In [1]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
```

In [2]:

```
hd=pd.read_csv(r'S:\DOCS\5th,6th\EDA_project\heart.csv')
```

In [3]:

```
hd
```

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
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 [4]:

```
hd.shape
```

Out[4]:

```
(303, 14)
```

In [5]:

```
hd.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 [6]:

```
hd.head() # exploring data
```

Out[6]:

	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 [7]:
```
hd.isnull().any().any() # checking null vaule
```

Out[7]:
```
False
```

In [8]:
```
hd.tail()
```

Out[8]:

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
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

In [9]:
```
hd.describe()
```

Out[9]:

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	o
count	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.0
mean	54.366337	0.683168	0.966997	131.623762	246.264026	0.148515	0.528053	149.646865	0.326733	1.0
std	9.082101	0.466011	1.032052	17.538143	51.830751	0.356198	0.525860	22.905161	0.469794	1.
min	29.000000	0.000000	0.000000	94.000000	126.000000	0.000000	0.000000	71.000000	0.000000	0.0
25%	47.500000	0.000000	0.000000	120.000000	211.000000	0.000000	0.000000	133.500000	0.000000	0.0
50%	55.000000	1.000000	1.000000	130.000000	240.000000	0.000000	1.000000	153.000000	0.000000	0.8
75%	61.000000	1.000000	2.000000	140.000000	274.500000	0.000000	1.000000	166.000000	1.000000	1.6
max	77.000000	1.000000	3.000000	200.000000	564.000000	1.000000	2.000000	202.000000	1.000000	6.2

In [10]:
```
# univariate analysis
```

In [11]:
```
hd['target'].unique()
```

Out[11]:
```
array([1, 0], dtype=int64)
```

In [12]:
```
hd['target'].nunique()
```

Out[12]:
```
2
```

In [13]:
```
hd['target'].value_counts()
```

Out[13]:
```
1    165
0    138
Name: target, dtype: int64
```

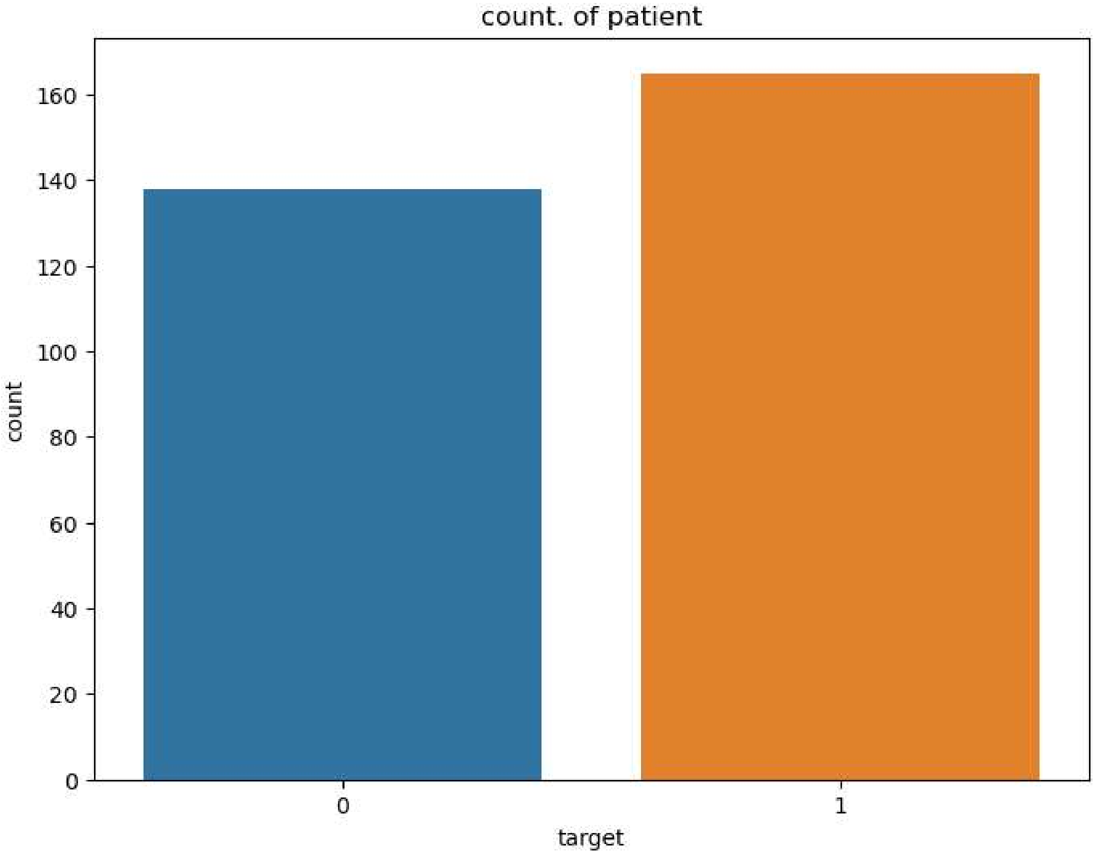
In [14]:
```
# so, patient with heart deases are 165 and without heart disease are 138
```

In [15]:

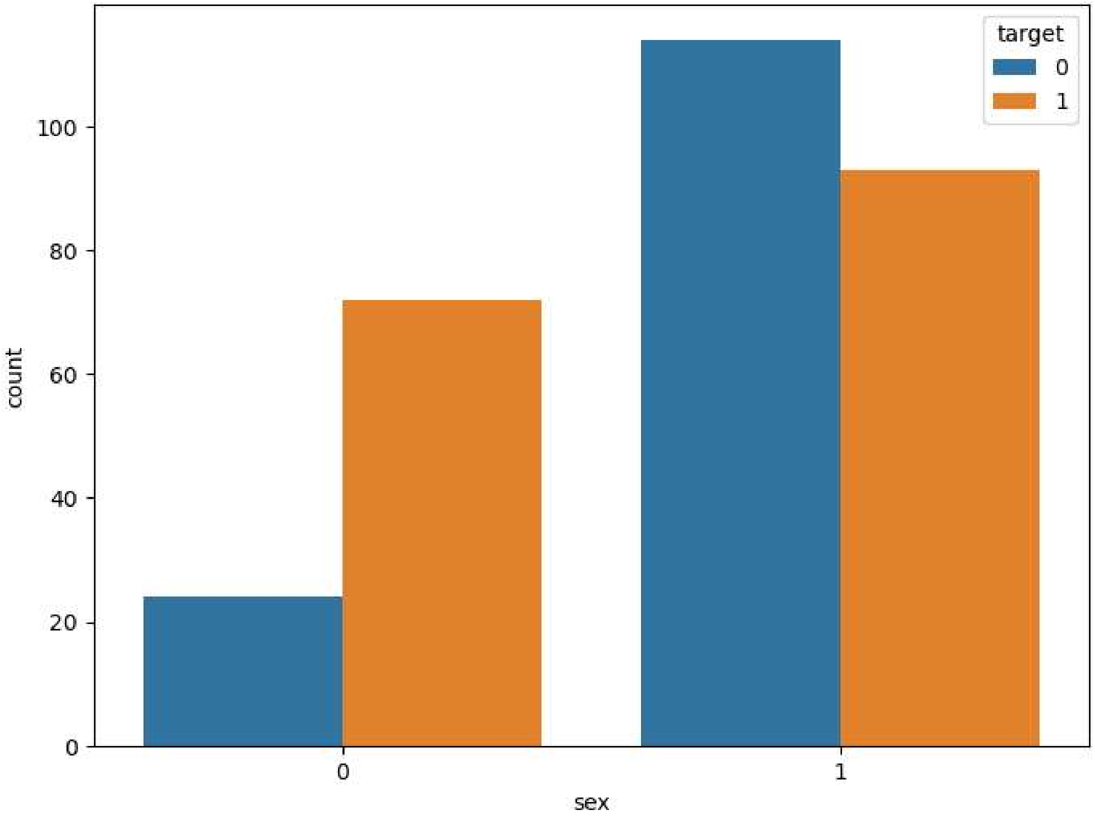
```
f, ax=plt.subplots(figsize=(8,6))
ax=sns.countplot(x=hd['target'] )
plt.title('count. of patient')

```

Out[15]:

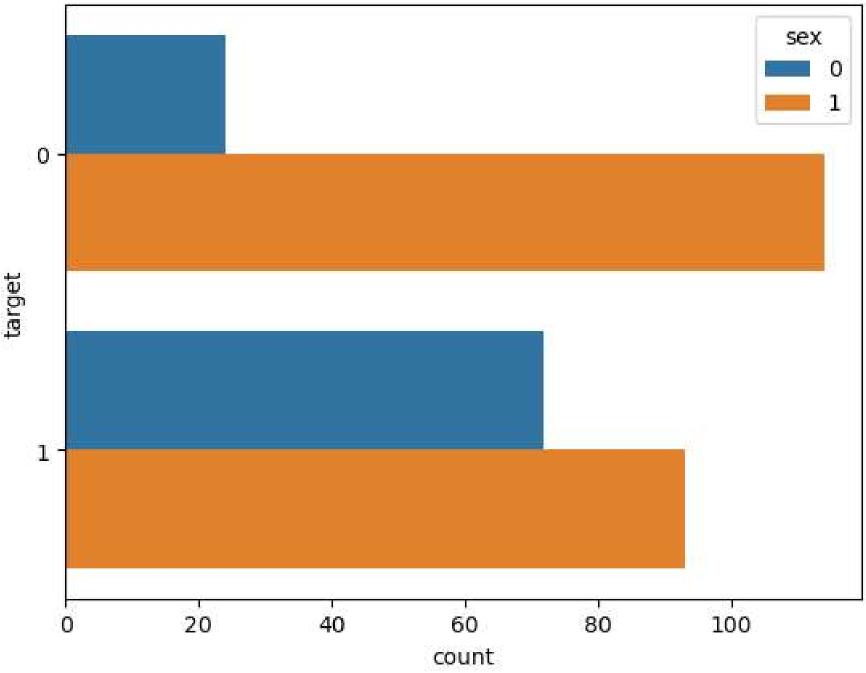
```
Text(0.5, 1.0, 'count. of patient')
```



```
In [16]: f,ax=plt.subplots(figsize=(8,6))
         ax=sns.countplot(data=hd, x='sex' ,hue='target')


```

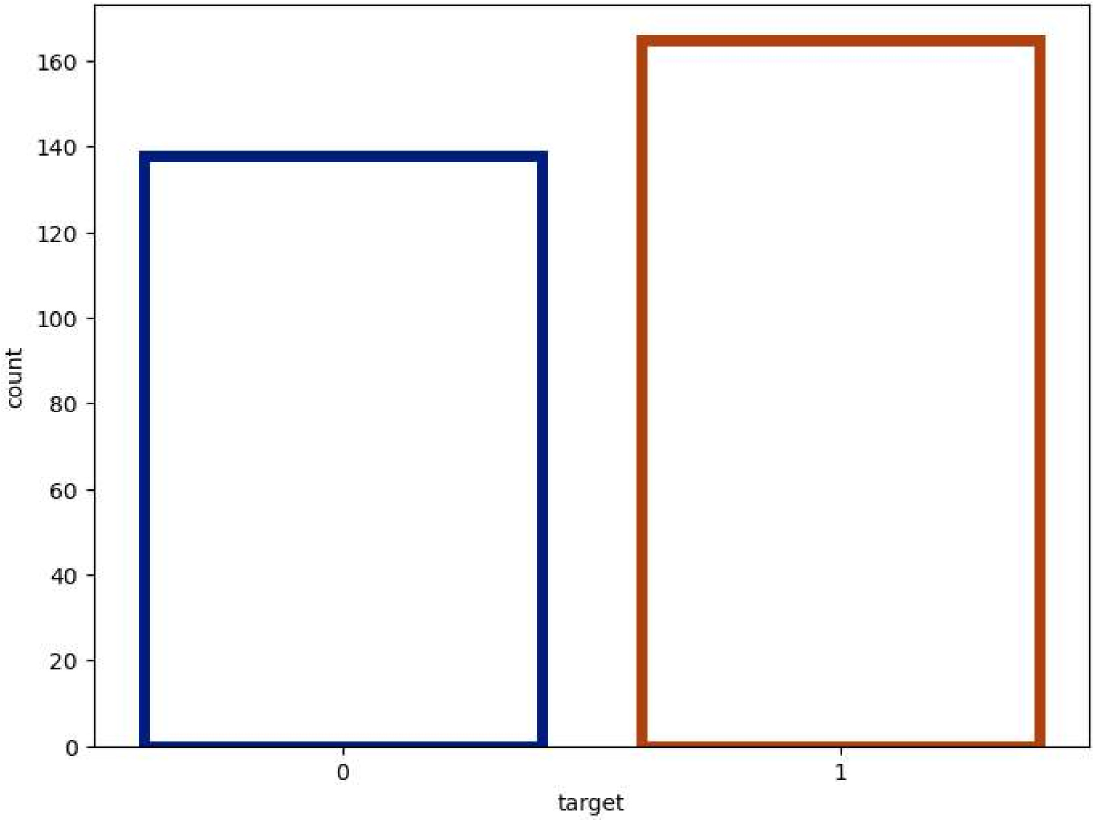


```
In [17]: sns.countplot(data=hd, y='target',  hue='sex')

Out[17]: <Axes: xlabel='count', ylabel='target'>
```



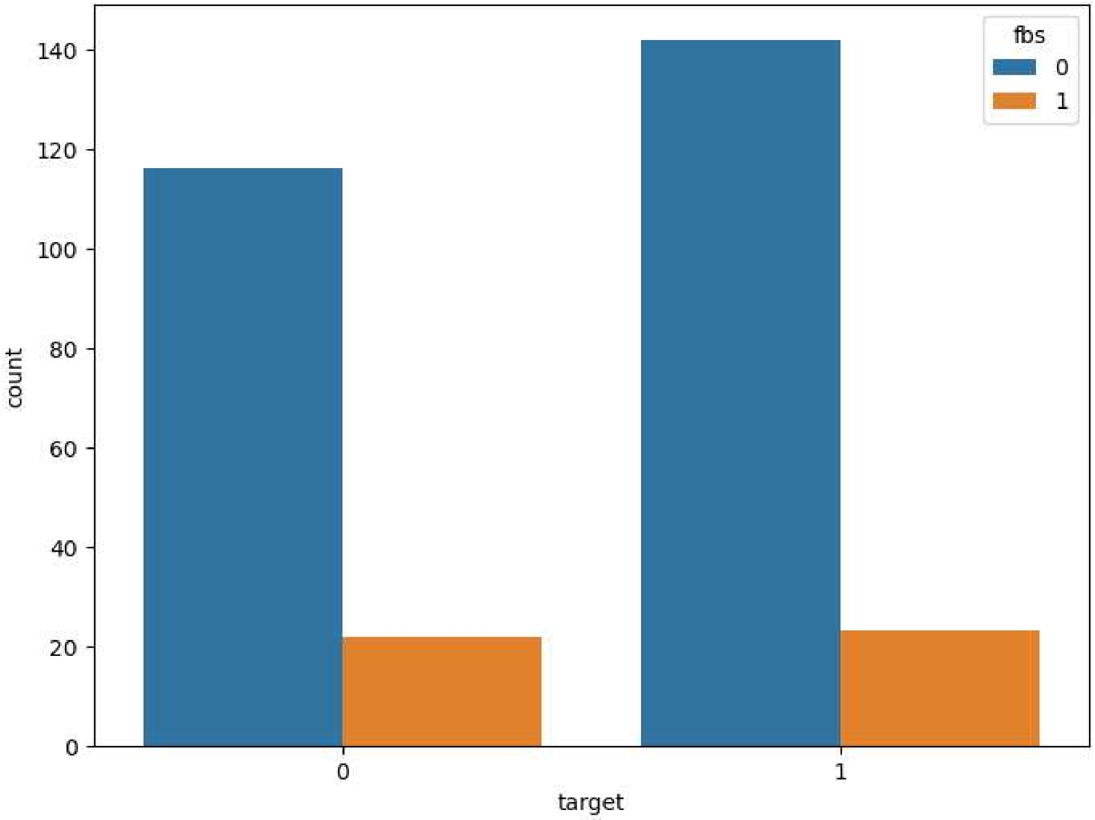
In [18]:

```
f, ax=plt.subplots(figsize=(8,6))
ax=sns.countplot(data=hd ,  x='target' , facecolor=(0,0,0,0) ,linewidth=5, edgecolor=sns.color
```



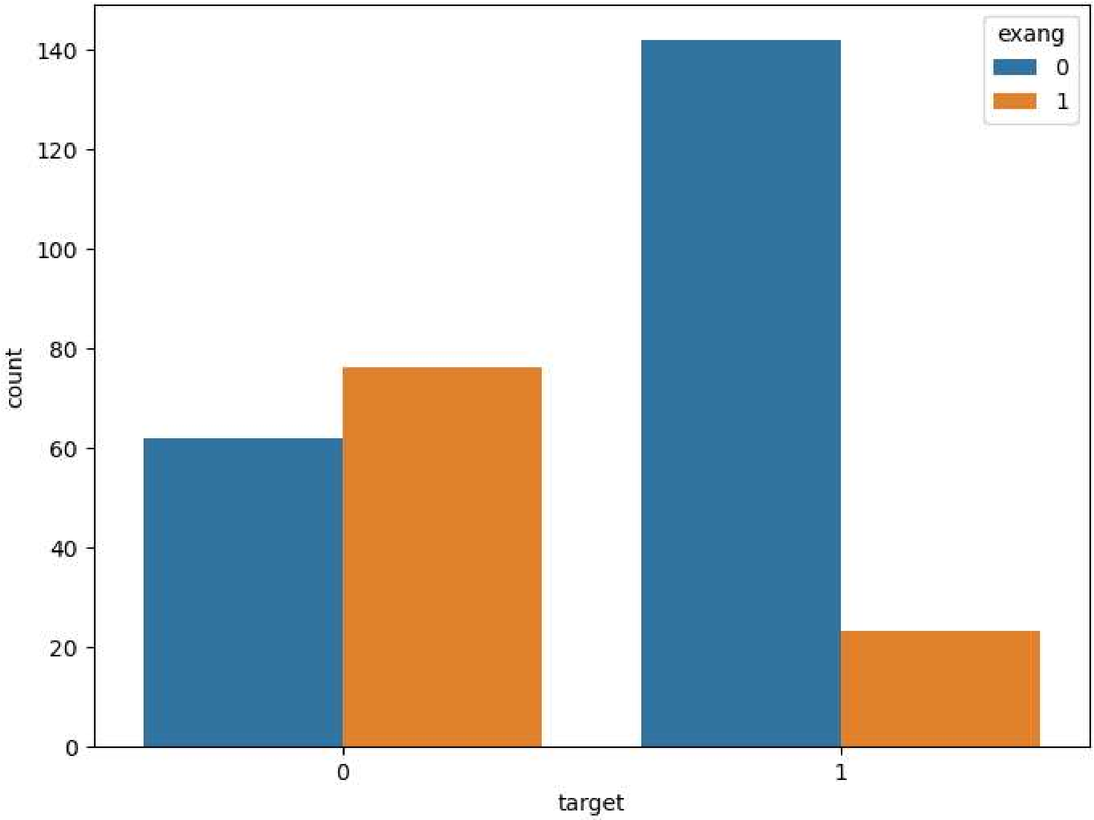
In [19]:

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



In [20]:

```python
f,ax=plt.subplots(figsize=(8,6))
ax=sns.countplot(data=hd, x='target', hue='exang')
```



```
# findinds of univariate analysis
''' refers to the presence of heart disease in the patient.

It is integer valued as it contains two integers 0 and 1 - (0 stands for absence of heart disease and 1 for presence of heart disease).

1 stands for presence of heart disease. So, there are 165 patients suffering from heart disease.

Similarly, 0 stands for absence of heart disease. So, there are 138 patients who do not have any heart disease.

There are 165 patients suffering from heart disease, and

There are 138 patients who do not have any heart disease.

Out of 96 females - 72 have heart disease and 24 do not have heart disease.

Similarly, out of 207 males - 93 have heart disease and 114 do not have heart disease.'''
```

In [22]:

```python
#bi variate analysis
```

In [26]:

```python
correlations=hd.corr()
```

In [27]:

```python
correlations['target']
```

Out[27]:

```
age        -0.225439
sex        -0.280937
cp          0.433798
trestbps   -0.144931
chol       -0.085239
fbs        -0.028046
restecg     0.137230
thalach     0.421741
exang      -0.436757
oldpeak    -0.430696
slope       0.345877
ca         -0.391724
thal       -0.344029
target      1.000000
Name: target, dtype: float64
```

```python
''' correlation ranges from -1 to +1, +1 indicates strong positive corelation but there is not
    any value with 1
    cp and thalach is near so  i will analyse these with target
'''
```

In [29]:

```python
hd['cp'].unique()
```

Out[29]:

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

In [30]:

```python
# cp is categorical variable
```

In [32]:

```python
hd['cp'].value_counts()
```

Out[32]:

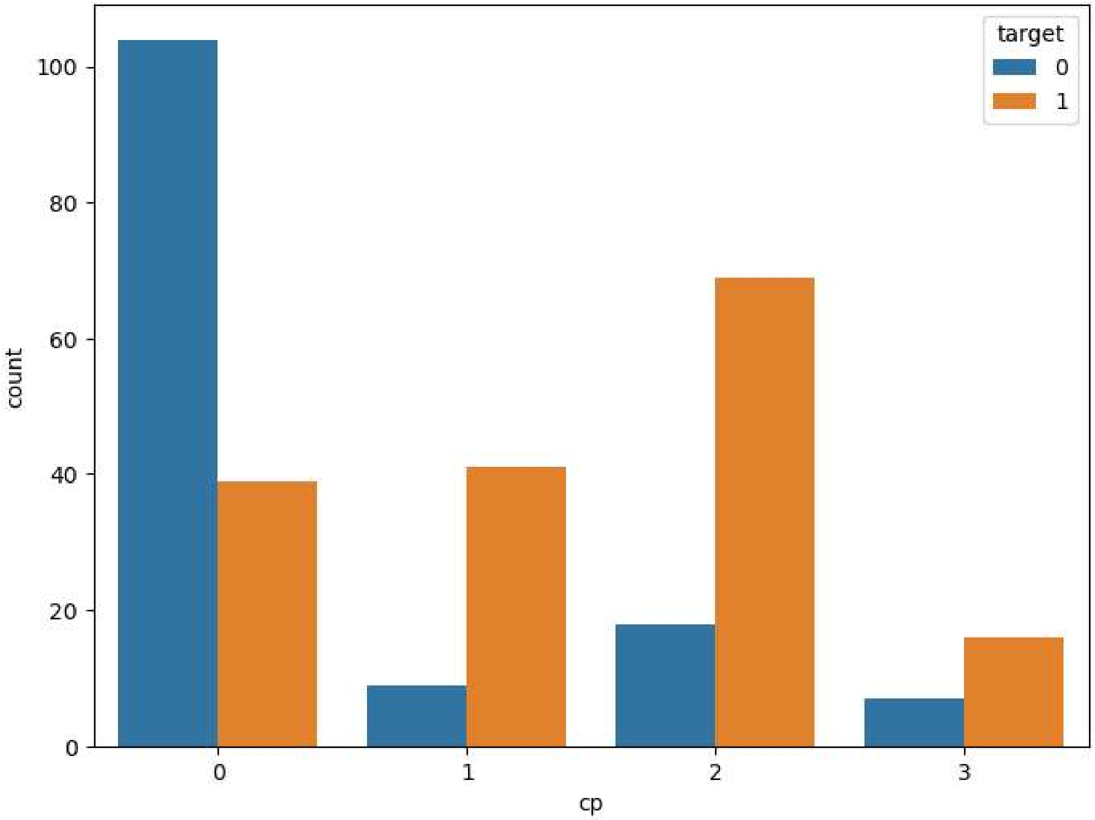
```
0    143
2     87
1     50
3     23
Name: cp, dtype: int64
```

In [ ]:

```python
# cp is categorical value contain only 4 types of balue 0,1,2,3
```

In [34]:

```
f, ax=plt.subplots(figsize=(8,6))
ax=sns.countplot(data=hd, x='cp' , hue='target')
```



In [35]:

```
# o means no chest pain while 1,2,3 are the severity of chest pain

```

In [40]:

```
hd.groupby('cp')['target'].value_counts()
```

Out[40]:

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

In [41]:

```
# target and thalach
```

In [42]:

```
hd['thalach'].unique()
```

Out[42]:

```
array([150, 187, 172, 178, 163, 148, 153, 173, 162, 174, 160, 139, 171,
       144, 158, 114, 151, 161, 179, 137, 157, 123, 152, 168, 140, 188,
       125, 170, 165, 142, 180, 143, 182, 156, 115, 149, 146, 175, 186,
       185, 159, 130, 190, 132, 147, 154, 202, 166, 164, 184, 122, 169,
       138, 111, 145, 194, 131, 133, 155, 167, 192, 121,  96, 126, 105,
       181, 116, 108, 129, 120, 112, 128, 109, 113,  99, 177, 141, 136,
        97, 127, 103, 124,  88, 195, 106,  95, 117,  71, 118, 134,  90],
      dtype=int64)
```

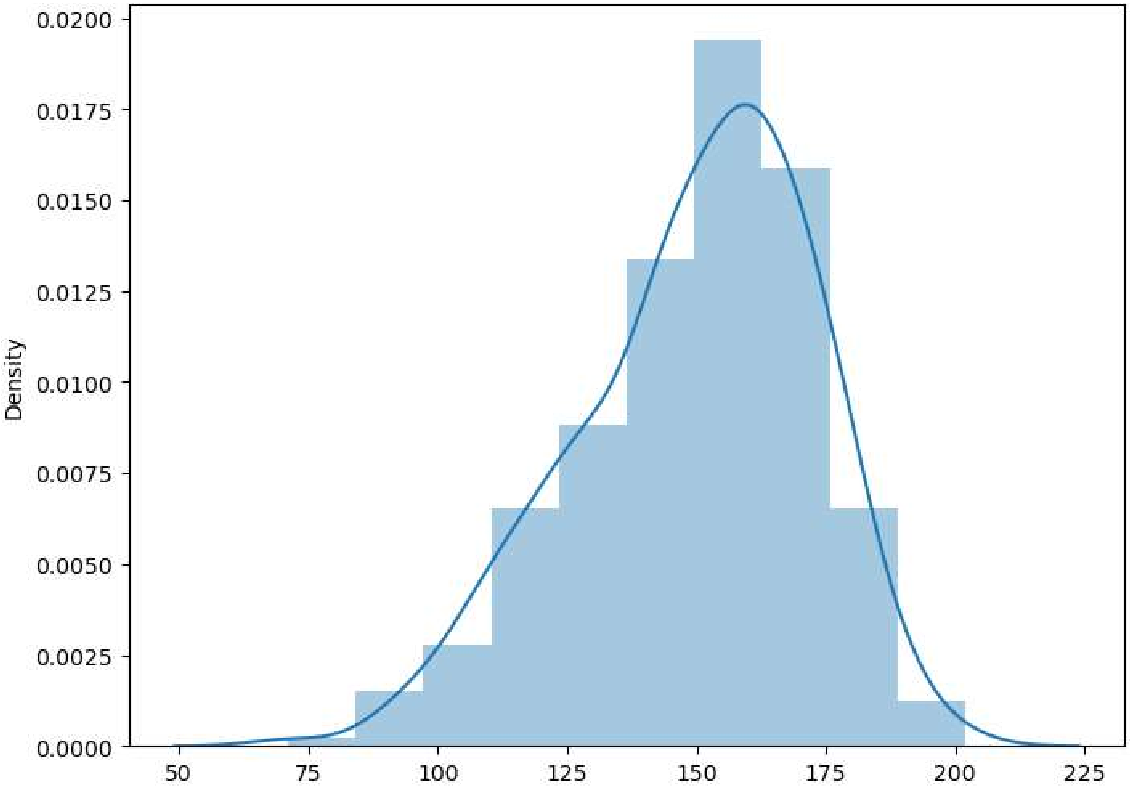
In [43]:

```
# the value in thalach are 91 so it is numerical variable
# we use frequency distribution
```

In [50]:

```
f, ax=plt.subplots(figsize=(8,6))
ax=sns.distplot( x=hd['thalach'] ,bins=10)
```

```
C:\Users\ASUS\AppData\Local\Temp\ipykernel_7172\3496898270.py:2: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751 (https://gist.github.com/mwasko
m/de44147ed2974457ad6372750bbe5751)

  ax=sns.distplot( x=hd['thalach'] ,bins=10)
```



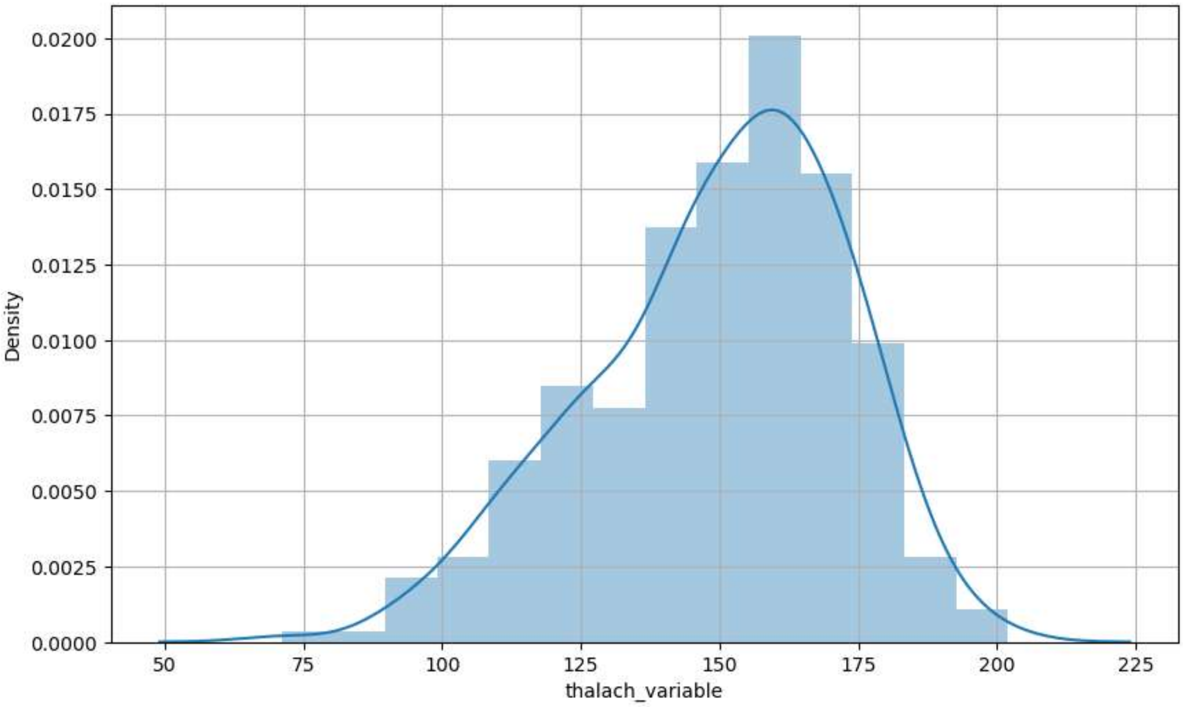
In [51]:

```
# distribution is -ve skewed
```

In [149]:

```
f,ax=plt.subplots(figsize=(10,6))
x=hd['thalach']
x=pd.Series(x,name='thalach_variable')
ax=sns.distplot(x)
plt.grid()
plt.show()
```

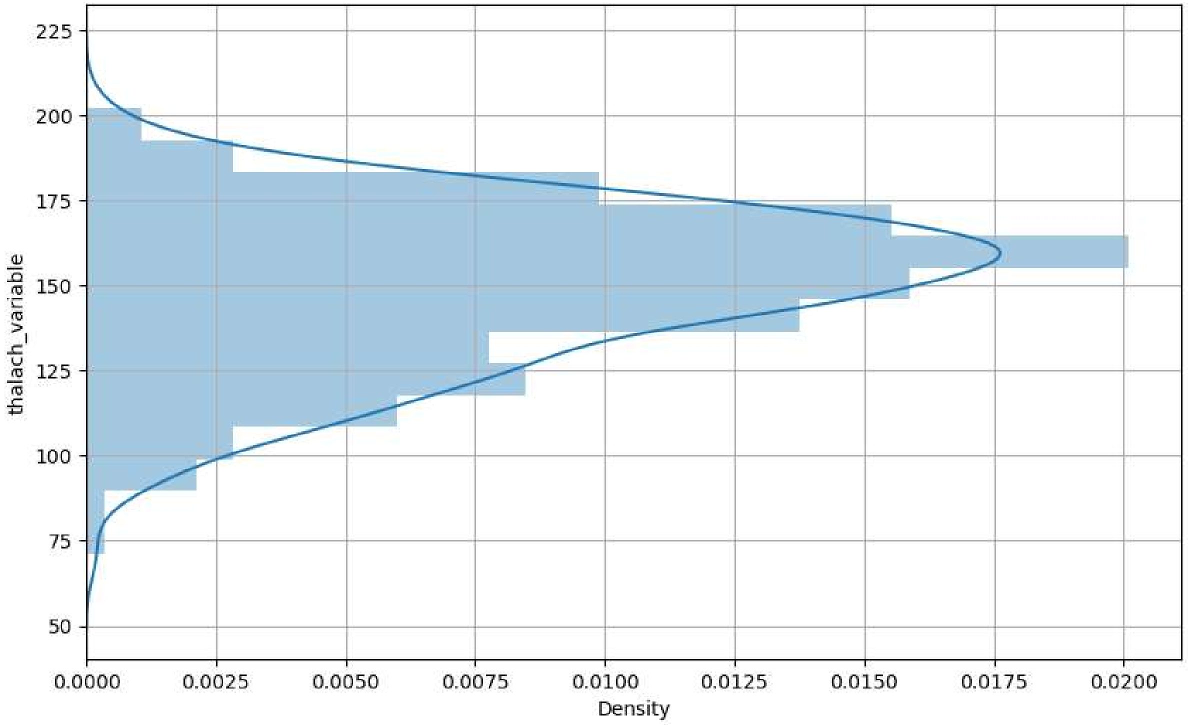
```
C:\Users\ASUS\AppData\Local\Temp\ipykernel_7172\626780498.py:4: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751 (https://gist.github.com/mwasko
m/de44147ed2974457ad6372750bbe5751)

  ax=sns.distplot(x)
```



In [58]:

```
f,ax=plt.subplots(figsize=(10,6))
x=hd['thalach']
x=pd.Series(x,name='thalach_variable')
ax=sns.distplot(x, vertical=True)
plt.grid()
```

```
C:\Users\ASUS\AppData\Local\Temp\ipykernel_7172\4142258308.py:4: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751 (https://gist.github.com/mwasko
m/de44147ed2974457ad6372750bbe5751)

  ax=sns.distplot(x, vertical=True)
```

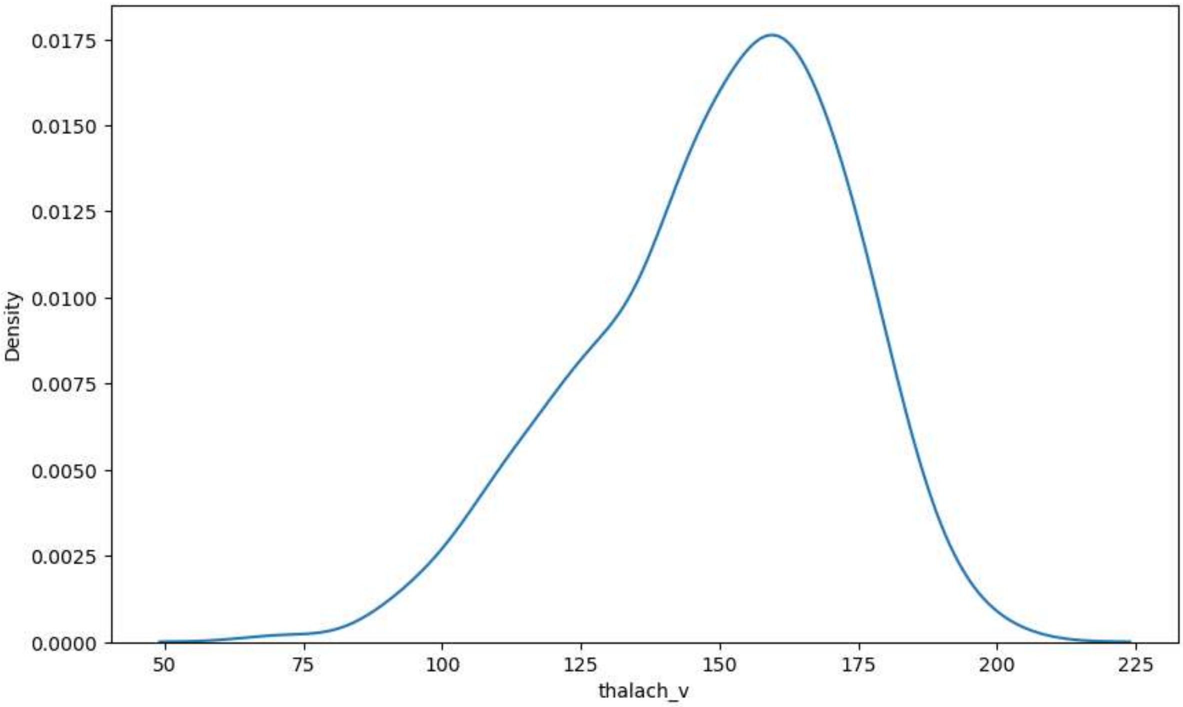


In [59]:

```
# kde(kernel density plot) useful for ploting shape of distribution
```

In [61]:

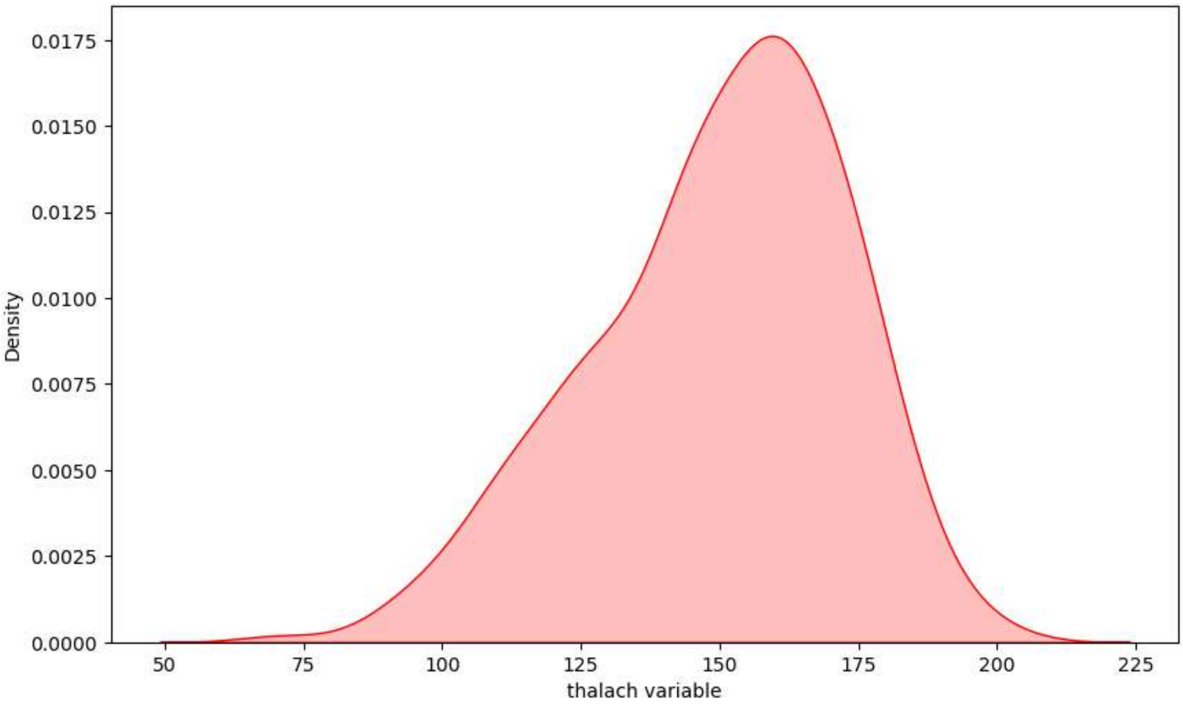
```
f,ax=plt.subplots(figsize=(10,6))
x=hd['thalach']
x=pd.Series(x, name='thalach_v')
ax=sns.kdeplot(x)
```



In [63]:

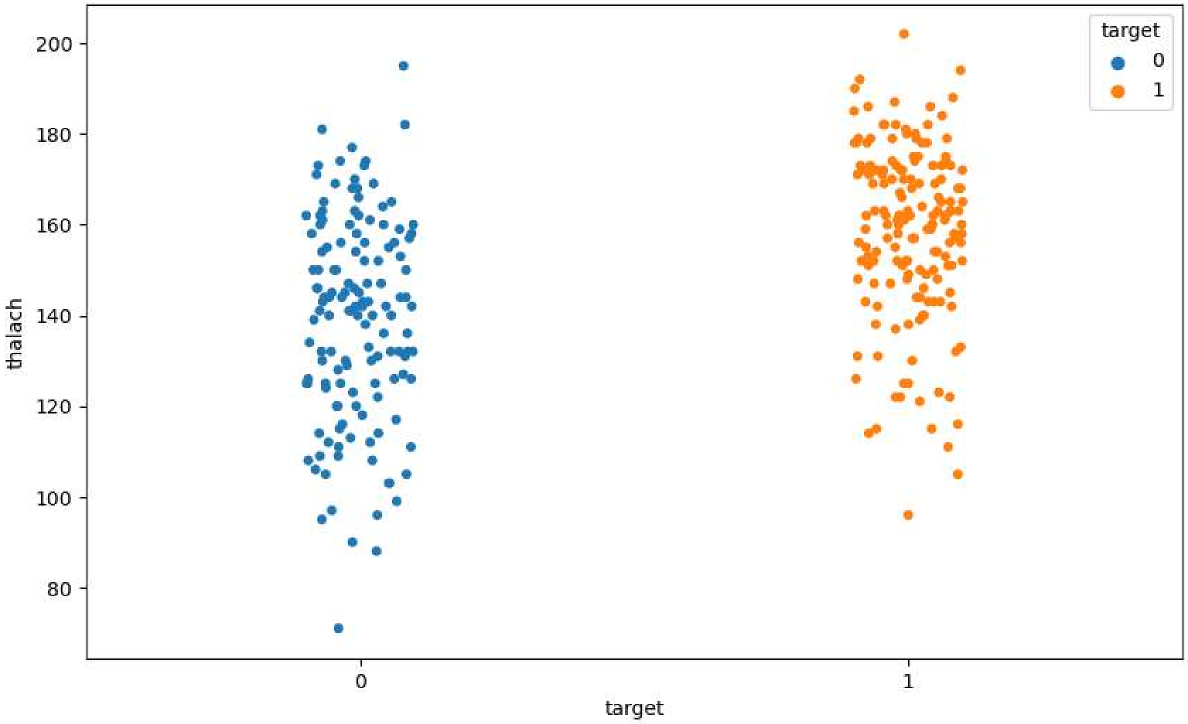
```
f, ax = plt.subplots(figsize=(10,6))
x = hd['thalach']
x = pd.Series(x, name="thalach variable")
ax = sns.kdeplot(x, shade=True, color='r')
plt.show()
```

```
C:\Users\ASUS\AppData\Local\Temp\ipykernel_7172\1191144267.py:4: FutureWarning:

`shade` is now deprecated in favor of `fill`; setting `fill=True`.
This will become an error in seaborn v0.14.0; please update your code.

  ax = sns.kdeplot(x, shade=True, color='r')
```



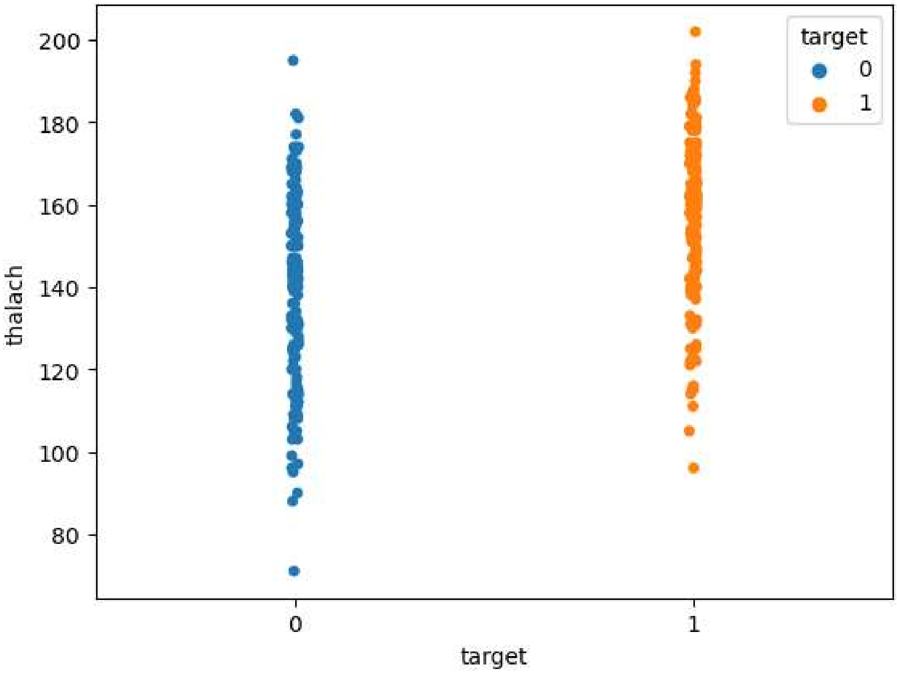
In [66]:

```
f,ax=plt.subplots(figsize=(10,6))
ax=sns.stripplot(data=hd, x='target', y='thalach' ,hue='target')
```



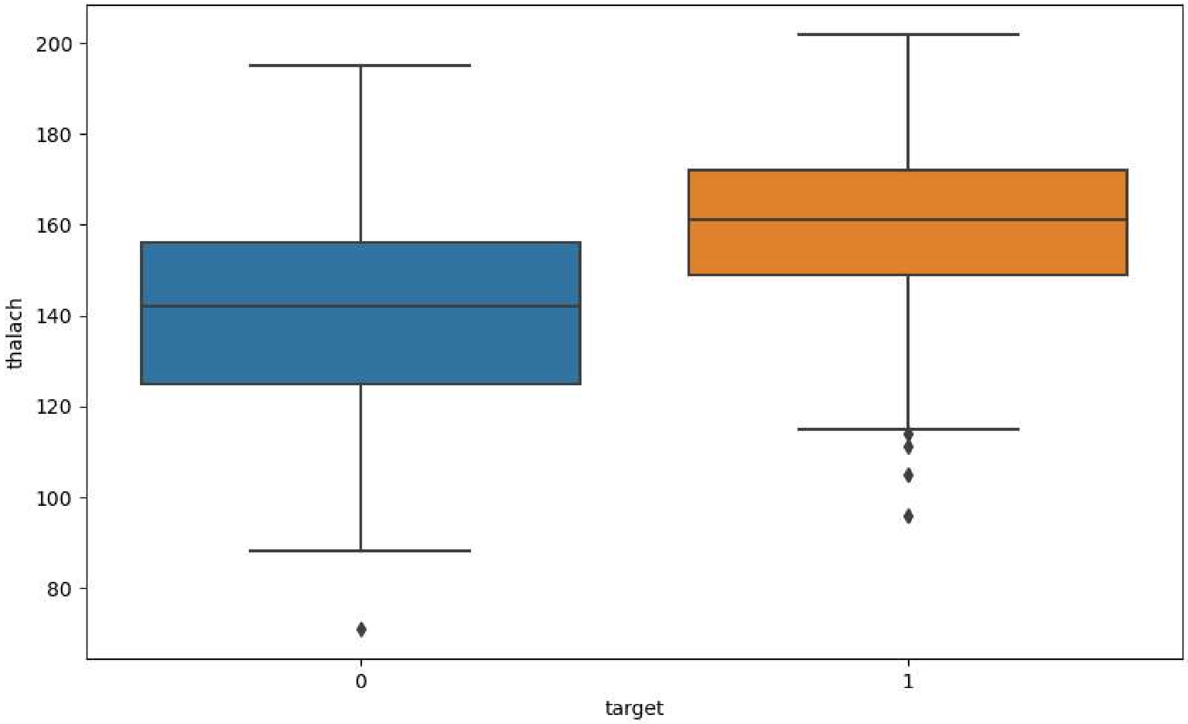
In [70]:

```
ax=sns.stripplot(data=hd, x='target', y='thalach' ,hue='target', jitter=0.01)
```



In [74]:

```
f, ax=plt.subplots(figsize=(10,6))
ax=sns.boxplot(data=hd, x='target', y='thalach')

```



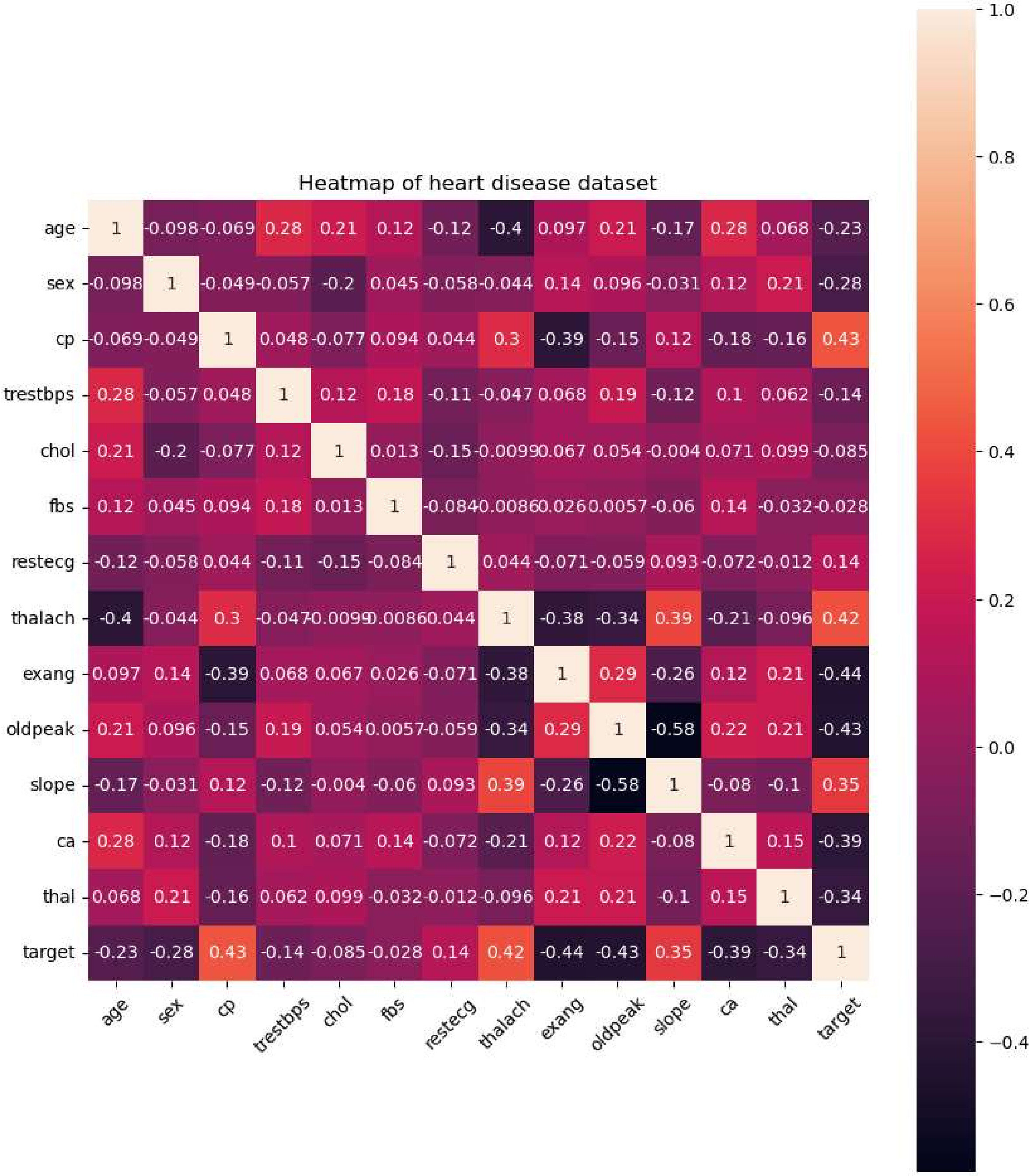
Findings of Bivariate Analysis

```
''' Findings of Bivariate Analysis are as follows -
There is no variable which has strong positive correlation with target variable.
There is no variable which has strong negative correlation with target variable.
There is no correlation between target and fbs.
The cp and thalach variables are mildly positively correlated with target variable.
We can see that the thalach variable is slightly negatively skewed.
The people suffering from heart disease (target = 1) have relatively higher heart rate (thalach)
as compared to people who are not suffering from heart disease (target = 0).
The people suffering from heart disease (target = 1) have relatively higher heart rate (thalach)
as compared to people who are not suffering from heart disease (target = 0) '''
```

univariate analysis

```
'''
An important step in EDA is to discover patterns and relationships between variables in the dataset.

I will use heat map and pair plot to discover the patterns and relationships in the dataset.

First of all, I will draw a heat map.

'''
```

In [93]:
```
correlation=hd.corr()
plt.figure(figsize=(10,12))
a=sns.heatmap(correlation, square=True,annot=True )
plt.title('Heatmap of heart disease dataset')
a.set_xticklabels(a.get_xticklabels(),rotation=45)
plt.show()
```



	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
age	1	-0.098	-0.069	0.28	0.21	0.12	-0.12	-0.4	0.097	0.21	-0.17	0.28	0.068	-0.23
sex	-0.098	1	-0.049	-0.057	-0.2	0.045	-0.058	-0.044	0.14	0.096	-0.031	0.12	0.21	-0.28
cp	-0.069	-0.049	1	0.048	-0.077	0.094	0.044	0.3	-0.39	-0.15	0.12	-0.18	-0.16	0.43
trestbps	0.28	-0.057	0.048	1	0.12	0.18	-0.11	-0.047	0.068	0.19	-0.12	0.1	0.062	-0.14
chol	0.21	-0.2	-0.077	0.12	1	0.013	-0.15	-0.0099	0.067	0.054	-0.004	0.071	0.099	-0.085
fbs	0.12	0.045	0.094	0.18	0.013	1	-0.084	0.0086	0.026	0.0057	-0.06	0.14	-0.032	-0.028
restecg	-0.12	-0.058	0.044	-0.11	-0.15	-0.084	1	0.044	-0.071	-0.059	0.093	-0.072	-0.012	0.14
thalach	-0.4	-0.044	0.3	-0.047	-0.0099	0.0086	0.044	1	-0.38	-0.34	0.39	-0.21	-0.096	0.42
exang	0.097	0.14	-0.39	0.068	0.067	0.026	-0.071	-0.38	1	0.29	-0.26	0.12	0.21	-0.44
oldpeak	0.21	0.096	-0.15	0.19	0.054	0.0057	-0.059	-0.34	0.29	1	-0.58	0.22	0.21	-0.43
slope	-0.17	-0.031	0.12	-0.12	-0.004	-0.06	0.093	0.39	-0.26	-0.58	1	-0.08	-0.1	0.35
ca	0.28	0.12	-0.18	0.1	0.071	0.14	-0.072	-0.21	0.12	0.22	-0.08	1	0.15	-0.39
thal	0.068	0.21	-0.16	0.062	0.099	-0.032	-0.012	-0.096	0.21	0.21	-0.1	0.15	1	-0.34
target	-0.23	-0.28	0.43	-0.14	-0.085	-0.028	0.14	0.42	-0.44	-0.43	0.35	-0.39	-0.34	1

```
'''
target and cp variable are mildly positively correlated (correlation coefficient = 0.43).
target and thalach variable are also mildly positively correlated (correlation coefficient = 0.42).
target and slope variable are weakly positively correlated (correlation coefficient = 0.35).
target and exang variable are mildly negatively correlated (correlation coefficient = -0.44).
target and oldpeak variable are also mildly negatively correlated (correlation coefficient = -0.43).
target and ca variable are weakly negatively correlated (correlation coefficient = -0.39).
target and thal variable are also waekly negatively correlated (correlation coefficient = -0.34).

```

```
'''
```

In [95]:

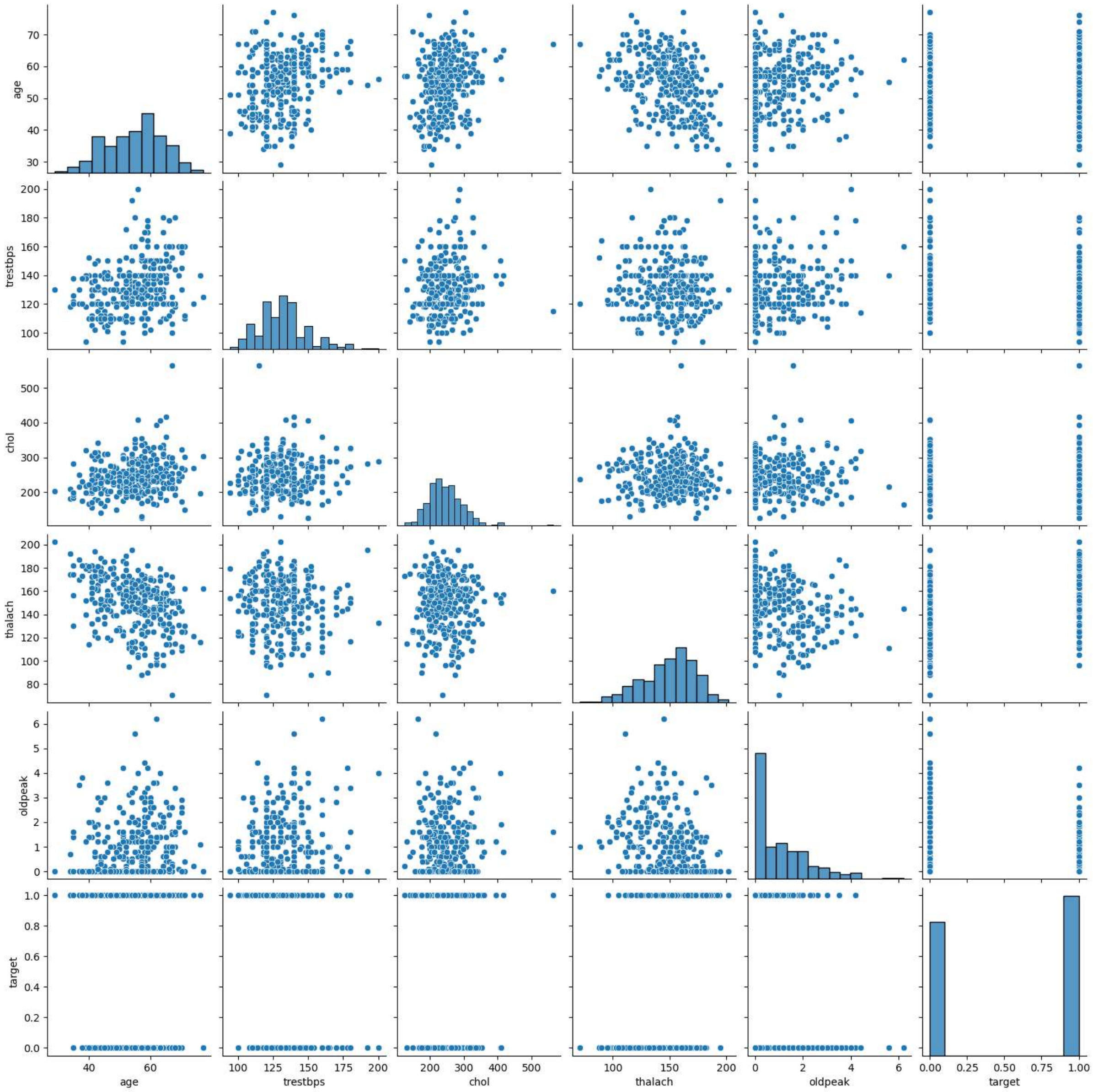
```
# pair plot
```

In [99]:

```
num_var=['age','trestbps','chol','thalach','oldpeak','target']
sns.pairplot(hd[num_var],kind='scatter', diag_kind='hist')
```

Out[99]:

```
<seaborn.axisgrid.PairGrid at 0x1c600dafa90>
```



In [100]:

```
# age with other variable
```

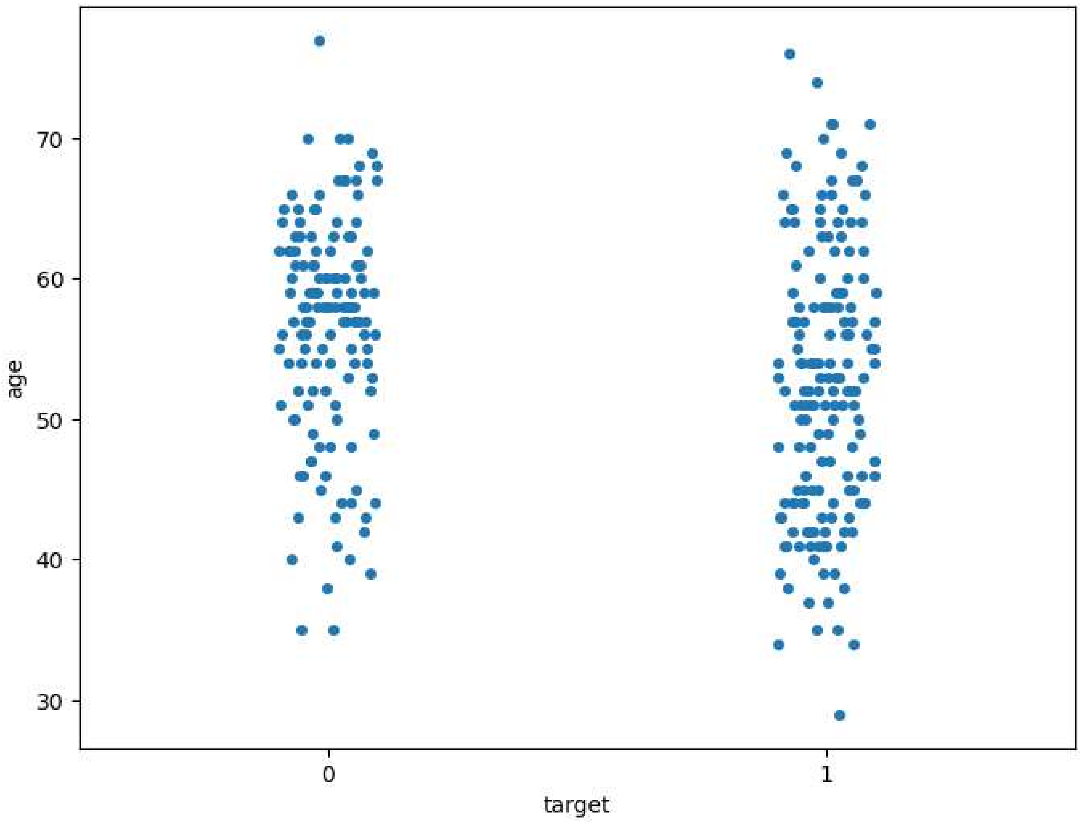
In [101]:

```
hd['age'].describe()
```

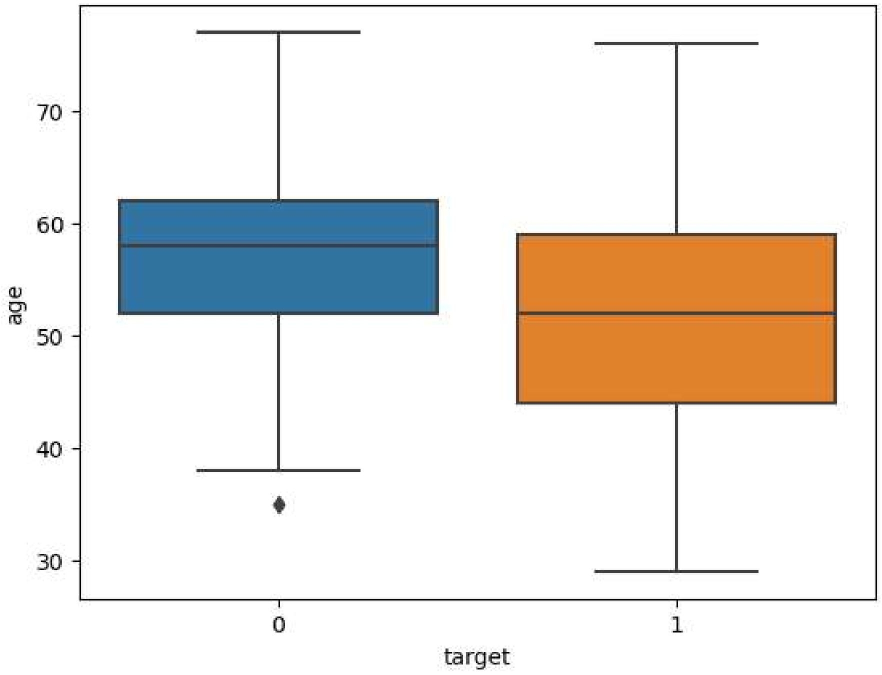
Out[101]:

```
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 [105]:
```
f,ax=plt.subplots(figsize=(8,6))
ax=sns.stripplot(data=hd,x='target',y='age')
```



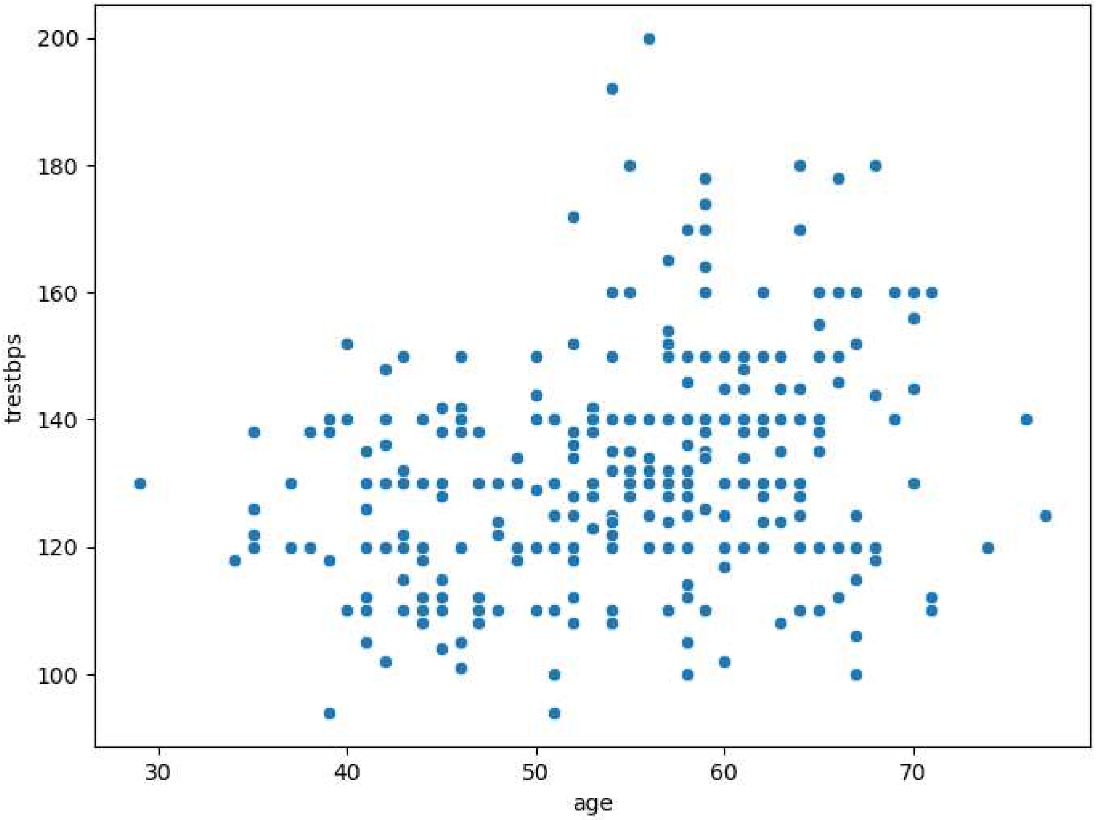
In [106]:
```
ax=sns.boxplot(data=hd, x='target', y='age')
```



```
'''
The average age of people who have heart disease is less than average age of people without any heart deases.

'''
```

In [110]:

```
f,ax=plt.subplots(figsize=(8,6))
ax=sns.scatterplot(data=hd,x='age', y='trestbps')
```

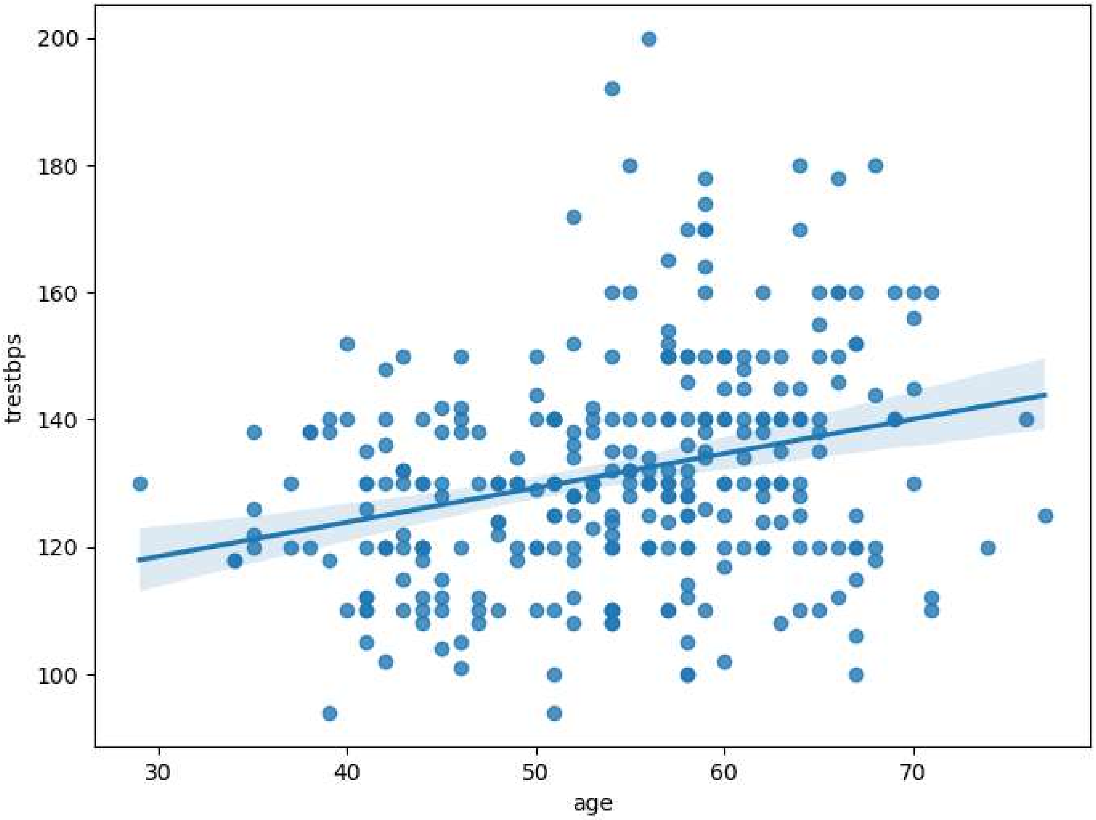


In [111]:

```
# above plot shows there is no co relation between trestbps and age
```

In [116]:

```
f,ax=plt.subplots(figsize=(8,6))
ax=sns.regplot(data=hd,x='age',y='trestbps' )
```

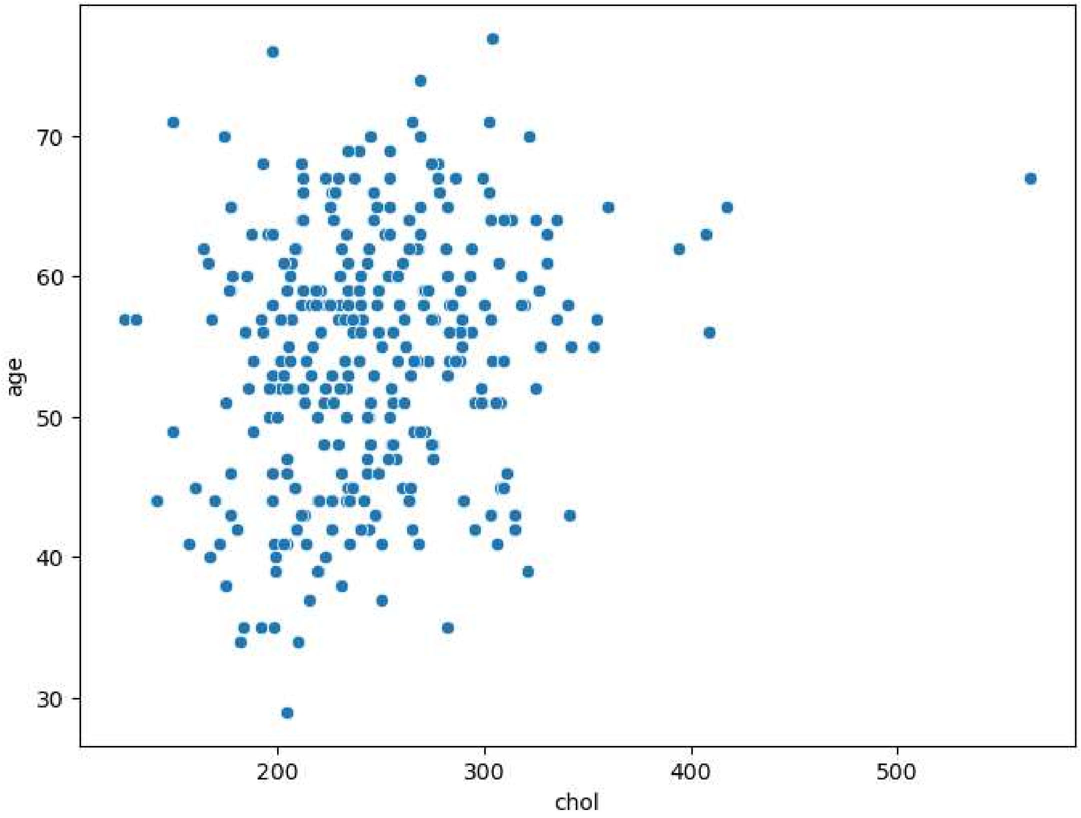


In [117]:

```
 # age vs cholestrol
```

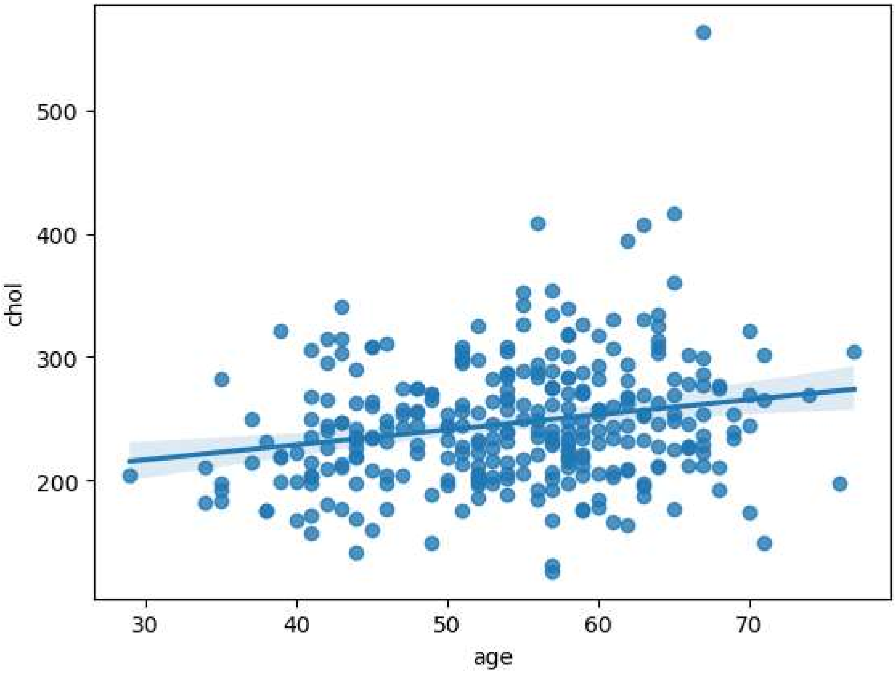
In [120]:

```
f, ax=plt.subplots(figsize=(8,6))
ax=sns.scatterplot(data=hd,x='chol',y='age')
```



In [121]:

```
ax=sns.regplot(data=hd,x='age',y='chol')
```



In [122]:

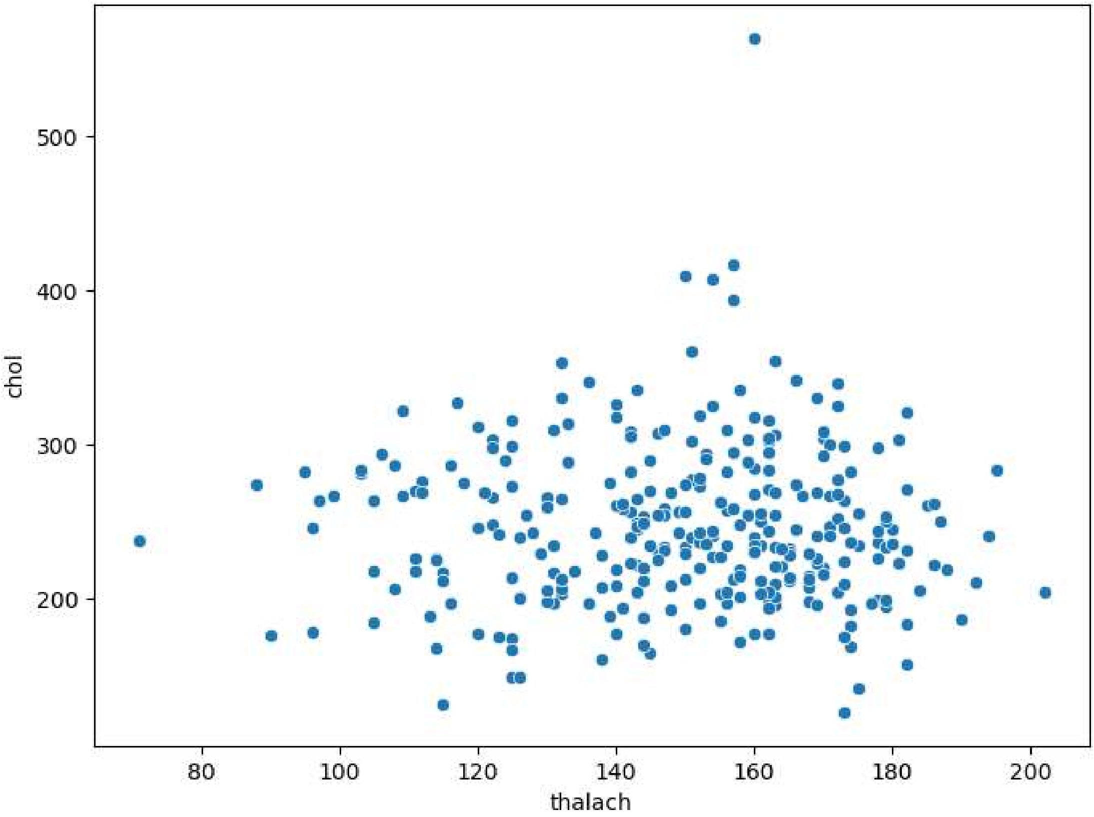
```
# above plot show slighly +ve relation between age and cholestrol
```

In [123]:

```
# thalach vs chol
```

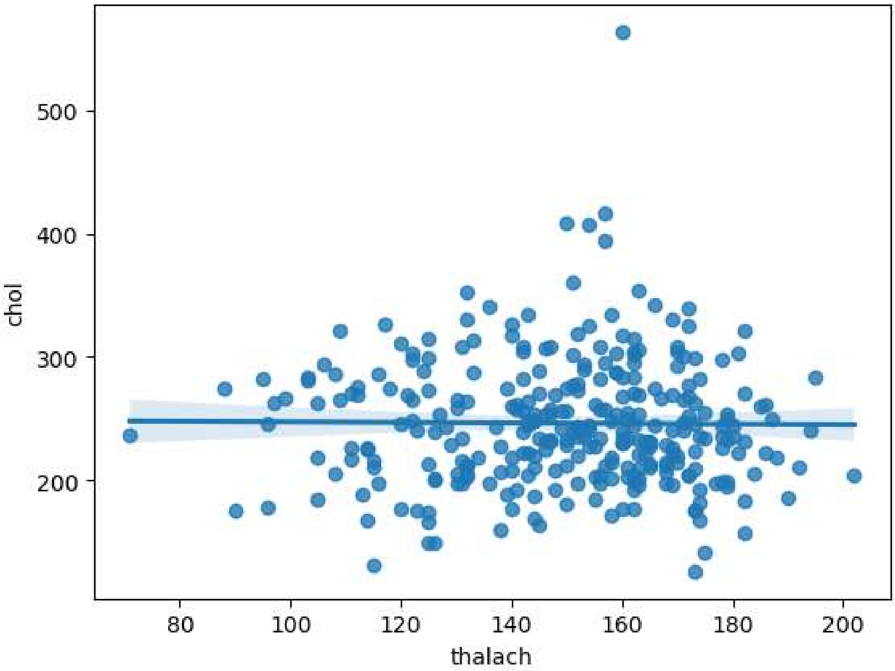
In [127]:

```
f,ax=plt.subplots(figsize=(8,6))
ax=sns.scatterplot(data=hd,x='thalach',y='chol')
```



In [129]:

```
ax=sns.regplot(data=hd, x='thalach',y='chol')
```



In [130]:

```
# no corelation between thalach and chol
```

```
# outlier detection
```

In [132]:

```
# age variable
```

In [134]:

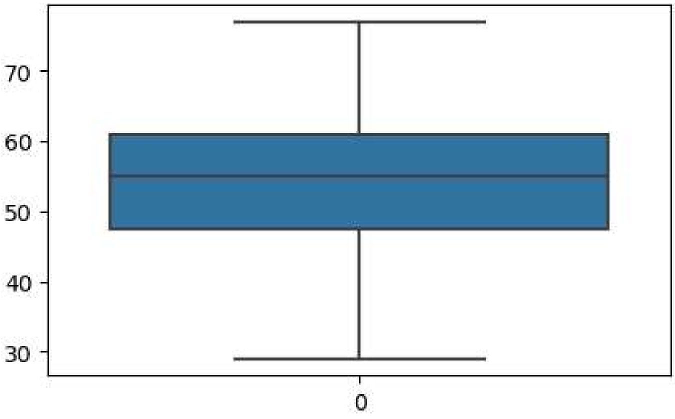
```
hd['age'].describe()
```

Out[134]:

```
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 [136]:

```
f, ax=plt.subplots(figsize=(5,3))
ax=sns.boxplot(hd['age'])
```



In [137]:

```
# tresbps varaible
```

In [138]:

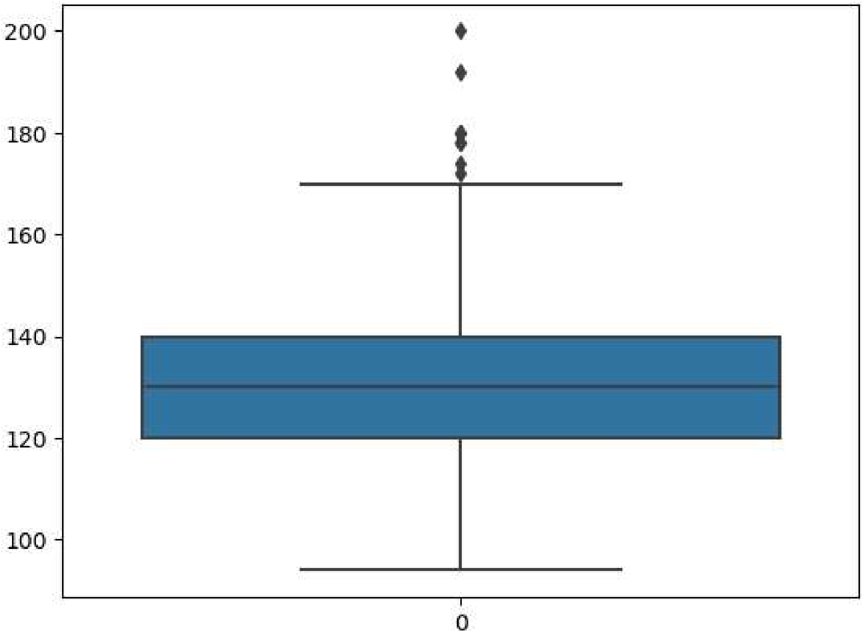
```
hd['trestbps'].describe()
```

Out[138]:

```
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 [140]:

```
ax=sns.boxplot(hd['trestbps'])
```



In [141]:

```
# thalach variable
```

In [142]:

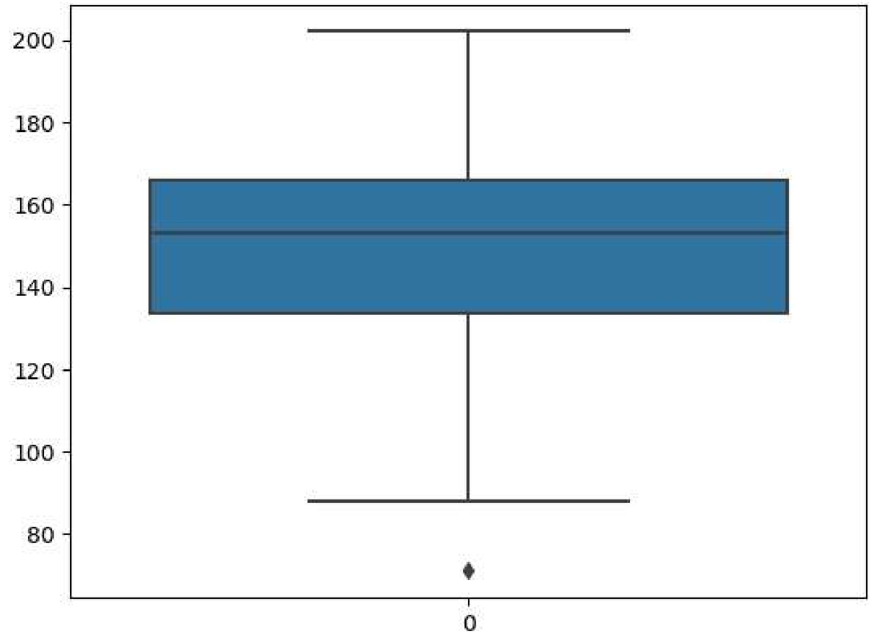
```
hd['thalach'].describe()
```

Out[142]:

```
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 [143]:

```
ax=sns.boxplot(hd['thalach'])
```



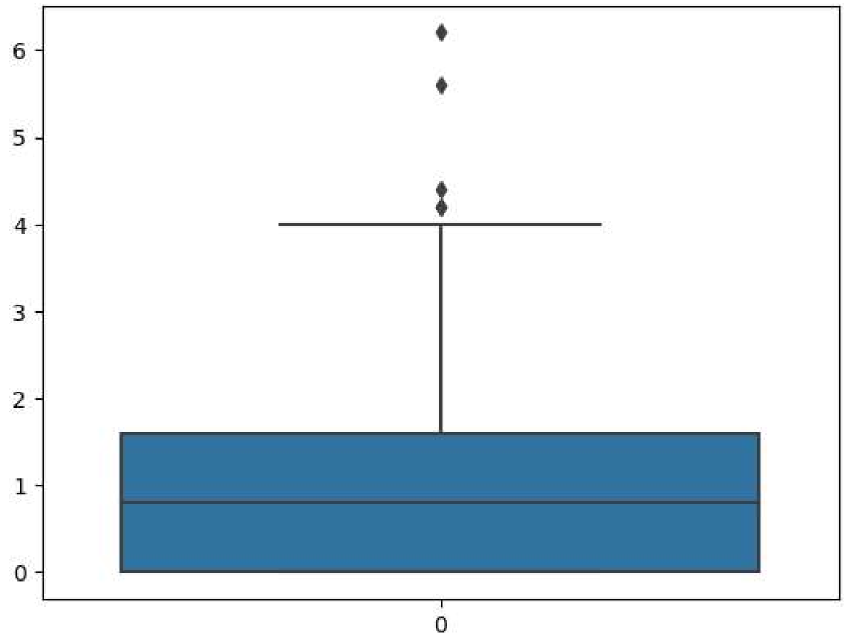
In [144]:

```
hd['oldpeak'].describe()
```

Out[144]:

```
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 [145]:
```
ax=sns.boxplot(hd['oldpeak'])
```



In [146]:
```
'''the variable age doesn't have any outlier
   thalach have outliers
   oldpeak have outlier
   chol have outlier
   Those variable have outlier needs further investigation
   '''
```

Out[146]:
```
"the variable age doesn't have any outlier\n   thalach have outliers\n   oldpeak have outlier\n
chol have outlier\n   Those variable have outlier needs further investigation\n   "
```

```
# conclusion
  I have performed EDA on heart analysis dataset wrt target variable.
1.explored the dataset -including domain knowledge, understanding each variable,
  knowing depedent(target) and independent variables.
2.finding missing value--not found.
3.performed univariate analysis on target vairable (graph).
4.bi variate analysis between many variables (graph).
5.multivariate analysis is also performed between multiple variable .
6.finding outliers in varaibles.
```

In [ ]:
```

```