In [45]:
```
# packages use for dataset import and basic data analysis
import numpy as np
import pandas as pd
```

In [46]:
```
# packages use for data visualization and advance statistical analysis
import seaborn as sns
import matplotlib.pyplot as plt
import scipy.stats as st
%matplotlib inline
sns.set(style="whitegrid")
```

In [47]:
```
# package for ignoring warning
import warnings
warnings.filterwarnings('ignore')
```

In [48]:
```
df = pd.read_csv("heart.csv")
df
```

Out[48]:

	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	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	
298	57	0	0	140	241	0	1	123	1	0.2	1	0	3	
299	45	1	3	110	264	0	1	132	0	1.2	1	0	3	
300	68	1	0	144	193	1	1	141	0	3.4	1	2	3	
301	57	1	0	130	131	0	1	115	1	1.2	1	1	3	
302	57	0	1	130	236	0	0	174	0	0.0	1	1	2	

303 rows × 14 columns

Exploraratory Data Analysis

In [49]:
```
print('The shape of the dataset : ', df.shape)
```

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

In [50]:
```
# Preview the dataset
df.head()
```

Out[50]:

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	targ
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 [51]:
```
# Summary of dataset
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 [52]:
```
df.dtypes
```

Out[52]:
```
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 [53]:
```
# statistical properties of dataset
df.describe()
```

Out[53]:

	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 [54]:
```
#View column names
df.columns
```

Out[54]:
```
Index(['age', 'sex', 'cp', 'trestbps', 'chol', 'fbs', 'restecg', 'thalac
h',
       'exang', 'oldpeak', 'slope', 'ca', 'thal', 'target'],
      dtype='object')
```

In [55]:
```
# Check the number of unique values in target variable
df['target'].nunique()
```

Out[55]:
```
2
```

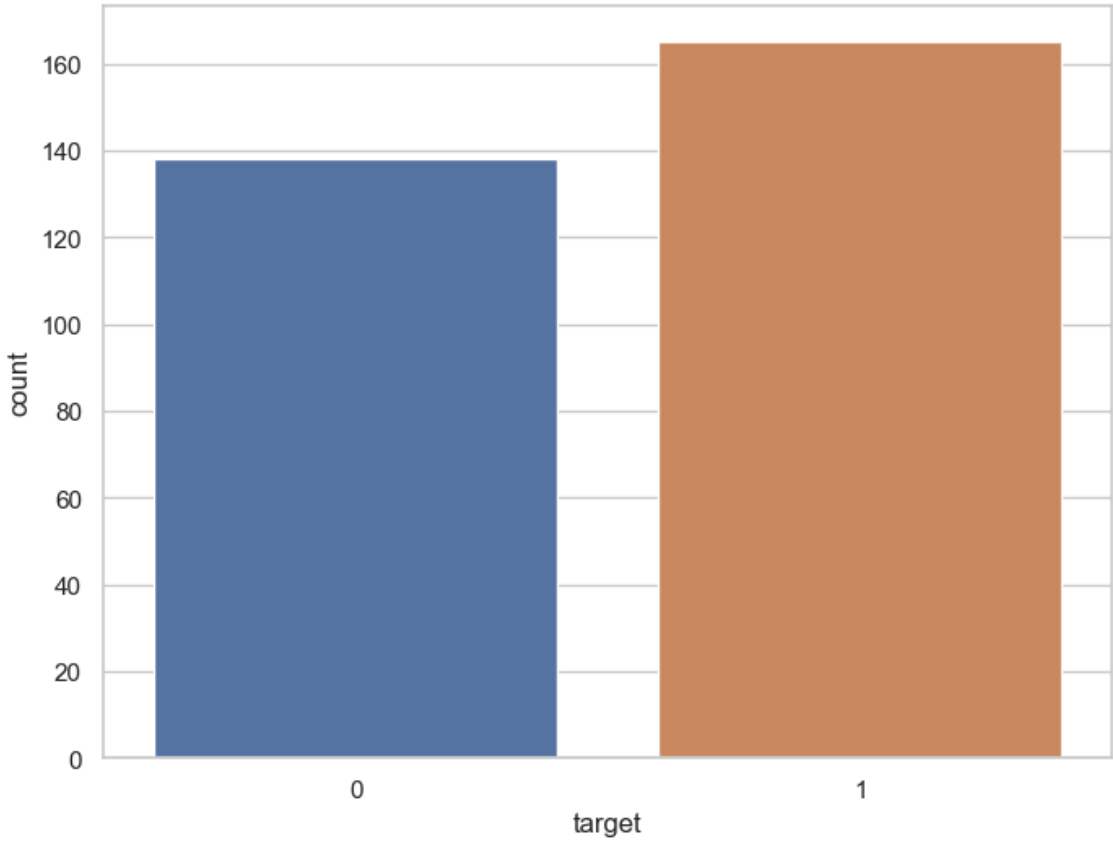
In [56]:
```
# View the unique values in target variable
df['target'].unique()
```

Out[56]:
```
array([1, 0], dtype=int64)
```

In [57]:
```
#Frequency distribution of target variable
df['target'].value_counts()
```

Out[57]:
```
target
1    165
0    138
Name: count, dtype: int64
```

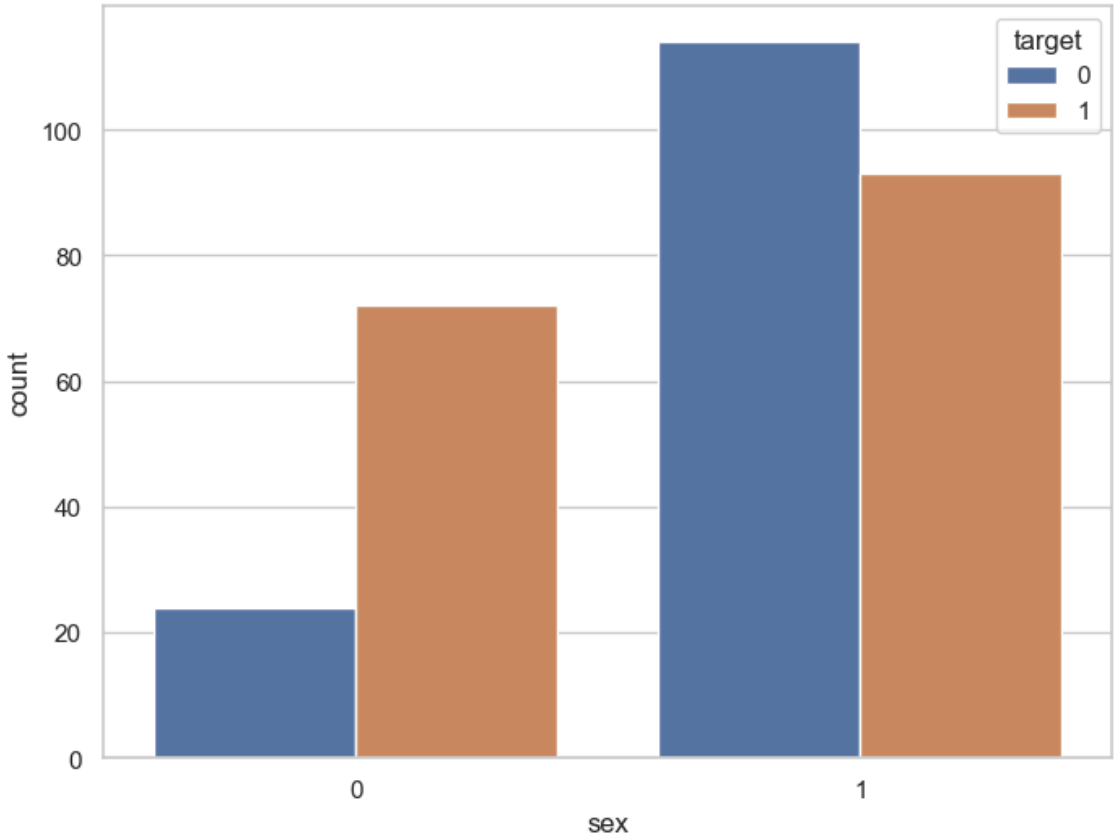
In [58]:
```
f, ax = plt.subplots(figsize = (8,6))
ax = sns.countplot(x="target",data=df)
plt.show()
```



In [59]:
```
#Frequency distribution of target variable wrt sex
df.groupby('sex')['target'].value_counts()
```

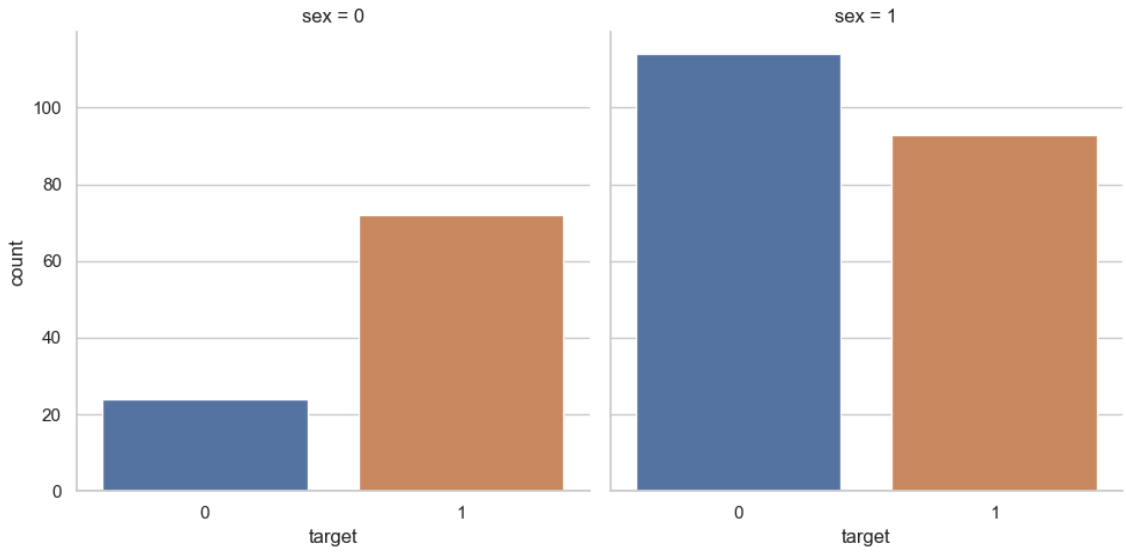
Out[59]:
```
sex  target
0    1          72
     0          24
1    0         114
     1          93
Name: count, dtype: int64
```

In [60]:
```
f, ax = plt.subplots(figsize=(8,6))
ax = sns.countplot(x = "sex", hue = "target",data = df)
plt.show()
```

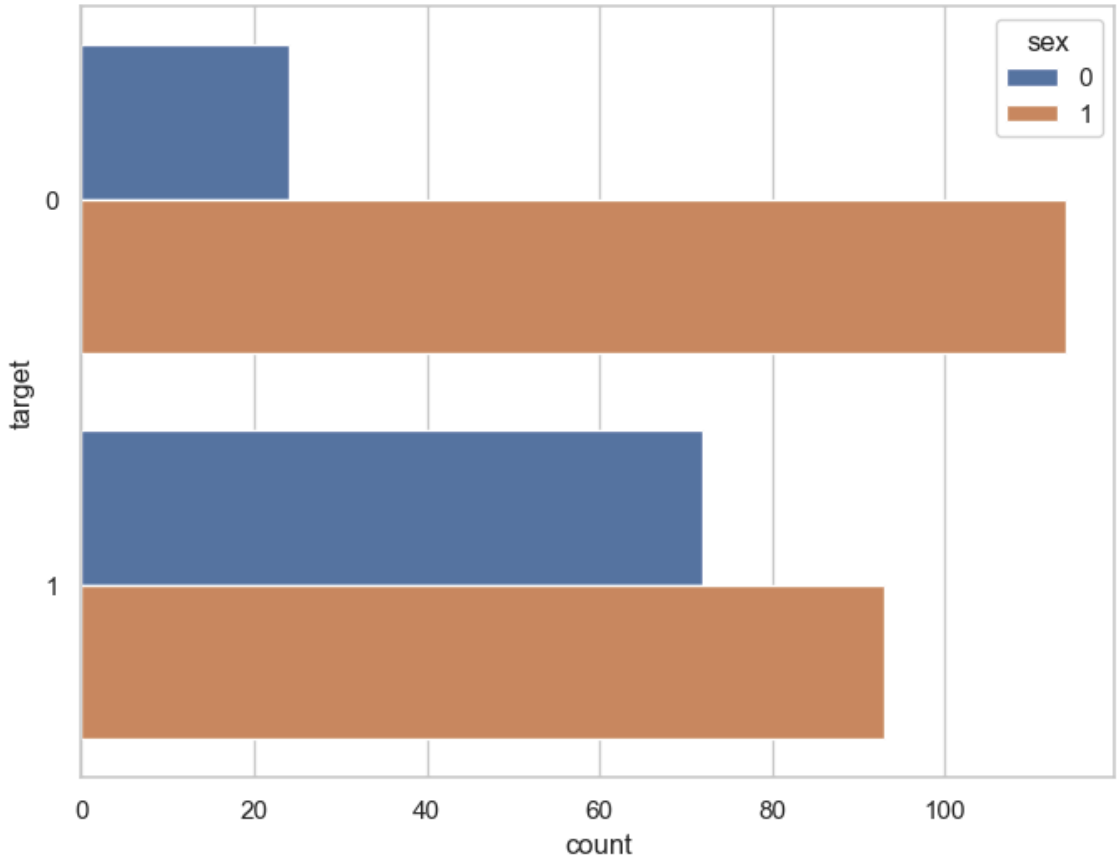


In [61]:
```
sns.catplot(x = "target", col = "sex", data = df, kind = "count", height =
```

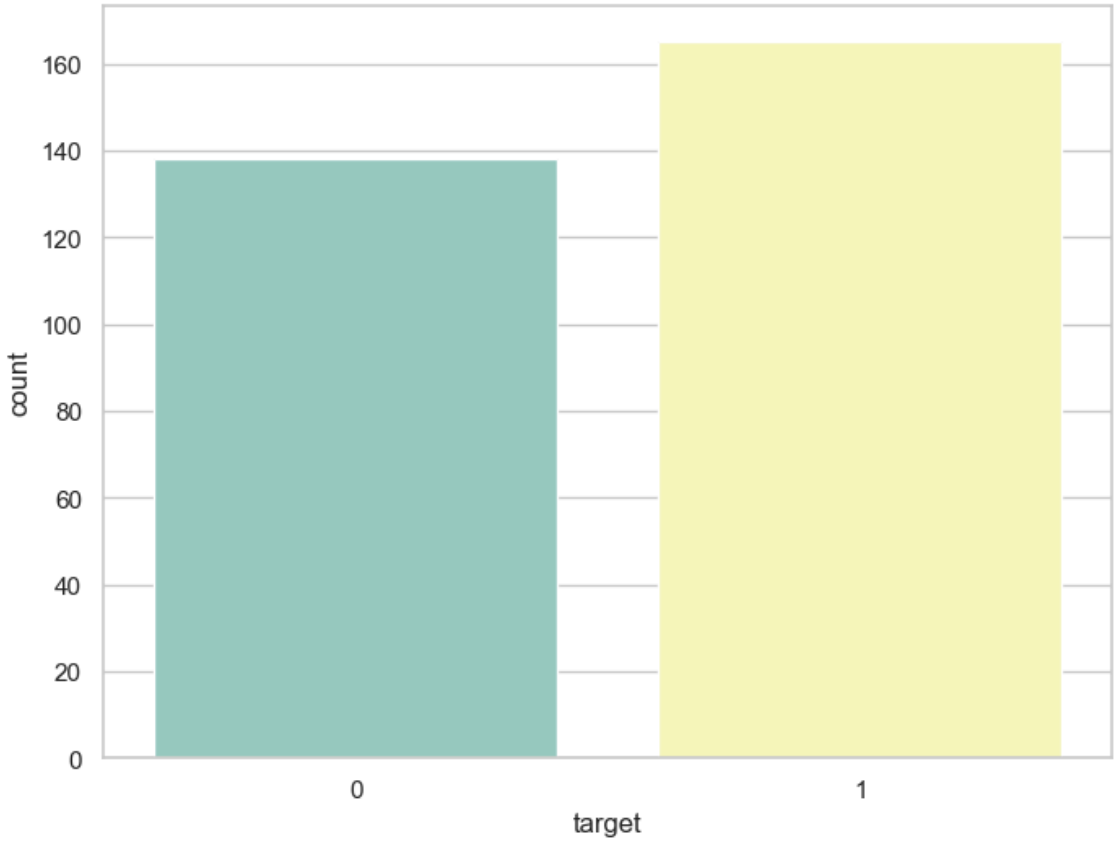
Out[61]: <seaborn.axisgrid.FacetGrid at 0x214741cb6d0>



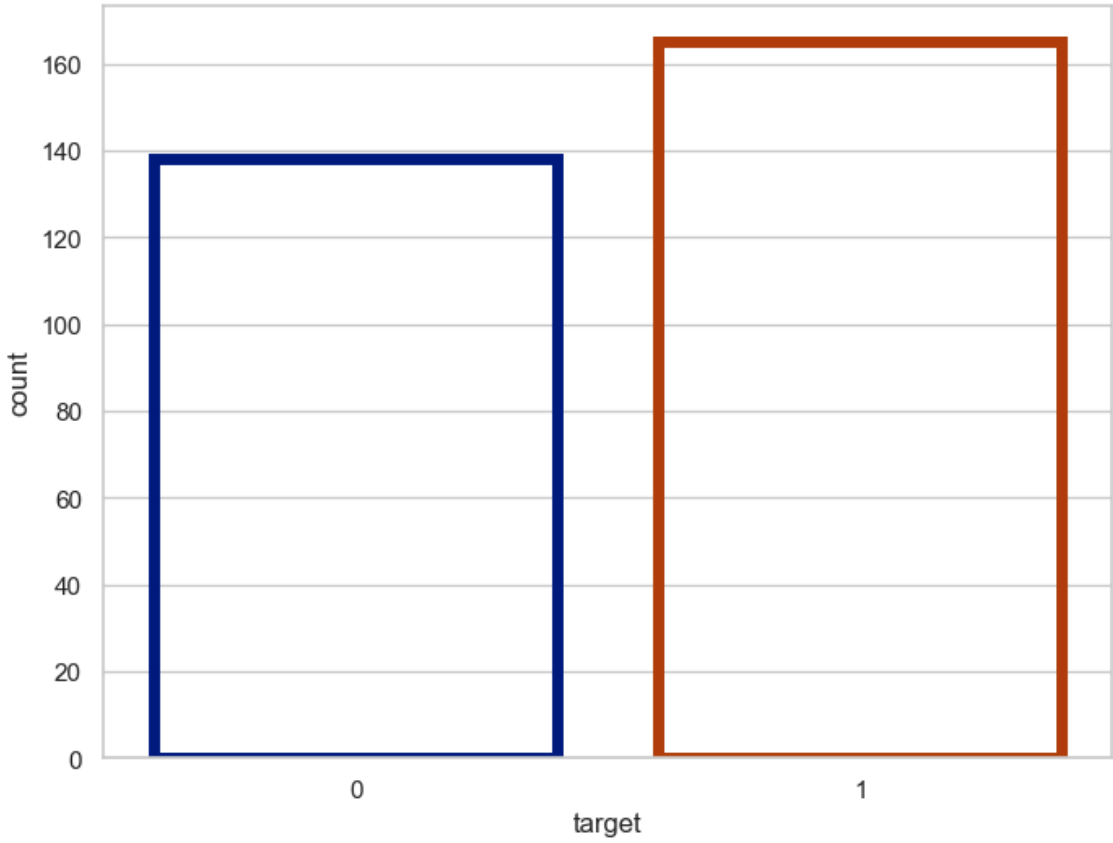
In [62]:
```
f, ax = plt.subplots(figsize=(8,6))
ax = sns.countplot(y="target",hue="sex",data=df)
plt.show()
```



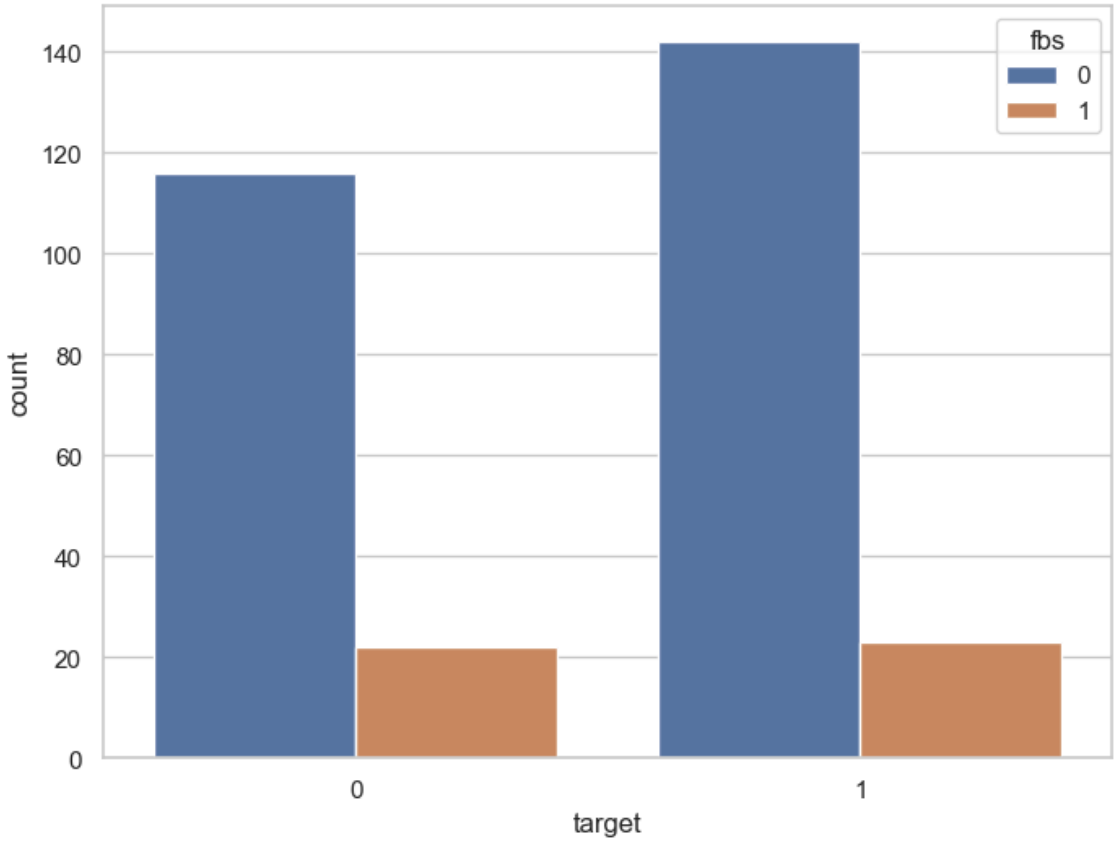
In [63]:
```
f, ax = plt.subplots(figsize = (8,6))
ax = sns.countplot(x="target", data=df, palette = "Set3")
plt.show()
```



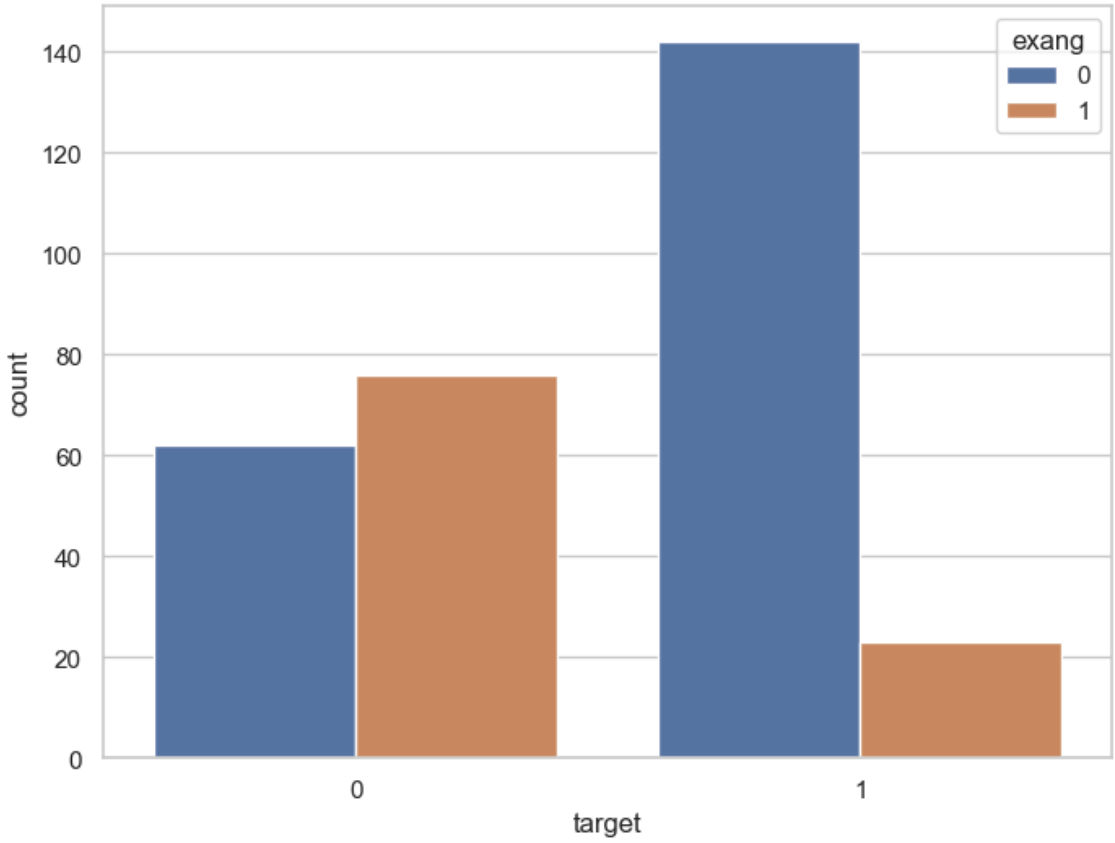
In [64]:
```
f, ax = plt.subplots(figsize=(8,6))
ax = sns.countplot(x = "target", data=df, facecolor = (0,0,0,0), linewidth
plt.show()
```



In [65]:
```
f, ax = plt.subplots(figsize=(8,6))
ax = sns.countplot(x = "target", hue = "fbs", data=df)
plt.show()
```



In [66]:
```
f, ax = plt.subplots(figsize=(8,6))
ax = sns.countplot(x = "target", hue = "exang", data=df)
plt.show()
```



In [67]:
```
correlation = df.corr()
correlation
```

Out[67]:

	age	sex	cp	trestbps	chol	fbs	restecg	thalac
age	1.000000	-0.098447	-0.068653	0.279351	0.213678	0.121308	-0.116211	-0.39852
sex	-0.098447	1.000000	-0.049353	-0.056769	-0.197912	0.045032	-0.058196	-0.04402
cp	-0.068653	-0.049353	1.000000	0.047608	-0.076904	0.094444	0.044421	0.29576
trestbps	0.279351	-0.056769	0.047608	1.000000	0.123174	0.177531	-0.114103	-0.04669
chol	0.213678	-0.197912	-0.076904	0.123174	1.000000	0.013294	-0.151040	-0.00994
fbs	0.121308	0.045032	0.094444	0.177531	0.013294	1.000000	-0.084189	-0.00856
restecg	-0.116211	-0.058196	0.044421	-0.114103	-0.151040	-0.084189	1.000000	0.04412
thalach	-0.398522	-0.044020	0.295762	-0.046698	-0.009940	-0.008567	0.044123	1.00000
exang	0.096801	0.141664	-0.394280	0.067616	0.067023	0.025665	-0.070733	-0.37881
oldpeak	0.210013	0.096093	-0.149230	0.193216	0.053952	0.005747	-0.058770	-0.34418
slope	-0.168814	-0.030711	0.119717	-0.121475	-0.004038	-0.059894	0.093045	0.38678
ca	0.276326	0.118261	-0.181053	0.101389	0.070511	0.137979	-0.072042	-0.21317
thal	0.068001	0.210041	-0.161736	0.062210	0.098803	-0.032019	-0.011981	-0.09643
target	-0.225439	-0.280937	0.433798	-0.144931	-0.085239	-0.028046	0.137230	0.42174

In [68]:
```
correlation['target'].sort_values(ascending = False)
```

Out[68]:
```
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
```

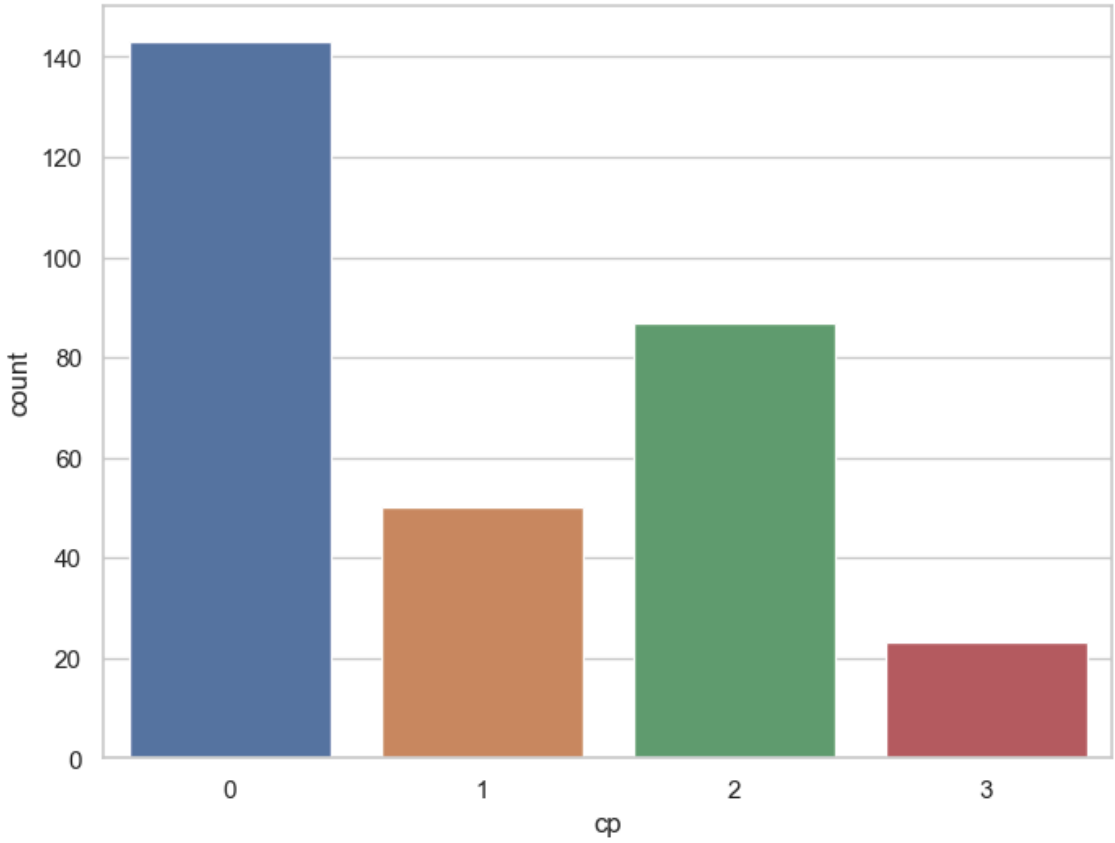
In [69]:
```
df['cp'].nunique()
```

Out[69]:
```
4
```

In [70]:
```
df['cp'].value_counts()
```

Out[70]:
```
cp
0    143
2     87
1     50
3     23
Name: count, dtype: int64
```

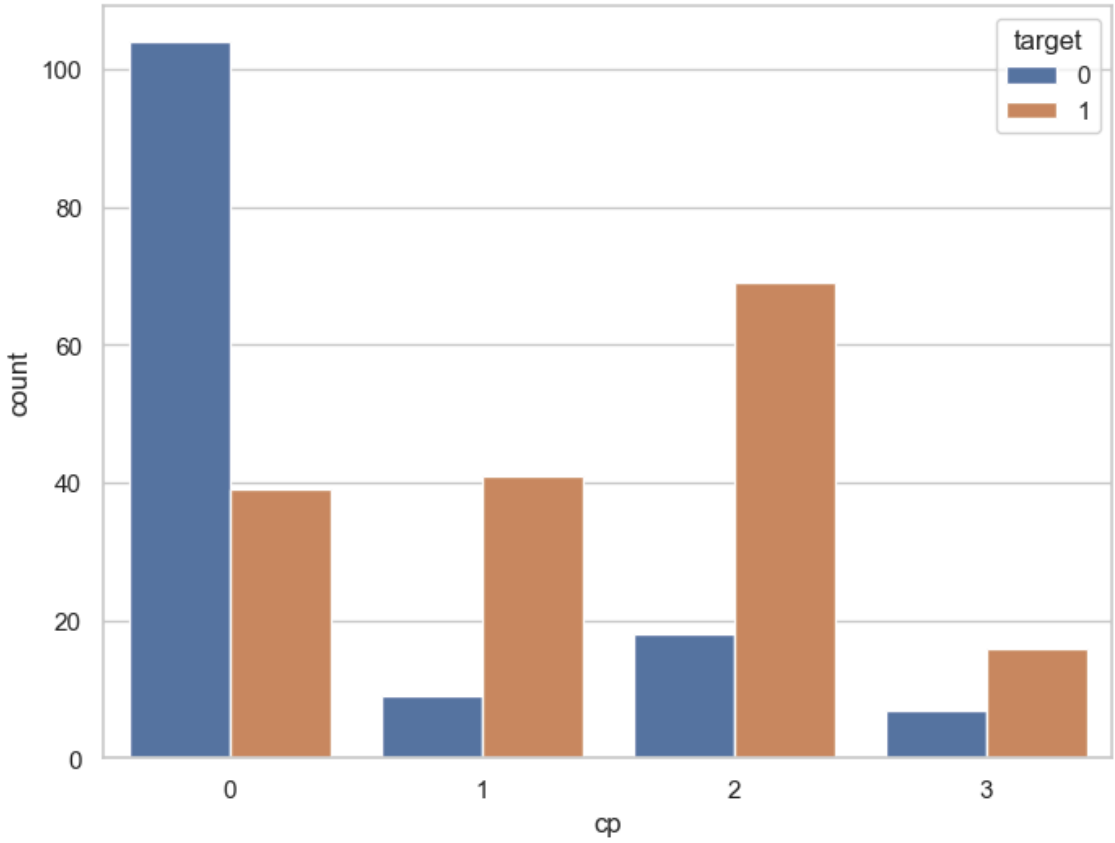
In [71]:
```
f, ax = plt.subplots(figsize = (8,6))
ax = sns.countplot(x = "cp", data=df)
plt.show()
```



In [72]:
```
df.groupby('cp')['target'].value_counts()
```

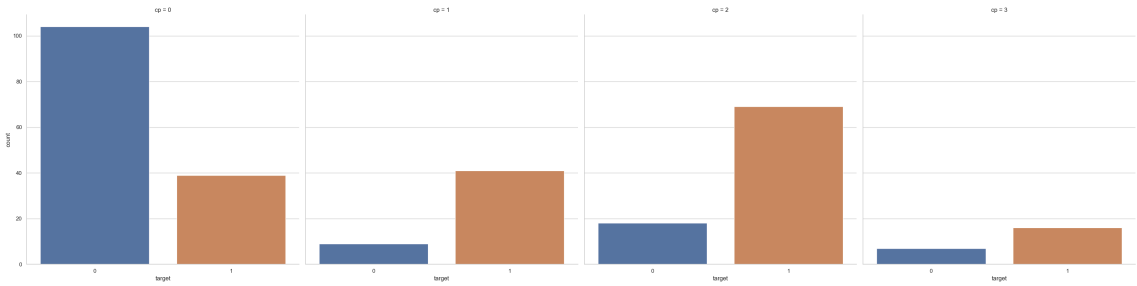
Out[72]:
```
cp  target
0   0         104
    1          39
1   1          41
    0           9
2   1          69
    0          18
3   1          16
    0           7
Name: count, dtype: int64
```

In [73]:
```
f, ax = plt.subplots(figsize = (8,6))
ax = sns.countplot(x="cp",hue="target",data =df)
plt.show()
```



In [74]:
```
sns.catplot(x="target", col ="cp", data=df, kind="count", height=8, aspect=
```

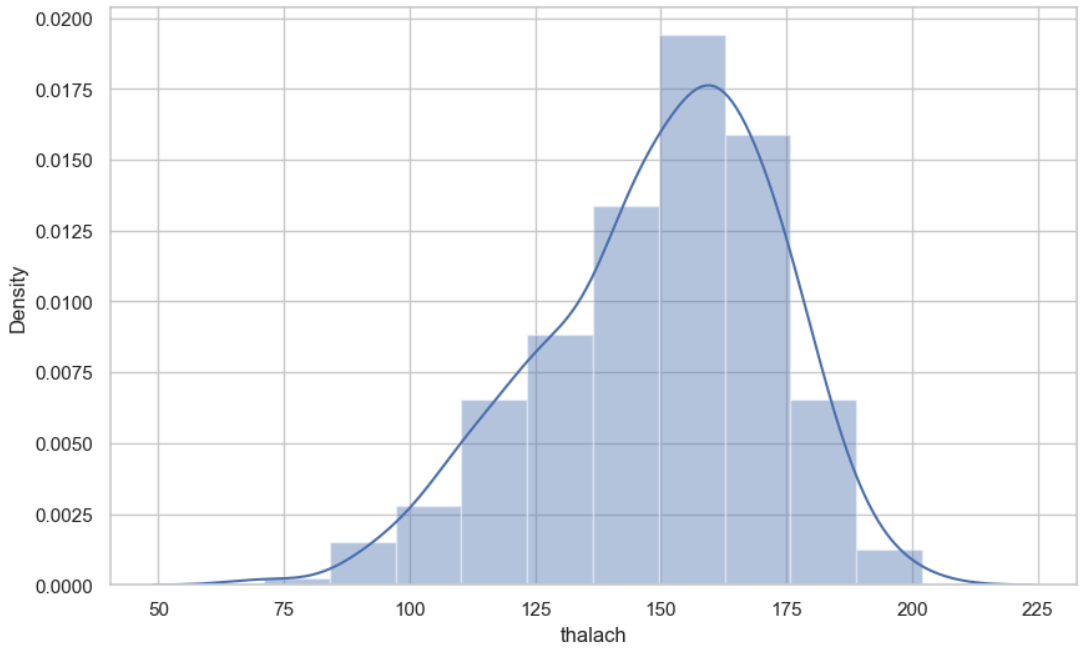
Out[74]: <seaborn.axisgrid.FacetGrid at 0x2147438ee50>



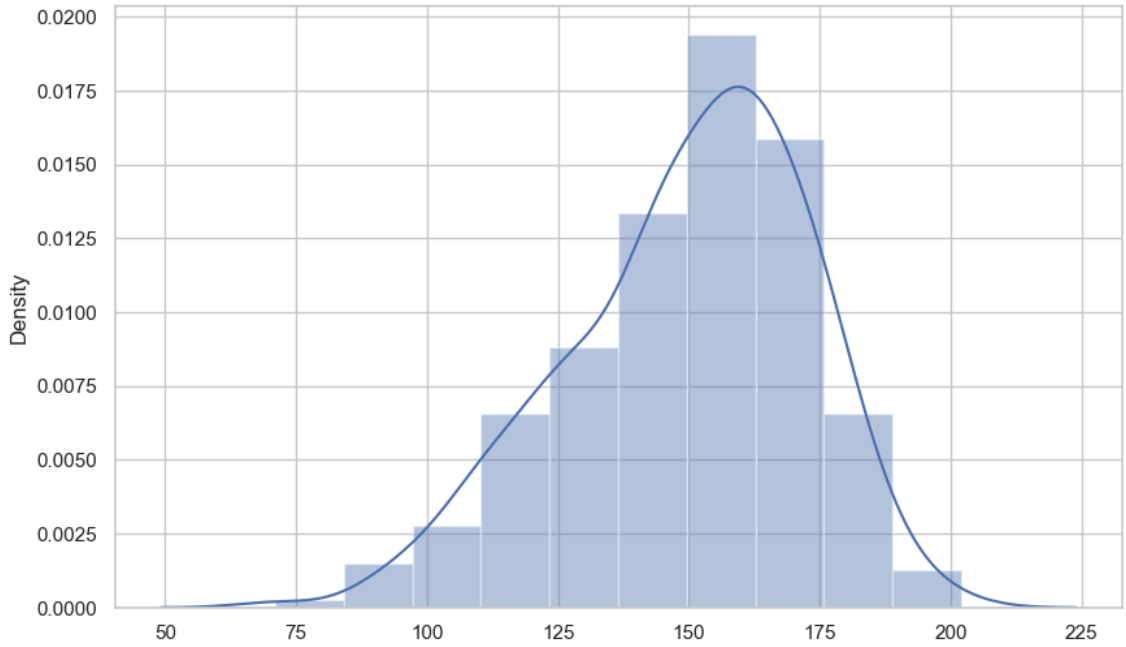
In [75]:
```
df['thalach'].nunique()
```

Out[75]: 91

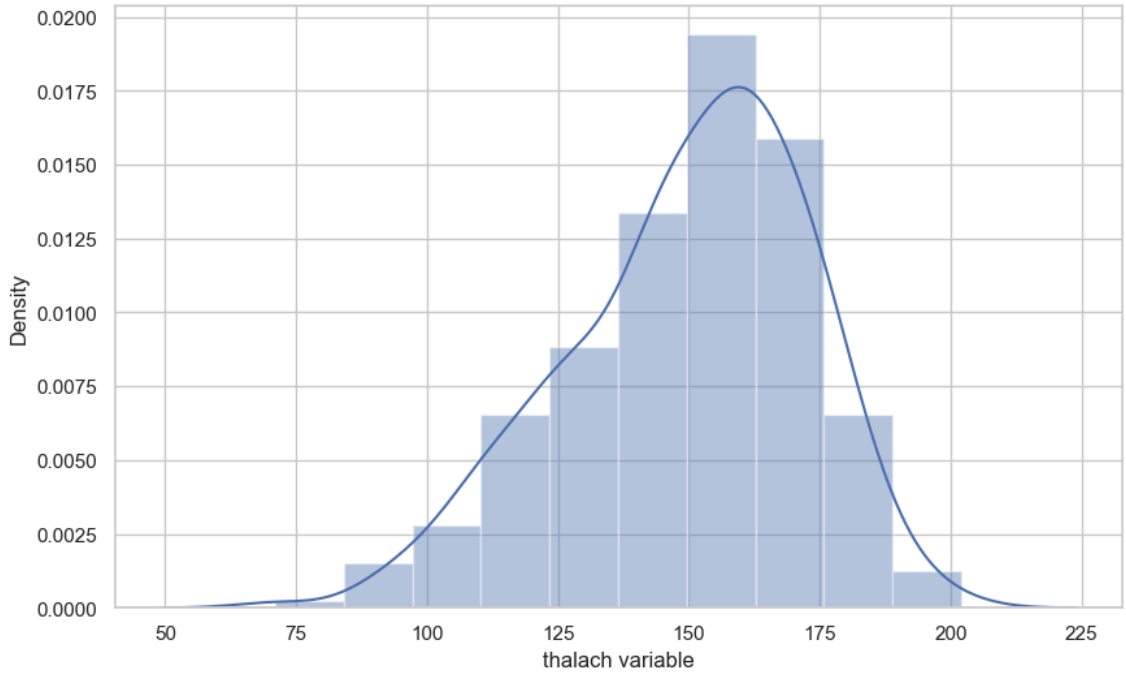
In [76]:
```
f, ax = plt.subplots(figsize=(10,6))
x = df['thalach']
ax = sns.distplot(x, bins = 10)
plt.show()
```



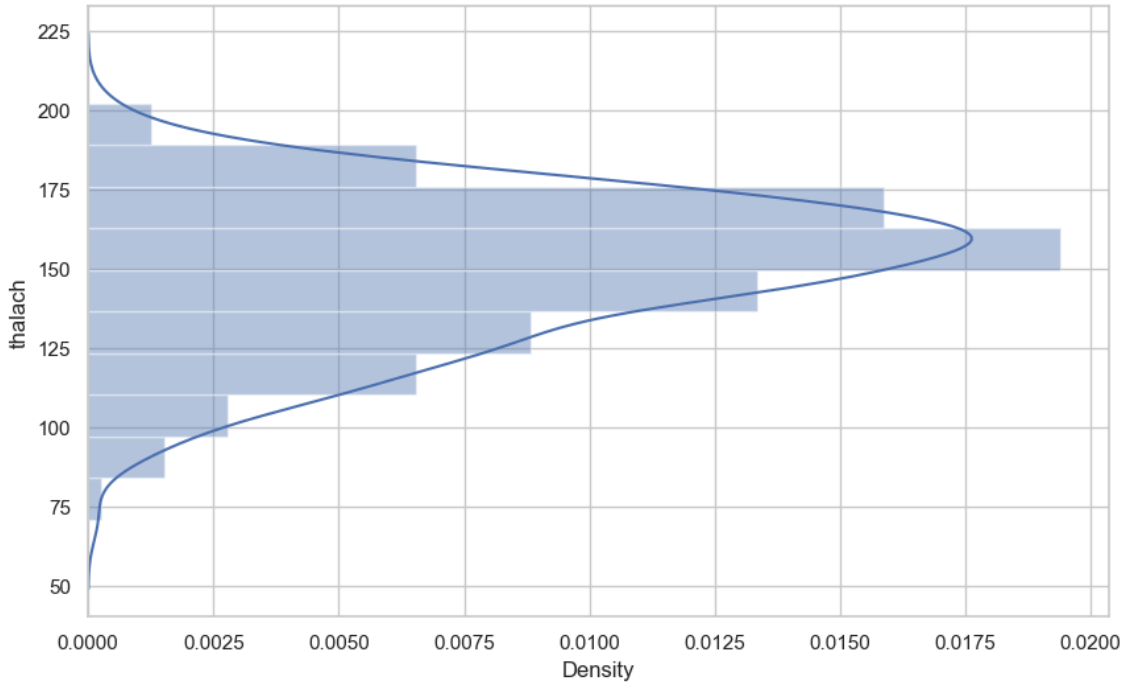
In [77]:
```
f, ax = plt.subplots(figsize=(10,6))
ax = sns.distplot(x=df['thalach'], bins = 10)
plt.show()
```



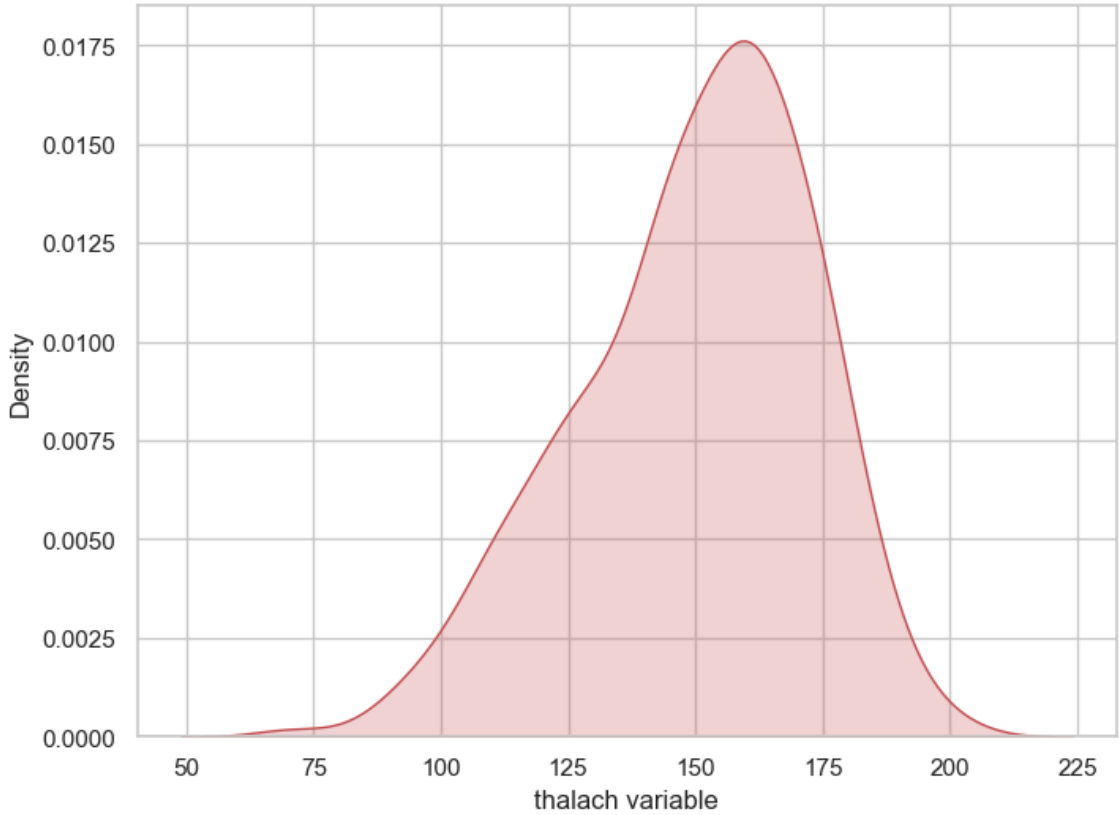
In [78]:
```
f, ax = plt.subplots(figsize=(10,6))
x = df['thalach']
x = pd.Series(x, name="thalach variable")
ax = sns.distplot(x, bins=10)
plt.show()
```



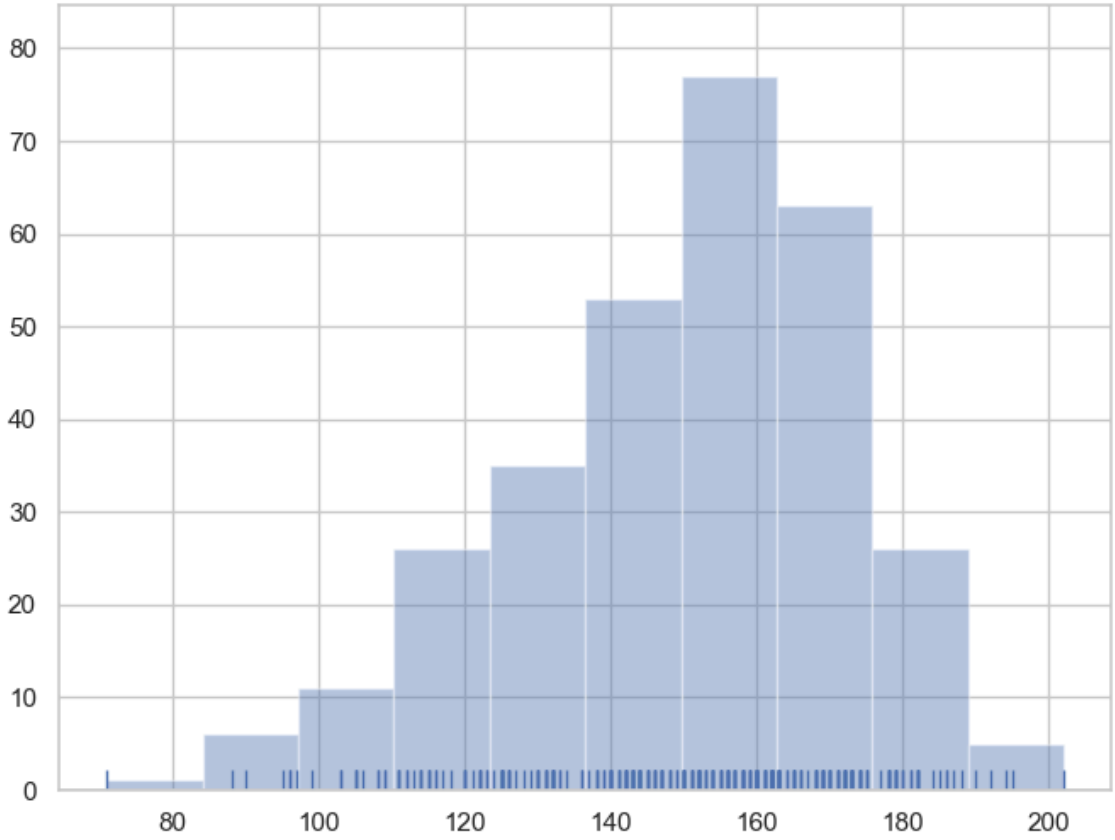
In [79]:
```
f, ax = plt.subplots(figsize=(10,6))
x = df['thalach']
ax = sns.distplot(x, bins=10, vertical=True)
plt.show()
```



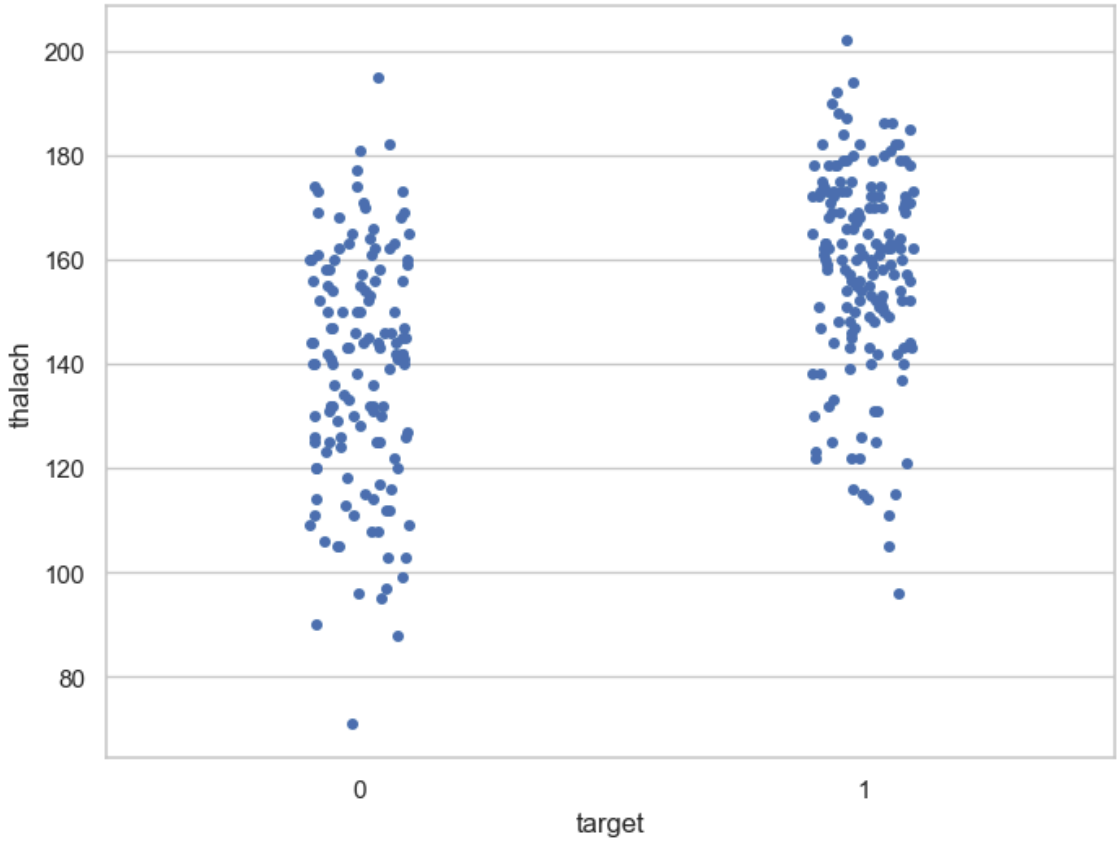
In [80]:
```
f, ax = plt.subplots(figsize = (8,6))
x = df['thalach']
x = pd.Series(x, name="thalach variable")
ax =  sns.kdeplot(x, shade = True, color = 'r')
plt.show()
```



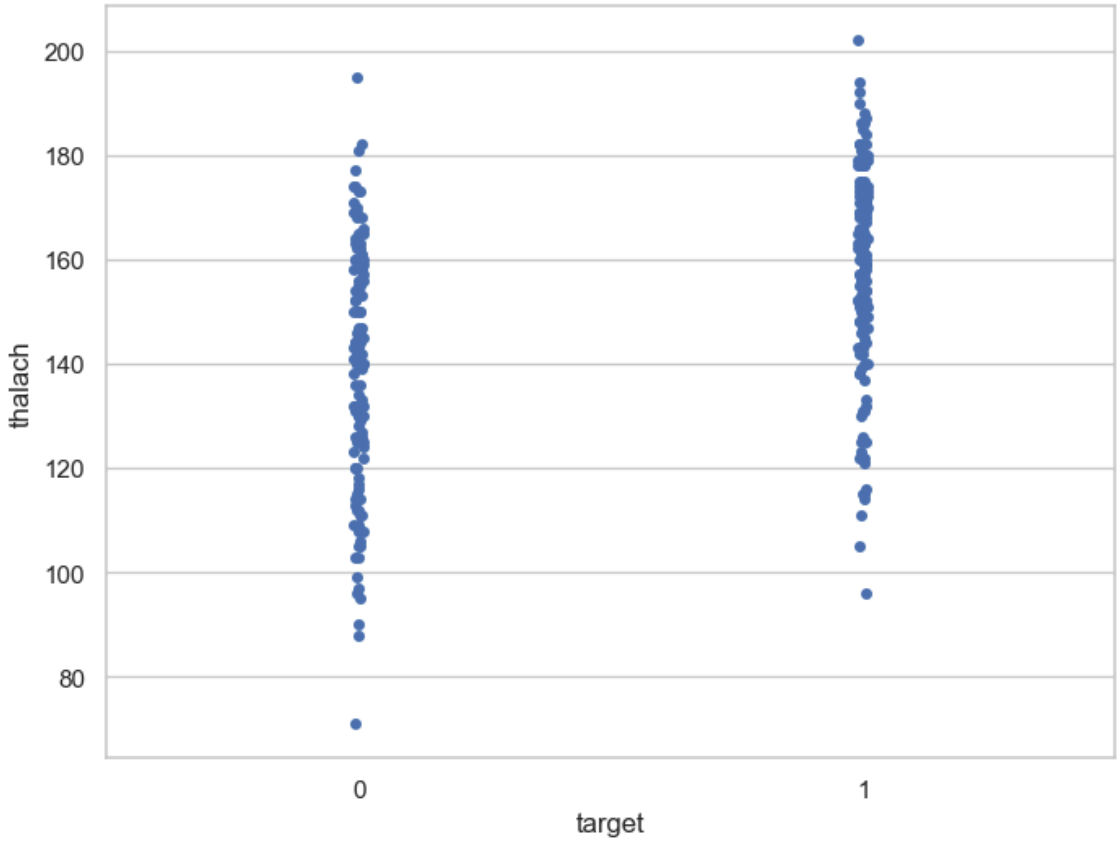
In [81]:
```
f, ax = plt.subplots(figsize=(8,6))
ax = sns.distplot(x = df['thalach'], kde = False, rug = True, bins = 10)
plt.show()
```



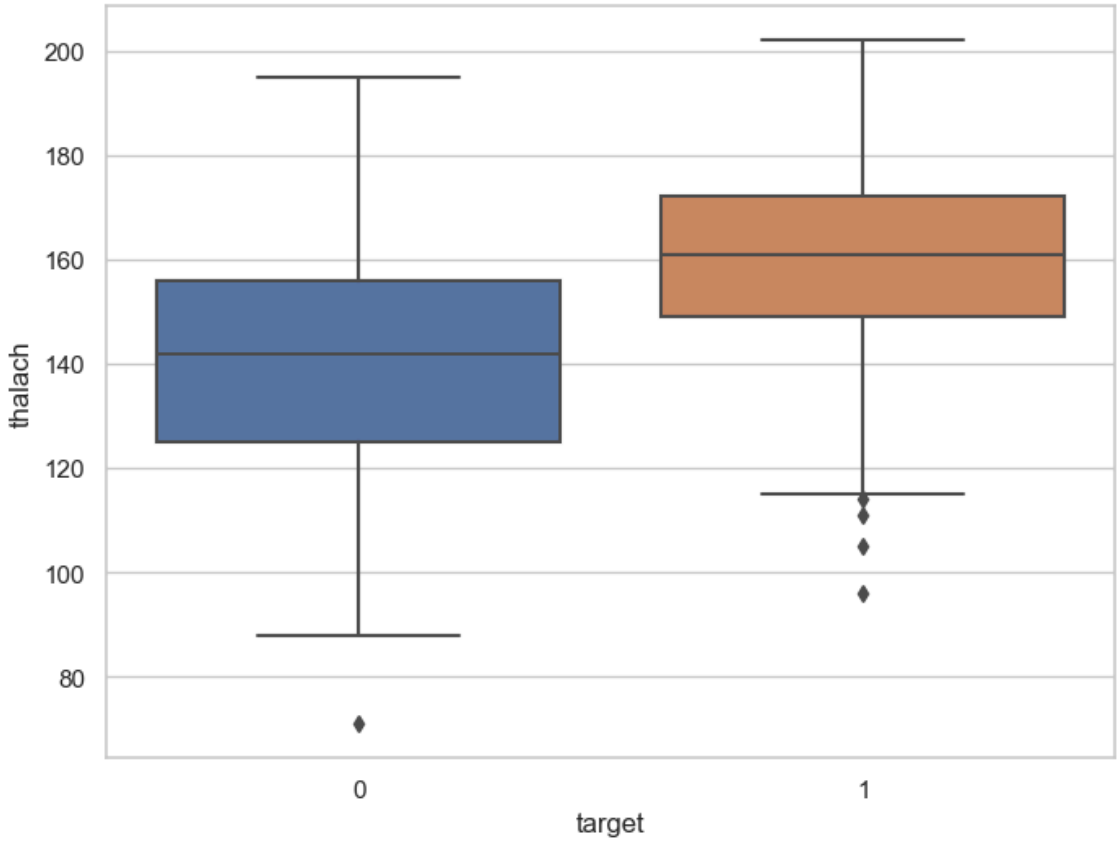
In [82]:
```
f, ax = plt.subplots(figsize=(8,6))
sns.stripplot(x="target",y="thalach", data=df)
plt.show()
```



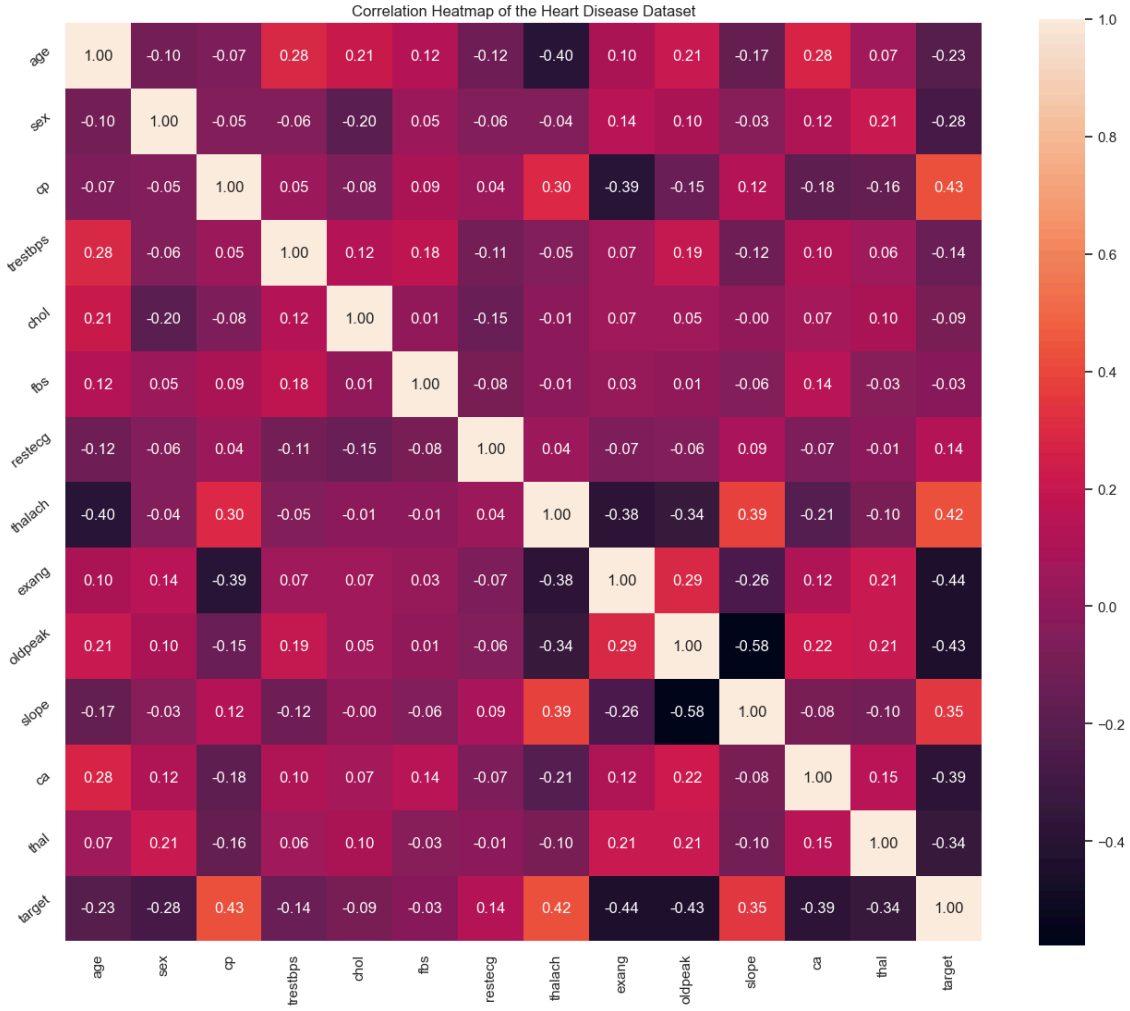
In [83]:
```
f, ax = plt.subplots(figsize=(8,6))
sns.stripplot(x="target",y="thalach", data = df,jitter = 0.01)
plt.show()
```



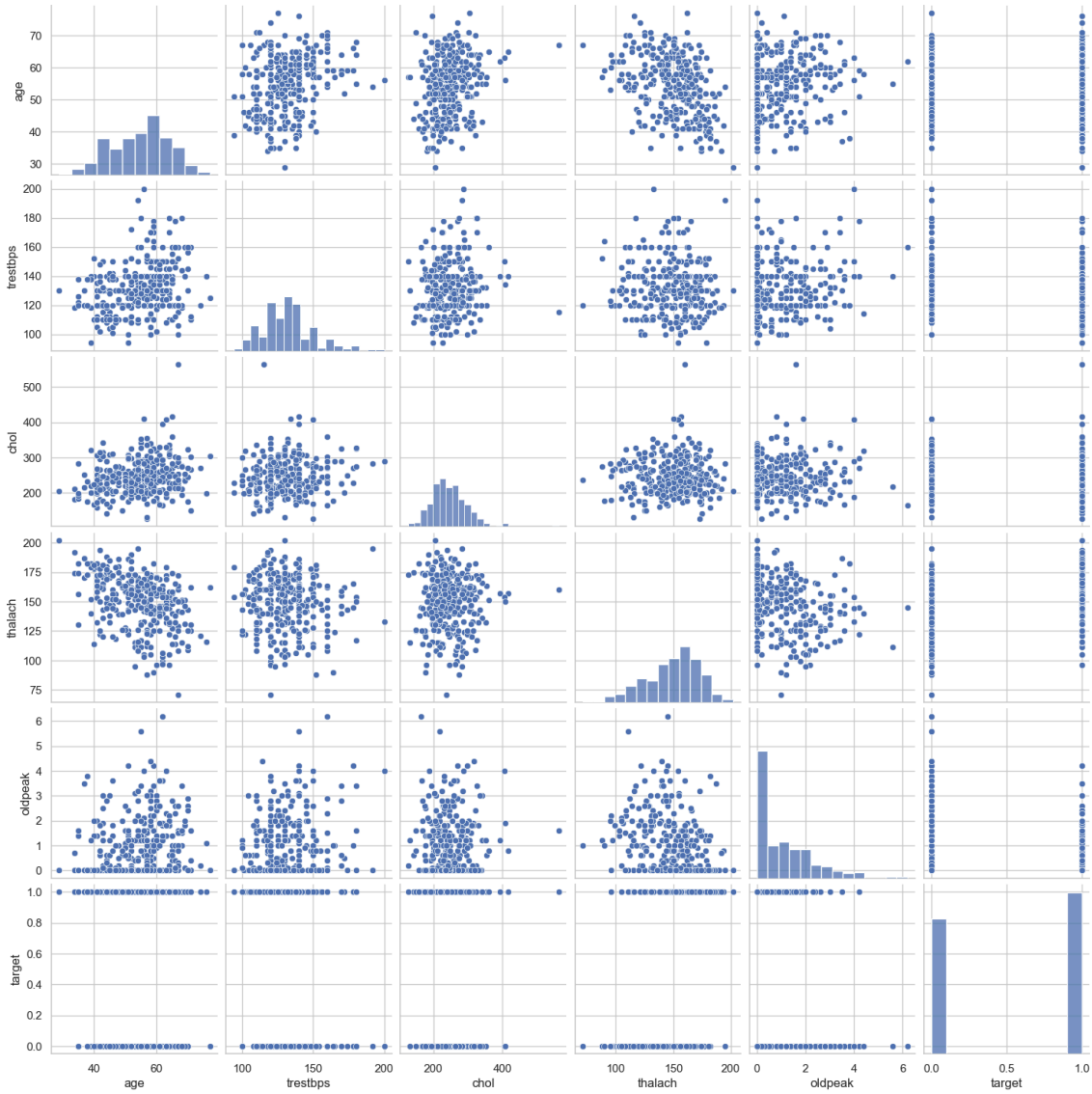
In [84]:
```
f, ax = plt.subplots(figsize=(8,6))
sns.boxplot(x = "target", y ="thalach", data=df)
plt.show()
```



In [87]:
```
plt.figure(figsize=(16,13))
plt.title('Correlation Heatmap of the Heart Disease Dataset')
a = sns.heatmap(correlation, square = True, annot = True, fmt = '.2f', line
a.set_xticklabels(a.get_xticklabels(),rotation=90)
a.set_yticklabels(a.get_yticklabels(),rotation=40)
plt.show()
```



In [88]:
```
# Pair plot
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

check the number of unique values in age variables

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

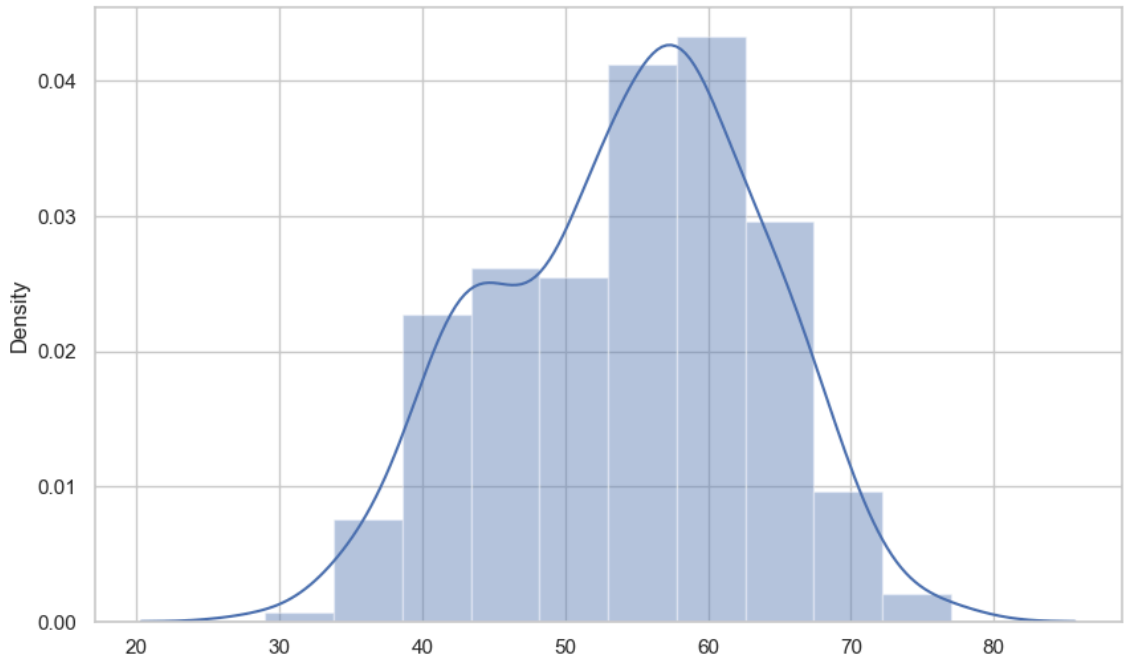
Out[89]: 41

View statistical summary of age variable

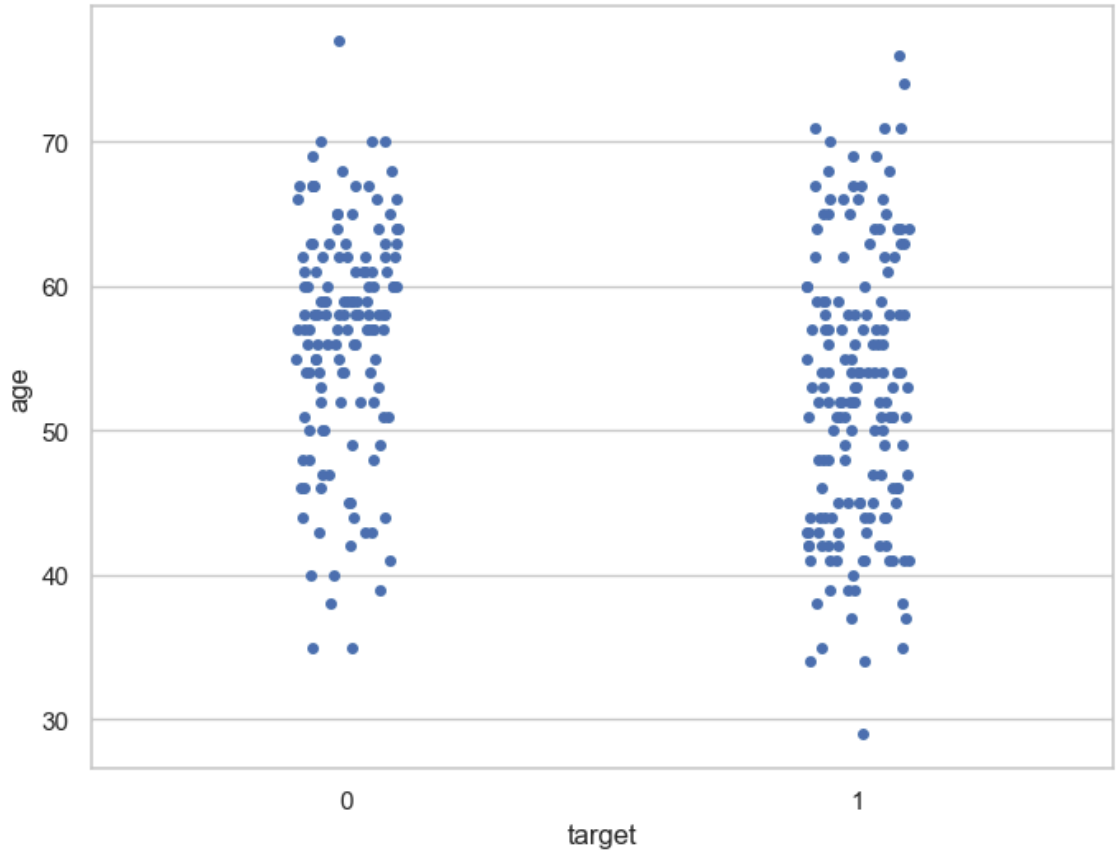
In [91]:
```
df['age'].describe()
```

Out[91]:
```
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
```

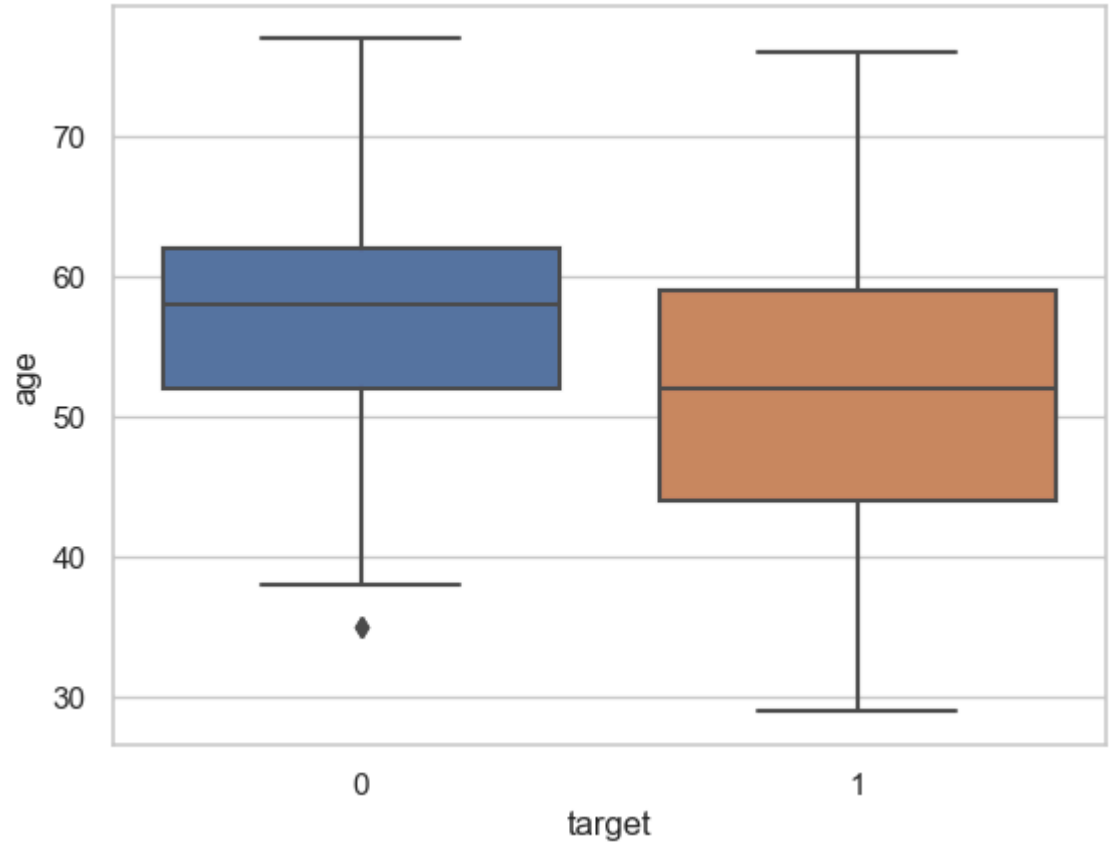
In [92]:
```
f, ax = plt.subplots(figsize=(10,6))
ax = sns.distplot(x = df['age'], bins = 10)
plt.show()
```



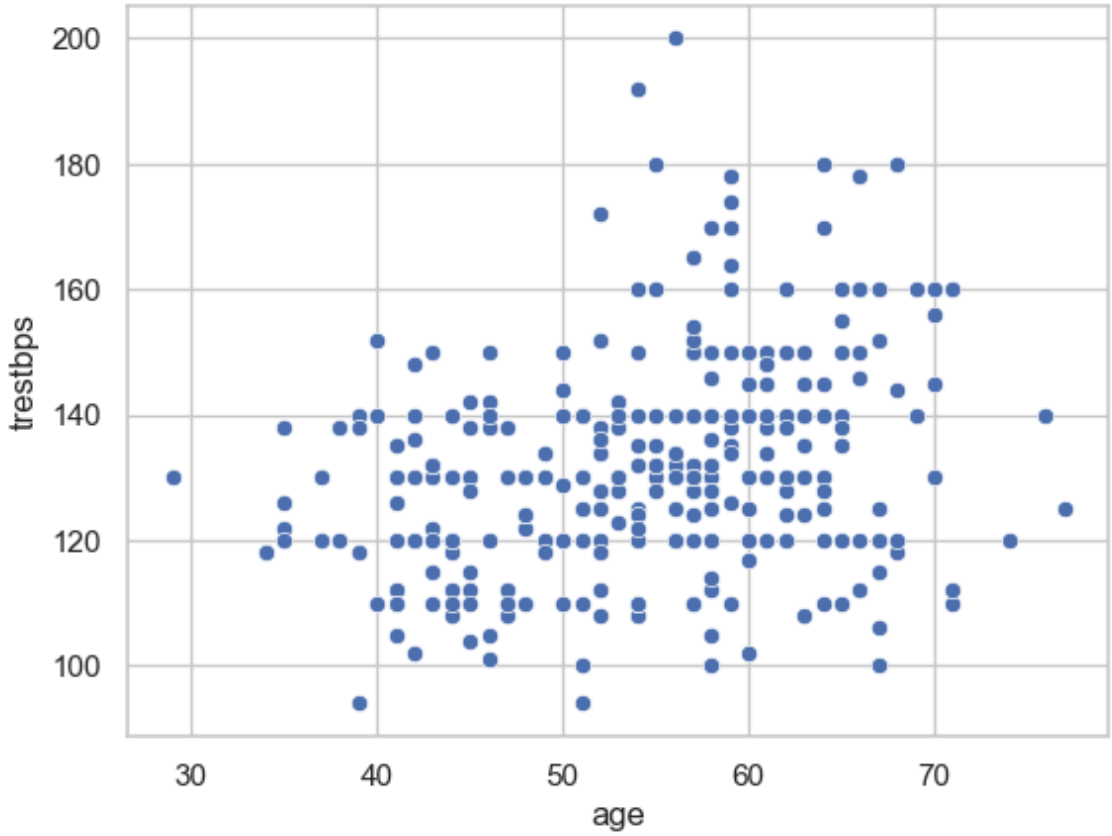
In [93]:
```
f, ax = plt.subplots(figsize=(8, 6))
sns.stripplot(x="target", y="age", data=df)
plt.show()
```



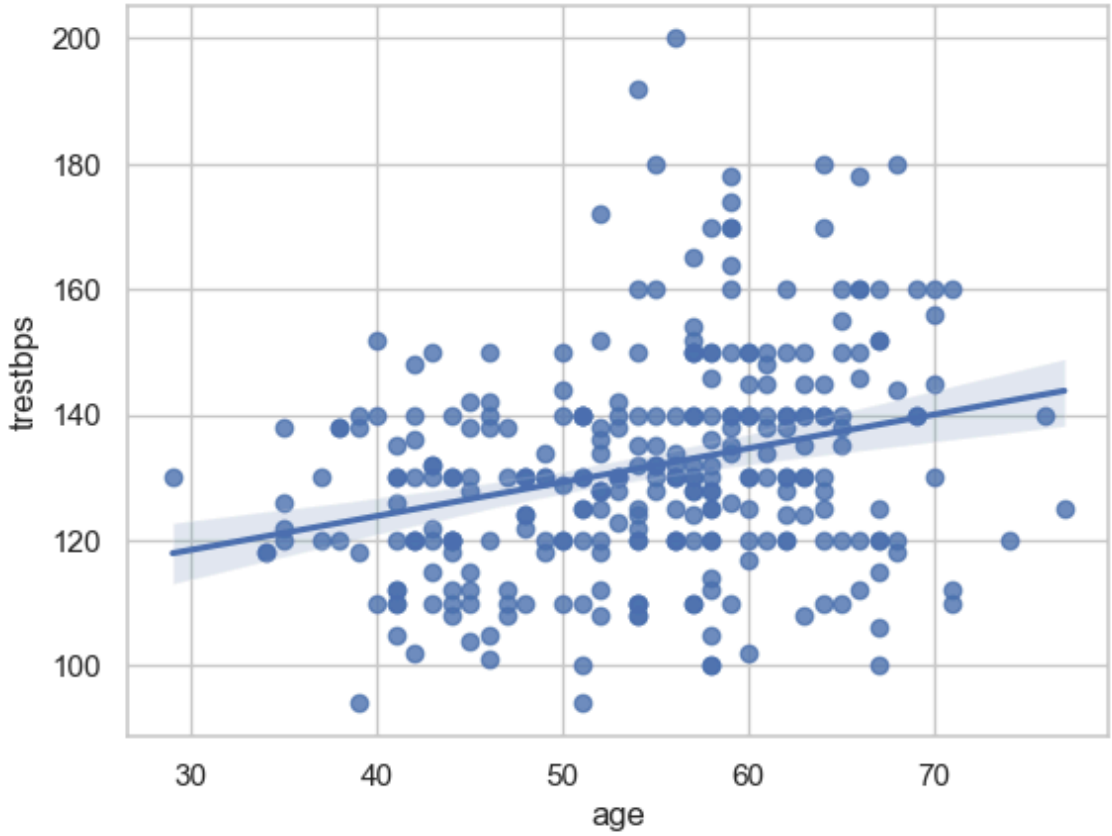
In [94]:
```
sns.boxplot(x="target", y="age", data=df)
plt.show()
```



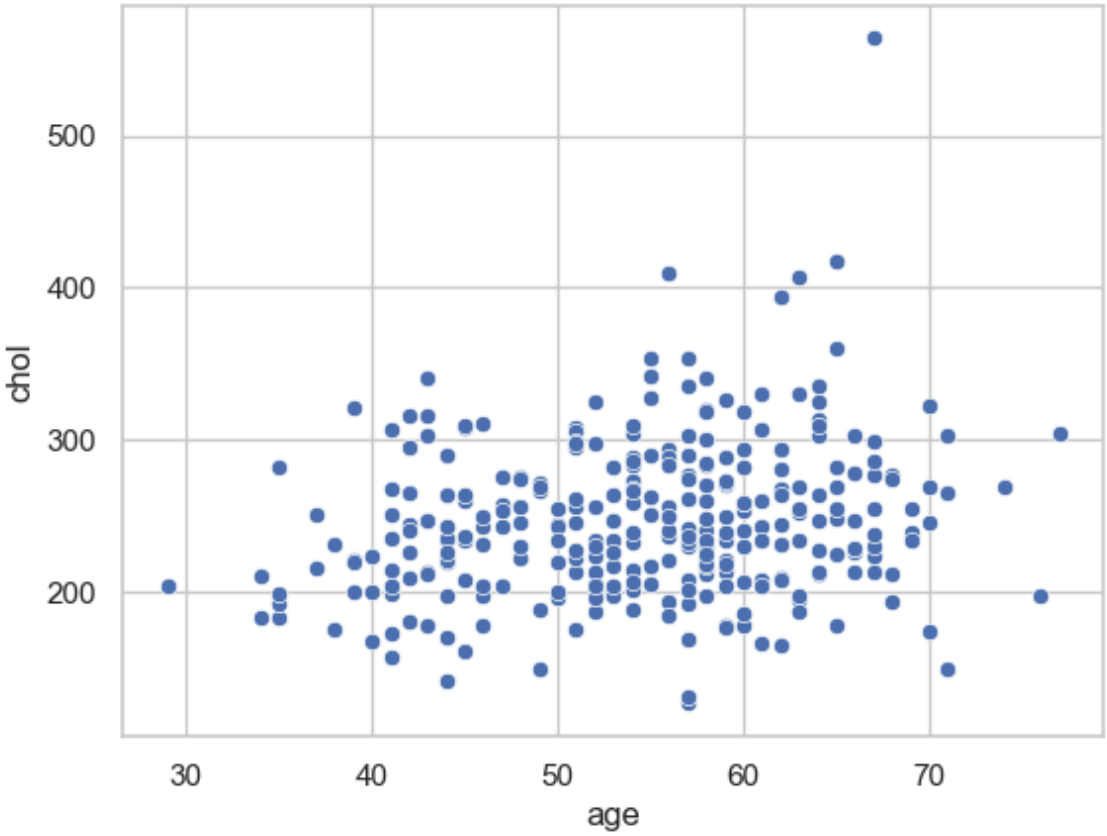
In [96]:
```
sns.scatterplot(x = "age", y = "trestbps", data=df)
plt.show()
```



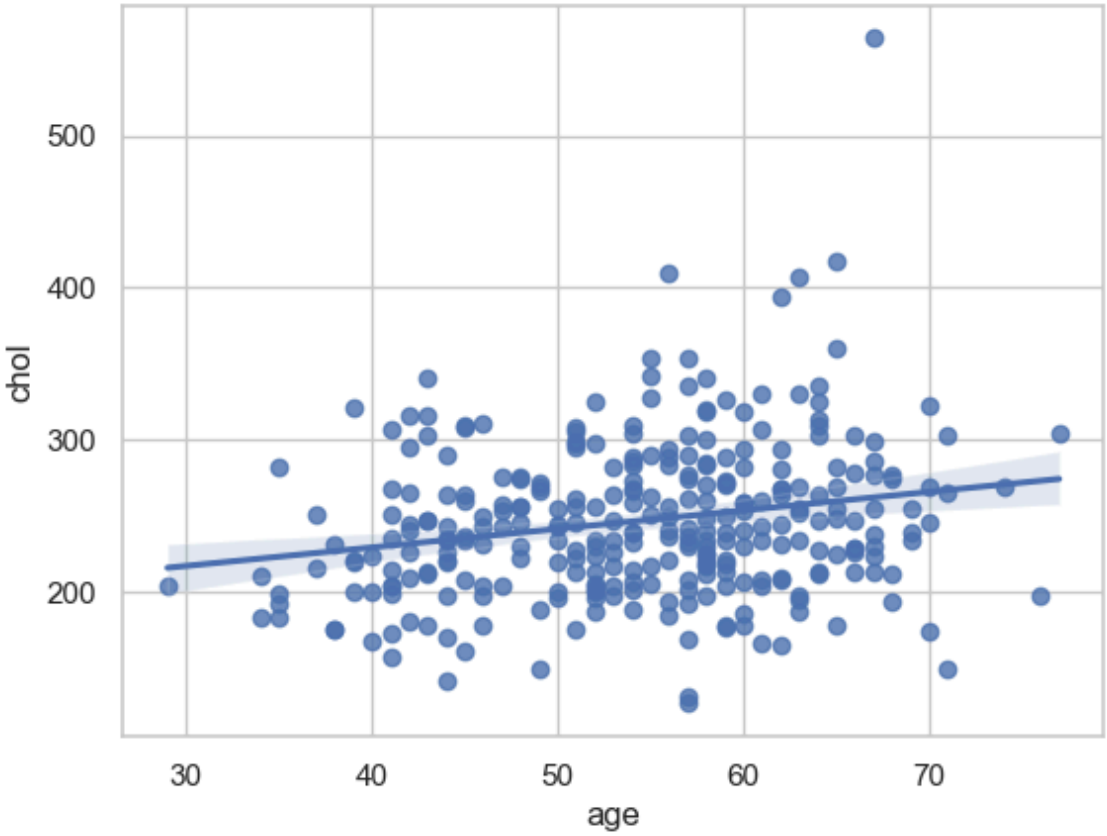
In [97]:
```
sns.regplot(x = "age", y = "trestbps", data=df)
plt.show()
```



In [98]:
```
sns.scatterplot(x = "age", y = "chol", data = df)
plt.show()
```

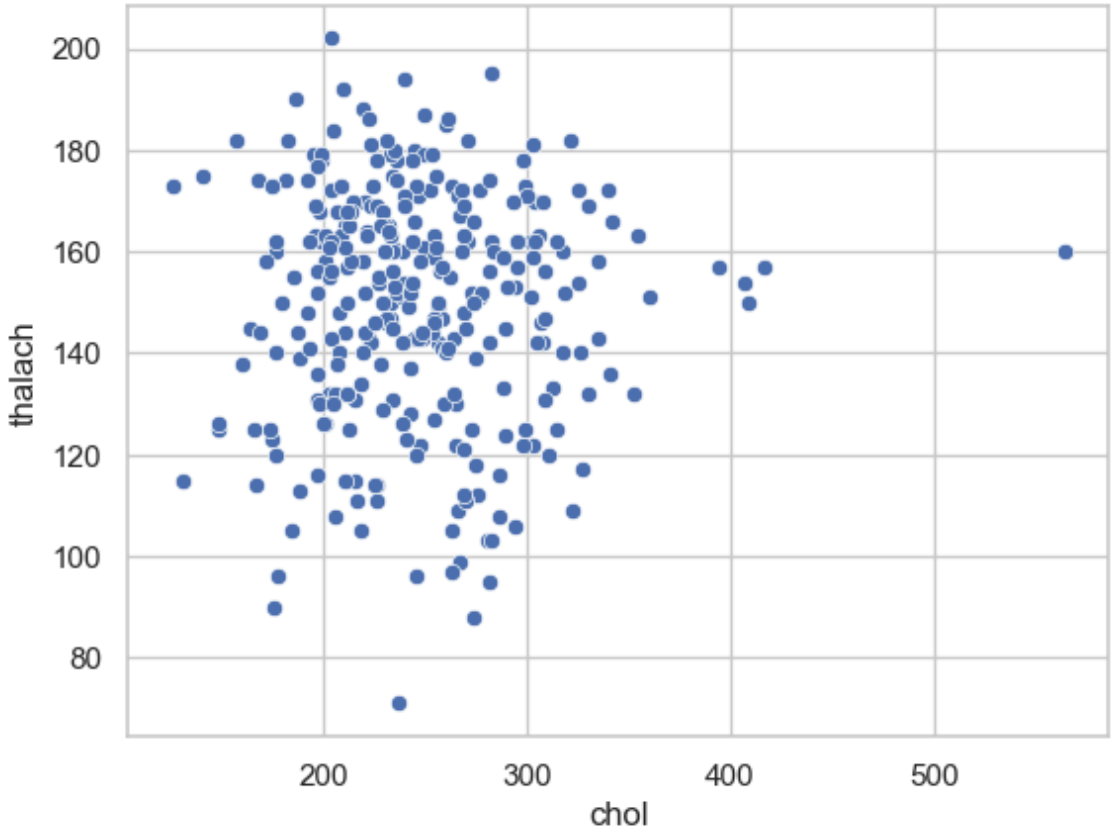


In [99]:
```
sns.regplot(x = "age", y ="chol", data = df)
plt.show()
```

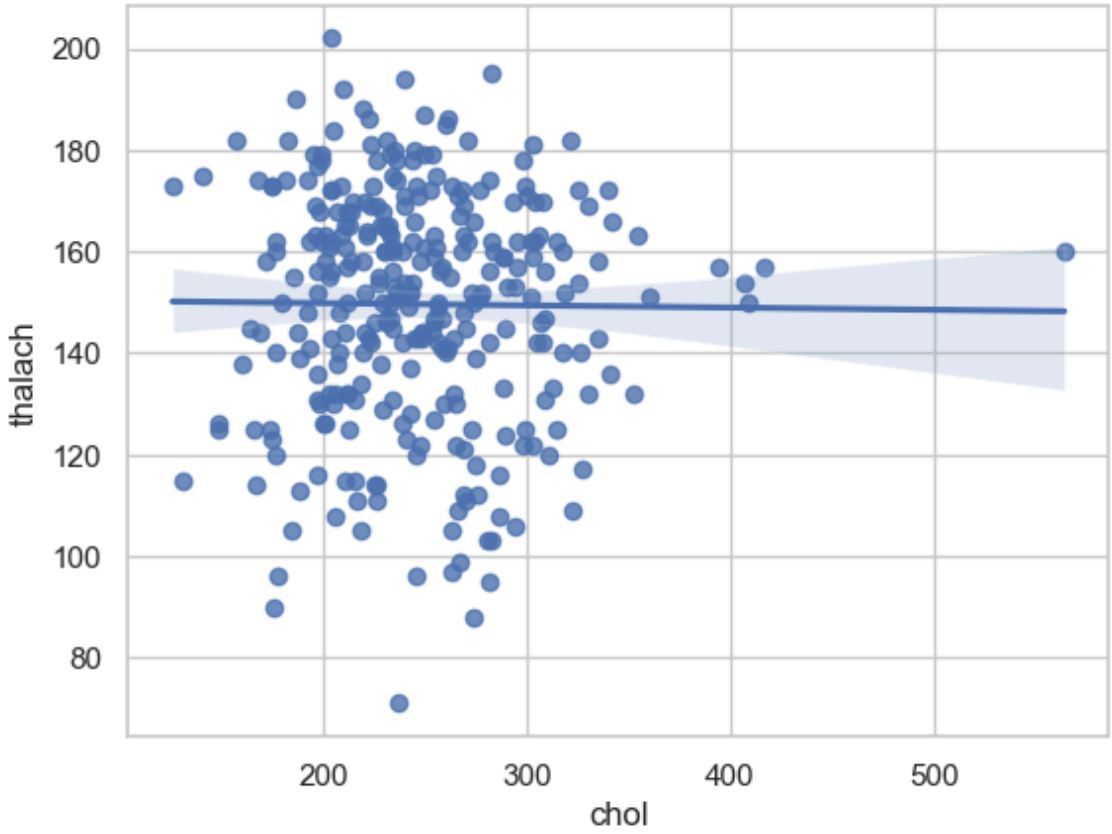


Analyze chol and thalach variable

In [100]:
```
sns.scatterplot(x = "chol", y = "thalach", data =df)
plt.show()
```



In [101]:
```
sns.regplot(x = "chol", y = "thalach", data =df)
plt.show()
```



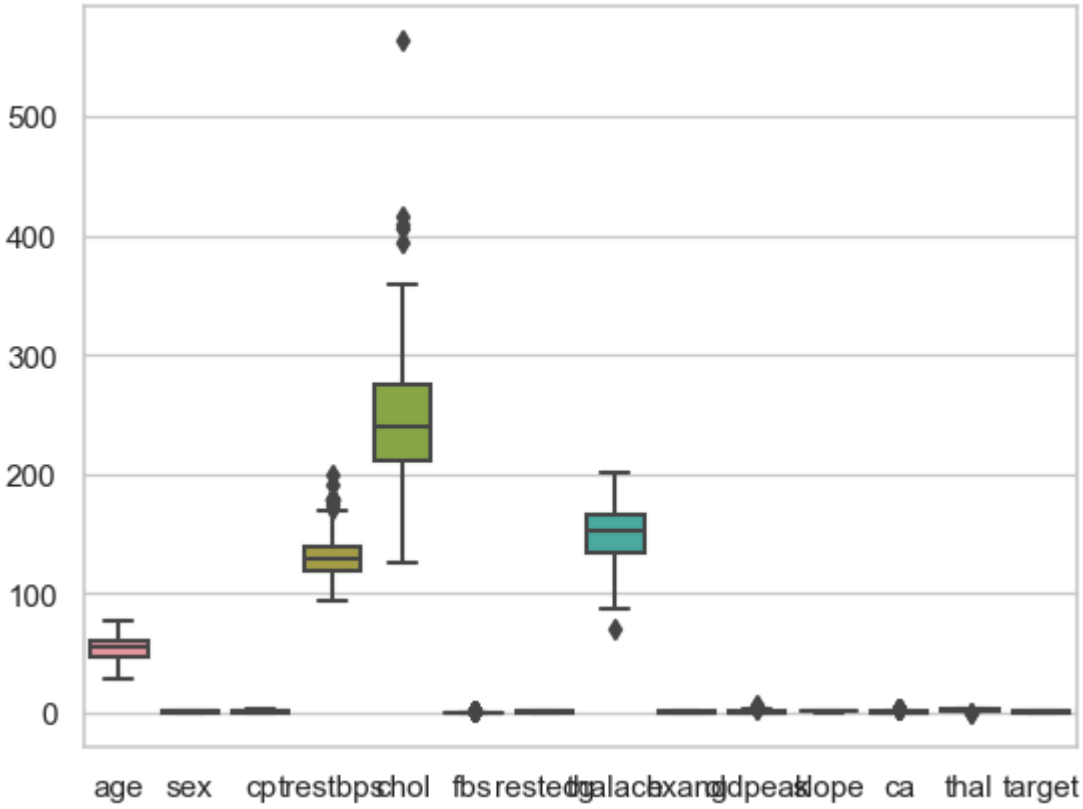
Outlier detection

Box - plot of data

In [108]:
```
df['age'].describe()
```

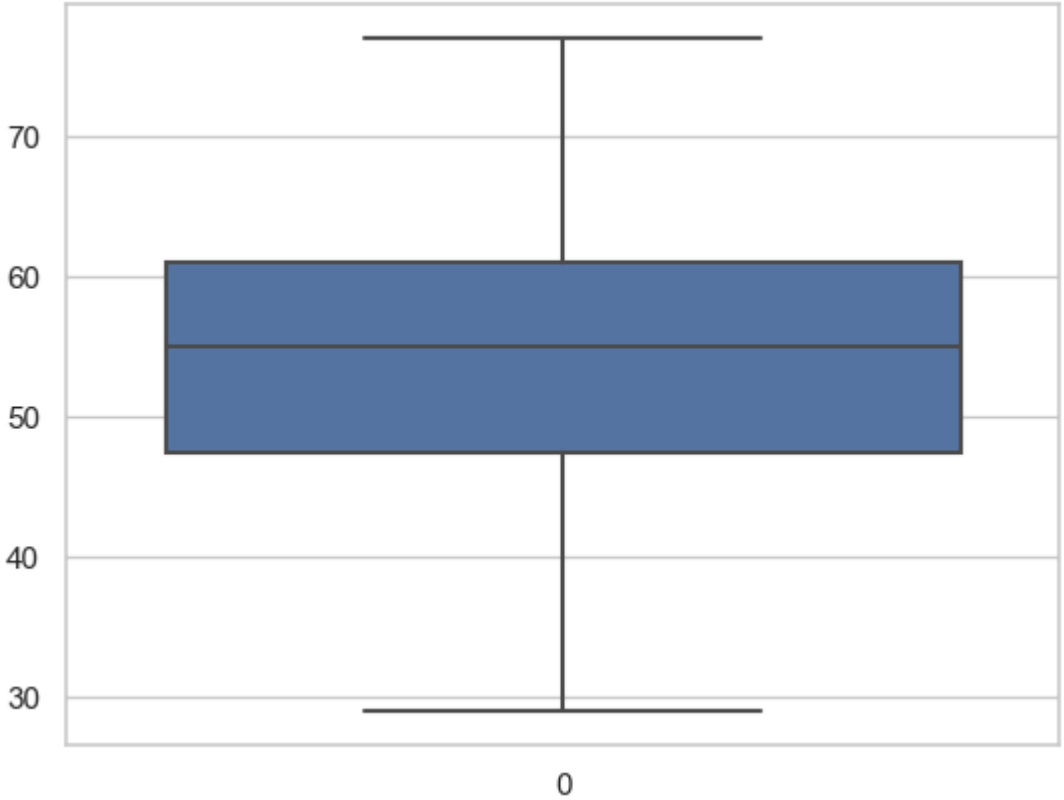
Out[108]:
```
count    303.000000
mean      54.366337
std        9.082101
min       29.000000
25%       47.500000
50%       55.000000
75%       61.000000
max       77.000000
Name: age, dtype: float64
```

In [102]:
```
sns.boxplot(df)
plt.show()
```



Box-plot of age variable

In [107]:
```
sns.boxplot(df['age'])
plt.show()
```

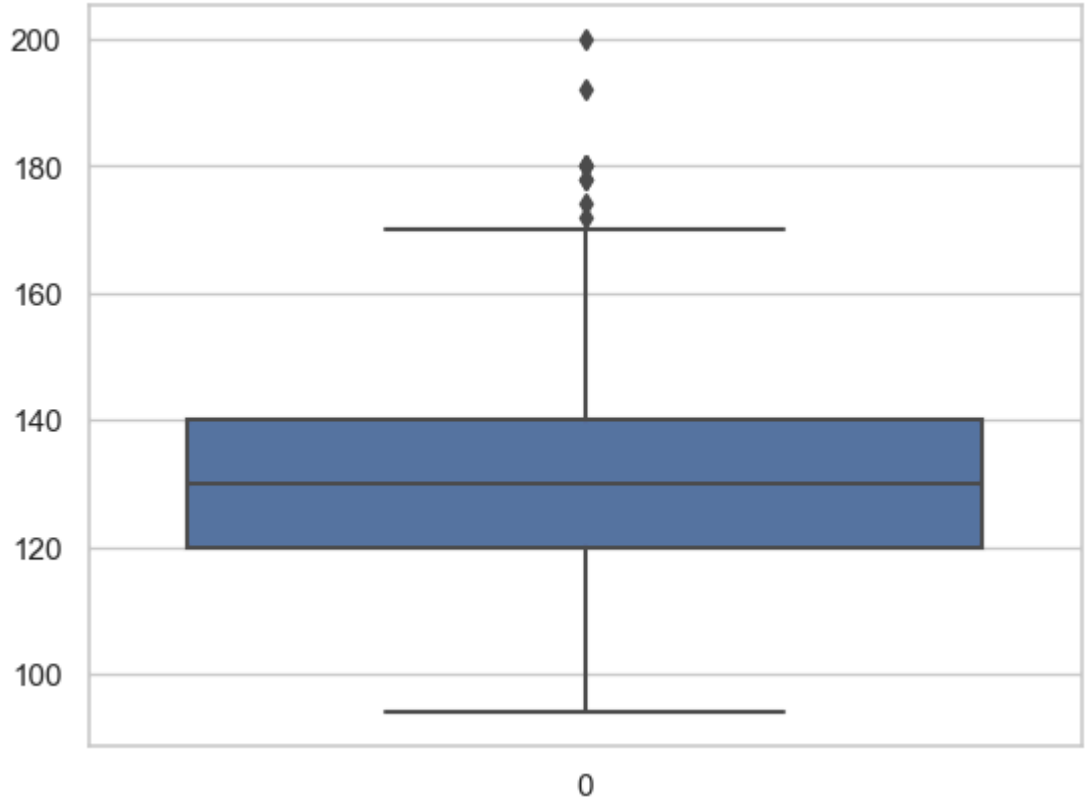


trestbps

In [109]:
```
df['trestbps'].describe()
```

Out[109]:
```
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 [111]:
```
sns.boxplot(df['trestbps'])
plt.show()
```

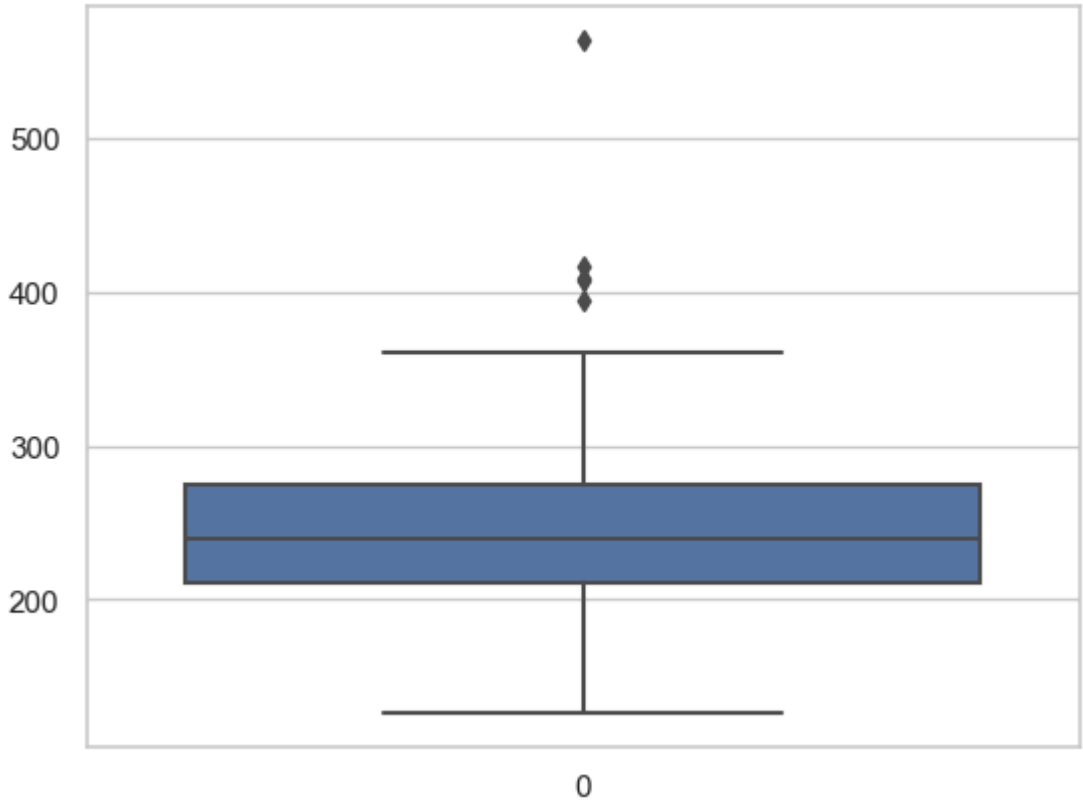


chol variable

In [112]:
```
df['chol'].describe()
```

Out[112]:
```
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 [113]:
```
sns.boxplot(df['chol'])
plt.show()
```

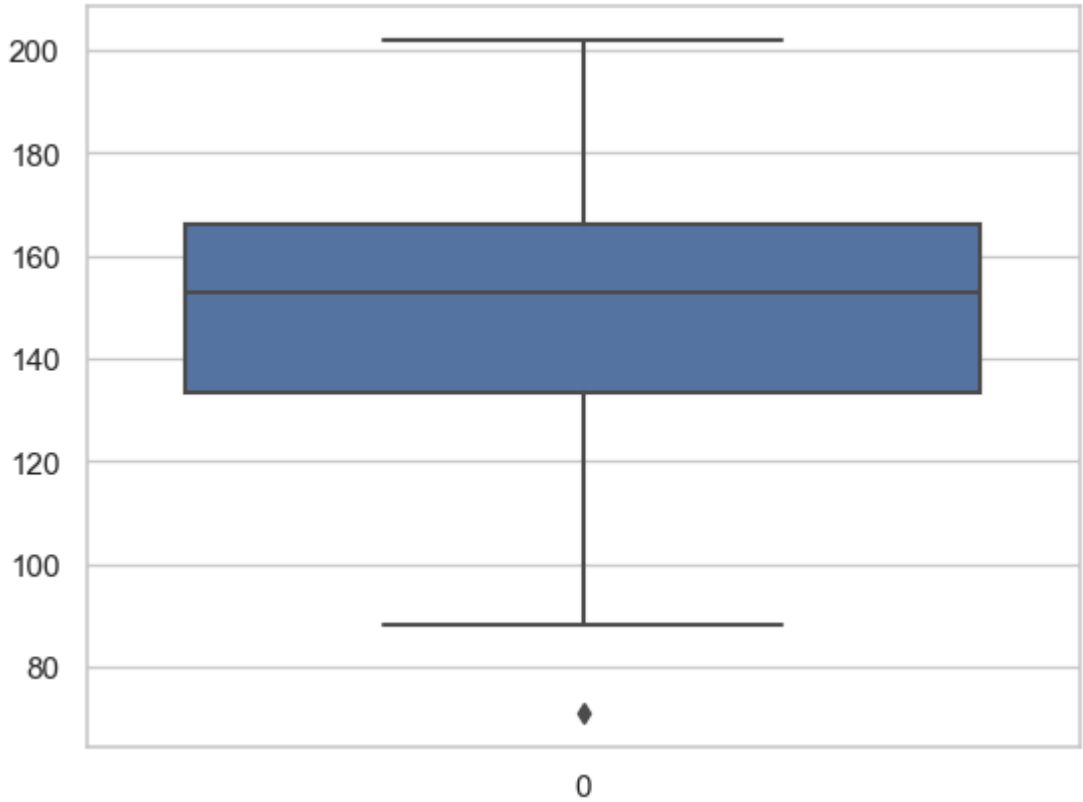


thalach variable

In [115]:
```
df['thalach'].describe()
```

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

In [116]:
```
sns.boxplot(df['thalach'])
plt.show()
```

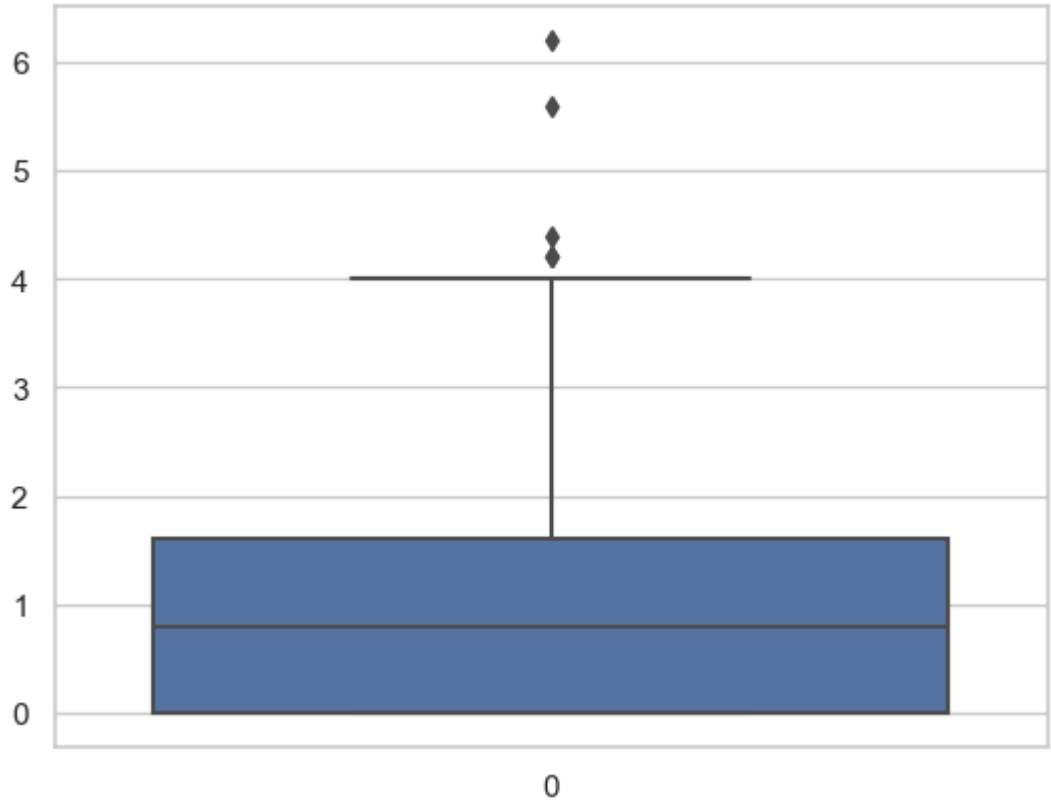


oldpeak variable

In [120]:
```
df['oldpeak'].describe()
```

Out[120]:
```
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 [121]:
```
sns.boxplot(df['oldpeak'])
plt.show()
```



In [ ]: