

```
In [8]: import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import scipy.stats as st
%matplotlib inline
```

```
In [10]: import warnings
warnings.filterwarnings("ignore")
```

```
In [12]: data=pd.read_csv(r'C:\Users\S SHYAMILI\OneDrive\Desktop\data science\28th - Seaborn
```

```
In [14]: data
```

```
Out[14]:
```

	age	sex	cp	trestbps	chol	fb	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
...
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



```
In [16]: data.shape
```

```
Out[16]: (303, 14)
```

```
In [20]: data.head()
```

Out[20]:

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



In [22]: `data.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
```

Dataset description

- The dataset contains several columns which are as follows -
 - age : age in years
 - sex : (1 = male; 0 = female)
 - cp : chest pain type
 - trestbps : resting blood pressure (in mm Hg on admission to the hospital)
 - chol : serum cholesterol in mg/dl
 - fbs : (fasting blood sugar > 120 mg/dl) (1 = true; 0 = false)
 - restecg : resting electrocardiographic results
 - thalach : maximum heart rate achieved
 - exang : exercise induced angina (1 = yes; 0 = no)
 - oldpeak : ST depression induced by exercise relative to rest
 - slope : the slope of the peak exercise ST segment

- ca : number of major vessels (0-3) colored by flourosopy
- thal : 3 = normal; 6 = fixed defect; 7 = reversable defect
- target : 1 or 0

```
In [26]: data.dtypes
```

```
Out[26]: 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 [30]: data.describe()
```

	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 [34]: data.describe(include='all')
```

	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 [38]: `data.columns`

Out[38]: `Index(['age', 'sex', 'cp', 'trestbps', 'chol', 'fbs', 'restecg', 'thalach', 'exang', 'oldpeak', 'slope', 'ca', 'thal', 'target'], dtype='object')`

In [40]: `data.nunique()`

Out[40]:

age	41
sex	2
cp	4
trestbps	49
chol	152
fbs	2
restecg	3
thalach	91
exang	2
oldpeak	40
slope	3
ca	5
thal	4
target	2
	dtype: int64

In [42]: `data.target.nunique()`

Out[42]: 2

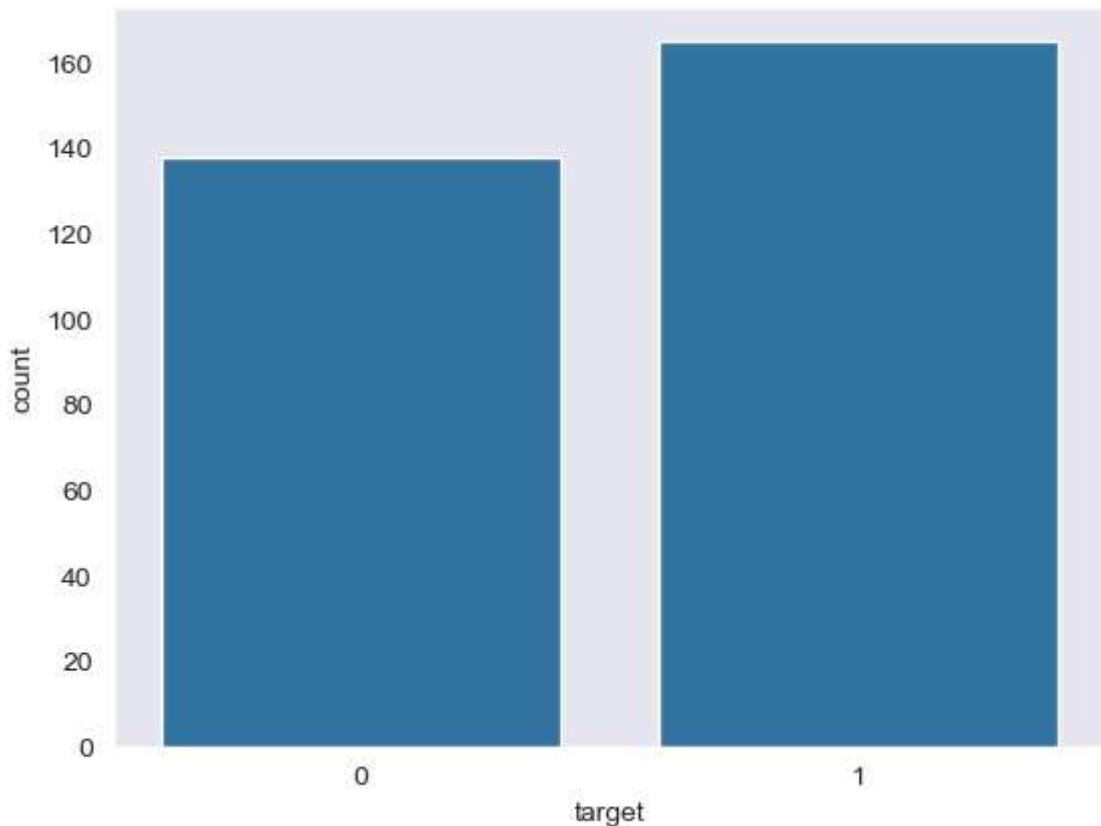
In [44]: `data.target.unique()`

Out[44]: `array([1, 0], dtype=int64)`

In [50]: `data.target.value_counts()`

```
Out[50]: target
1    165
0    138
Name: count, dtype: int64
```

```
In [68]: sns.set_style('dark')
plt.subplots()
sns.countplot(data=data,x='target')
plt.show()
```

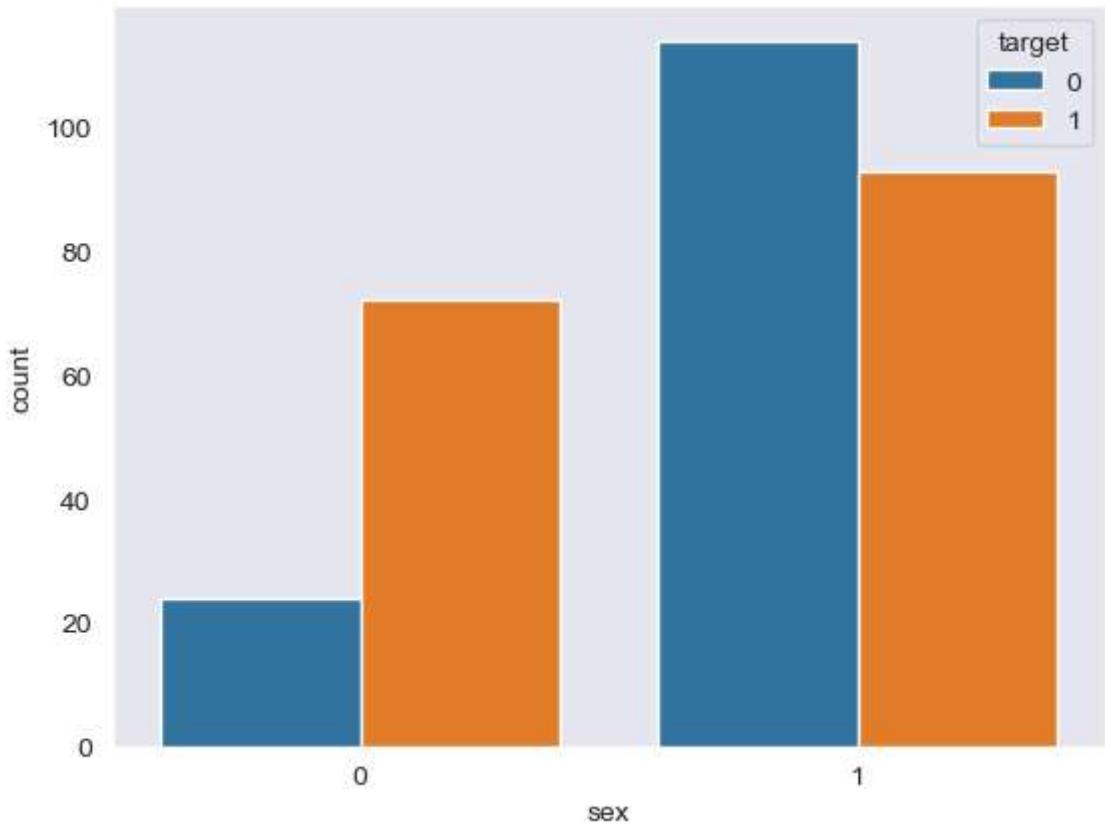


```
In [72]: data.groupby('sex').target.value_counts()
```

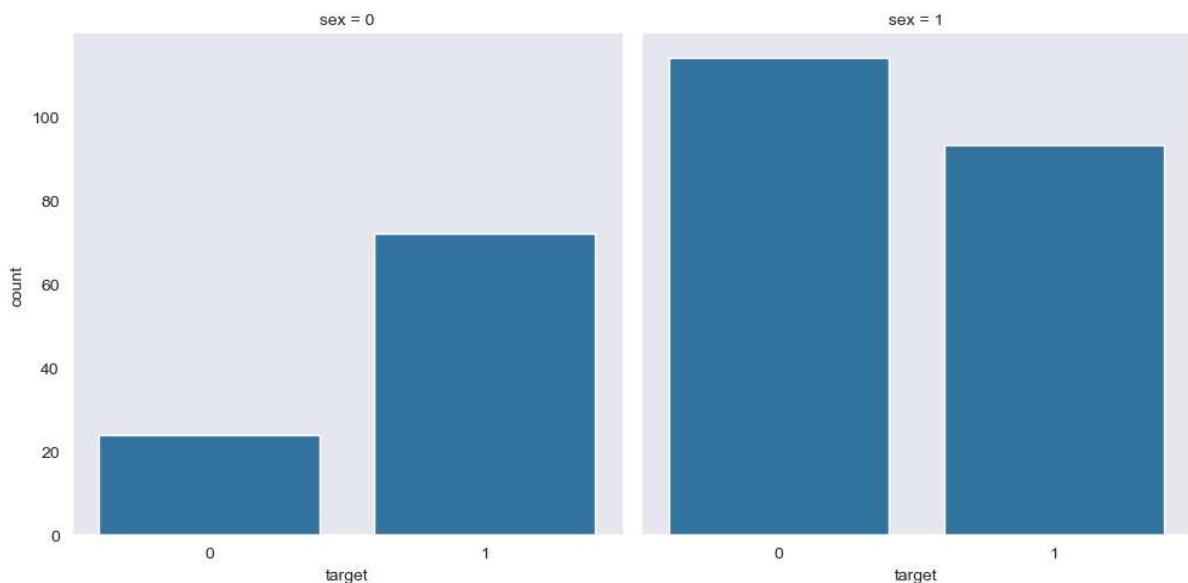
```
Out[72]: sex  target
0    1        72
      0        24
1    0       114
      1        93
Name: count, dtype: int64
```

```
In [80]: plt.subplots()
sns.countplot(data=data,x='sex',hue='target')
```

```
Out[80]: <Axes: xlabel='sex', ylabel='count'>
```

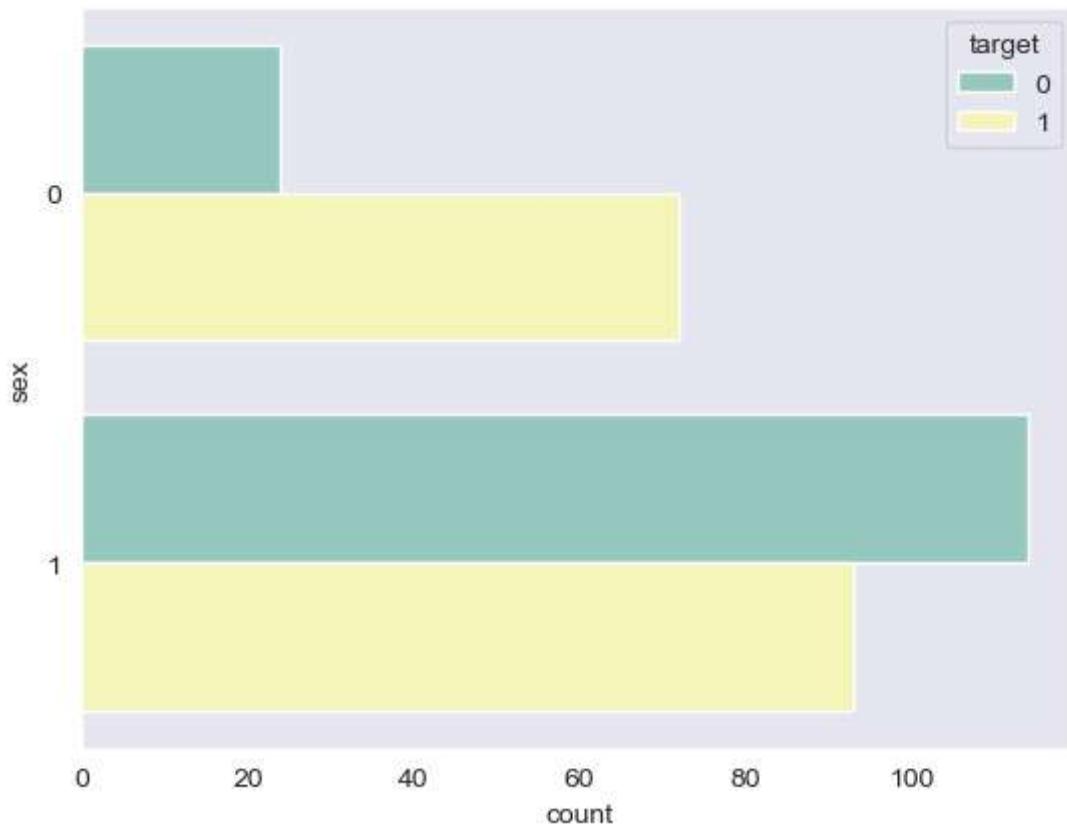


```
In [84]: ax = sns.catplot(x="target", col="sex", data=data, kind="count", height=5, aspect=1)
```



```
In [90]: plt.subplots()  
sns.countplot(data=data,y='sex',hue='target',palette='Set3')
```

```
Out[90]: <Axes: xlabel='count', ylabel='sex'>
```



```
In [92]: correlation = data.corr()
```

```
In [94]: correlation
```

Out[94]:

	age	sex	cp	trestbps	chol	fbs	restecg	thalach
age	1.000000	-0.098447	-0.068653	0.279351	0.213678	0.121308	-0.116211	-0.398522
sex	-0.098447	1.000000	-0.049353	-0.056769	-0.197912	0.045032	-0.058196	-0.044020
cp	-0.068653	-0.049353	1.000000	0.047608	-0.076904	0.094444	0.044421	0.295762
trestbps	0.279351	-0.056769	0.047608	1.000000	0.123174	0.177531	-0.114103	-0.046698
chol	0.213678	-0.197912	-0.076904	0.123174	1.000000	0.013294	-0.151040	-0.00940
fbs	0.121308	0.045032	0.094444	0.177531	0.013294	1.000000	-0.084189	-0.008196
restecg	-0.116211	-0.058196	0.044421	-0.114103	-0.151040	-0.084189	1.000000	0.044123
thalach	-0.398522	-0.044020	0.295762	-0.046698	-0.009940	-0.008567	0.044123	1.000000
exang	0.096801	0.141664	-0.394280	0.067616	0.067023	0.025665	-0.070733	-0.378522
oldpeak	0.210013	0.096093	-0.149230	0.193216	0.053952	0.005747	-0.058770	-0.344020
slope	-0.168814	-0.030711	0.119717	-0.121475	-0.004038	-0.059894	0.093045	0.386765
ca	0.276326	0.118261	-0.181053	0.101389	0.070511	0.137979	-0.072042	-0.213402
thal	0.068001	0.210041	-0.161736	0.062210	0.098803	-0.032019	-0.011981	-0.096698
target	-0.225439	-0.280937	0.433798	-0.144931	-0.085239	-0.028046	0.137230	0.421340



In [96]:

correlation.target.sort_values(ascending=False)

Out[96]:

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 [100...]

data.cp.unique()

Out[100...]

4

In [102...]

data.cp.unique()

Out[102...]

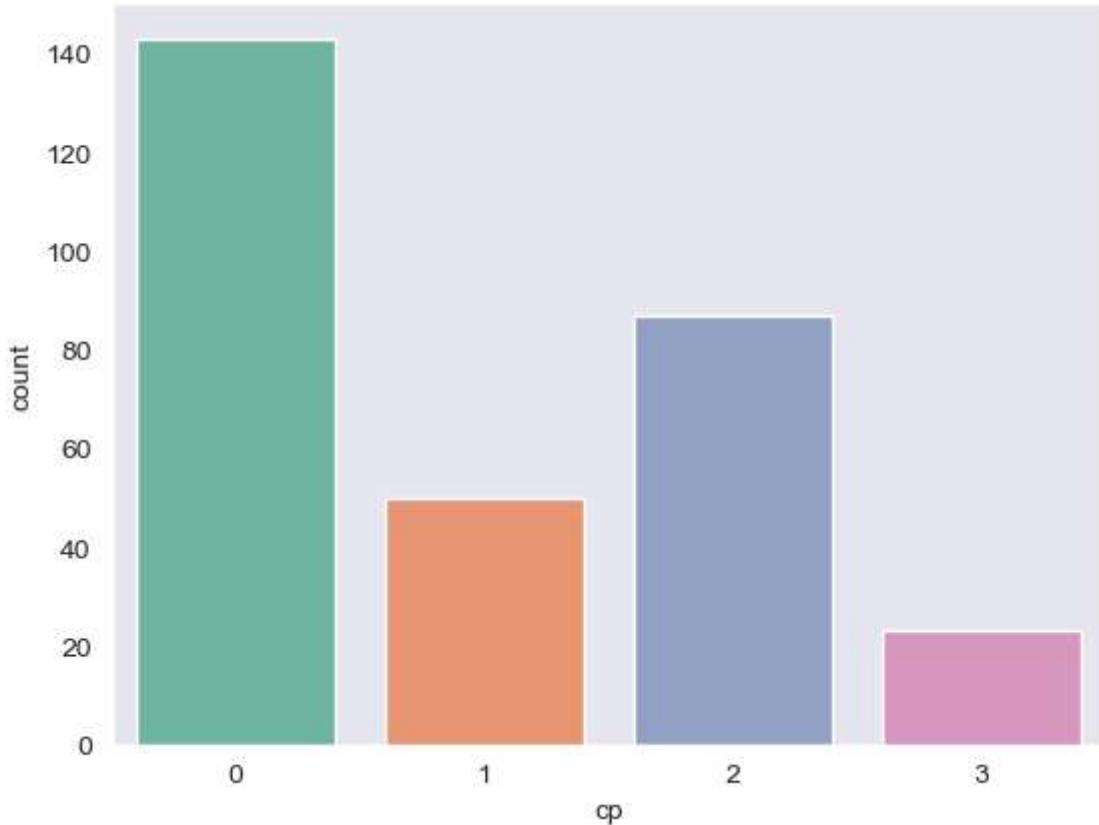
array([3, 2, 1, 0], dtype=int64)

```
In [104... data.cp.value_counts()
```

```
Out[104... cp
0    143
2     87
1     50
3     23
Name: count, dtype: int64
```

```
In [110... plt.subplot()
sns.countplot(data=data,x='cp',palette='Set2')
```

```
Out[110... <Axes: xlabel='cp', ylabel='count'>
```

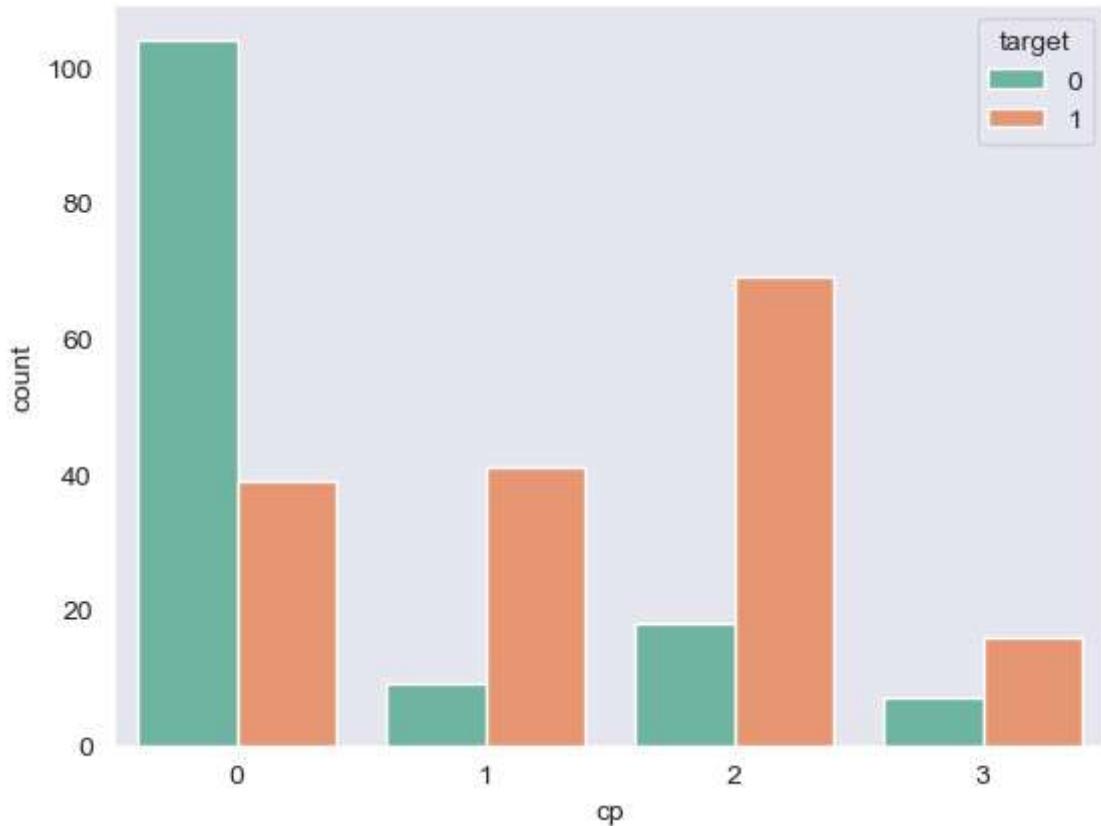


```
In [114... data.groupby('cp').target.value_counts()
```

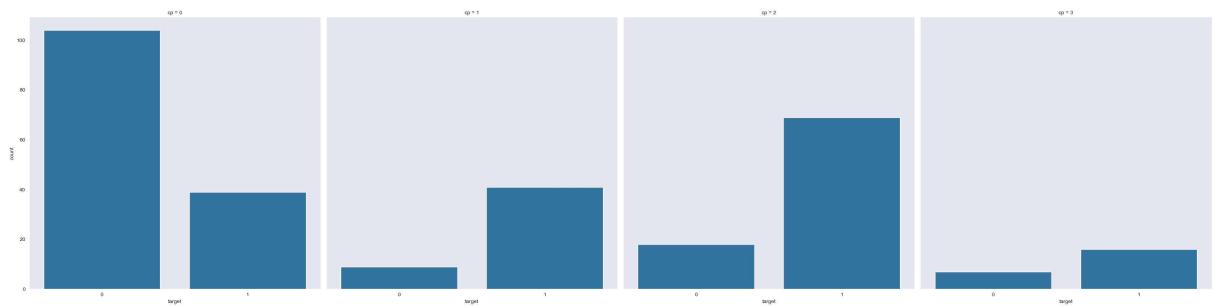
```
Out[114... cp  target
0   0      104
     1      39
1   1      41
     0       9
2   1      69
     0      18
3   1      16
     0       7
Name: count, dtype: int64
```

```
In [116... plt.subplot()
sns.countplot(data=data,x='cp',hue='target',palette='Set2')
```

```
Out[116... <Axes: xlabel='cp', ylabel='count'>
```



```
In [120... ax = sns.catplot(x="target", col="cp", data=data, kind="count", height=8, aspect=1)
```



```
In [124... data.thalach.value_counts()
```

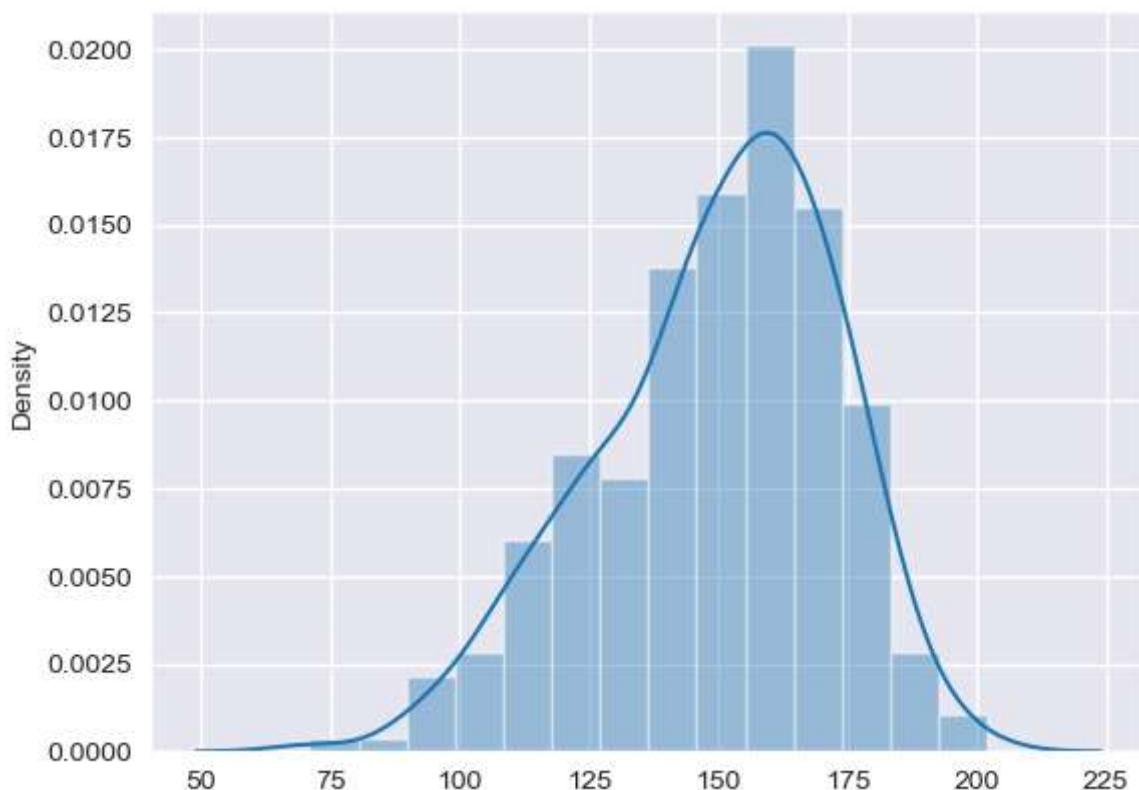
```
Out[124... thalach
162    11
160     9
163     9
152     8
173     8
...
202     1
184     1
121     1
192     1
90      1
Name: count, Length: 91, dtype: int64
```

```
In [126...]: data.thalach.unique()
```

```
Out[126...]: 91
```

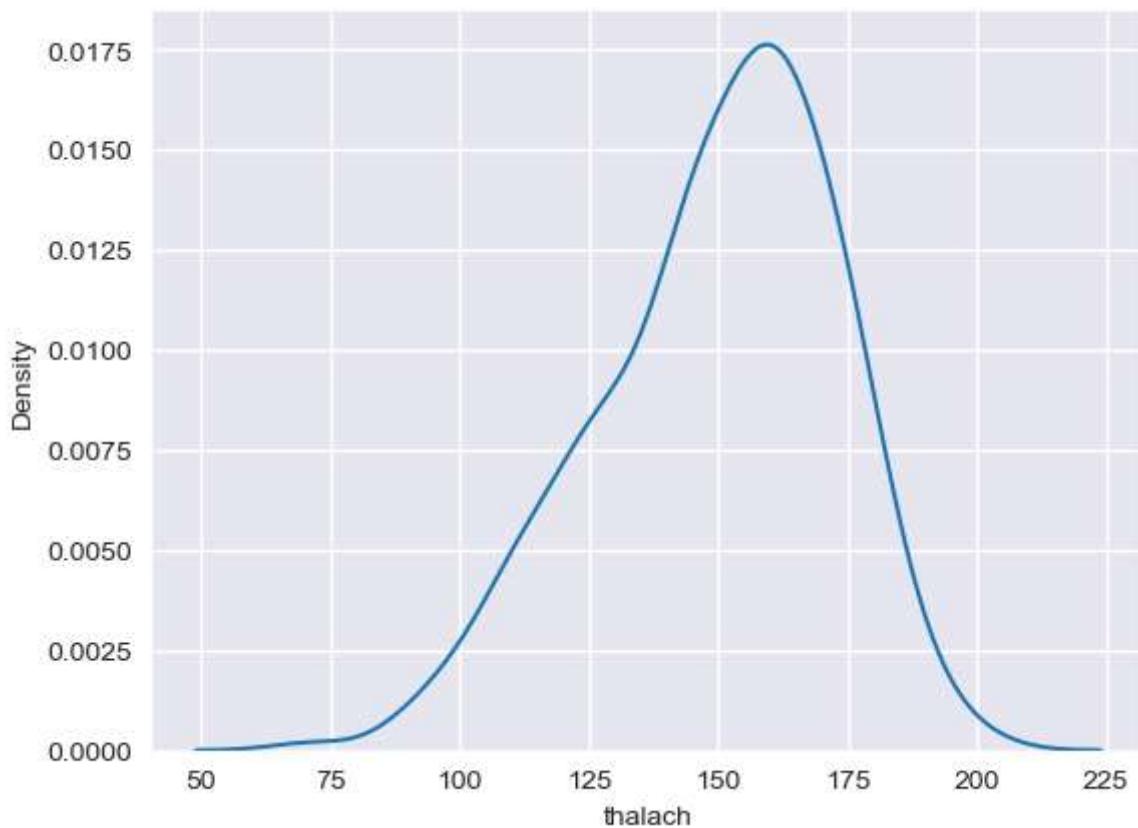
```
In [144...]: sns.set_style("darkgrid")
plt.subplots()
pd.Series(data['thalach'], name="thalach variable")
sns.distplot(x=data['thalach'])
```

```
Out[144...]: <Axes: ylabel='Density'>
```



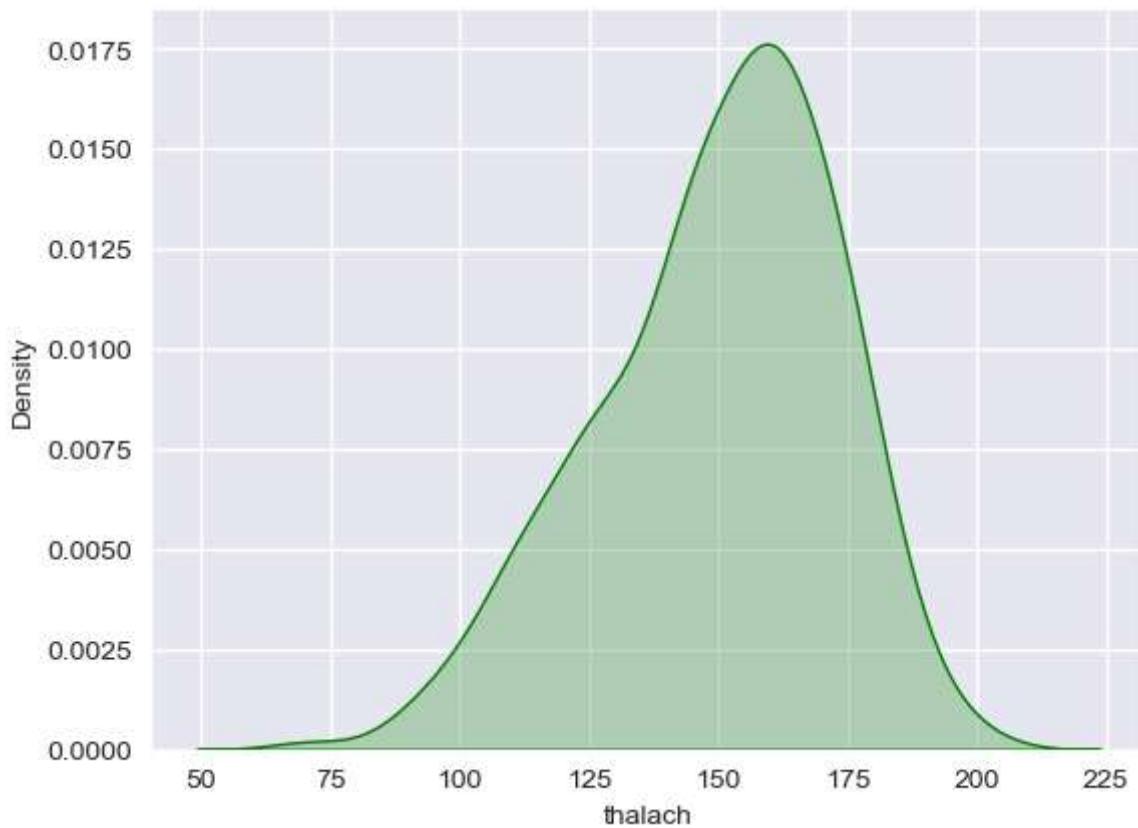
```
In [156...]: sns.kdeplot(data.thalach)
```

```
plt.show()
```



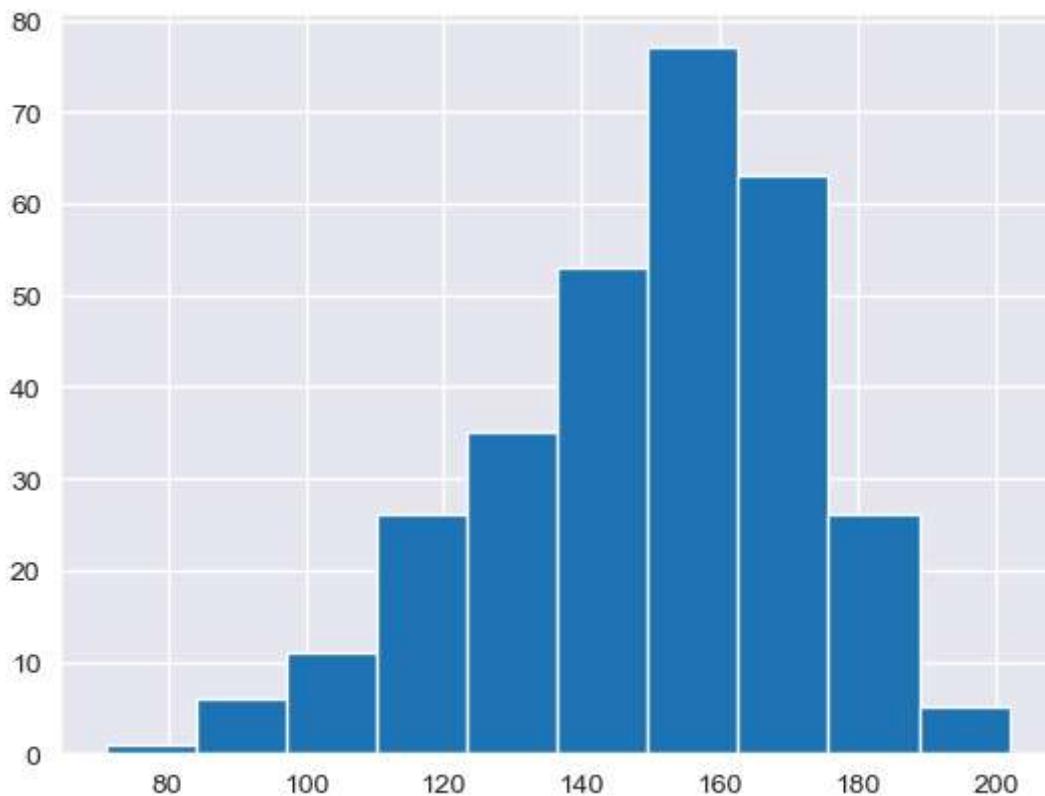
```
In [160]: sns.kdeplot(data.thalach, shade=True, color='g')
```

```
Out[160]: <Axes: xlabel='thalach', ylabel='Density'>
```



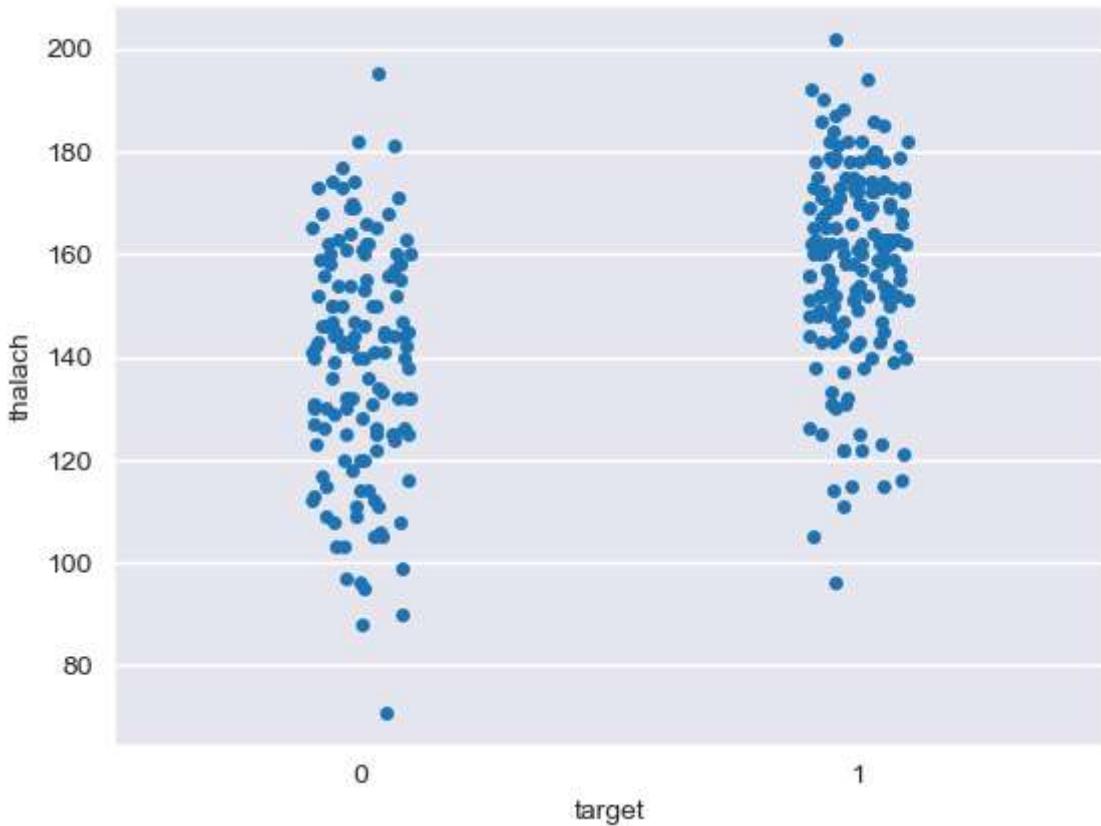
```
In [164... plt.hist(data.thalach)
```

```
Out[164... (array([ 1.,  6., 11., 26., 35., 53., 77., 63., 26.,  5.]),
 array([ 71. ,  84.1,  97.2, 110.3, 123.4, 136.5, 149.6, 162.7, 175.8,
        188.9, 202. ]),
 <BarContainer object of 10 artists>)
```



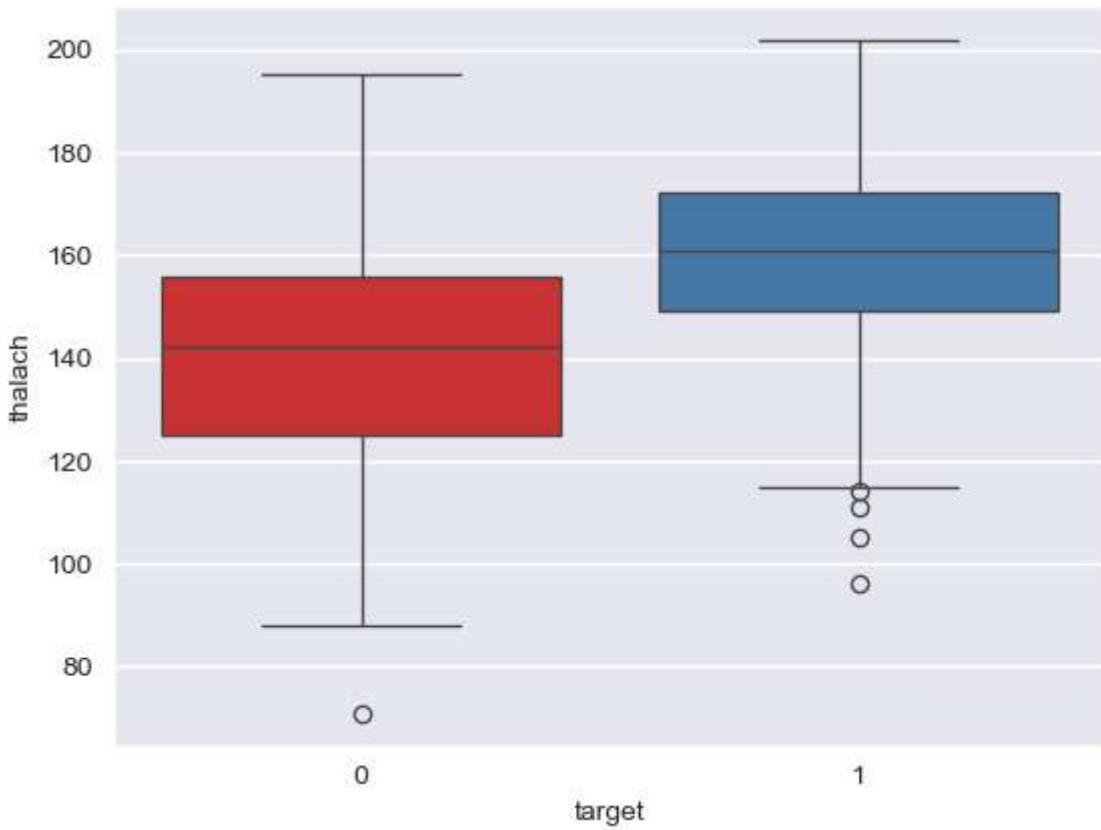
```
In [168... plt.subplots()
sns.stripplot(data=data,x='target',y='thalach')
```

```
Out[168... <Axes: xlabel='target', ylabel='thalach'>
```



```
In [174]: sns.boxplot(data=data,x='target',y='thalach',palette='Set1')
```

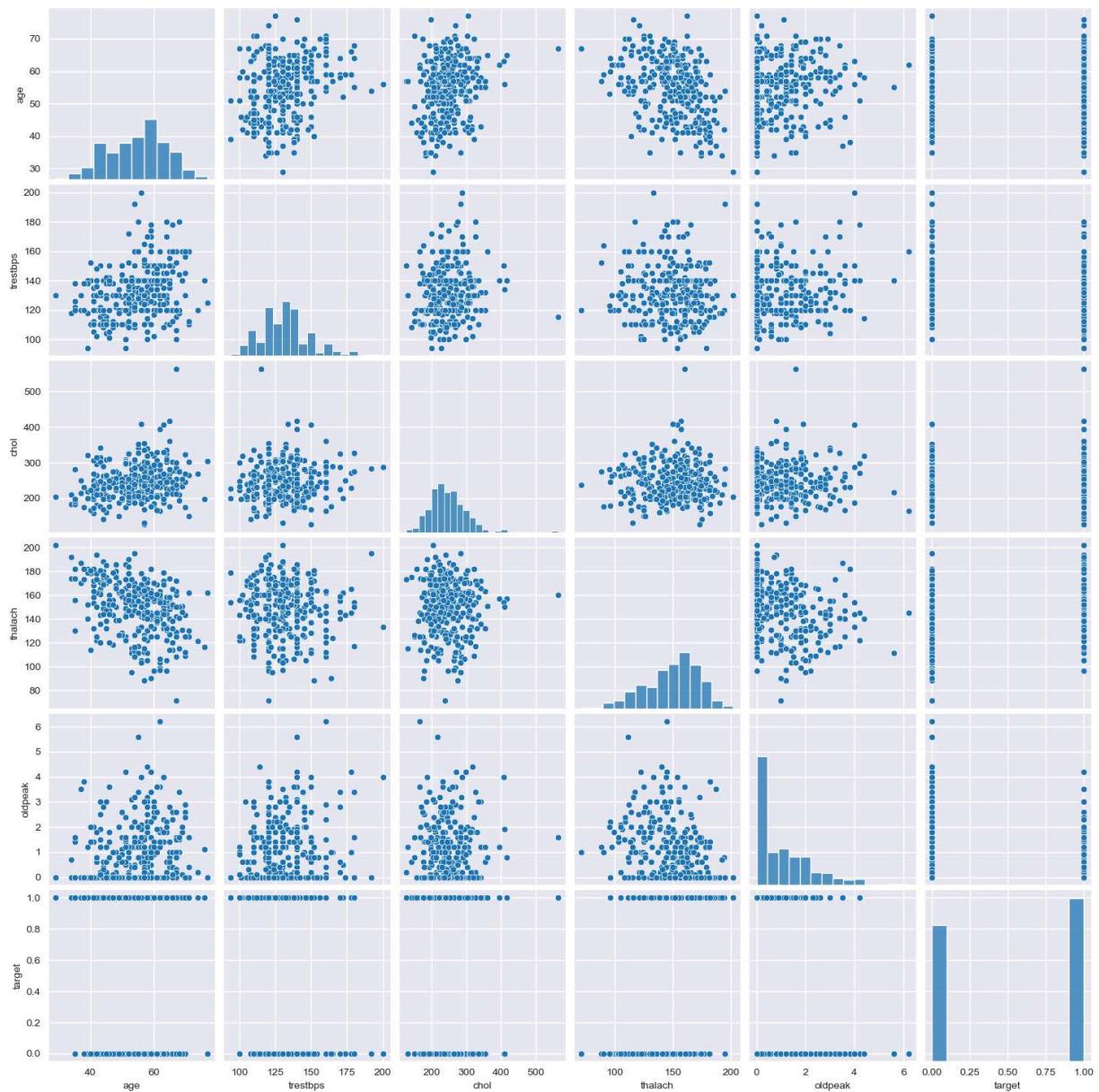
```
Out[174]: <Axes: xlabel='target', ylabel='thalach'>
```



```
In [200... plt.figure(figsize=(18,6))
plt.title('Correlation heatmap of Heart disease data set')
a=sns.heatmap(correlation,square=True,annot=True,fmt='.2f',linecolor='white')
```



```
In [209... num_var = ['age', 'trestbps', 'chol', 'thalach', 'oldpeak', 'target' ]
sns.pairplot(data[num_var],kind='scatter',diag_kind='hist')
plt.show()
```



In [211...]: `data.age.nunique()`

Out[211...]: 41

In [213...]: `data.age.describe()`

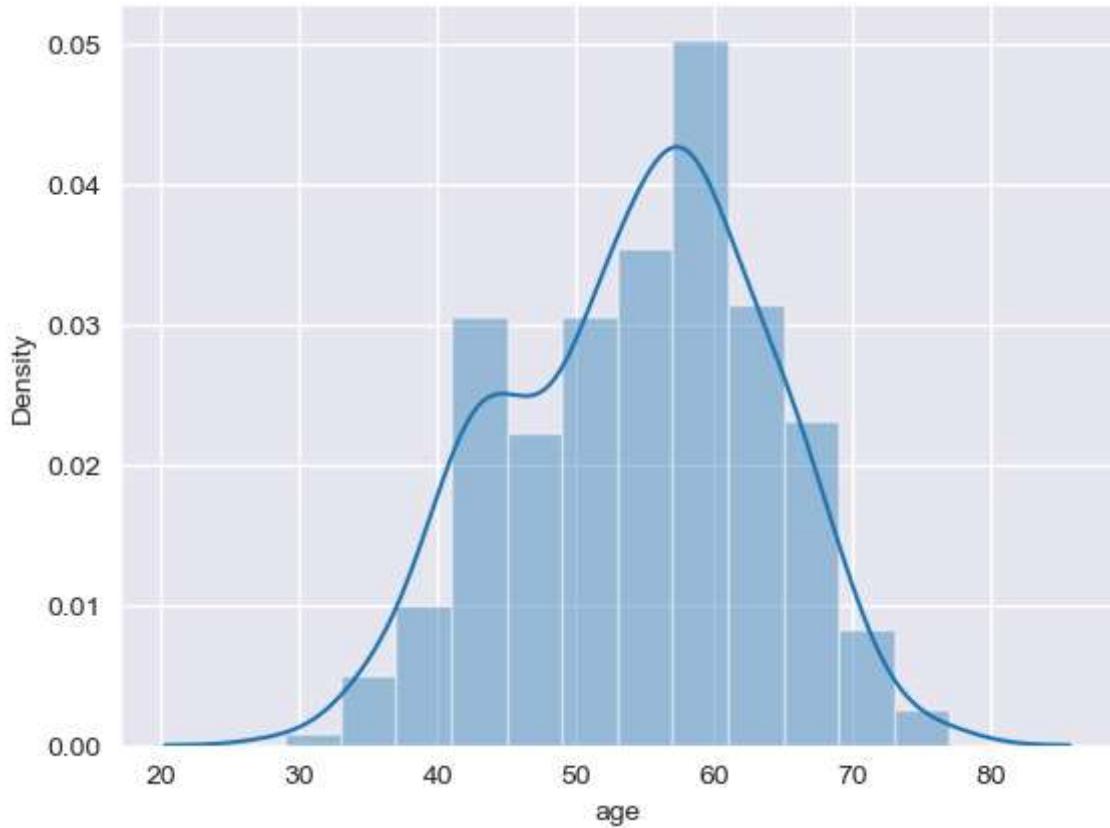
Out[213...]:

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

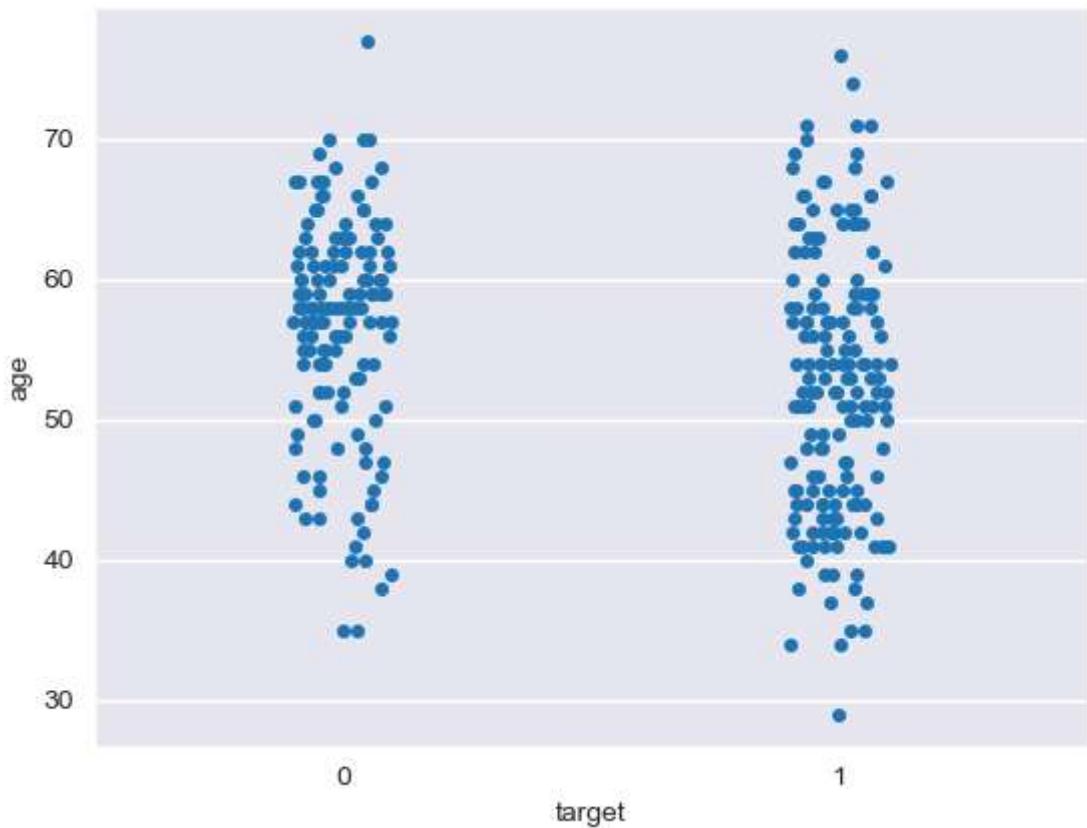
In [221...]: `sns.distplot(data.age)`

```
Out[221... <Axes: xlabel='age', ylabel='Density'>
```



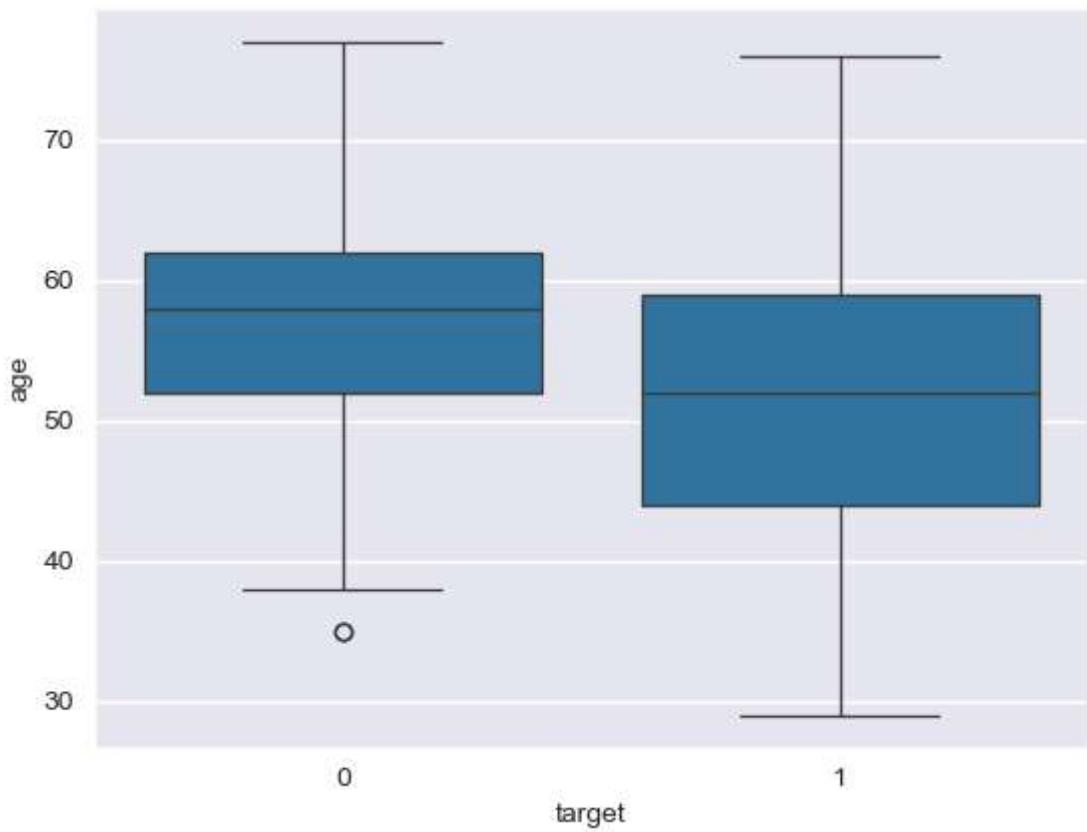
```
In [227... sns.stripplot(data=data,x='target',y='age')
```

```
Out[227... <Axes: xlabel='target', ylabel='age'>
```

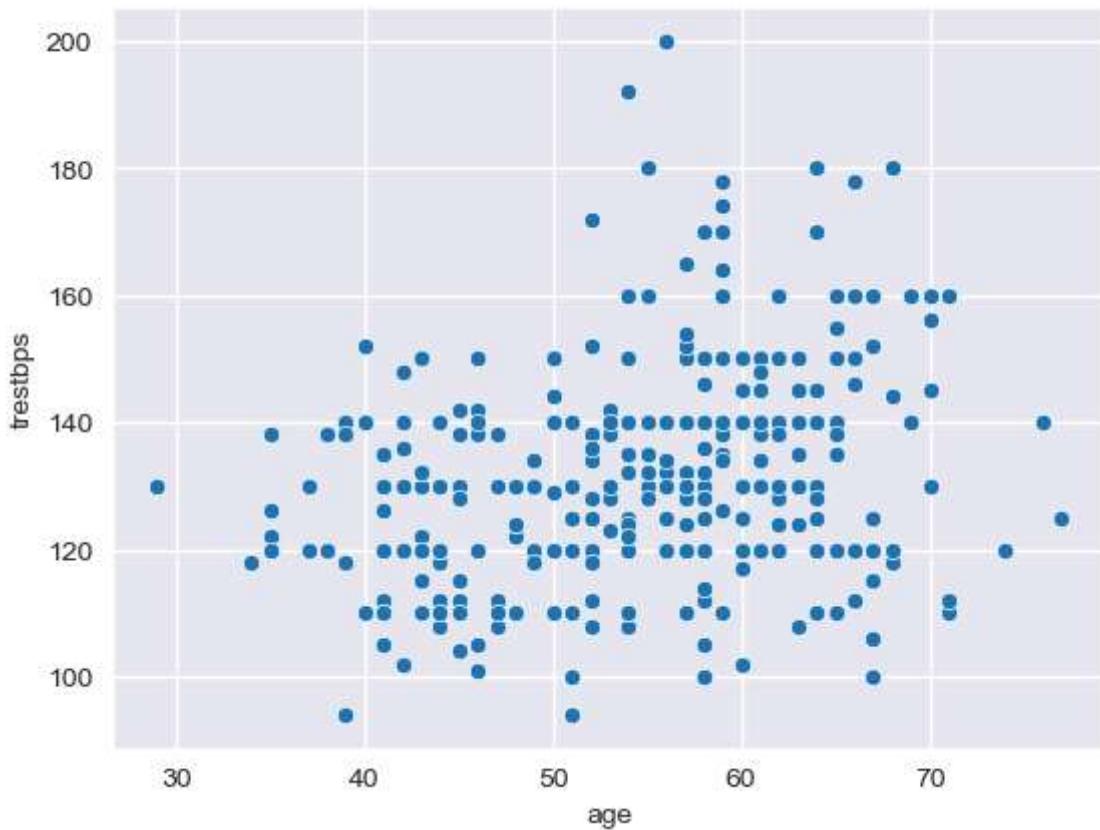


```
In [229]: sns.boxplot(x="target", y="age", data=data)
```

```
Out[229]: <Axes: xlabel='target', ylabel='age'>
```

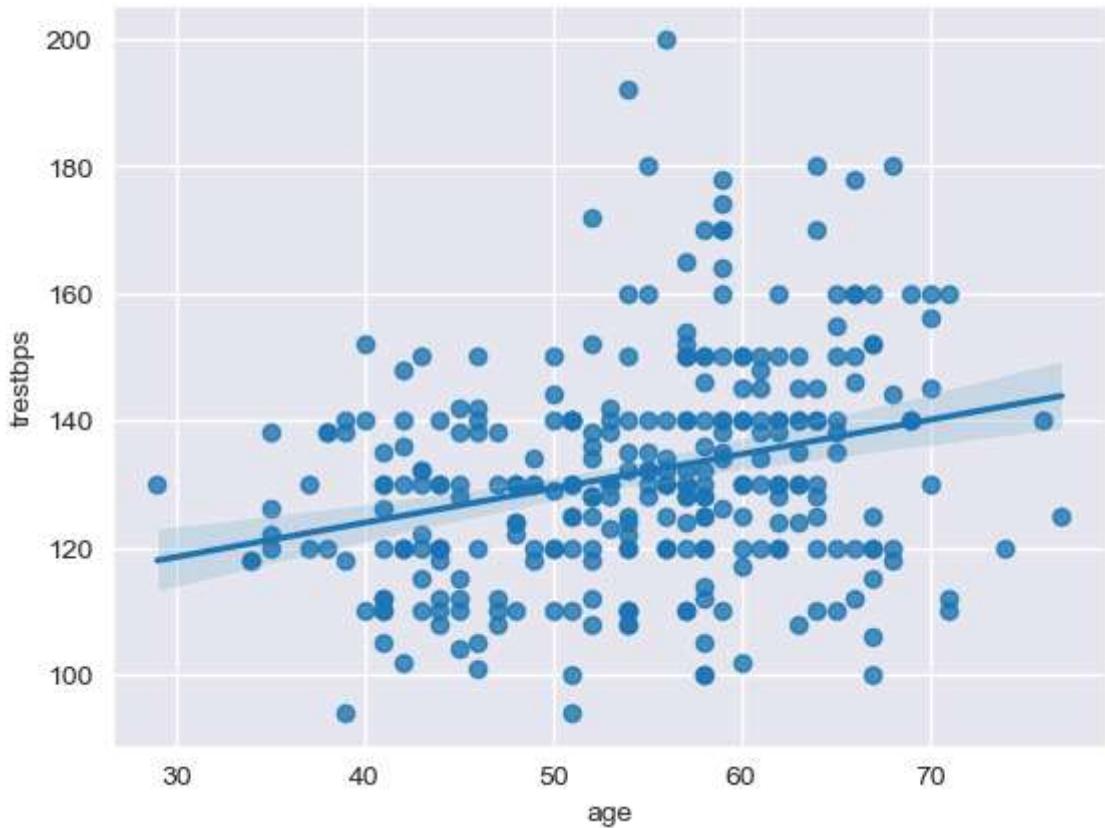


```
In [231... ax = sns.scatterplot(x="age", y="trestbps", data=data)
```



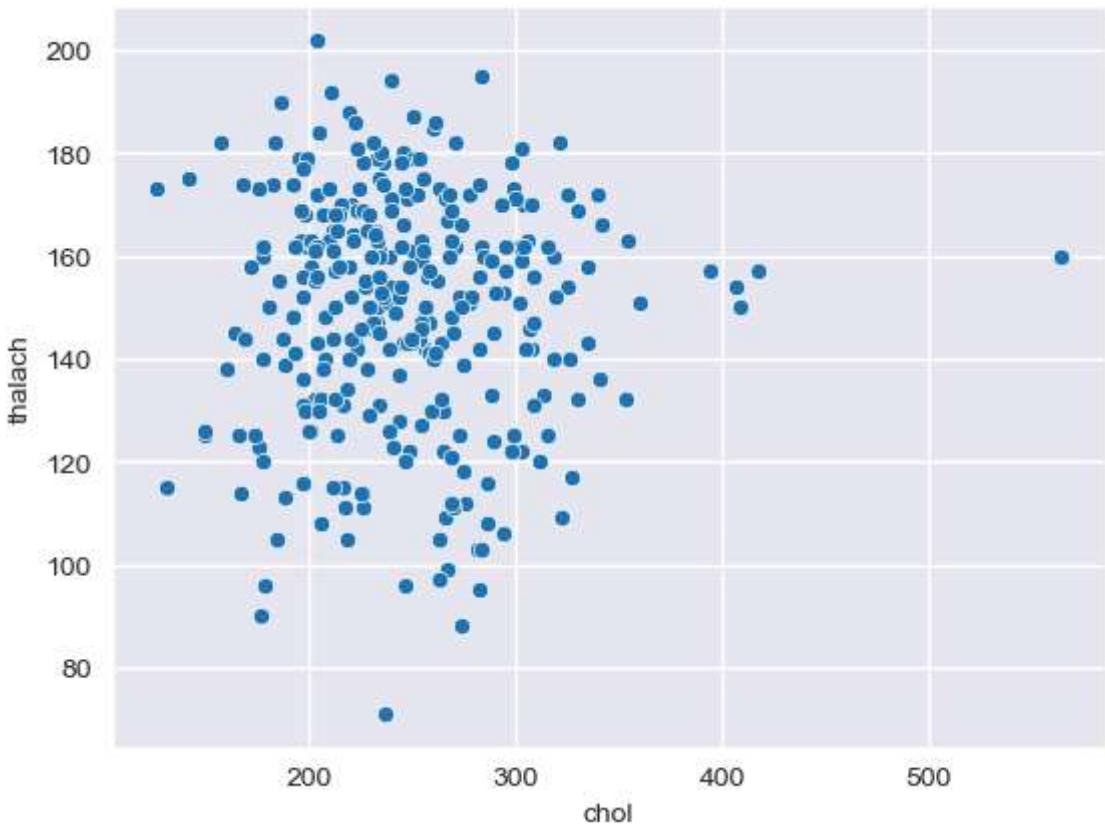
```
In [233... sns.regplot(x="age", y="trestbps", data=data)
```

```
Out[233... <Axes: xlabel='age', ylabel='trestbps'>
```



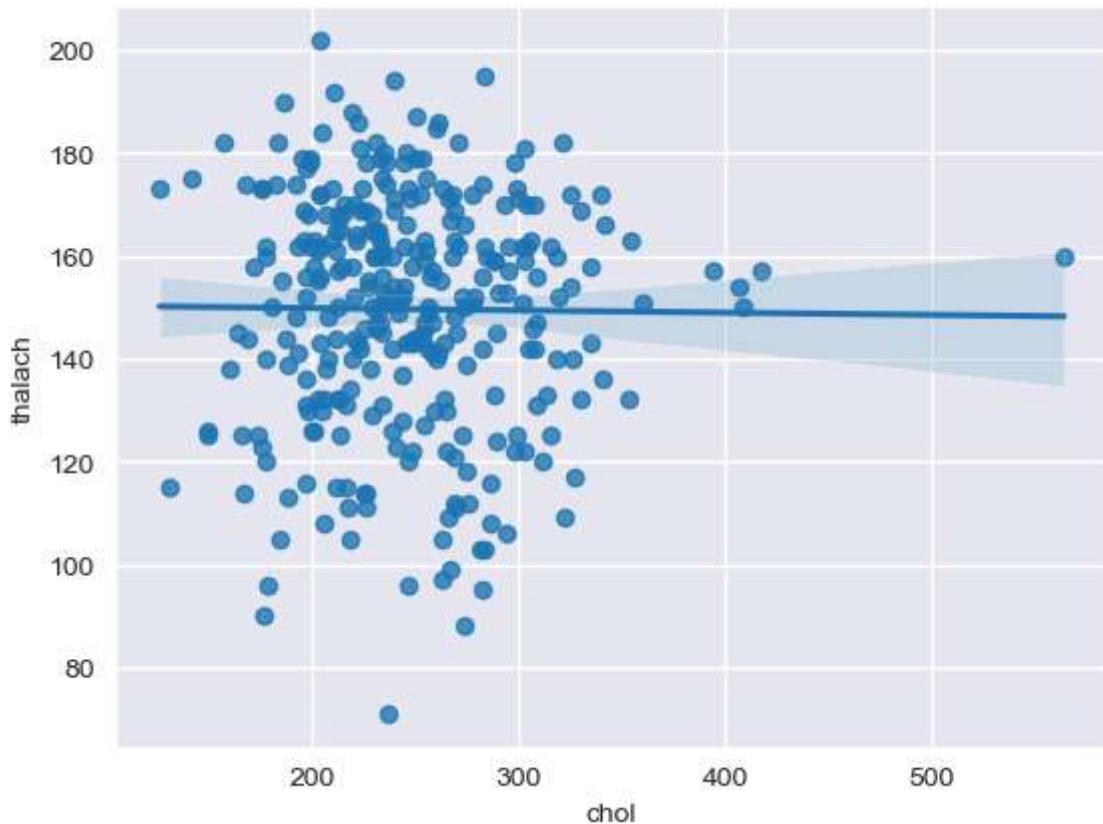
```
In [235]: sns.scatterplot(x="chol", y = "thalach", data=data)
```

```
Out[235]: <Axes: xlabel='chol', ylabel='thalach'>
```



```
In [239... sns.regplot(x="chol", y="thalach", data=data)
```

```
Out[239... <Axes: xlabel='chol', ylabel='thalach'>
```



```
In [243... data.isnull()
```

Out[243...]

	age	sex	cp	trestbps	chol	fb	restecg	thalach	exang	oldpeak	slope	ca	thal	target
0	False	False	False	False	False	False	False	False	False	False	False	False	False	False
1	False	False	False	False	False	False	False	False	False	False	False	False	False	False
2	False	False	False	False	False	False	False	False	False	False	False	False	False	False
3	False	False	False	False	False	False	False	False	False	False	False	False	False	False
4	False	False	False	False	False	False	False	False	False	False	False	False	False	False
...
298	False	False	False	False	False	False	False	False	False	False	False	False	False	False
299	False	False	False	False	False	False	False	False	False	False	False	False	False	False
300	False	False	False	False	False	False	False	False	False	False	False	False	False	False
301	False	False	False	False	False	False	False	False	False	False	False	False	False	False
302	False	False	False	False	False	False	False	False	False	False	False	False	False	False

303 rows × 14 columns



In [245...]

data.isnull().sum()

Out[245...]

```

age          0
sex          0
cp           0
trestbps    0
chol         0
fb           0
restecg     0
thalach     0
exang        0
oldpeak     0
slope        0
ca           0
thal         0
target       0
dtype: int64

```

In [249...]

assert pd.notnull(data).all().all()

In [257...]

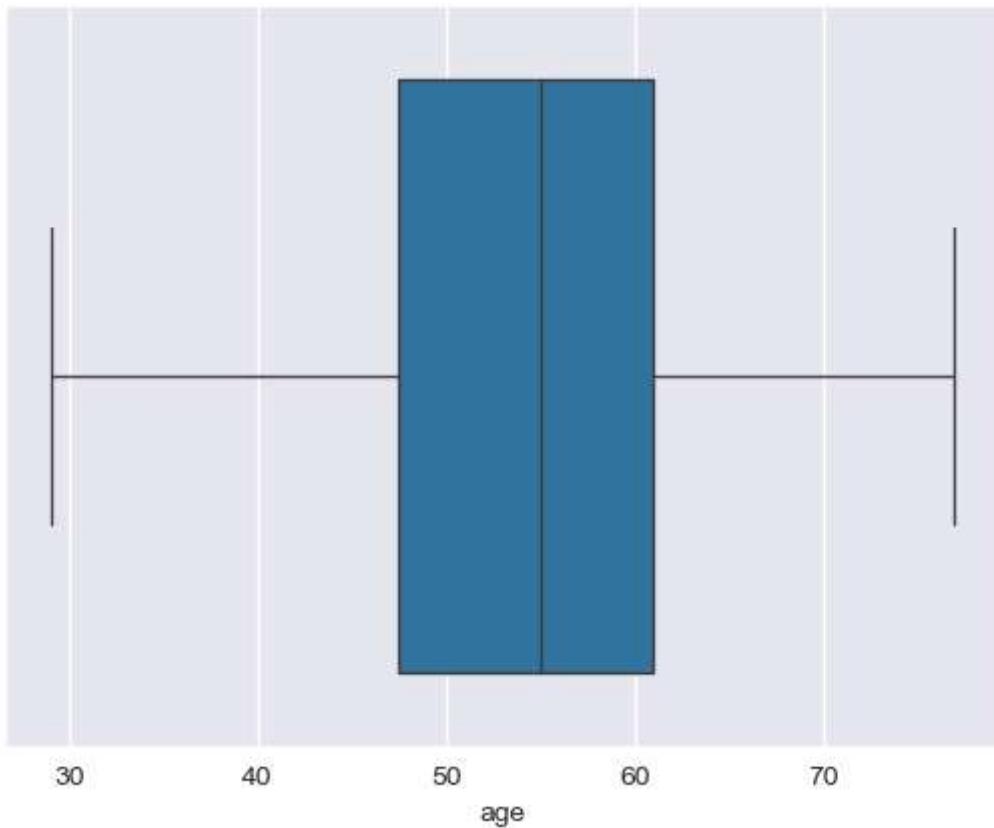
assert (data>=0).all().all()

In [261...]

sns.boxplot(x=data["age"])

Out[261...]

<Axes: xlabel='age'>

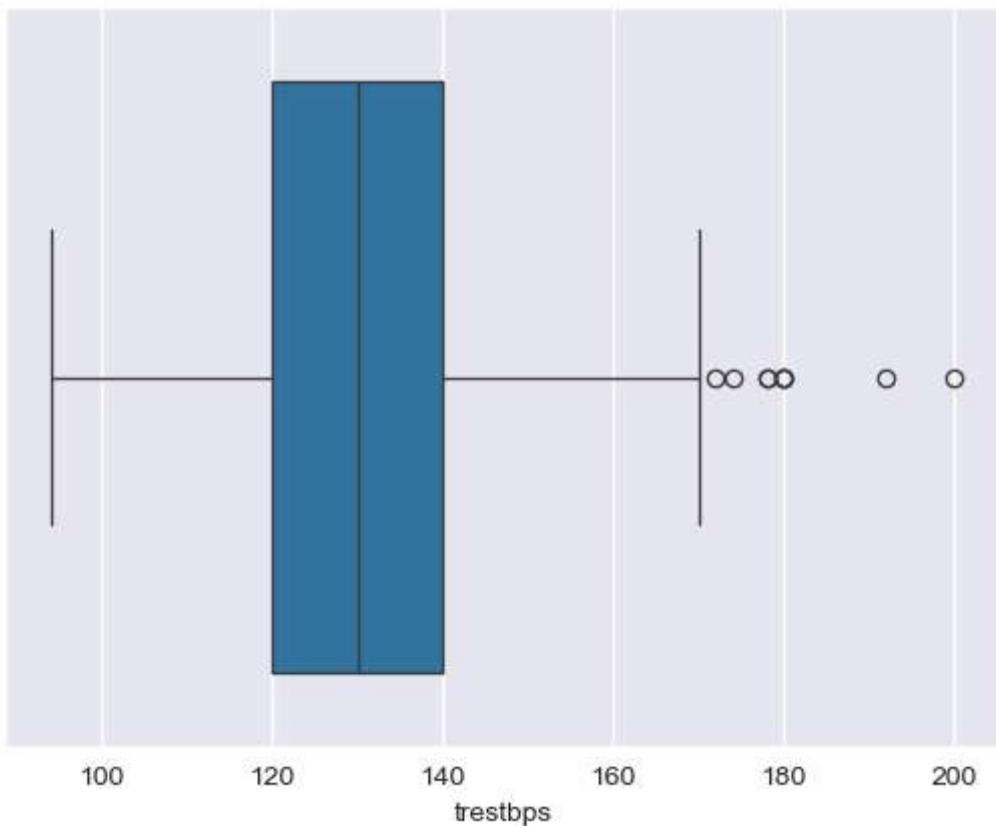


```
In [263...]: data['trestbps'].describe()
```

```
Out[263...]: 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 [265...]: sns.boxplot(x=data["trestbps"])
```

```
Out[265...]: <Axes: xlabel='trestbps'>
```

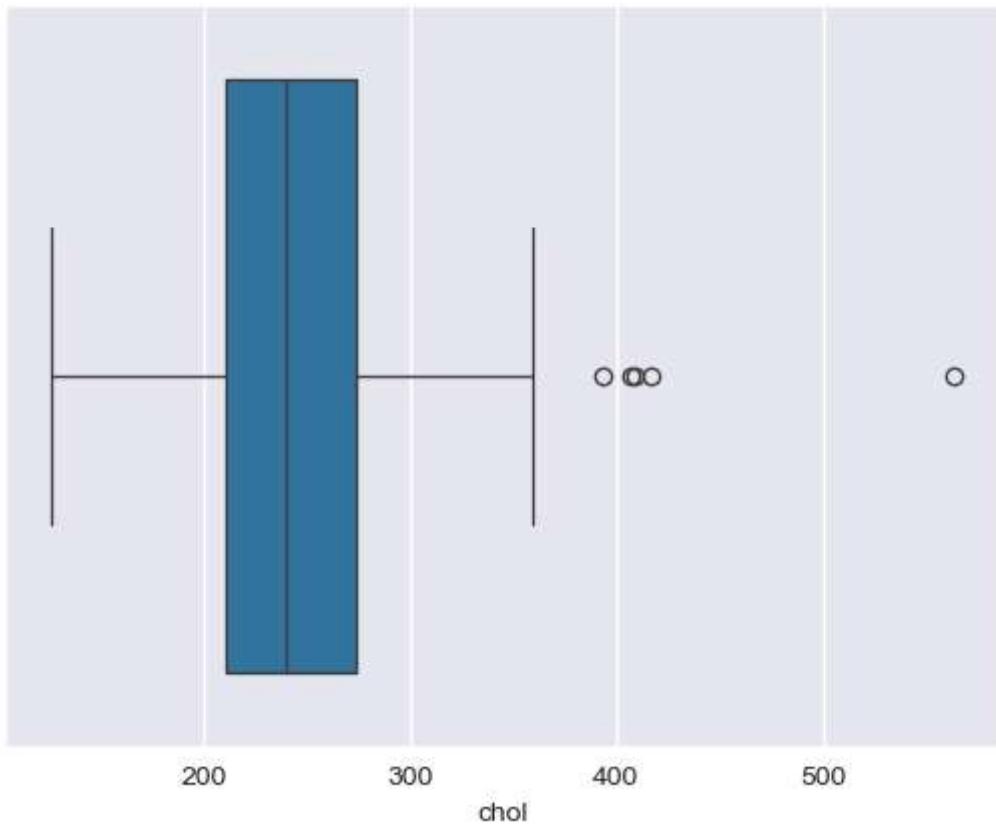


```
In [267...]: data['chol'].describe()
```

```
Out[267...]: 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 [269...]: sns.boxplot(x=data["chol"])
```

```
Out[269...]: <Axes: xlabel='chol'>
```

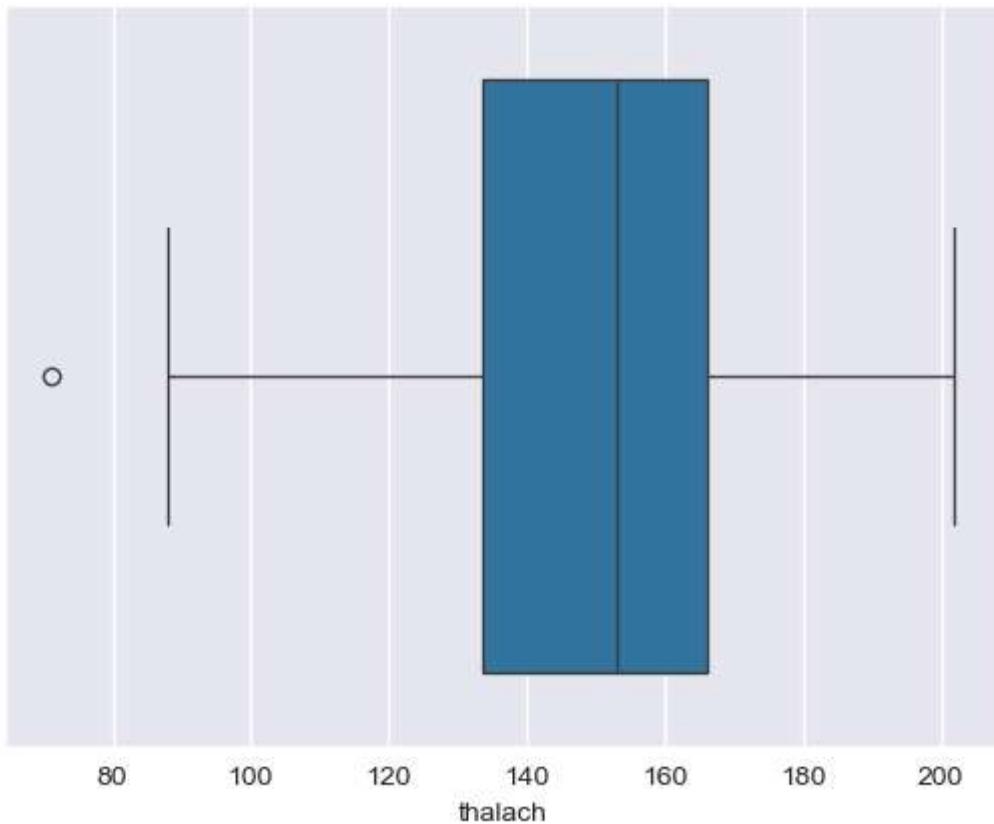


```
In [271...]: data['thalach'].describe()
```

```
Out[271...]: 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 [275...]: sns.boxplot(x=data["thalach"])
```

```
Out[275...]: <Axes: xlabel='thalach'>
```

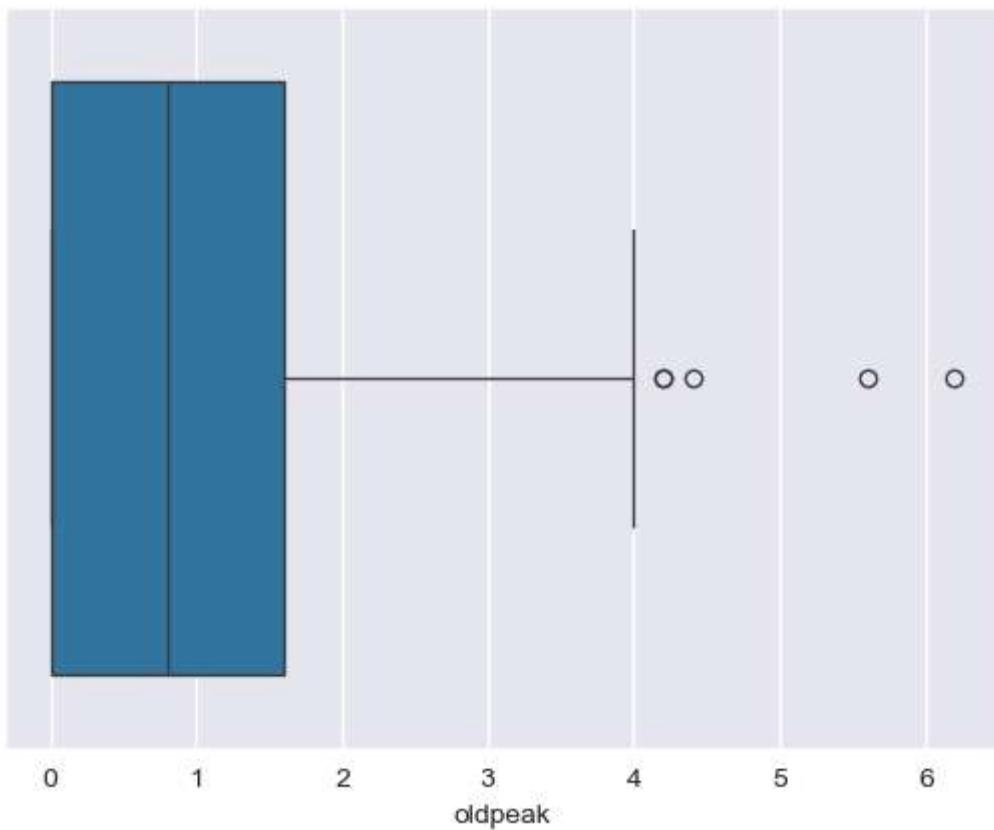


```
In [279...]: data['oldpeak'].describe()
```

```
Out[279...]: 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 [281...]: sns.boxplot(x=data["oldpeak"])
```

```
Out[281...]: <Axes: xlabel='oldpeak'>
```



In []: