In [1]:
```python
import pandas as pd
```
In [2]:
```python
import numpy as np
```
In [3]:
```python
import matplotlib.pyplot as plt
```
In [5]:
```python
import seaborn as sns
```
In [6]:
```python
data=pd.read_excel('data.xlsx')
```
In [7]:
```python
data.head()
```
Out[7]:

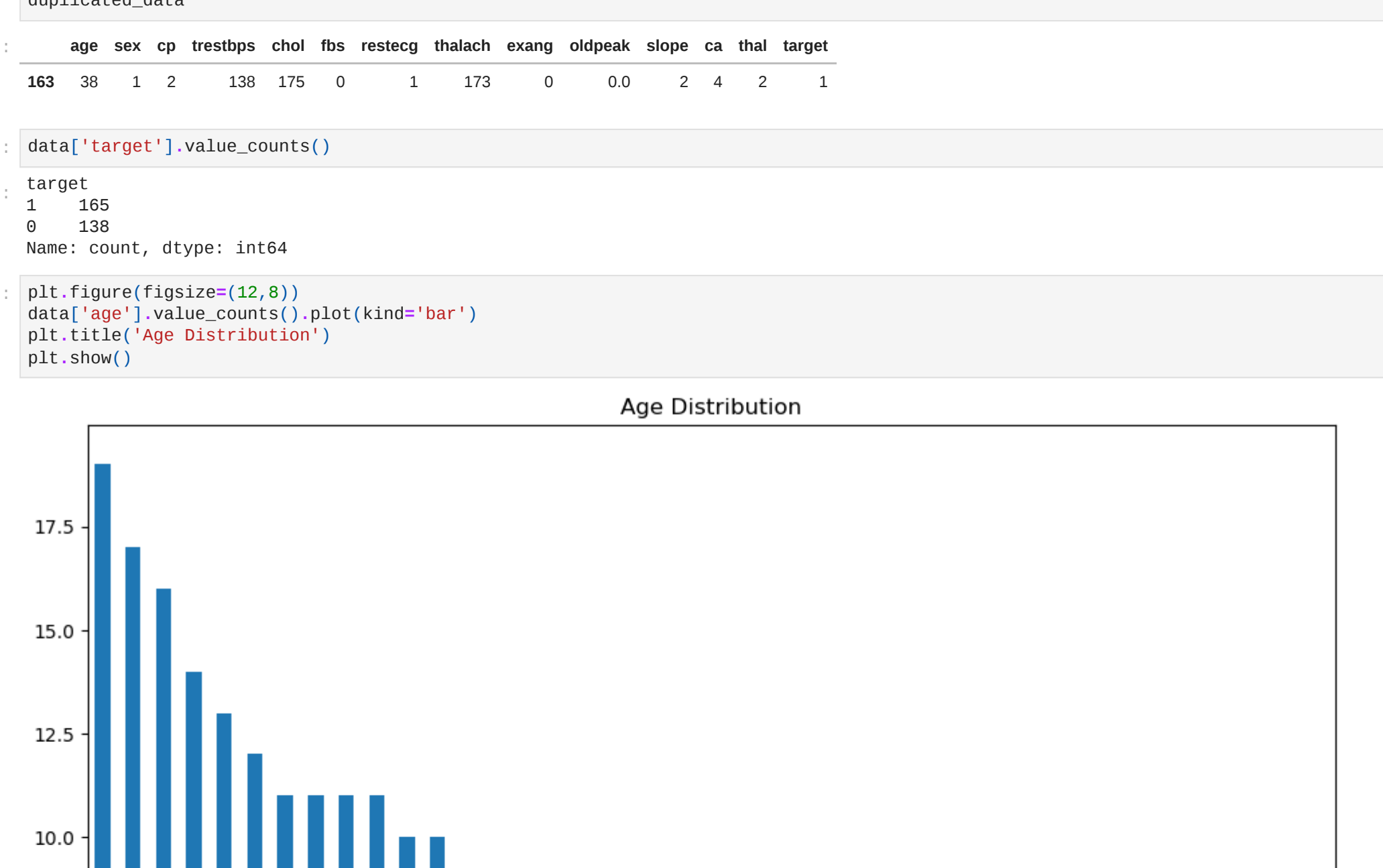
	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
0	63	1	3	145	233	1	0	150	0	2.3	0	0	1	1
1	37	1	2	130	250	0	1	187	0	3.5	0	0	2	1
2	41	0	1	130	204	0	0	172	0	1.4	2	0	2	1
3	56	1	1	120	236	0	1	178	0	0.8	2	0	2	1
4	57	0	0	120	354	0	1	163	1	0.6	2	0	2	1

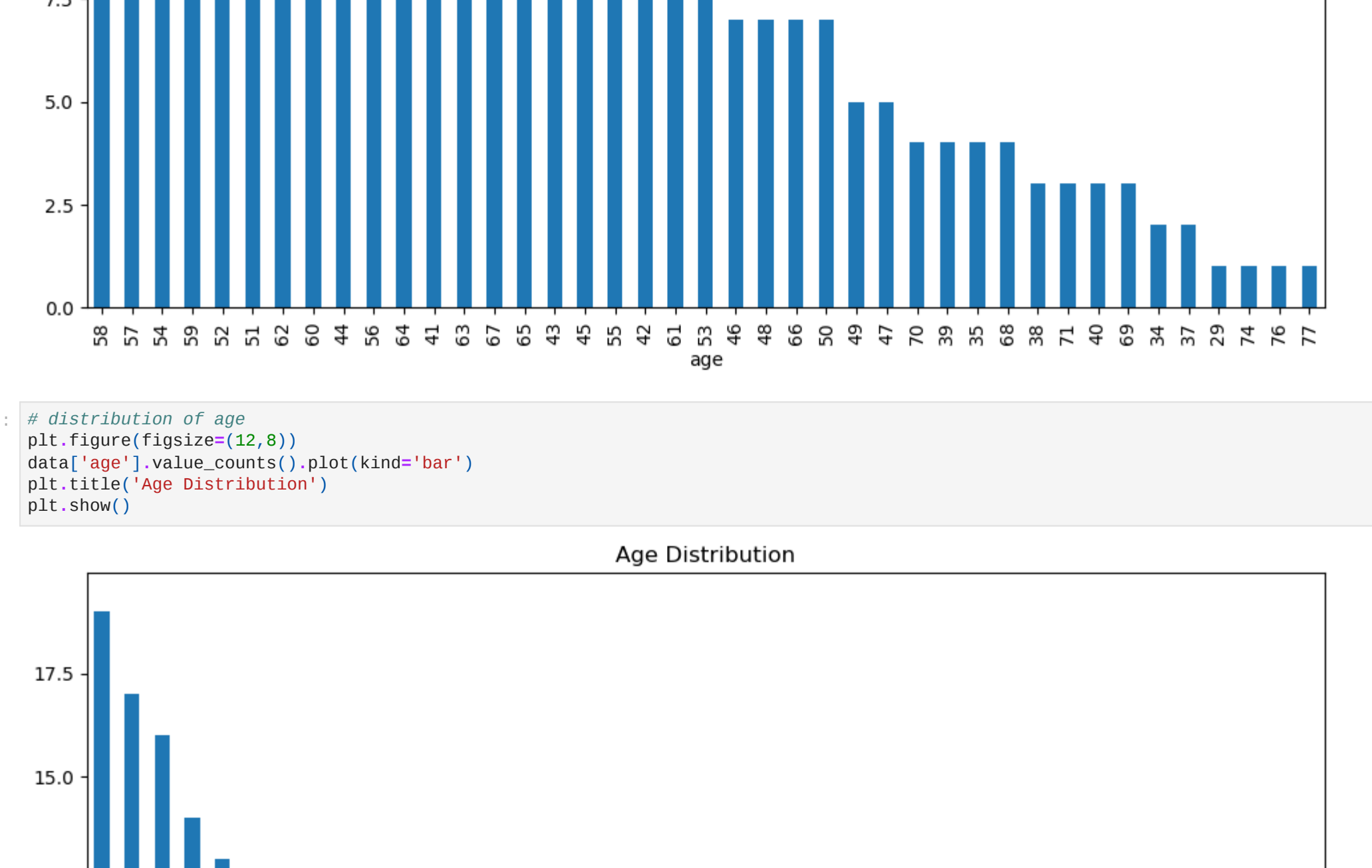
In [8]:
```python
data.shape
```
Out[8]:
```
(303, 14)
```
In [9]:
```python
data.info()
```
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 303 entries, 0 to 302
Data columns (total 14 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   age       303 non-null    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
```
In [10]:
```python
data.isnull().sum()
```
Out[10]:
```
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 [11]:
```python
data.describe()
```
Out[11]:

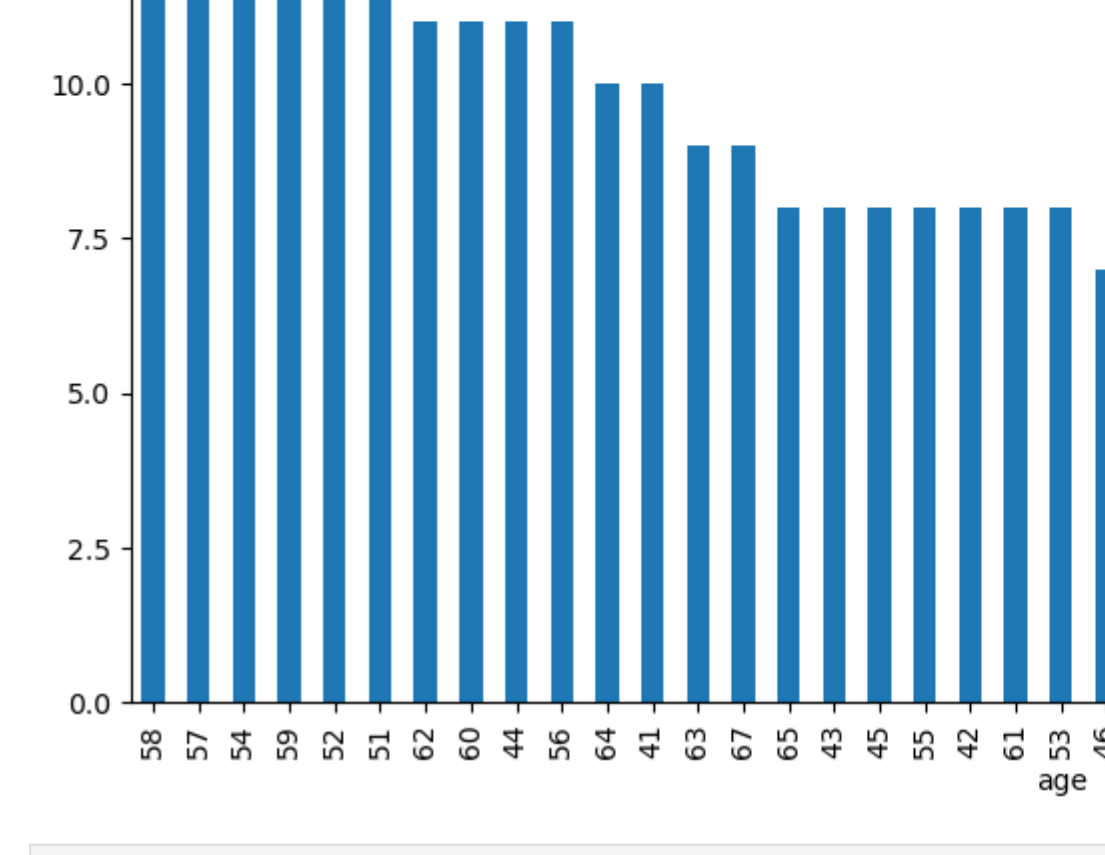
	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
count	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000
mean	54.366337	0.683168	0.966997	131.623762	246.264026	0.148515	0.528053	149.646865	0.326733	1.039604	1.399340	0.729373	2.313531	0.544554
std	9.082101	0.466011	1.032052	17.538143	51.830751	0.356198	0.525860	22.905161	0.469794	1.161075	0.616226	1.022606	0.612277	0.498835
min	29.000000	0.000000	0.000000	94.000000	126.000000	0.000000	0.000000	71.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	47.500000	0.000000	0.000000	120.000000	211.000000	0.000000	0.000000	133.500000	0.000000	0.000000	1.000000	0.000000	2.000000	0.000000
50%	55.000000	1.000000	1.000000	130.000000	240.000000	0.000000	1.000000	153.000000	0.000000	0.800000	1.000000	0.000000	2.000000	1.000000
75%	61.000000	1.000000	2.000000	140.000000	274.500000	0.000000	1.000000	166.000000	1.000000	1.600000	2.000000	1.000000	3.000000	1.000000
max	77.000000	1.000000	3.000000	200.000000	564.000000	1.000000	2.000000	202.000000	1.000000	6.200000	2.000000	4.000000	3.000000	1.000000

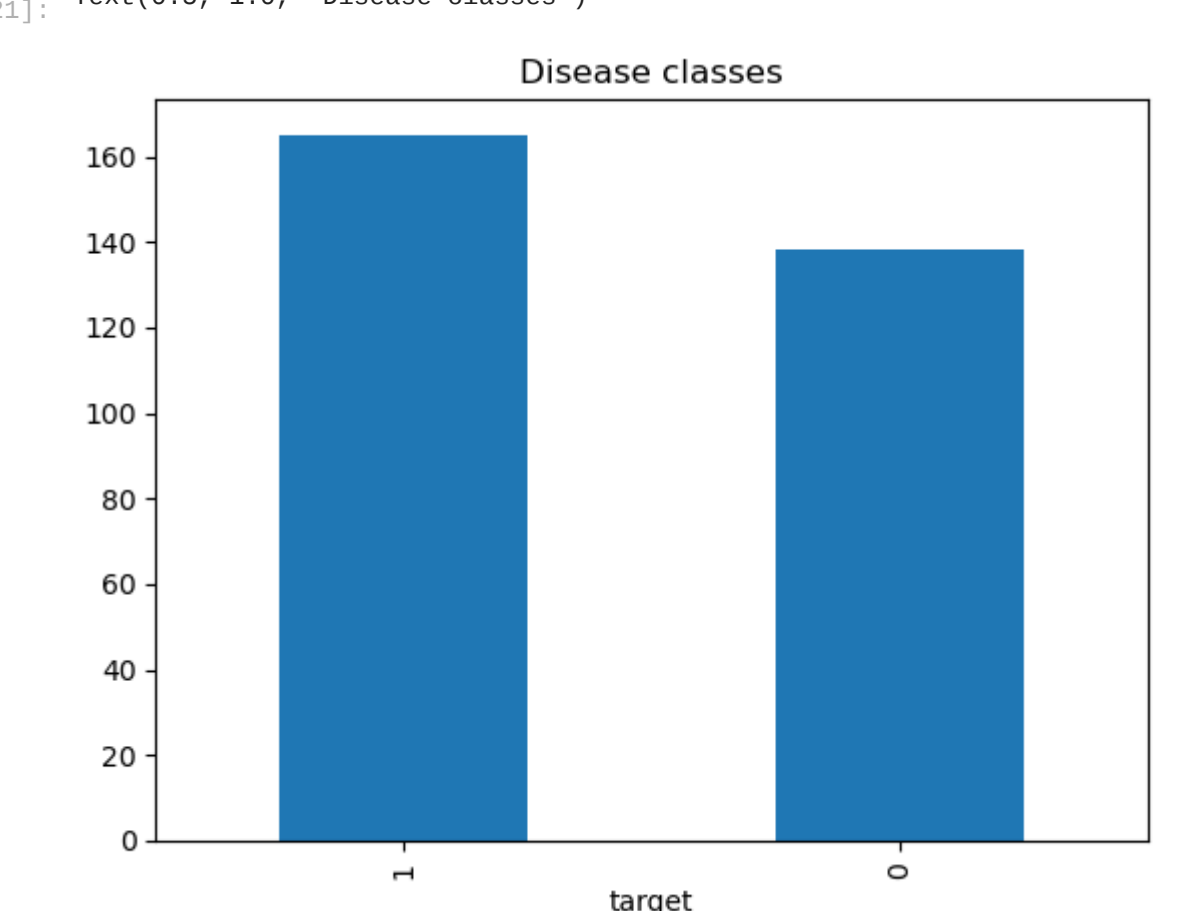
In [12]:
```python
data.nunique()
```
Out[12]:
```
age          41
sex           2
cp            4
trestbps     49
chol        152
fbs           2
restecg       3
thalach      91
exang         2
oldpeak      40
slope         3
ca            5
thal          4
target        2
dtype: int64
```
In [14]:
```python
duplicated_data=data[data.duplicated(keep='last')]
duplicated_data
```
Out[14]:

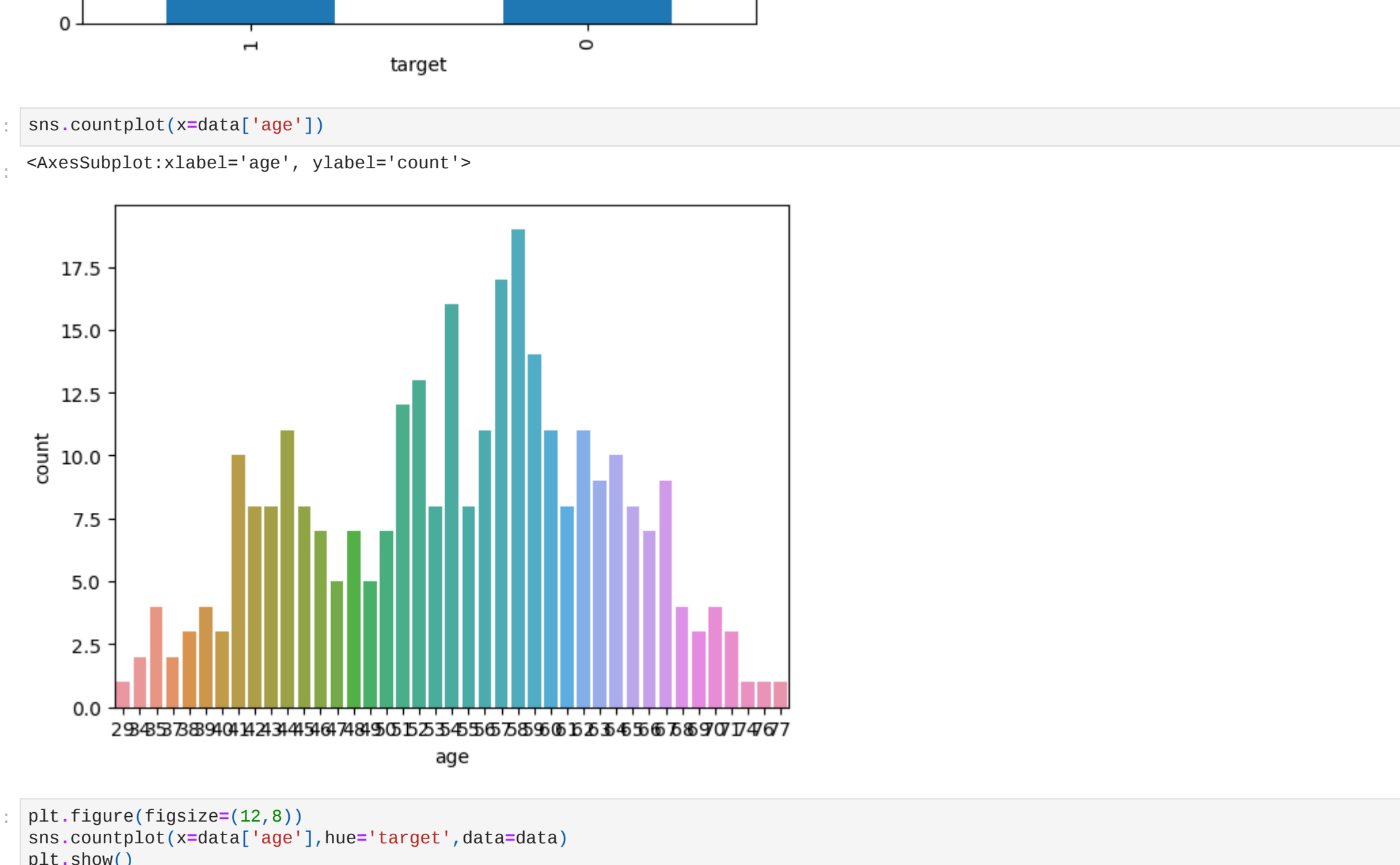
	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
163	38	1	2	138	175	0	1	173	0	0.0	2	4	2	1

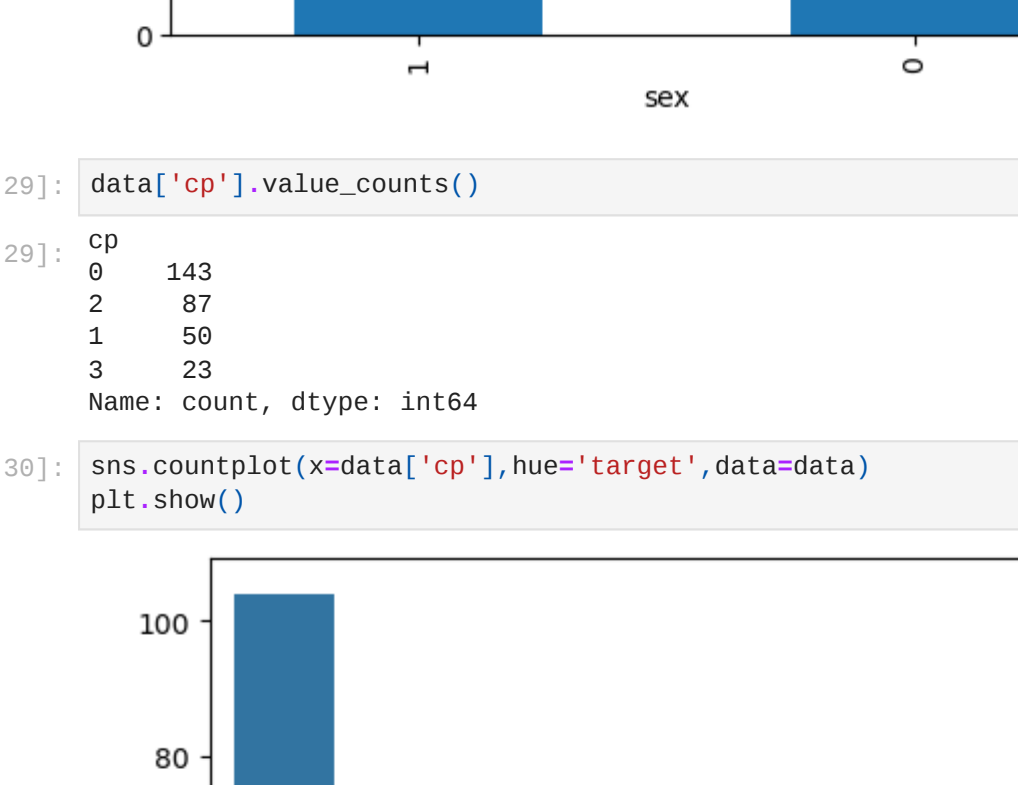
In [15]:
```python
data['target'].value_counts()
```
```
target
1    165
0    138
Name: count, dtype: int64
```
In [20]:
```python
plt.figure(figsize=(12,8))
data['age'].value_counts().plot(kind='bar')
plt.title('Age Distribution')
plt.show()
```


In [18]:
```python
# distribution of age
plt.figure(figsize=(12,8))
data['age'].value_counts().plot(kind='bar')
plt.title('Age Distribution')
plt.show()
```


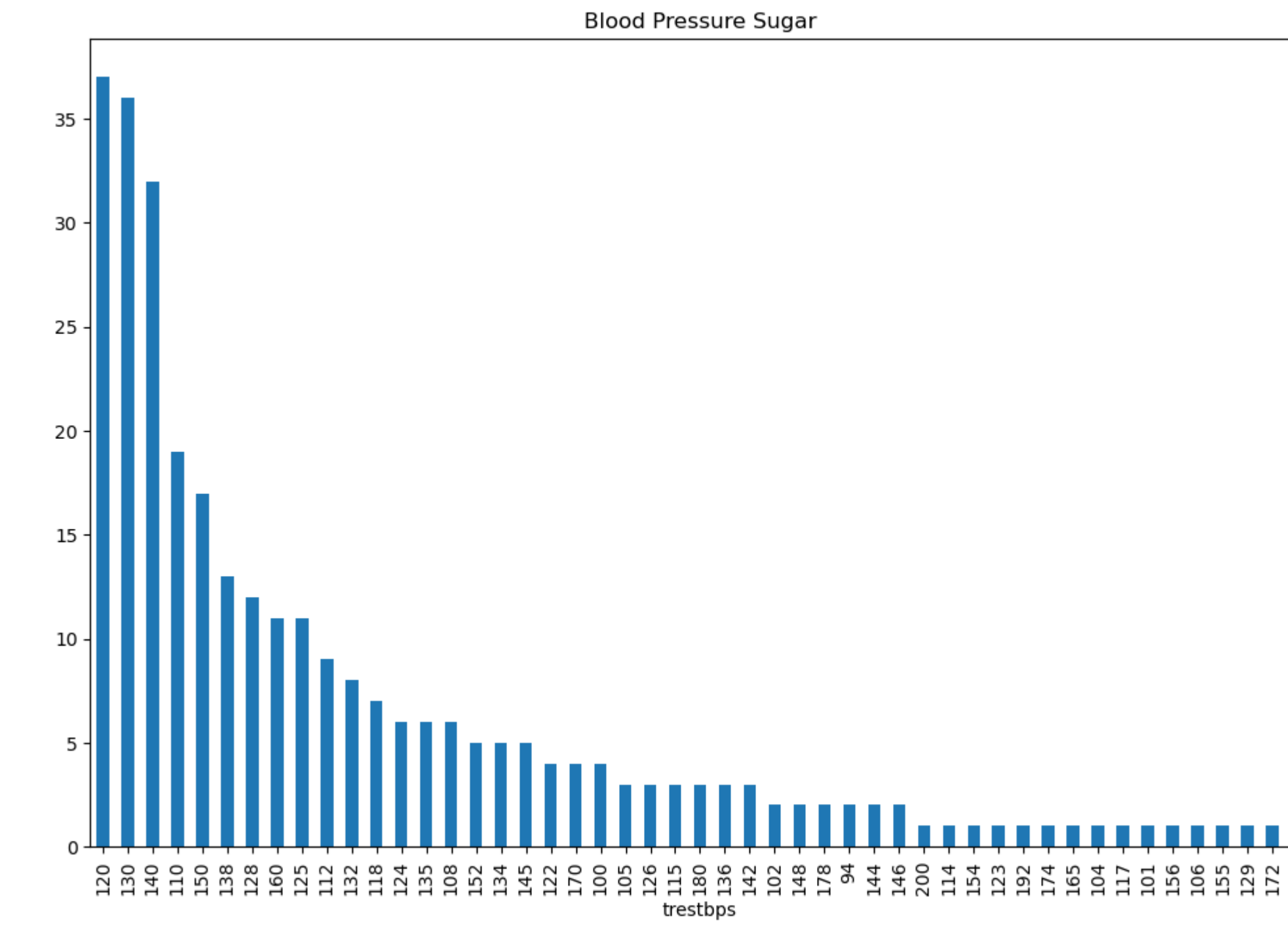
In [21]:
```python
#EDA on Heart Diseases
data['target'].value_counts().plot(kind='bar')
plt.title('Disease classes')
```
Out[21]:
```
Text(0.5, 1.0, 'Disease classes')
```


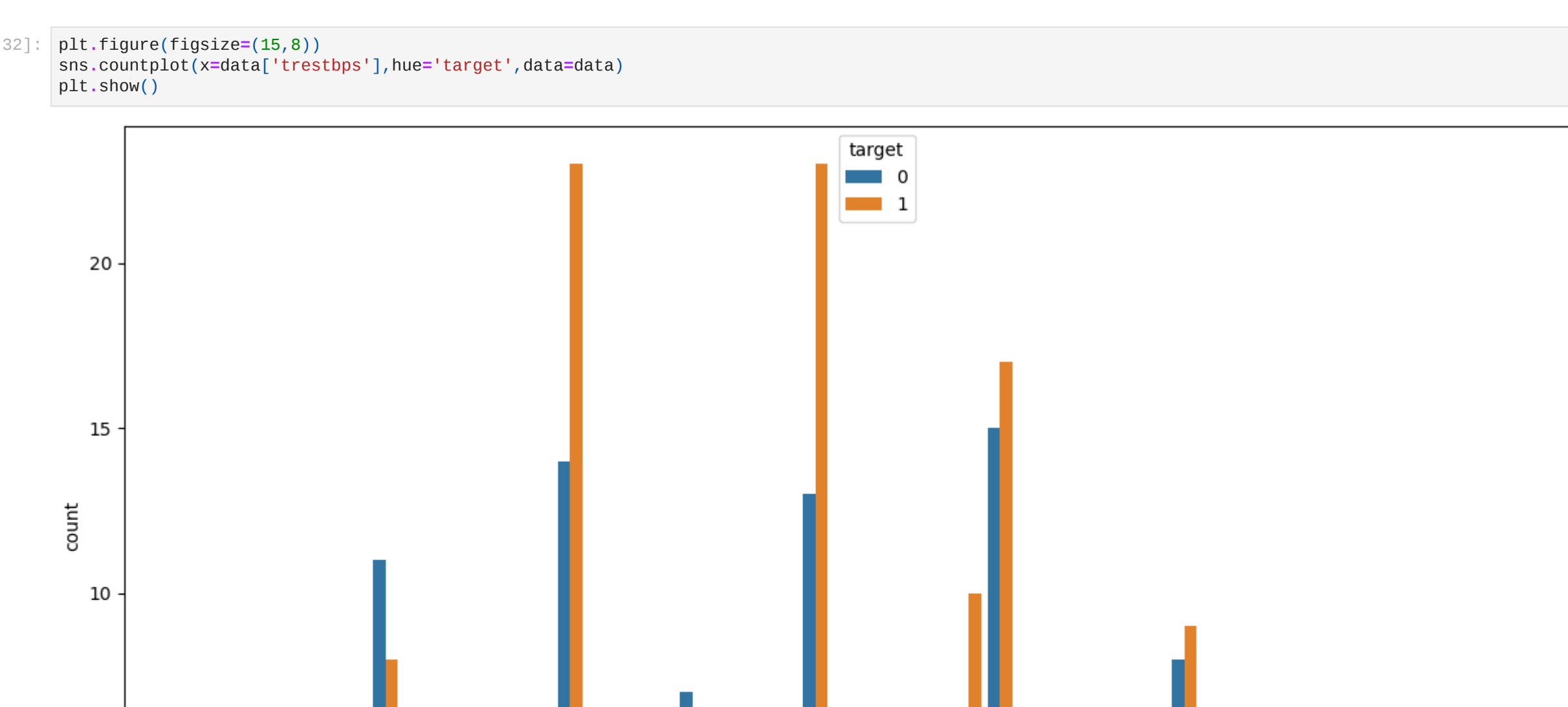
In [22]:
```python
sns.countplot(x=data['age'])
```
Out[22]:
```
<AxesSubplot:xlabel='age', ylabel='count'>
```


In [23]:
```python
plt.figure(figsize=(12,8))
sns.countplot(x=data['age'],hue='target',data=data)
plt.show()
```


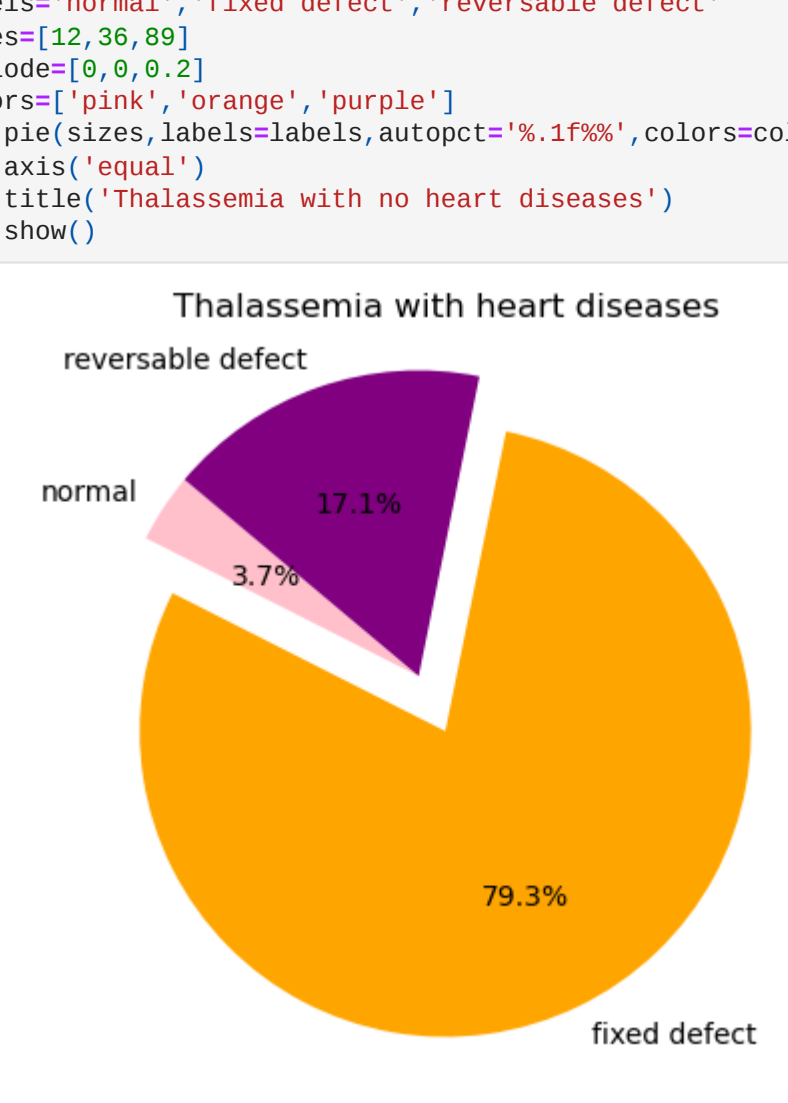
In [26]:
```python
data.columns
```
Out[26]:
```
Index(['age', 'sex', 'cp', 'trestbps', 'chol', 'fbs', 'restecg', 'thalach',
       'exang', 'oldpeak', 'slope', 'ca', 'thal', 'target'],
      dtype='object')
```
In [27]:
```python
data['sex'].value_counts().plot(kind='bar')
```
Out[27]:
```
<AxesSubplot:xlabel='sex'>
```


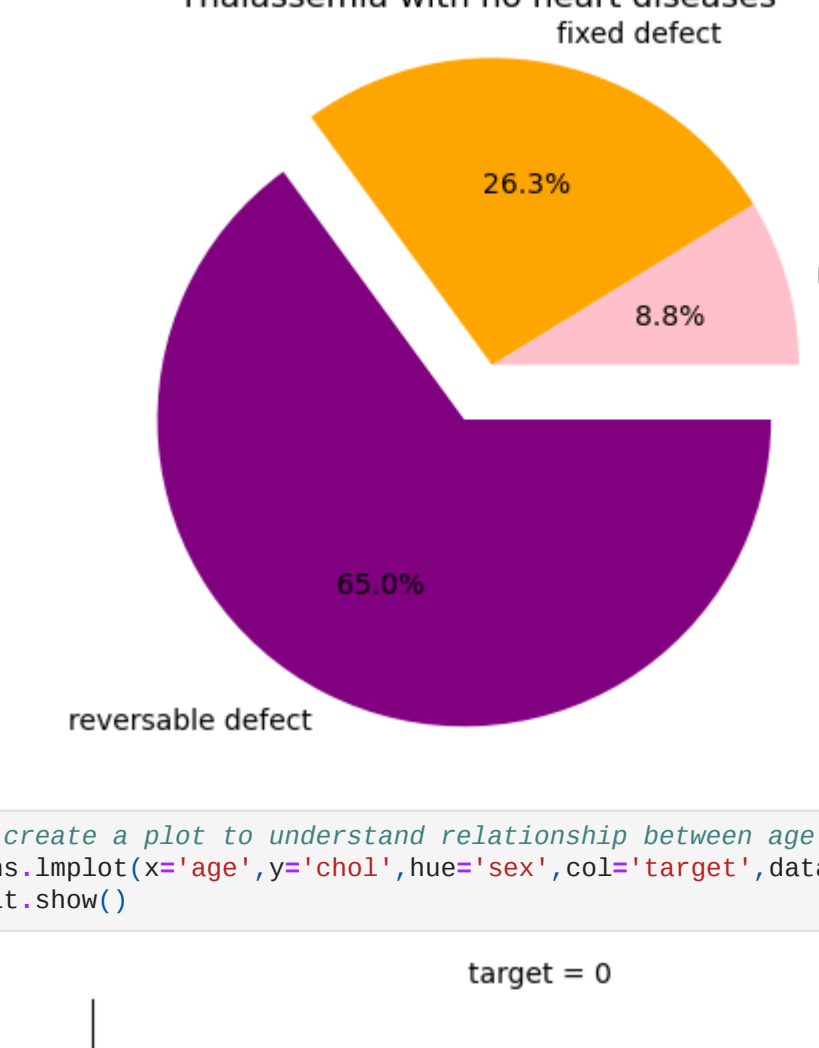
In [29]:
```python
data['cp'].value_counts()
```
Out[29]:
```
cp
0    143
2     87
1     50
3     23
Name: count, dtype: int64
```
In [30]:
```python
sns.countplot(x=data['cp'],hue='target',data=data)
plt.show()
```

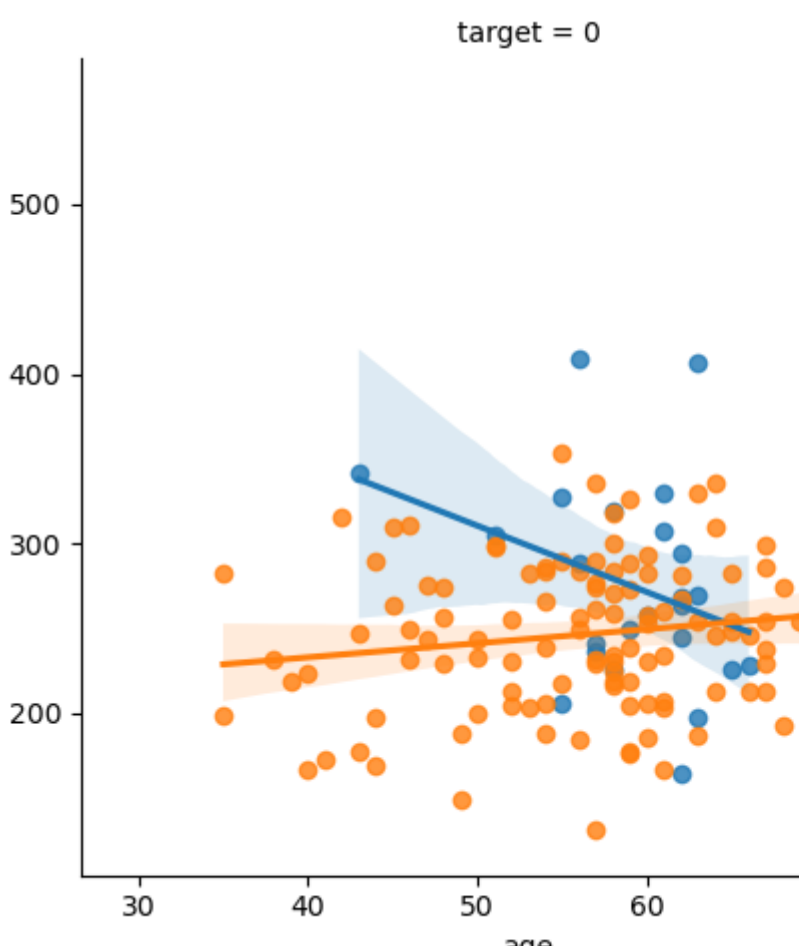

In [31]:
```python
plt.figure(figsize=(12,8))
data['trestbps'].value_counts().plot(kind='bar')
plt.title('Blood Pressure Sugar')
plt.show()
```


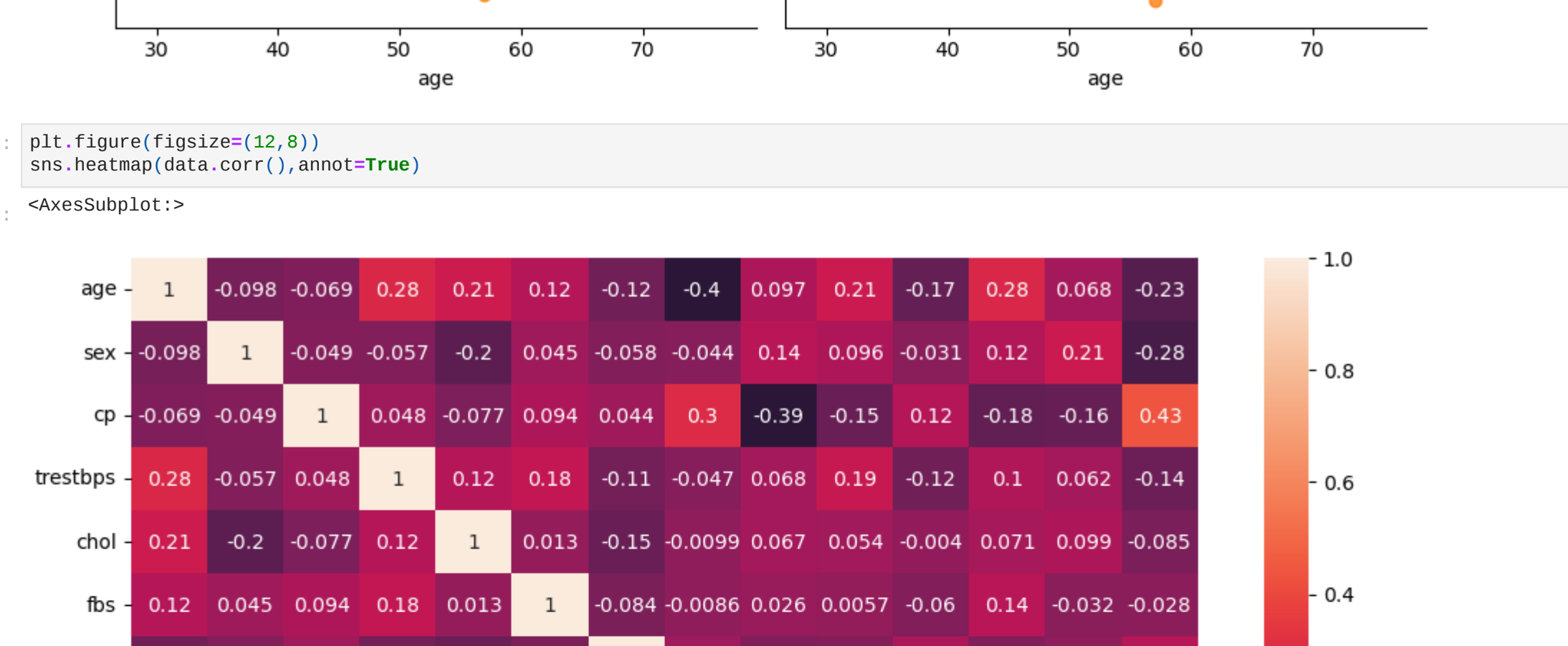
In [32]:
```python
plt.figure(figsize=(15,8))
sns.countplot(x=data['trestbps'],hue='target',data=data)
plt.show()
```


In [36]:
```python
# Create a subplot
fig,axes=plt.subplots(nrows=3,ncols=3,figsize=(15,10))
cat_features=['sex','cp','fbs','restecg','exang','slope','ca','thal','target']
for idx,feature in enumerate(cat_features):
    if feature!='target':
        ax=axes[int(idx/3),idx%3]
        sns.countplot(x=feature,hue='target',ax=ax,data=data)
```

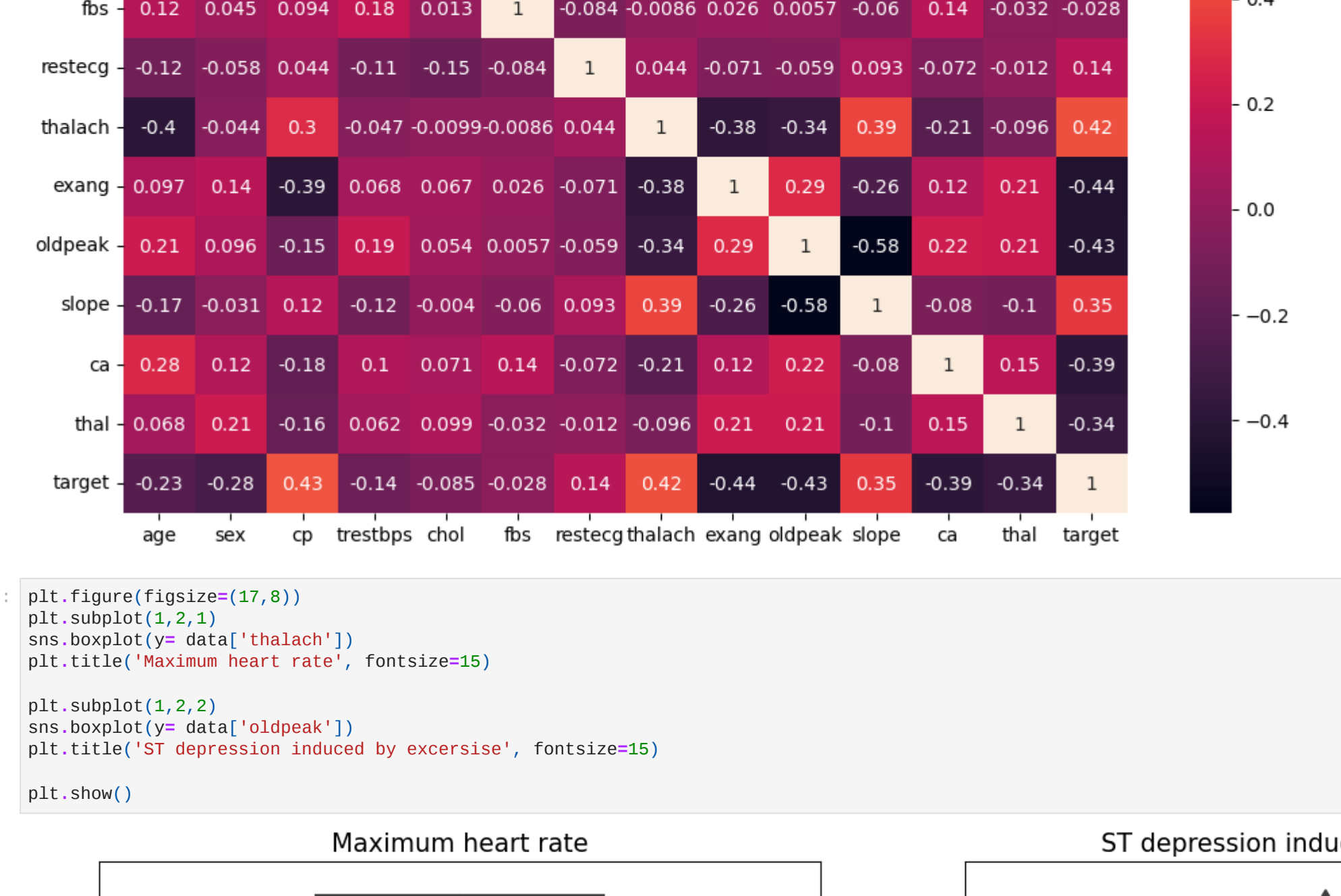

In [37]:
```python
# Create pie chart- have heart diseases
labels='normal','fixed defect','reversable defect'
sizes=[6,130,28]
colors=['pink','orange','purple']
plt.pie(sizes,labels=labels,autopct='%.1f%%',colors=colors,startangle=140)
plt.axis('equal')
plt.title('Thalassemia with heart diseases')
plt.show()
```


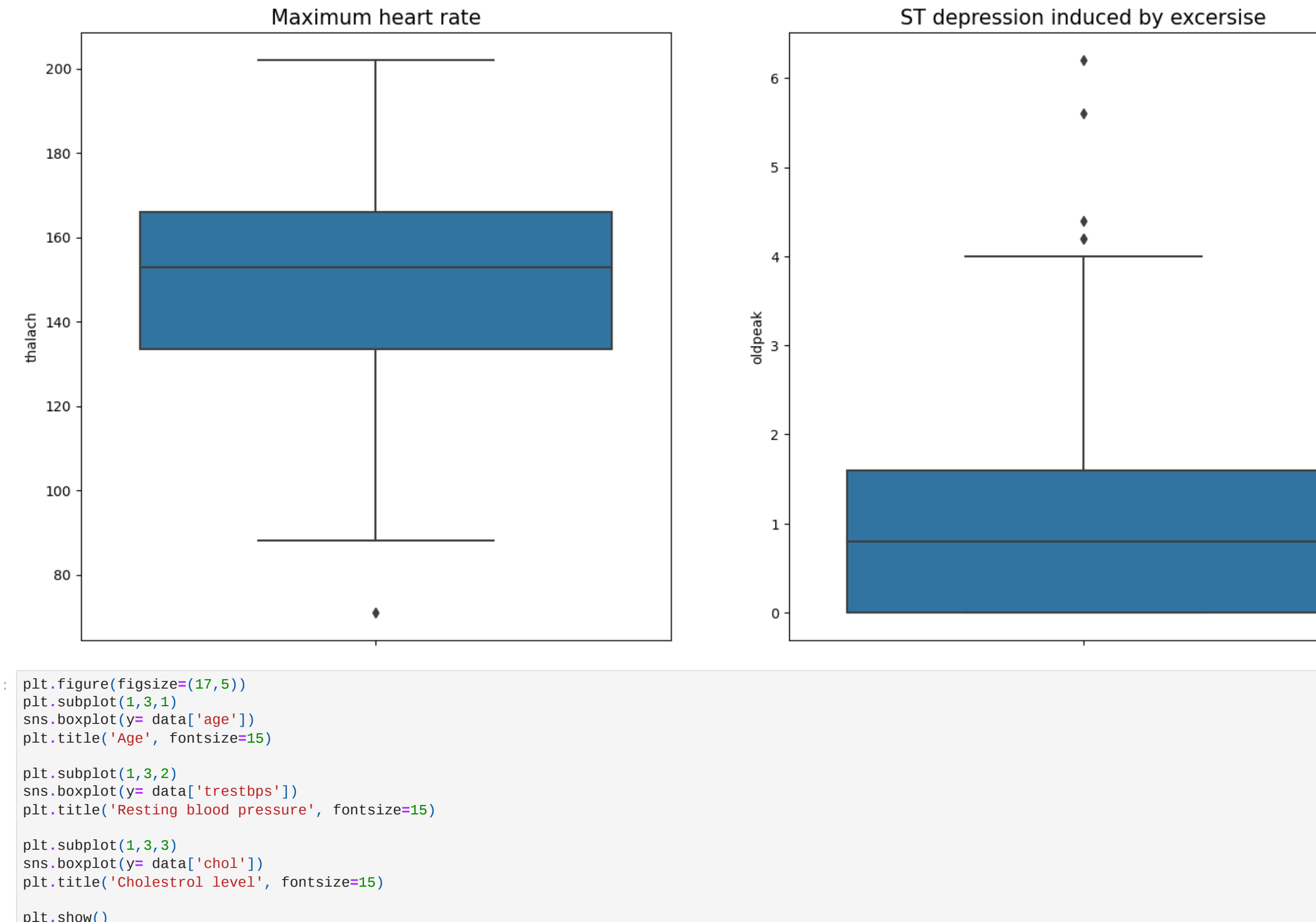
In [38]:
```python
# Create pie chart- have heart diseases
#plt.figure(figsize=(12,8))
labels='normal','fixed defect','reversable defect'
sizes=[6,130,28]
explode=[0,0.2,0]
colors=['pink','orange','purple']
plt.pie(sizes,labels=labels,autopct='%.1f%%',colors=colors,startangle=140,explode=explode)
plt.axis('equal')
plt.title('Thalassemia with heart diseases')
plt.show()

# Create pie chart- have no heart diseases
#plt.figure(figsize=(12,8))
labels='normal','fixed defect','reversable defect'
sizes=[12,36,89]
explode=[0,0,0.2]
colors=['pink','orange','purple']
plt.pie(sizes,labels=labels,autopct='%.1f%%',colors=colors,explode=explode)
plt.axis('equal')
plt.title('Thalassemia with no heart diseases')
plt.show()
```


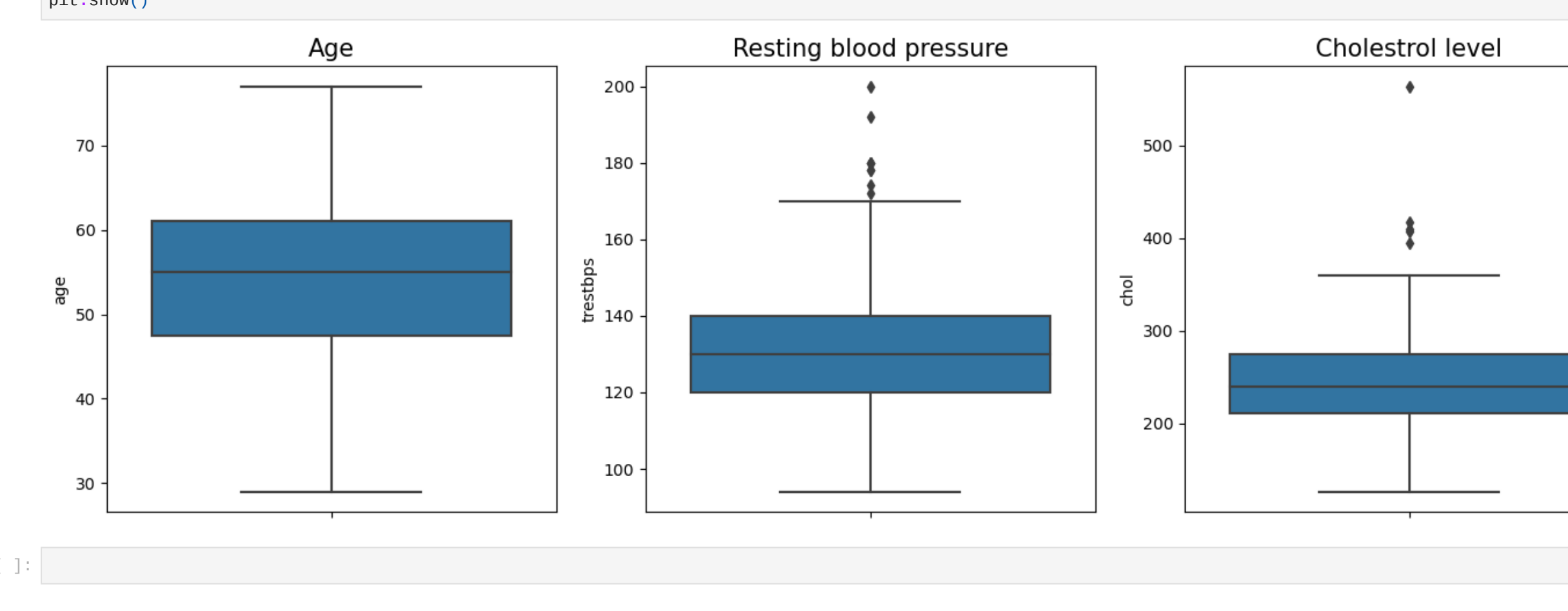


In [45]:
```python
# create a plot to understand relationship between age & chol, according to target
sns.lmplot(x='age',y='chol',hue='sex',col='target',data=data)
plt.show()
```


In [46]:
```python
plt.figure(figsize=(12,8))
sns.heatmap(data.corr(),annot=True)
```
Out[46]:
```
<AxesSubplot:>
```


In [47]:
```python
plt.figure(figsize=(12,8))
sns.heatmap(data.corr(),annot=True)
```
Out[47]:
```
<AxesSubplot:>
```


In [48]:
```python
plt.figure(figsize=(12,8))
plt.subplot(1,2,1)
sns.boxplot(y= data['thalach'])
plt.title('Maximum heart rate', fontsize=15)

plt.subplot(1,2,2)
sns.boxplot(y= data['oldpeak'])
plt.title('ST depression induced by excersise', fontsize=15)

plt.show()
```


In [49]:
```python
plt.figure(figsize=(17,5))
plt.subplot(1,3,1)
sns.boxplot(y= data['age'])
plt.title('Age', fontsize=15)

plt.subplot(1,3,2)
sns.boxplot(y= data['trestbps'])
plt.title('Resting blood pressure', fontsize=15)

plt.subplot(1,3,3)
sns.boxplot(y= data['chol'])
plt.title('Cholestrol level', fontsize=15)

plt.show()
```


In [ ]: