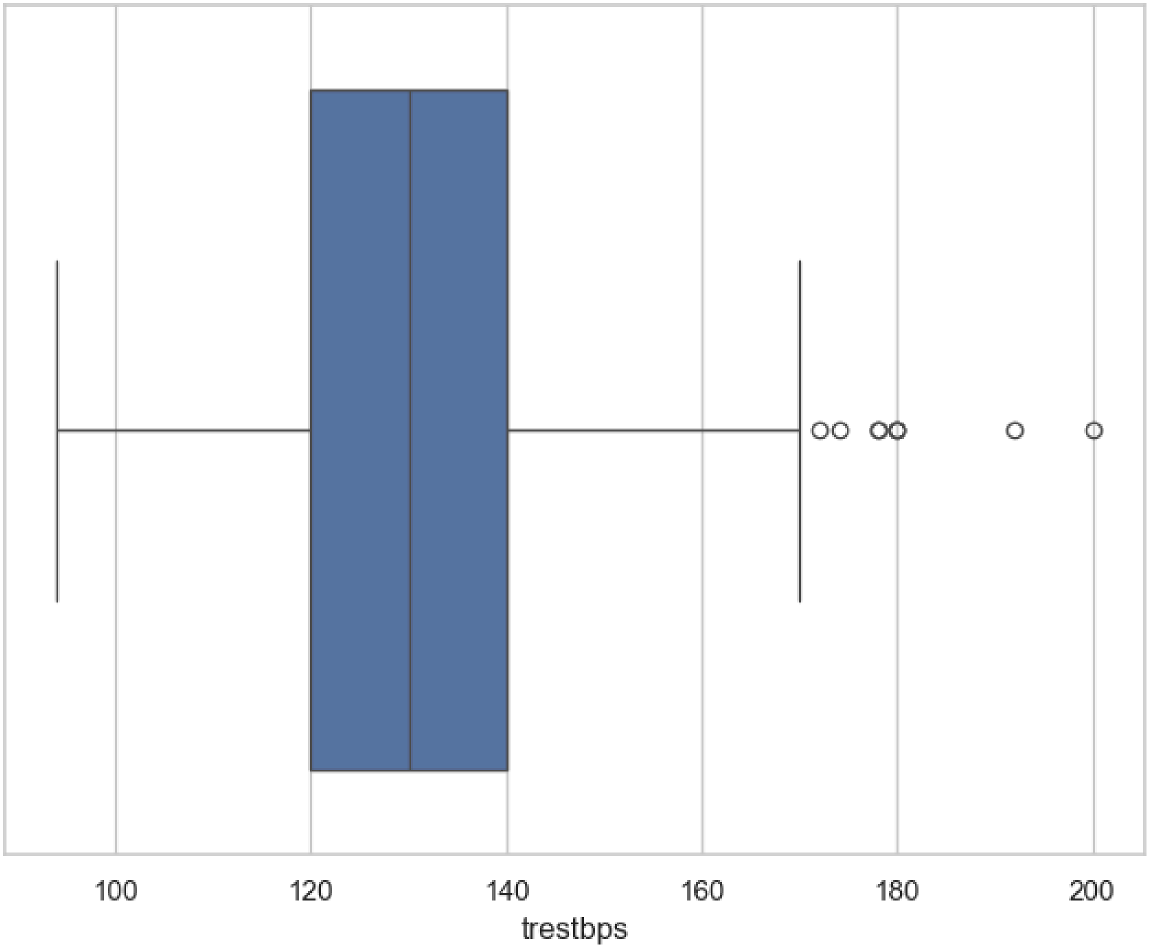
```
df['trestbps'].describe()
```

Out[69]:

```
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 [70]:

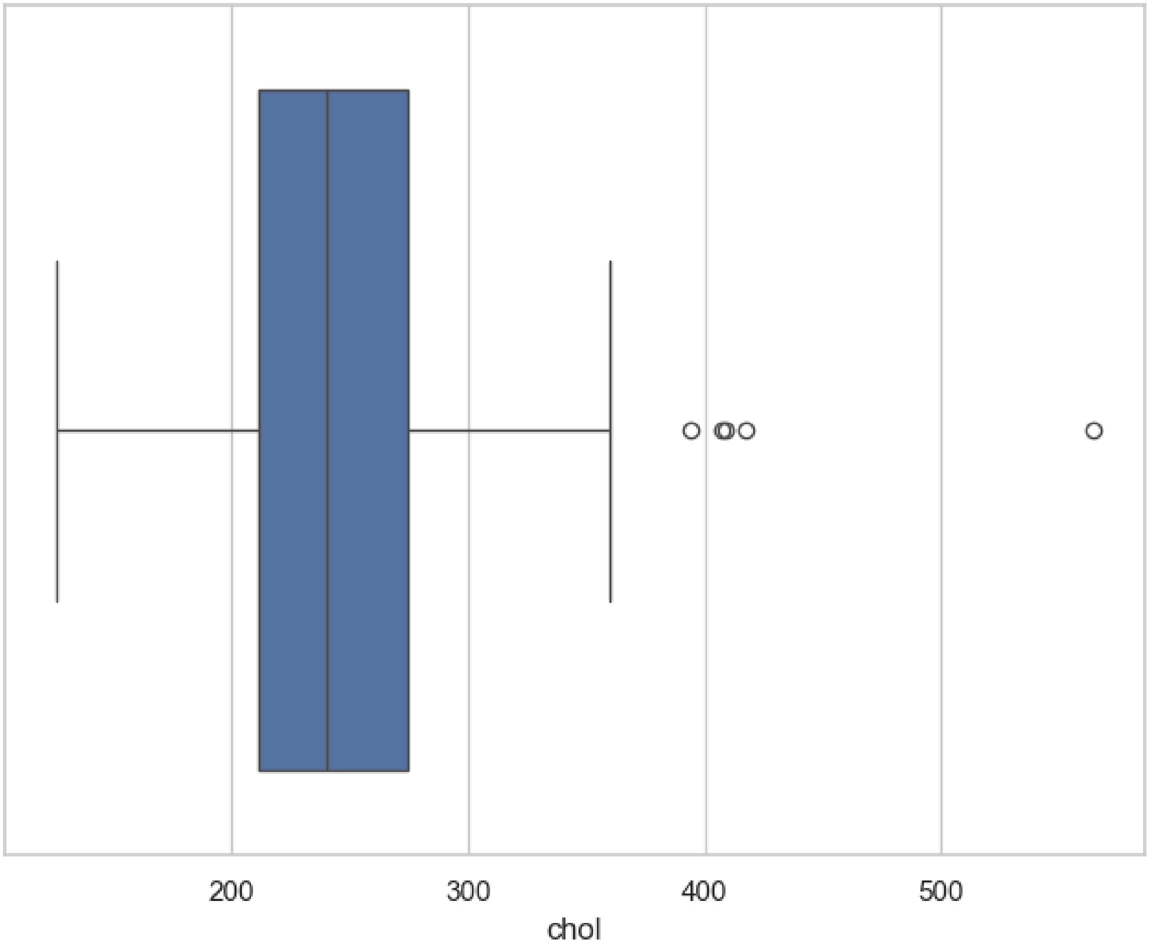
```
f, ax = plt.subplots(figsize=(8, 6))
sns.boxplot(x=df["trestbps"])
plt.show()
```



In [71]:
```
df['chol'].describe() #chol variable
```

Out[71]:
```
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
```

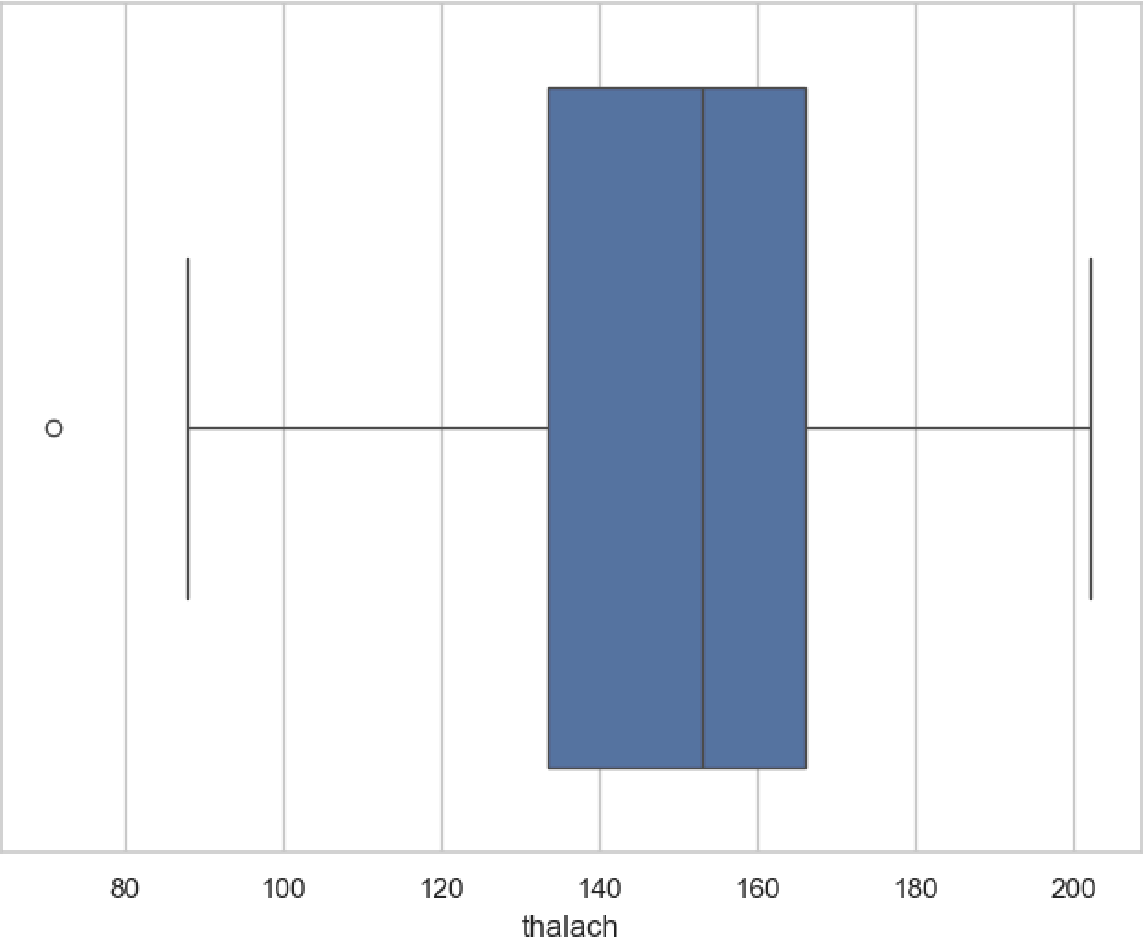
In [72]:
```
f, ax = plt.subplots(figsize=(8, 6))
sns.boxplot(x=df["chol"])
plt.show()
```



In [73]:
```
df['thalach'].describe()
```

Out[73]:
```
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
```

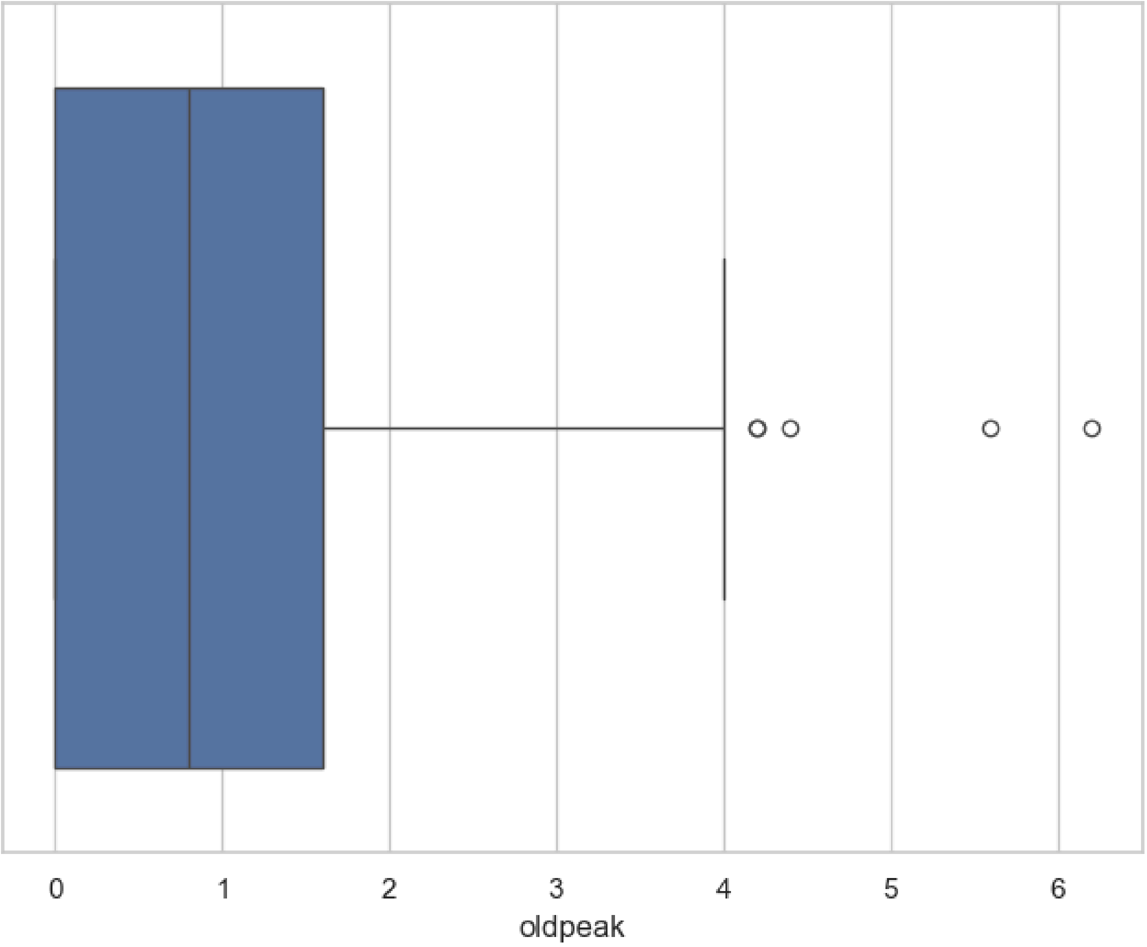
In [74]:
```
f, ax = plt.subplots(figsize=(8, 6))
sns.boxplot(x=df["thalach"]) #thalach variable
plt.show()
```



```
In [75]: df['oldpeak'].describe()
```

```
Out[75]: 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 [76]: f, ax = plt.subplots(figsize=(8, 6))
         sns.boxplot(x=df["oldpeak"])
         plt.show()
```



```
In [ ]: # The End
```