In [1]:

```
import numpy as np

import pandas as pd
```

In [6]:

```
df=pd.read_csv("heartattack.csv") # reading file
```

In [7]:

```
df
```

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

303 rows × 14 columns

In [9]:

```
df.head()
```

Out[9]:

	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 [10]:

```
df.info() #  original summary of the data
```

```
<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 [11]:

```
df.isna().sum() # finding null values
```

Out[11]:

```
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 [12]:

```
df.duplicated() # duplicate found
```

Out[12]:

```
0      False
1      False
2      False
3      False
4      False
       ...  
298    False
299    False
300    False
301    False
302    False
Length: 303, dtype: bool
```

In [13]:

```
df.drop_duplicates() # 1 duplicate removed
```

Out[13]:

	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

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
4	57	0	0	120	354	0	1	163	1	0.6	2	0	2	1
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
298	57	0	0	140	241	0	1	123	1	0.2	1	0	3	0
299	45	1	3	110	264	0	1	132	0	1.2	1	0	3	0
300	68	1	0	144	193	1	1	141	0	3.4	1	2	3	0
301	57	1	0	130	131	0	1	115	1	1.2	1	1	3	0
302	57	0	1	130	236	0	0	174	0	0.0	1	1	2	0

302 rows × 14 columns

In [14]:

```
df.describe() #  after removing duplicates summary of data
```

Out[14]:

	age	sex	cp	trestbps	chol	fbs	restecg	thalach
count	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
std	9.082101	0.466011	1.032052	17.538143	51.830751	0.356198	0.525860	22.905161
min	29.000000	0.000000	0.000000	94.000000	126.000000	0.000000	0.000000	71.000000
25%	47.500000	0.000000	0.000000	120.000000	211.000000	0.000000	0.000000	133.500000
50%	55.000000	1.000000	1.000000	130.000000	240.000000	0.000000	1.000000	153.000000
75%	61.000000	1.000000	2.000000	140.000000	274.500000	0.000000	1.000000	166.000000
max	77.000000	1.000000	3.000000	200.000000	564.000000	1.000000	2.000000	202.000000

In [17]:

```
df.mean() # central tendencies
```

Out[17]:

```
age          54.366337
sex           0.683168
cp            0.966997
trestbps    131.623762
chol        246.264026
fbs           0.148515
restecg       0.528053
thalach     149.646865
exang         0.326733
oldpeak       1.039604
slope         1.399340
ca            0.729373
thal          2.313531
target        0.544554
dtype: float64
```

In [18]:

```
df.mode() # central tendencies
```

Out[18]:

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
0	58.0	1.0	0.0	120.0	197	0.0	1.0	162.0	0.0	0.0	2.0	0.0	2.0	1.0
1	NaN	NaN	NaN	NaN	204	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2	NaN	NaN	NaN	NaN	234	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

In [19]:

```
df.median() # central tendencies
```

Out[19]:

```
age          55.0
sex           1.0
cp            1.0
trestbps    130.0
chol        240.0
fbs           0.0
restecg       1.0
thalach     153.0
exang         0.0
oldpeak       0.8
slope         1.0
ca            0.0
thal          2.0
target        1.0
dtype: float64
```

In [26]:

```
# spreading of the data
```

In [20]:

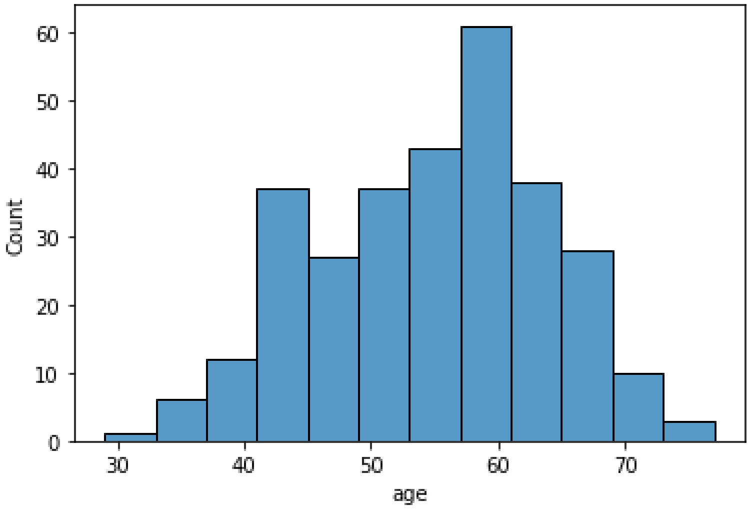
```
import seaborn as  sns
from matplotlib import pyplot as plt
```

In [21]:

```
sns.histplot(data=df,x="age")
```

Out[21]:

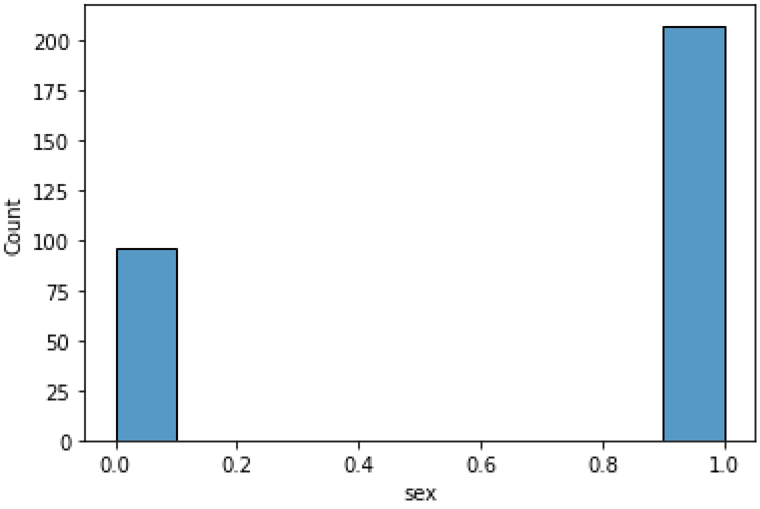
```
<AxesSubplot:xlabel='age', ylabel='Count'>
```



In [22]:

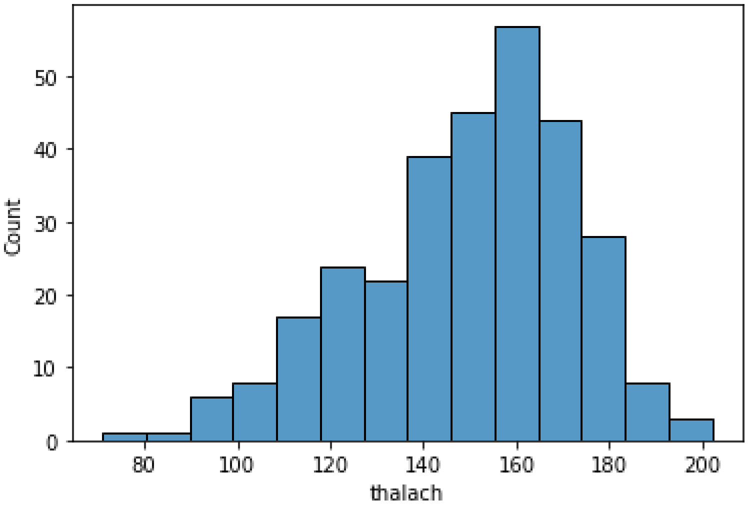
```
sns.histplot(data=df,x="sex")
```

Out[22]:
```
<AxesSubplot:xlabel='sex', ylabel='Count'>
```



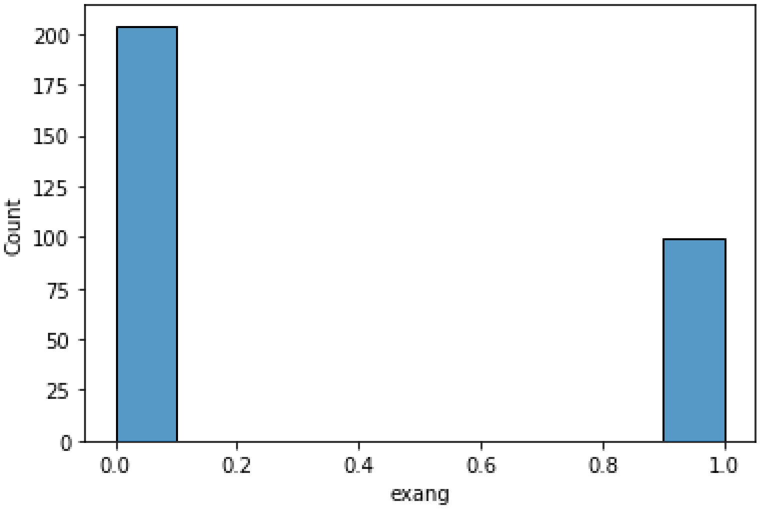
In [23]:
```python
sns.histplot(data=df,x="thalach")
```

Out[23]:
```
<AxesSubplot:xlabel='thalach', ylabel='Count'>
```



In [24]:
```python
sns.histplot(data=df,x="exang")
```

Out[24]:
```
<AxesSubplot:xlabel='exang', ylabel='Count'>
```

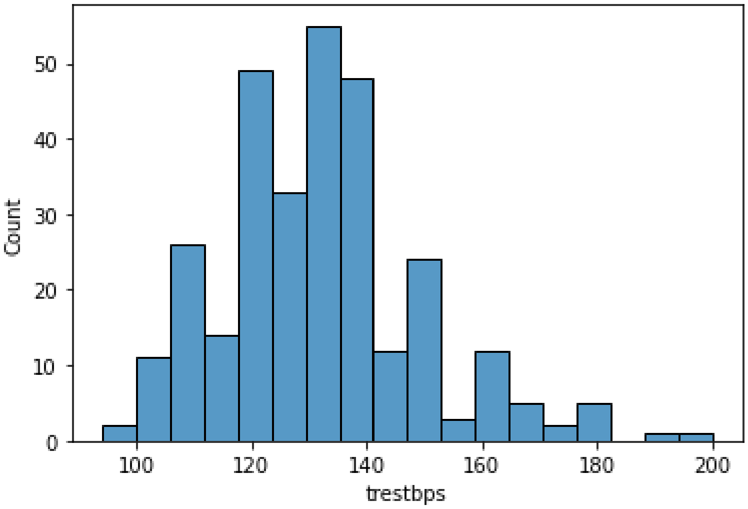


In [25]:

```
sns.histplot(data=df,x="trestbps")
```

Out[25]:

```
<AxesSubplot:xlabel='trestbps', ylabel='Count'>
```

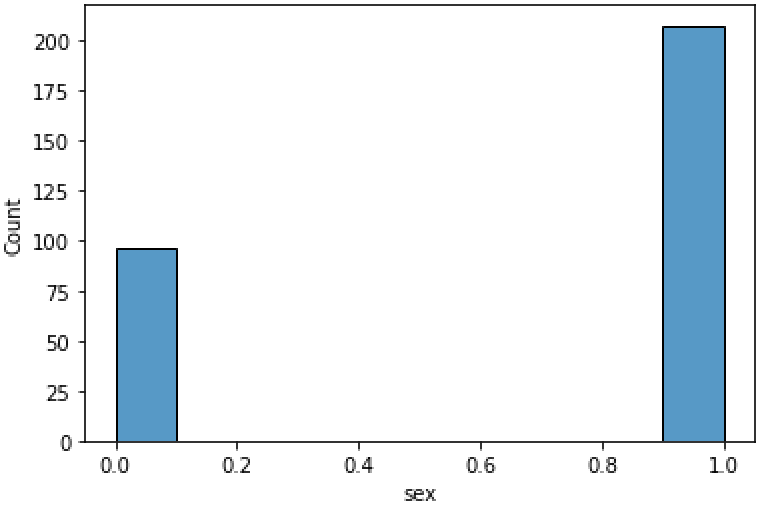


In [26]:

```
sns.histplot(data=df,x="sex")
```

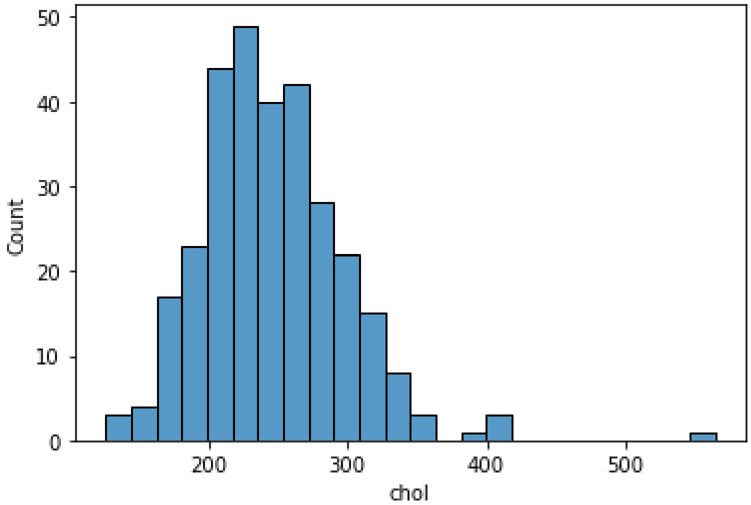
Out[26]:

```
<AxesSubplot:xlabel='sex', ylabel='Count'>
```



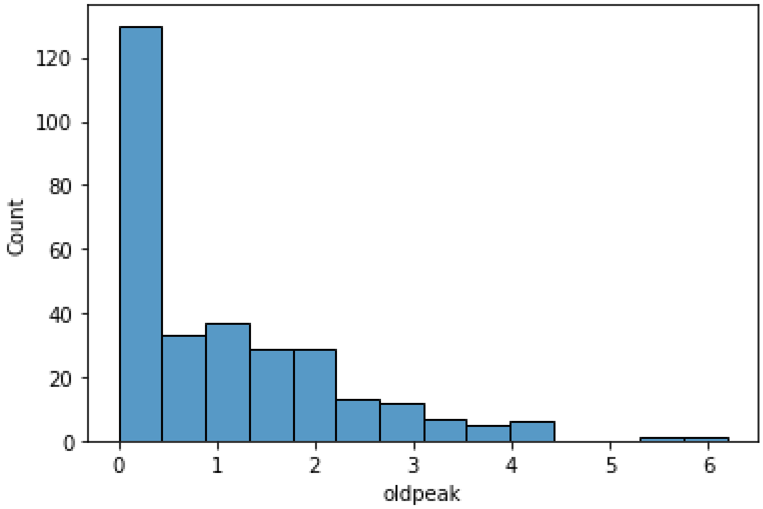
```
In [20]: sns.histplot(data=df,x="chol")
```

```
Out[20]: <AxesSubplot:xlabel='chol', ylabel='Count'>
```



```
In [28]: sns.histplot(data=df,x="oldpeak")
```

```
Out[28]: <AxesSubplot:xlabel='oldpeak', ylabel='Count'>
```



```
In [29]: df.sex
```

```
Out[29]: 0      1
         1      1
         2      0
         3      1
         4      0
               ..
         298    0
         299    1
         300    1
         301    1
         302    0
         Name: sex, Length: 303, dtype: int64
```

```
In [ ]: # Data variables which might be categorical in nature
```

```
In [30]: df.thalach.value_counts()
```

```
Out[30]: 162    11
163     9
160     9
152     8
173     8
       ..
128     1
129     1
134     1
137     1
202     1
Name: thalach, Length: 91, dtype: int64
```

```
In [31]: df.sex.value_counts()
```

```
Out[31]: 1    207
0     96
Name: sex, dtype: int64
```

```
In [27]: df.trestbps.value_counts()
```

```
Out[27]: 120    37
130    36
140    32
110    19
150    17
138    13
128    12
125    11
160    11
112     9
132     8
118     7
108     6
135     6
124     6
152     5
145     5
134     5
100     4
122     4
170     4
126     3
115     3
105     3
136     3
180     3
142     3
146     2
148     2
178     2
94      2
144     2
102     2
129     1
192     1
```

```
101     1
174     1
172     1
104     1
165     1
164     1
106     1
156     1
155     1
154     1
114     1
117     1
123     1
200     1
Name: trestbps, dtype: int64
```

In [32]:
```
df.chol.value_counts()
```

Out[32]:
```
204    6
197    6
234    6
269    5
212    5
      ..
215    1
210    1
200    1
195    1
417    1
Name: chol, Length: 152, dtype: int64
```

In [34]:
```
df.fbs.value_counts()
```

Out[34]:
```
0    258
1     45
Name: fbs, dtype: int64
```

In [35]:
```
df.exang.value_counts()
```

Out[35]:
```
0    204
1     99
Name: exang, dtype: int64
```

In [37]:
```
df.ca.value_counts()
```

Out[37]:
```
0    175
1     65
2     38
3     20
4      5
Name: ca, dtype: int64
```

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

Out[38]:
```
1    165
0    138
Name: target, dtype: int64
```

In [39]:

```
df.loc[df.sex==1,'sex']='male'
```

In [40]:

```
df.loc[df.sex==0,'sex']="female"
```

In [41]:

```
df
```

Out[41]:

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
0	63	male	3	145	233	1	0	150	0	2.3	0	0	1	1
1	37	male	2	130	250	0	1	187	0	3.5	0	0	2	1
2	41	female	1	130	204	0	0	172	0	1.4	2	0	2	1
3	56	male	1	120	236	0	1	178	0	0.8	2	0	2	1
4	57	female	0	120	354	0	1	163	1	0.6	2	0	2	1
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
298	57	female	0	140	241	0	1	123	1	0.2	1	0	3	0
299	45	male	3	110	264	0	1	132	0	1.2	1	0	3	0
300	68	male	0	144	193	1	1	141	0	3.4	1	2	3	0
301	57	male	0	130	131	0	1	115	1	1.2	1	1	3	0
302	57	female	1	130	236	0	0	174	0	0.0	1	1	2	0

303 rows × 14 columns

In [42]:

```
df.loc[df.cp==0,'cp']='no chest pain'
```

In [43]:

```
df.loc[df.cp==1,'cp']='low chest pain'
```

In [44]:

```
df.loc[df.cp==2,'cp']='medium chest pain'
```

In [45]:

```
df.loc[df.cp==3,'cp']='high chest pain'
```

In [46]:

```
df
```

Out[46]:

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	targ
0	63	male	high chest pain	145	233	1	0	150	0	2.3	0	0	1	
1	37	male	medium chest pain	130	250	0	1	187	0	3.5	0	0	2	

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	targ
2	41	female	low chest pain	130	204	0	0	172	0	1.4	2	0	2	
3	56	male	low chest pain	120	236	0	1	178	0	0.8	2	0	2	
4	57	female	no chest pain	120	354	0	1	163	1	0.6	2	0	2	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	
298	57	female	no chest pain	140	241	0	1	123	1	0.2	1	0	3	
299	45	male	high chest pain	110	264	0	1	132	0	1.2	1	0	3	
300	68	male	no chest pain	144	193	1	1	141	0	3.4	1	2	3	
301	57	male	no chest pain	130	131	0	1	115	1	1.2	1	1	3	
302	57	female	low chest pain	130	236	0	0	174	0	0.0	1	1	2	

303 rows × 14 columns

In [47]:

```
df.loc[df.fbs==0,'fbs']='<120mg/ml'
```

In [48]:

```
df.loc[df.fbs==1,'fbs']='>120mg/ml'
```

In [49]:

```
df
```

Out[49]:

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	th
0	63	male	high chest pain	145	233	>120mg/ml	0	150	0	2.3	0	0	
1	37	male	medium chest pain	130	250	<120mg/ml	1	187	0	3.5	0	0	
2	41	female	low chest pain	130	204	<120mg/ml	0	172	0	1.4	2	0	

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	th
3	56	male	low chest pain	120	236	<120mg/ml	1	178	0	0.8	2	0	
4	57	female	no chest pain	120	354	<120mg/ml	1	163	1	0.6	2	0	
...	...	...	...	...	...	...	...	...	...	...	...	...	
298	57	female	no chest pain	140	241	<120mg/ml	1	123	1	0.2	1	0	
299	45	male	high chest pain	110	264	<120mg/ml	1	132	0	1.2	1	0	
300	68	male	no chest pain	144	193	>120mg/ml	1	141	0	3.4	1	2	
301	57	male	no chest pain	130	131	<120mg/ml	1	115	1	1.2	1	1	
302	57	female	low chest pain	130	236	<120mg/ml	0	174	0	0.0	1	1	

303 rows × 14 columns

In [50]:

```
df.loc[df.restecg==0,'restecg']='normal ecg '
```

In [51]:

```
df.loc[df.restecg==1,'restecg']='not normal ecg'
```

In [52]:

```
df.loc[df.restecg==2,'restecg']='high ecg'
```

In [53]:

```
df
```

Out[53]:

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	th
0	63	male	high chest pain	145	233	>120mg/ml	normal ecg	150	0	2.3	0	0	
1	37	male	medium chest pain	130	250	<120mg/ml	not normal ecg	187	0	3.5	0	0	
2	41	female	low chest pain	130	204	<120mg/ml	normal ecg	172	0	1.4	2	0	

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	th
3	56	male	low chest pain	120	236	<120mg/ml	not normal ecg	178	0	0.8	2	0	
4	57	female	no chest pain	120	354	<120mg/ml	not normal ecg	163	1	0.6	2	0	
...	...	...	...	...	...	...	...	...	...	...	...	...	
298	57	female	no chest pain	140	241	<120mg/ml	not normal ecg	123	1	0.2	1	0	
299	45	male	high chest pain	110	264	<120mg/ml	not normal ecg	132	0	1.2	1	0	
300	68	male	no chest pain	144	193	>120mg/ml	not normal ecg	141	0	3.4	1	2	
301	57	male	no chest pain	130	131	<120mg/ml	not normal ecg	115	1	1.2	1	1	
302	57	female	low chest pain	130	236	<120mg/ml	normal ecg	174	0	0.0	1	1	

303 rows × 14 columns

In [54]:

```
df.loc[df.exang==0,'exang']='no'
```

In [55]:

```
df.loc[df.exang==1,'exang']='yes'
```

In [56]:

```
df
```

Out[56]:

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	th
0	63	male	high chest pain	145	233	>120mg/ml	normal ecg	150	no	2.3	0	0	
1	37	male	medium chest pain	130	250	<120mg/ml	not normal ecg	187	no	3.5	0	0	
2	41	female	low chest pain	130	204	<120mg/ml	normal ecg	172	no	1.4	2	0	
3	56	male	low chest pain	120	236	<120mg/ml	not normal ecg	178	no	0.8	2	0	

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	th
4	57	female	no chest pain	120	354	<120mg/ml	not normal ecg	163	yes	0.6	2	0	
...	...	...	...	...	...	...	...	...	...	...	...	...	
298	57	female	no chest pain	140	241	<120mg/ml	not normal ecg	123	yes	0.2	1	0	
299	45	male	high chest pain	110	264	<120mg/ml	not normal ecg	132	no	1.2	1	0	
300	68	male	no chest pain	144	193	>120mg/ml	not normal ecg	141	no	3.4	1	2	
301	57	male	no chest pain	130	131	<120mg/ml	not normal ecg	115	yes	1.2	1	1	
302	57	female	low chest pain	130	236	<120mg/ml	normal ecg	174	no	0.0	1	1	

303 rows × 14 columns

In [57]:

```
df.loc[df.slope==0,'slope']='upsloping'
```

In [58]:

```
df.loc[df.slope==1,'slope']='flat'
```

In [59]:

```
df.loc[df.slope==2,'slope']='downsloping'
```

In [60]:

```
df
```

Out[60]:

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope
0	63	male	high chest pain	145	233	>120mg/ml	normal ecg	150	no	2.3	upsloping
1	37	male	medium chest pain	130	250	<120mg/ml	not normal ecg	187	no	3.5	upsloping
2	41	female	low chest pain	130	204	<120mg/ml	normal ecg	172	no	1.4	downsloping
3	56	male	low chest pain	120	236	<120mg/ml	not normal ecg	178	no	0.8	downsloping

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope
4	57	female	no chest pain	120	354	<120mg/ml	not normal ecg	163	yes	0.6	downsloping
...	...	...	...	...	...	...	...	...	...	...	...
298	57	female	no chest pain	140	241	<120mg/ml	not normal ecg	123	yes	0.2	flat
299	45	male	high chest pain	110	264	<120mg/ml	not normal ecg	132	no	1.2	flat
300	68	male	no chest pain	144	193	>120mg/ml	not normal ecg	141	no	3.4	flat
301	57	male	no chest pain	130	131	<120mg/ml	not normal ecg	115	yes	1.2	flat
302	57	female	low chest pain	130	236	<120mg/ml	normal ecg	174	no	0.0	flat

303 rows × 14 columns

In [61]:

```
df.loc[df.thal==0,'thal']='normal'
```

In [62]:

```
df.loc[df.thal==1,'thal']='fixed defect'
```

In [63]:

```
df.loc[df.thal==2,'thal']='reversable defect'
```

In [64]:

```
df.loc[df.thal==3,'thal']='major defect'
```

In [65]:

```
df
```

Out[65]:

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope
0	63	male	high chest pain	145	233	>120mg/ml	normal ecg	150	no	2.3	upsloping
1	37	male	medium chest pain	130	250	<120mg/ml	not normal ecg	187	no	3.5	upsloping
2	41	female	low chest pain	130	204	<120mg/ml	normal ecg	172	no	1.4	downsloping

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope
3	56	male	low chest pain	120	236	<120mg/ml	not normal ecg	178	no	0.8	downsloping
4	57	female	no chest pain	120	354	<120mg/ml	not normal ecg	163	yes	0.6	downsloping
...	...	...	...	...	...	...	...	...	...	...	...
298	57	female	no chest pain	140	241	<120mg/ml	not normal ecg	123	yes	0.2	flat
299	45	male	high chest pain	110	264	<120mg/ml	not normal ecg	132	no	1.2	flat
300	68	male	no chest pain	144	193	>120mg/ml	not normal ecg	141	no	3.4	flat
301	57	male	no chest pain	130	131	<120mg/ml	not normal ecg	115	yes	1.2	flat
302	57	female	low chest pain	130	236	<120mg/ml	normal ecg	174	no	0.0	flat

303 rows × 14 columns

In [66]:

```
df.loc[df.target==0,'target']='Negative disease'
```

In [67]:

```
df.loc[df.target==1,'target']='Positive disease'
```

In [68]:

```
df
```

Out[68]:

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope
0	63	male	high chest pain	145	233	>120mg/ml	normal ecg	150	no	2.3	upsloping
1	37	male	medium chest pain	130	250	<120mg/ml	not normal ecg	187	no	3.5	upsloping
2	41	female	low chest pain	130	204	<120mg/ml	normal ecg	172	no	1.4	downsloping
3	56	male	low chest pain	120	236	<120mg/ml	not normal ecg	178	no	0.8	downsloping

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope
4	57	female	no chest pain	120	354	<120mg/ml	not normal ecg	163	yes	0.6	downsloping
...	...	...	...	...	...	...	...	...	...	...	...
298	57	female	no chest pain	140	241	<120mg/ml	not normal ecg	123	yes	0.2	flat
299	45	male	high chest pain	110	264	<120mg/ml	not normal ecg	132	no	1.2	flat
300	68	male	no chest pain	144	193	>120mg/ml	not normal ecg	141	no	3.4	flat
301	57	male	no chest pain	130	131	<120mg/ml	not normal ecg	115	yes	1.2	flat
302	57	female	low chest pain	130	236	<120mg/ml	normal ecg	174	no	0.0	flat

303 rows × 14 columns

In [69]:

```
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    object 
 2   cp        303 non-null    object 
 3   trestbps  303 non-null    int64  
 4   chol      303 non-null    int64  
 5   fbs       303 non-null    object 
 6   restecg   303 non-null    object 
 7   thalach   303 non-null    int64  
 8   exang     303 non-null    object 
 9   oldpeak   303 non-null    float64
 10  slope     303 non-null    object 
 11  ca        303 non-null    int64  
 12  thal      303 non-null    object 
 13  target    303 non-null    object 
dtypes: float64(1), int64(5), object(8)
memory usage: 33.3+ KB
```

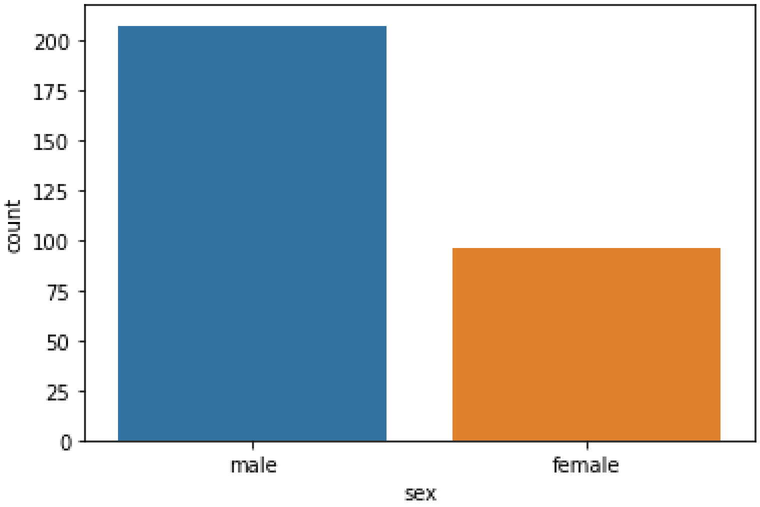
In [ ]:

```
# countplot for showing categorical columns in visualization pattern
```

In [70]:

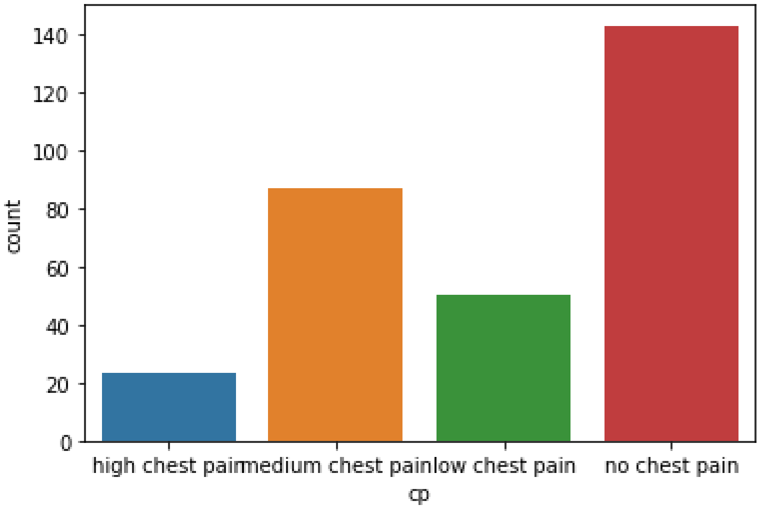
```
sns.countplot(x="sex",data=df)
```

```
Out[70]: <AxesSubplot:xlabel='sex', ylabel='count'>
```



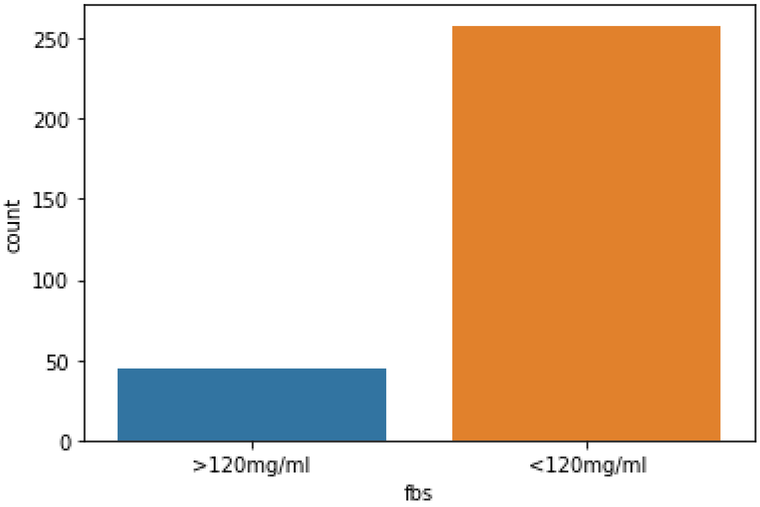
```
In [71]: sns.countplot(x='cp',data=df)
```

```
Out[71]: <AxesSubplot:xlabel='cp', ylabel='count'>
```



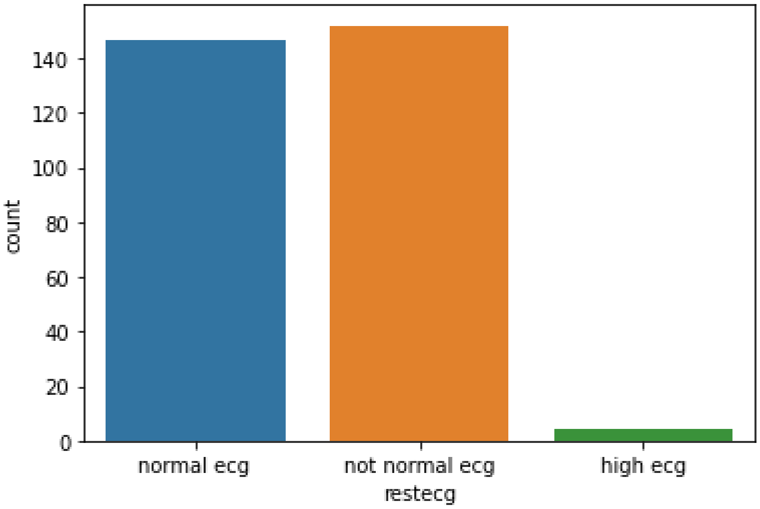
```
In [72]: sns.countplot(x='fbs',data=df)
```

```
Out[72]: <AxesSubplot:xlabel='fbs', ylabel='count'>
```



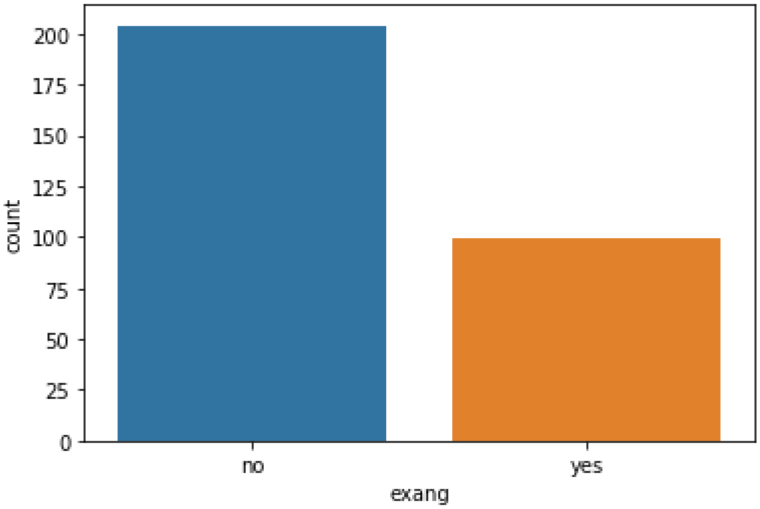
In [73]:
```
sns.countplot(x='restecg',data=df)
```

Out[73]:
```
<AxesSubplot:xlabel='restecg', ylabel='count'>
```



In [74]:
```
sns.countplot(x='exang',data=df)
```

Out[74]:
```
<AxesSubplot:xlabel='exang', ylabel='count'>
```

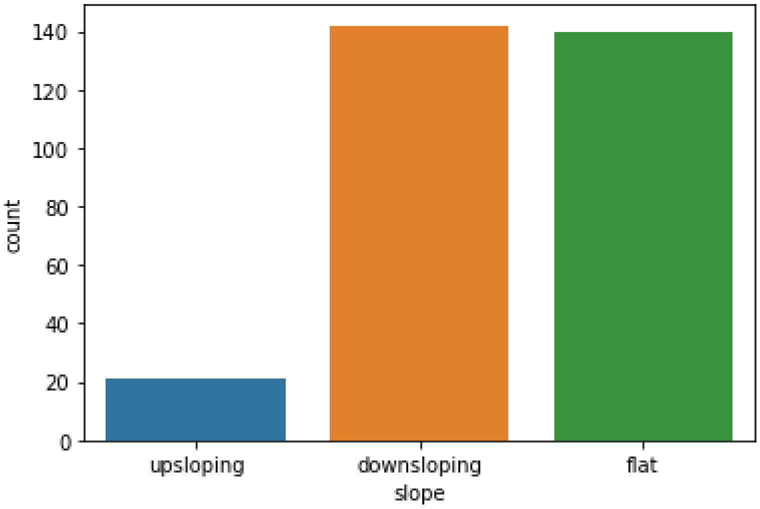


In [75]:

```
sns.countplot(x='slope',data=df)
```

Out[75]:

```
<AxesSubplot:xlabel='slope', ylabel='count'>
```

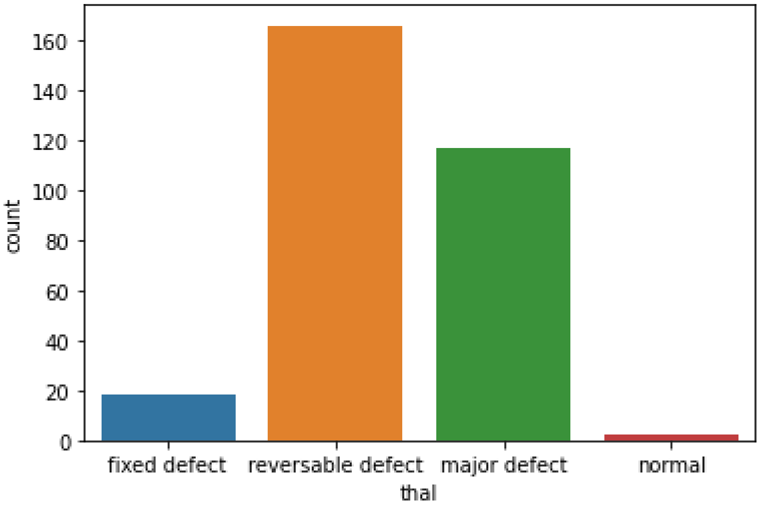


In [76]:

```
sns.countplot(x='thal',data=df)
```

Out[76]:

```
<AxesSubplot:xlabel='thal', ylabel='count'>
```

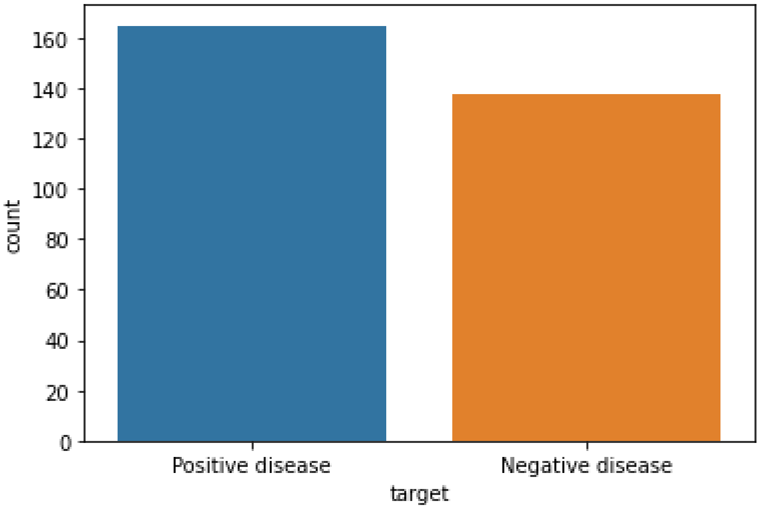


In [77]:

```
sns.countplot(x='target',data=df)
```

Out[77]:

```
<AxesSubplot:xlabel='target', ylabel='count'>
```



In [ ]:

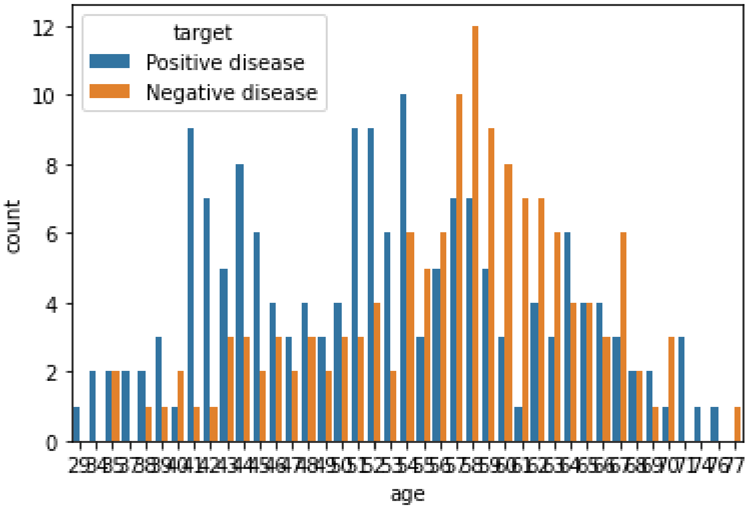
```
#Study the occurrence of CVD across different ages.
```

In [78]:

```
sns.countplot(x='age',data=df,hue="target")
```

Out[78]:

```
<AxesSubplot:xlabel='age', ylabel='count'>
```



In [ ]:

```
Can we detect heart attack based on anomalies in resting blood pressure of the patient?
```

In [79]:

```
df.trestbps.value_counts()
```

Out[79]:

```
120    37
130    36
140    32
110    19
150    17
138    13
128    12
125    11
160    11
112     9
132     8
118     7
```

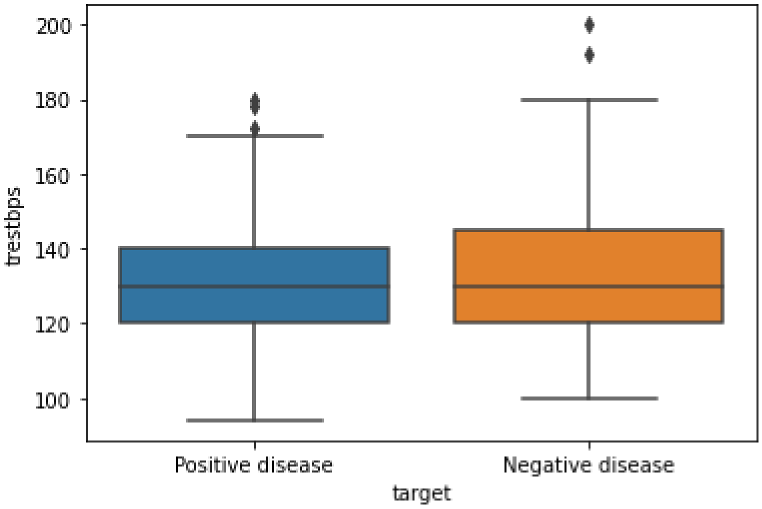
```
108     6
135     6
124     6
152     5
145     5
134     5
100     4
122     4
170     4
126     3
115     3
105     3
136     3
180     3
142     3
146     2
148     2
178     2
94      2
144     2
102     2
129     1
192     1
101     1
174     1
172     1
104     1
165     1
164     1
106     1
156     1
155     1
154     1
114     1
117     1
123     1
200     1
Name: trestbps, dtype: int64
```

In [80]:

```
sns.boxplot(x="target",y="trestbps",data=df)
```

Out[80]:

```
<AxesSubplot:xlabel='target', ylabel='trestbps'>
```



In [ ]:

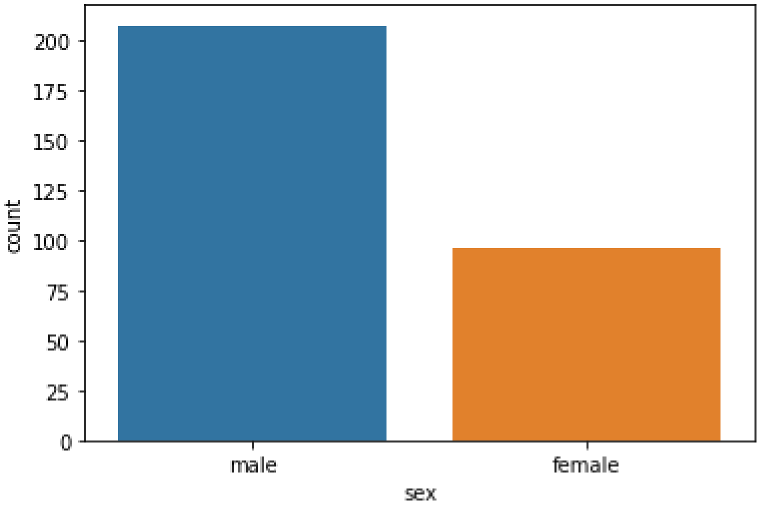
```
#Study the composition of overall patients w.r.t . gender.
```

In [81]:

```
sns.countplot(x='sex',data=df)
```

Out[81]:

```
<AxesSubplot:xlabel='sex', ylabel='count'>
```



In [ ]:

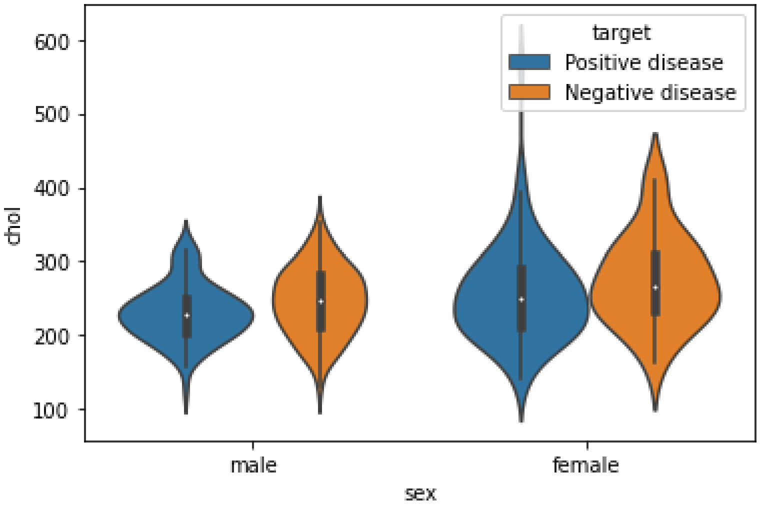
```
# relationship between cholesterol levels and our target variable
```

In [85]:

```
sns.violinplot(x='sex',y='chol',hue='target',data=df)
```

Out[85]:

```
<AxesSubplot:xlabel='sex', ylabel='chol'>
```



In [ ]:

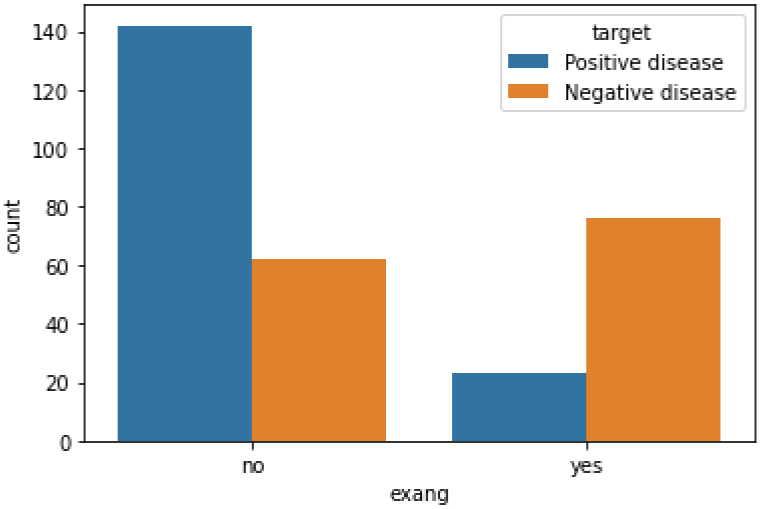
```
#relationship between peak exercising and occurrence of heart attack?
```

In [94]:

```
sns.countplot(x='exang',data=df,hue='target')
```

Out[94]:

```
<AxesSubplot:xlabel='exang', ylabel='count'>
```



In [ ]:

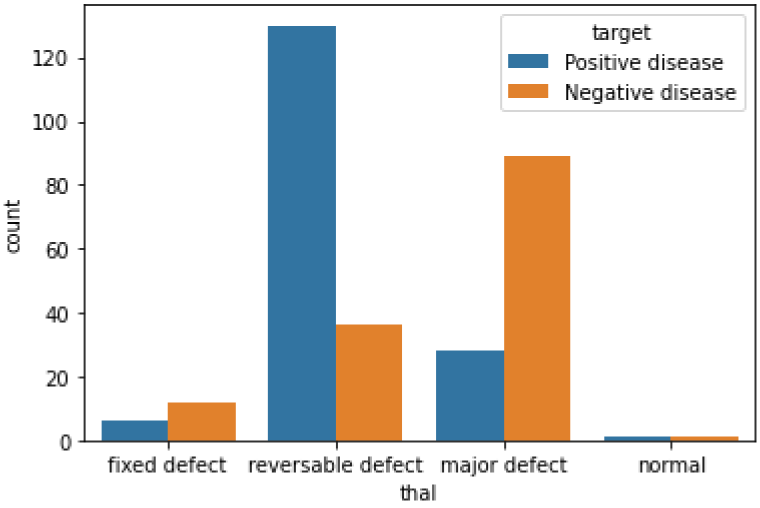
```
# Is thalassemia a major cause of CVD? How are the other factors determining the occurr
```

In [109…

```
sns.countplot(x='thal',data=df,hue='target')
```

Out[109…

```
<AxesSubplot:xlabel='thal', ylabel='count'>
```



In [115…

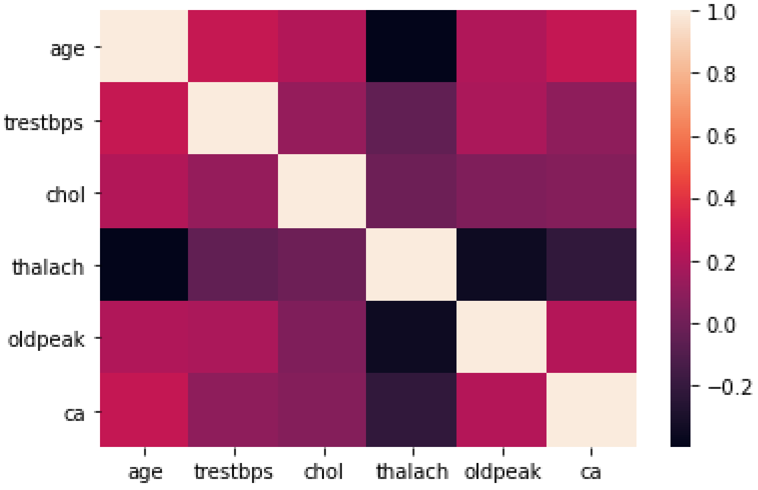
```
tc=df.corr()
```

In [116…

```
sns.heatmap(tc)
```

Out[116…

```
<AxesSubplot:>
```



In [ ]:

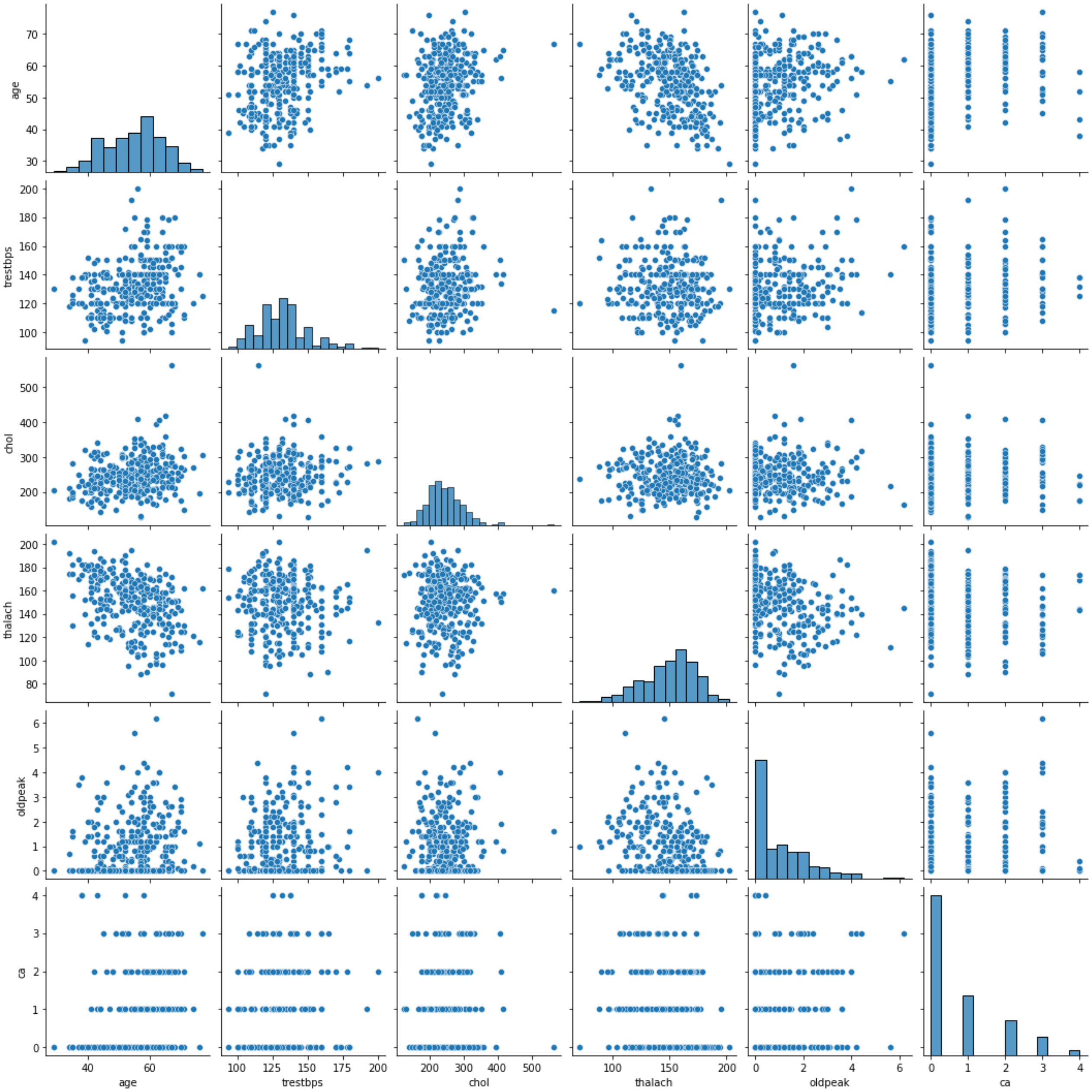
```
# pair plot to understand the relationship between all the given variables
```

In [92]:

```
sns.pairplot(df)
```

Out[92]:

```
<seaborn.axisgrid.PairGrid at 0x20c9d216ee0>
```



In [ ]:

```
# Logistic Regression Model
```

In [1]:

```
import pandas as pd
import numpy as np
```

In [2]:

```
data=pd.read_csv('heartattack.csv')
```

In [3]:

```
data
```

Out[3]:

	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

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
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
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
298	57	0	0	140	241	0	1	123	1	0.2	1	0	3	0
299	45	1	3	110	264	0	1	132	0	1.2	1	0	3	0
300	68	1	0	144	193	1	1	141	0	3.4	1	2	3	0
301	57	1	0	130	131	0	1	115	1	1.2	1	1	3	0
302	57	0	1	130	236	0	0	174	0	0.0	1	1	2	0

303 rows × 14 columns

In [5]:

```
from sklearn.model_selection import train_test_split as split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import classification_report,accuracy_score
```

In [7]:

```
data_dummy=pd.get_dummies(data)
data_dummy.columns=data_dummy.columns.str.replace(' ','_')
train,test = split (data_dummy,test_size=.30,random_state=12)
train.shape
train.head(2)
X_train=train.drop('target',axis=1)
Y_train=train.target
X_test=test.drop('target',axis=1)

Y_test=test.target
lr=LogisticRegression()
lr.fit(X_train,Y_train)
pred=lr.predict(X_test)
accuracy_score(y_true=Y_test,y_pred=pred)
print (classification_report(y_true=Y_test,y_pred=pred))
```

```
              precision    recall  f1-score   support

           0       0.84      0.80      0.82        45
           1       0.81      0.85      0.83        46

    accuracy                           0.82        91
   macro avg       0.82      0.82      0.82        91
weighted avg       0.82      0.82      0.82        91

C:\Users\pinku\anaconda3\lib\site-packages\sklearn\linear_model\_logistic.py:763: Conver
genceWarning: lbfgs failed to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
  n_iter_i = _check_optimize_result(
```

In [9]:

```
from sklearn.metrics import confusion_matrix
print( confusion_matrix(Y_test,pred))
```

```
[[36  9]
 [ 7 39]]
```

In [ ]:

```
 
```