In [4]:

```
import pandas as pd

df = pd.read_csv('heart_85a1a8fc00992a3d2b498875744c6df3.csv')
df.rename(columns = {'ï»¿age': 'age'}, inplace= True)
```

In [5]:

```
df.head()
```

Out[5]:

	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 [6]:
```
1 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 [5]:
```
1 df.columns
```

Out[5]:
```
Index(['age', 'sex', 'cp', 'trestbps', 'chol', 'fbs', 'restecg', 'thalach',
       'exang', 'oldpeak', 'slope', 'ca', 'thal', 'target'],
      dtype='object')
```

In [6]:
```
1 df.dtypes
```

Out[6]:
```
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 [8]:
```
1 df.shape
```

Out[8]:
```
(303, 14)
```

In [9]:
```
1 duplicate = df.duplicated(keep= False).sum()
2 duplicate
```

Out[9]:
```
2
```

In [13]:
```
missing_value = df.isna().sum()
missing_value
```

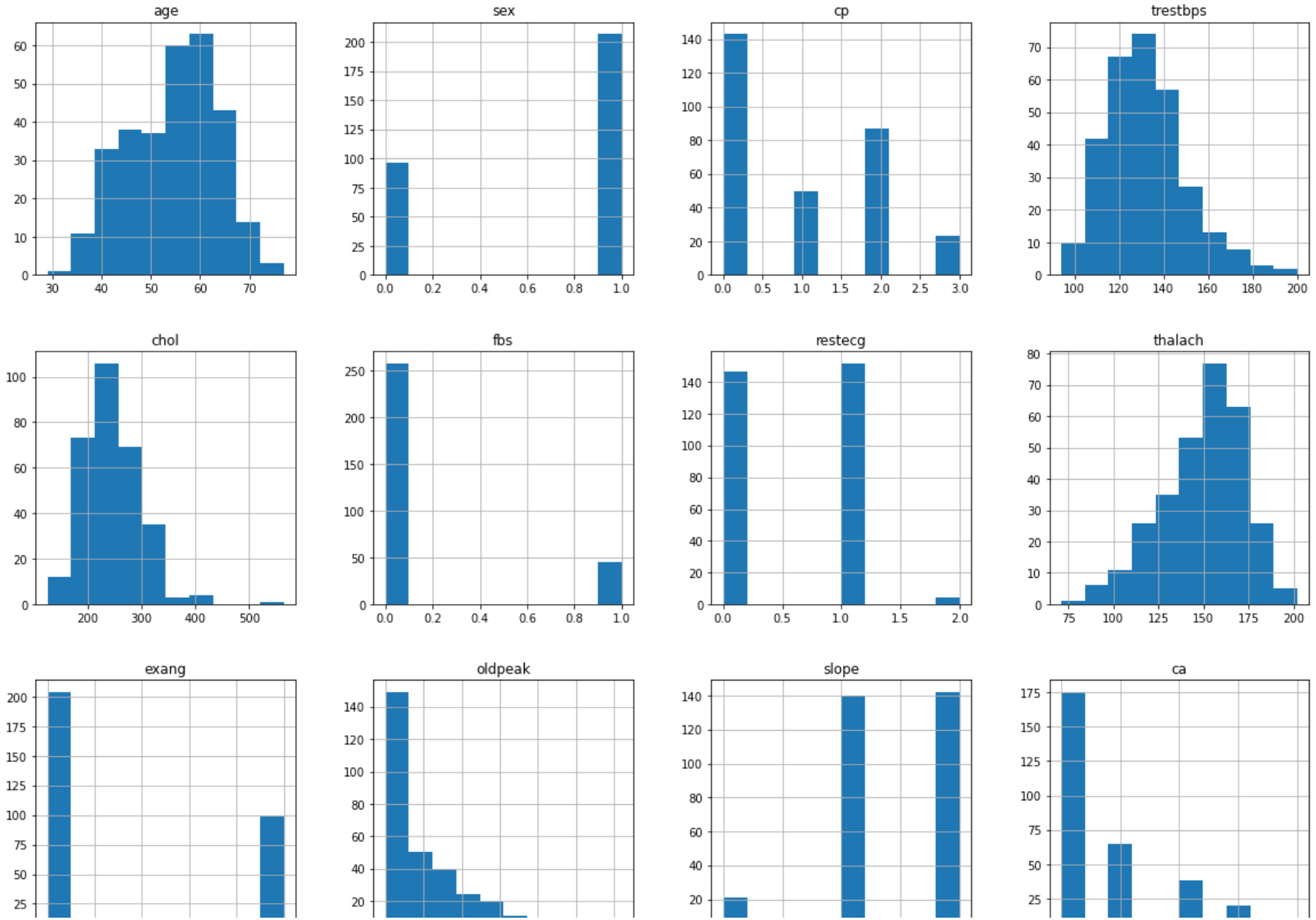
Out[13]:
```
age         0
sex         0
cp          0
trestbps    0
chol        0
fbs         0
restecg     0
thalach     0
exang       0
oldpeak     0
slope       0
ca          0
thal        0
target      0
dtype: int64
```

In [14]:
```
# there is no missing value in the data set
```

In [ ]:
```

```

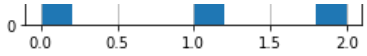
In [11]:

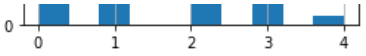
```
# univariate

import matplotlib.pyplot as plt

df.hist()
plt.gcf().set_size_inches(20,20)
plt.show()
```

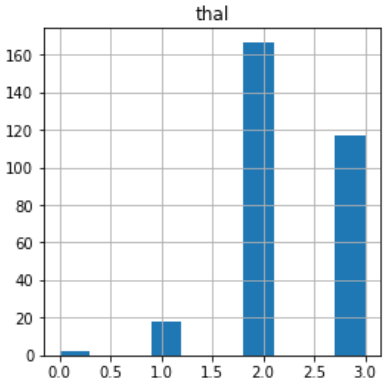


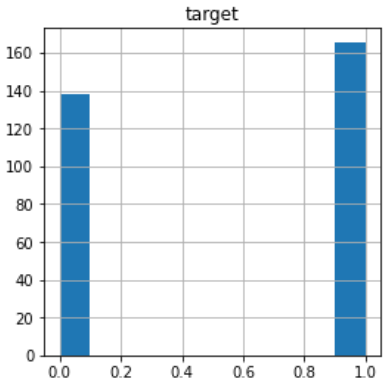

0
0.0
0.2
0.4
0.6
0.8
1.0


0
0
1
2
3
4
5
6


0
0.0
0.5
1.0
1.5
2.0

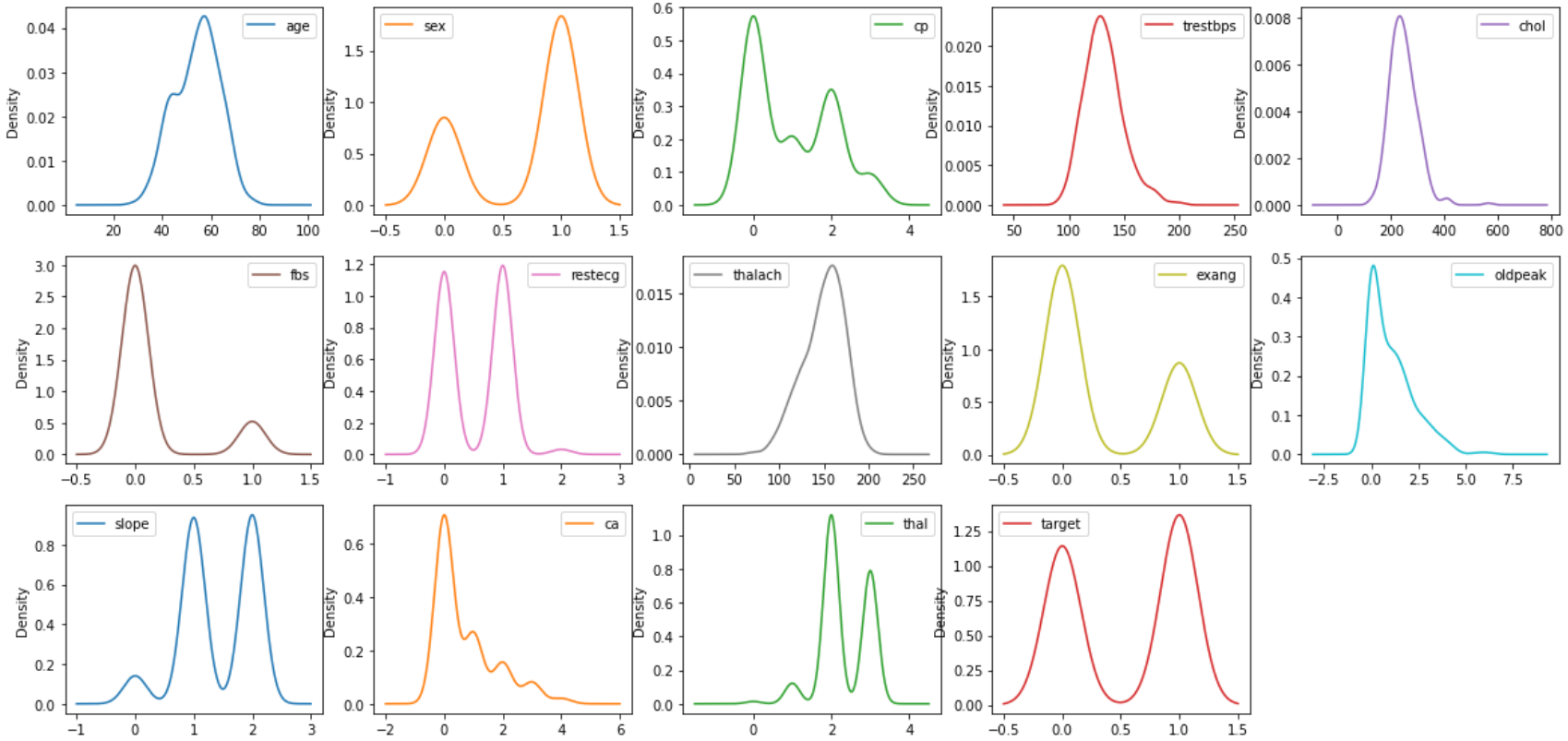

0
0
1
2
3
4


thal
160
140
120
100
80
60
40
20
0
0.0
0.5
1.0
1.5
2.0
2.5
3.0


target
160
140
120
100
80
60
40
20
0
0.0
0.2
0.4
0.6
0.8
1.0

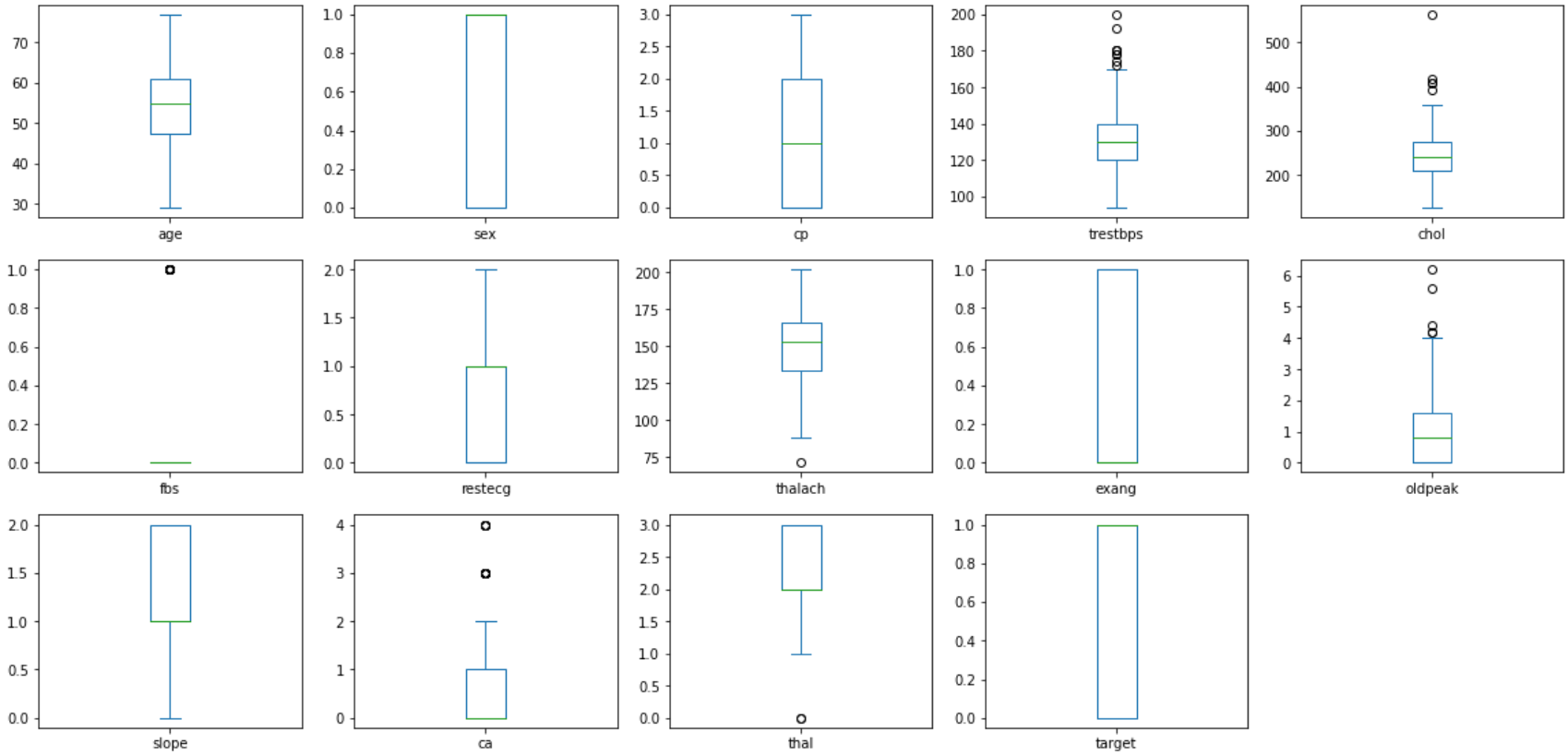
In [7]:

```
import matplotlib.pyplot as plt
from matplotlib import pyplot

df.plot(kind= 'density', subplots= True, layout= (6, 5), sharex= False)
plt.gcf().set_size_inches(20, 20)
pyplot.show()
```



In [12]:

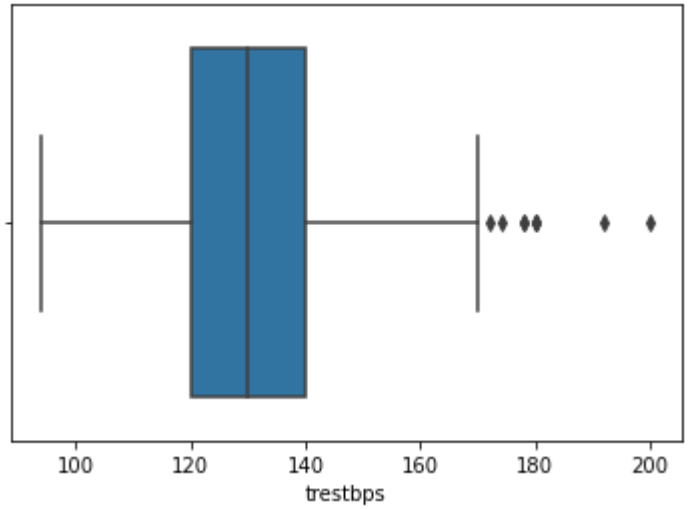
```
# univariate
from matplotlib import pyplot

df.plot(kind = 'box', subplots = True, layout=(6, 5), sharex = False)
plt.gcf().set_size_inches(20, 20)
pyplot.show()
```



In [23]:
```
# check for outliers
import seaborn as sns
import numpy as np

sns.boxplot(df['trestbps'])
np.where(df['trestbps'] > 170)
```

```
C:\Users\Shehu\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variab
le as a keyword arg: x. From version 0.12, the only valid positional argument will be `data`, and passing othe
r arguments without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```

Out[23]:
```
(array([  8, 101, 110, 203, 223, 241, 248, 260, 266], dtype=int64),)
```

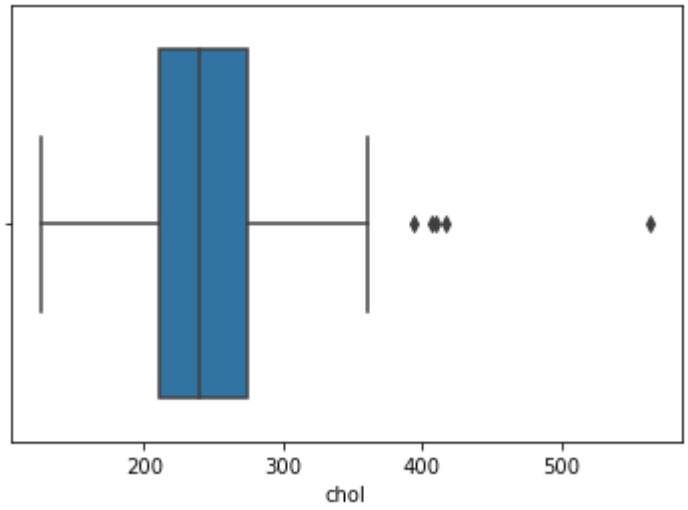


In [ ]:
```
values above 2.5 are acting as the outliers.
position of the outlier
```

In [22]:
```
1 sns.boxplot(df['chol'])
2 np.where(df['chol'] > 360)
```

```
C:\Users\Shehu\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variab
le as a keyword arg: x. From version 0.12, the only valid positional argument will be `data`, and passing othe
r arguments without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```

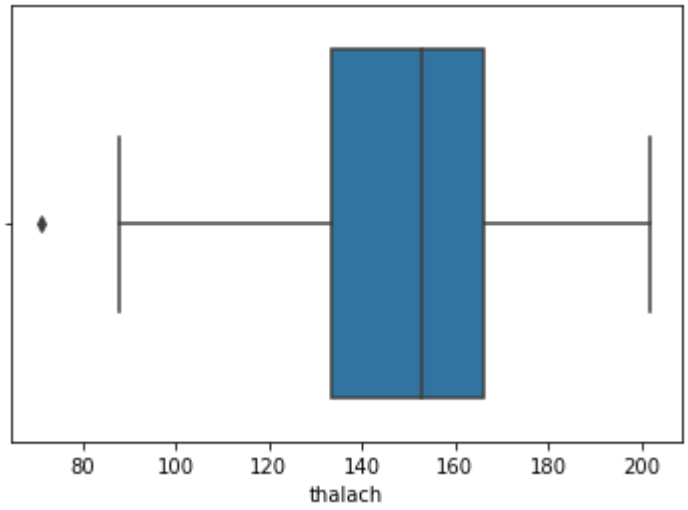
Out[22]:
```
(array([ 28,  85,  96, 220, 246], dtype=int64),)
```



In [26]:
```
1 sns.boxplot(df['thalach'])
2 np.where(df['thalach'] < 80)
```

```
C:\Users\Shehu\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variab
le as a keyword arg: x. From version 0.12, the only valid positional argument will be `data`, and passing othe
r arguments without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```

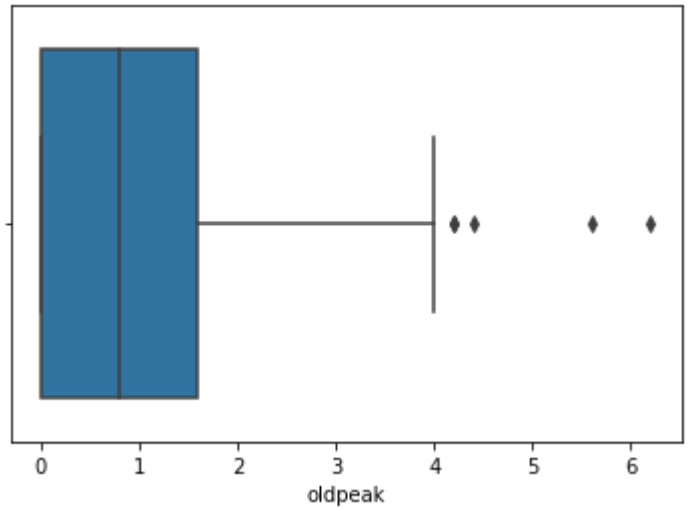
Out[26]:
```
(array([272], dtype=int64),)
```



In [28]:
```
1 sns.boxplot(df['oldpeak'])
2 np.where(df['oldpeak'] > 4)
```

```
C:\Users\Shehu\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variab
le as a keyword arg: x. From version 0.12, the only valid positional argument will be `data`, and passing othe
r arguments without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```

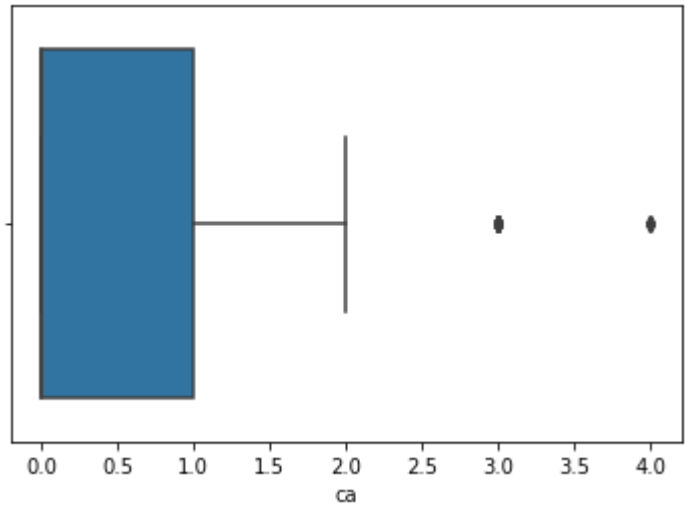
Out[28]:
```
(array([101, 204, 221, 250, 291], dtype=int64),)
```



In [32]:
```
1 sns.boxplot(df['ca'])
2 np.where(df['ca'] > 2.5)
```

```
C:\Users\Shehu\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variab
le as a keyword arg: x. From version 0.12, the only valid positional argument will be `data`, and passing othe
r arguments without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```

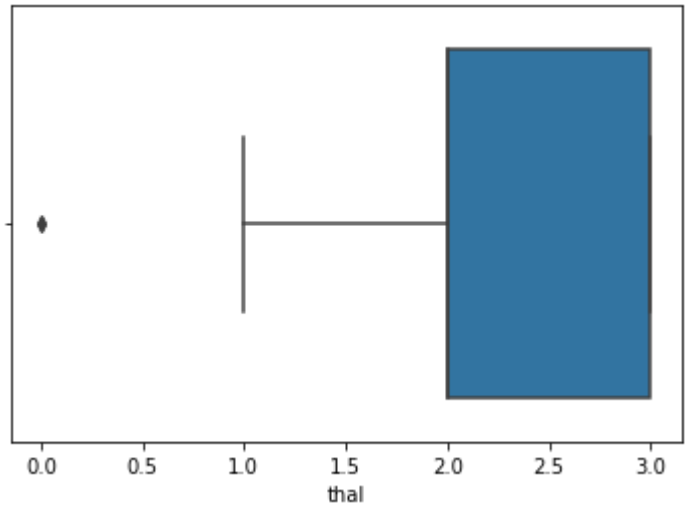
Out[32]:
```
(array([ 52,  92,  97,  99, 158, 163, 164, 165, 181, 191, 204, 208, 217,
        220, 231, 234, 238, 247, 249, 250, 251, 252, 255, 267, 291],
       dtype=int64),)
```



In [34]:
```
1 sns.boxplot(df['thal'])
2 np.where(df['thal'] < 1)
```

```
C:\Users\Shehu\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variab
le as a keyword arg: x. From version 0.12, the only valid positional argument will be `data`, and passing othe
r arguments without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```

Out[34]:
```
(array([ 48, 281], dtype=int64),)
```



In [3]:

```
# identifying outliers using mae

from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_absolute_error
from sklearn.neighbors import LocalOutlierFactor

data = df.values
x, y = data[:,0:13], data[:, 13]

print(x.shape, y.shape)

x_train, x_test, y_train, y_test = train_test_split(x, y, test_size= 0.33, random_state= 1)
print(x_train.shape, y_train.shape)

lof = LocalOutlierFactor()
preds = lof.fit_predict(x_train)

mask = preds != -1
x_train, y_train = x_train[mask, :], y_train[mask]
print(x_train.shape, y_train.shape)

model = LinearRegression()
model.fit(x_train, y_train)
pred = model.predict(x_test)
mae = mean_absolute_error(y_test, pred)

print(x_train.shape,  x_test.shape, y_train.shape, y_test.shape)
print(mae)
```

```
(303, 13) (303,)
(203, 13) (203,)
(198, 13) (198,)
(198, 13) (100, 13) (198,) (100,)
0.29524851067392677
```

In [ ]:

```

```

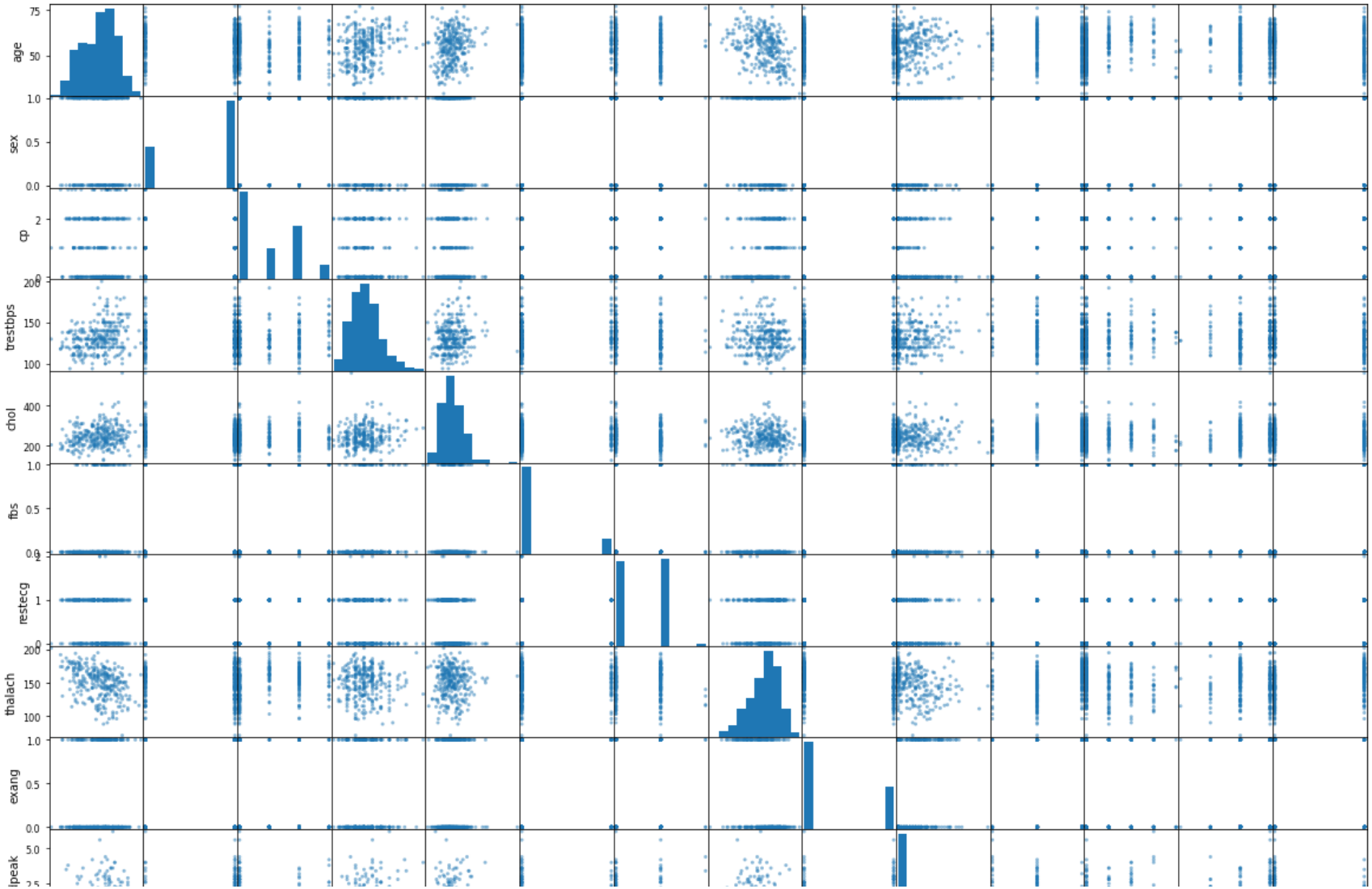
In [ ]:

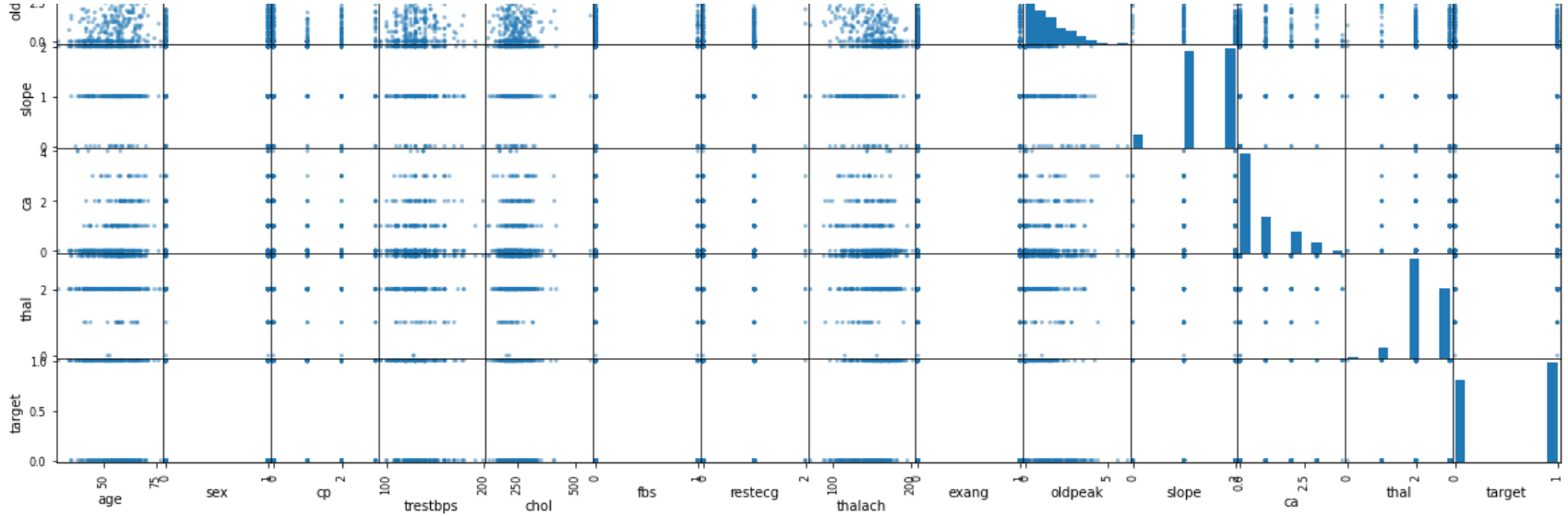
```
# sns.pairplot(df,kind= 'scatter')
# plt.show()
```

In [39]:

```
# relationship between two variables

from pandas.plotting import scatter_matrix

scatter_matrix(df)
plt.gcf().set_size_inches(20,20)
plt.show
```

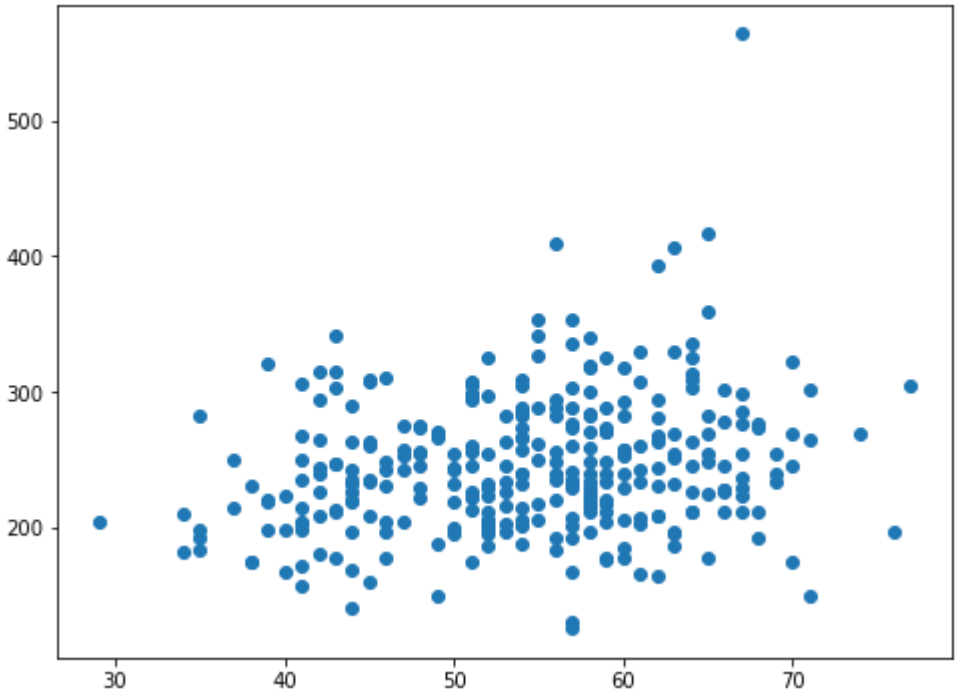
Out[39]: <function matplotlib.pyplot.show(close=None, block=None)>




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

In [41]:
```
1 fig, ax = plt.subplots(figsize=(8,6))
2 ax.scatter(df['age'], df['chol'])
```

Out[41]:
```
<matplotlib.collections.PathCollection at 0x2b382fdaa00>
```

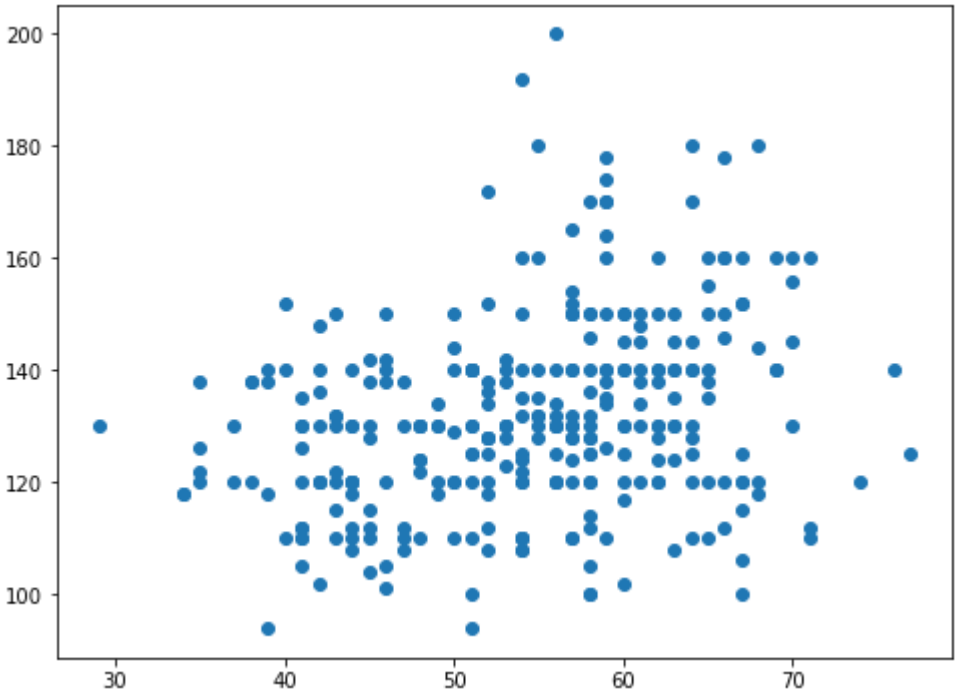


In [43]:
```
1 df.columns
```

Out[43]:
```
Index(['age', 'sex', 'cp', 'trestbps', 'chol', 'fbs', 'restecg', 'thalach',
       'exang', 'oldpeak', 'slope', 'ca', 'thal', 'target'],
      dtype='object')
```

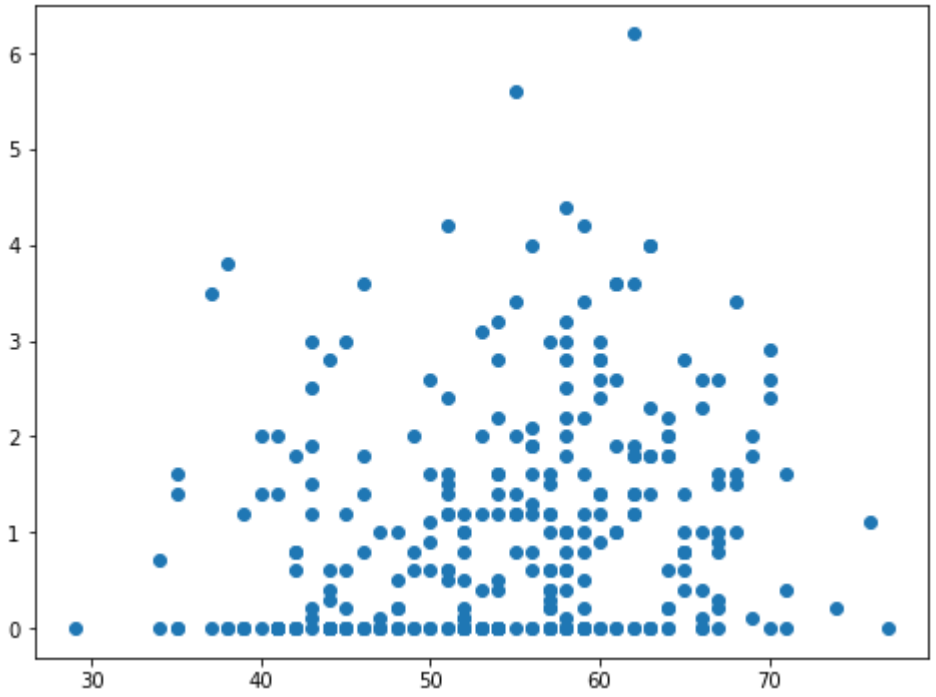
In [46]:
```
1 fig, ax = plt.subplots(figsize=(8,6))
2 ax.scatter(df['age'], df['trestbps'])
```

Out[46]: <matplotlib.collections.PathCollection at 0x2b38277dbe0>



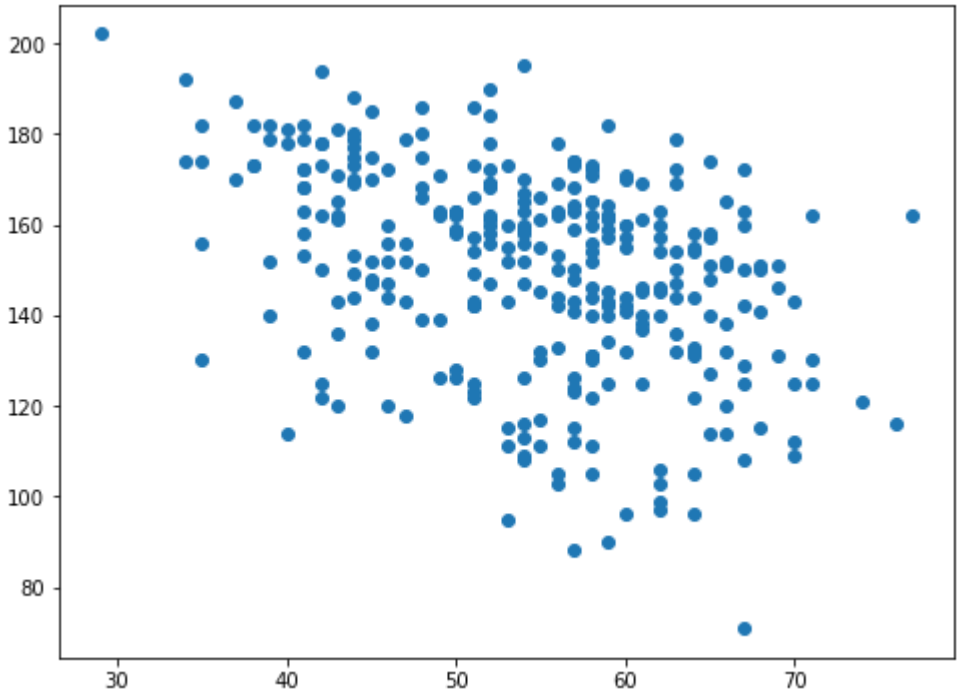
In [52]:
```
1 fig, ax = plt.subplots(figsize=(8,6))
2 ax.scatter(df['age'], df['oldpeak'])
```

Out[52]: <matplotlib.collections.PathCollection at 0x2b3fcacaa60>



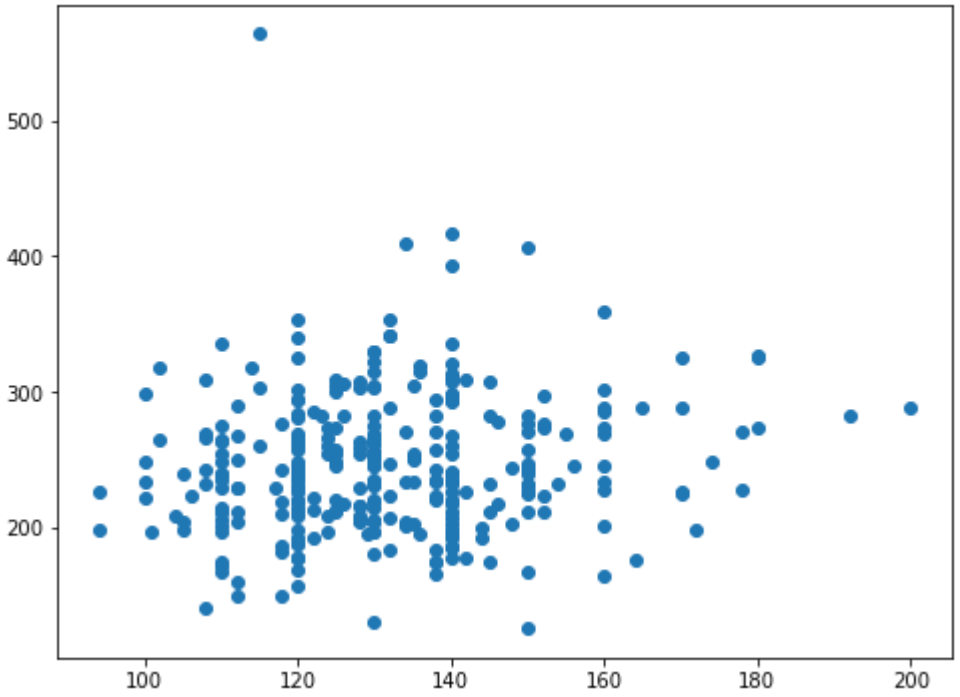
In [50]:
```
1 fig, ax = plt.subplots(figsize=(8,6))
2 ax.scatter(df['age'], df['thalach'])
```

Out[50]: <matplotlib.collections.PathCollection at 0x2b3fca35fd0>



In [42]:
```
1 fig, ax = plt.subplots(figsize=(8,6))
2 ax.scatter(df['trestbps'], df['chol'])
```

Out[42]:
```
<matplotlib.collections.PathCollection at 0x2b383016880>
```

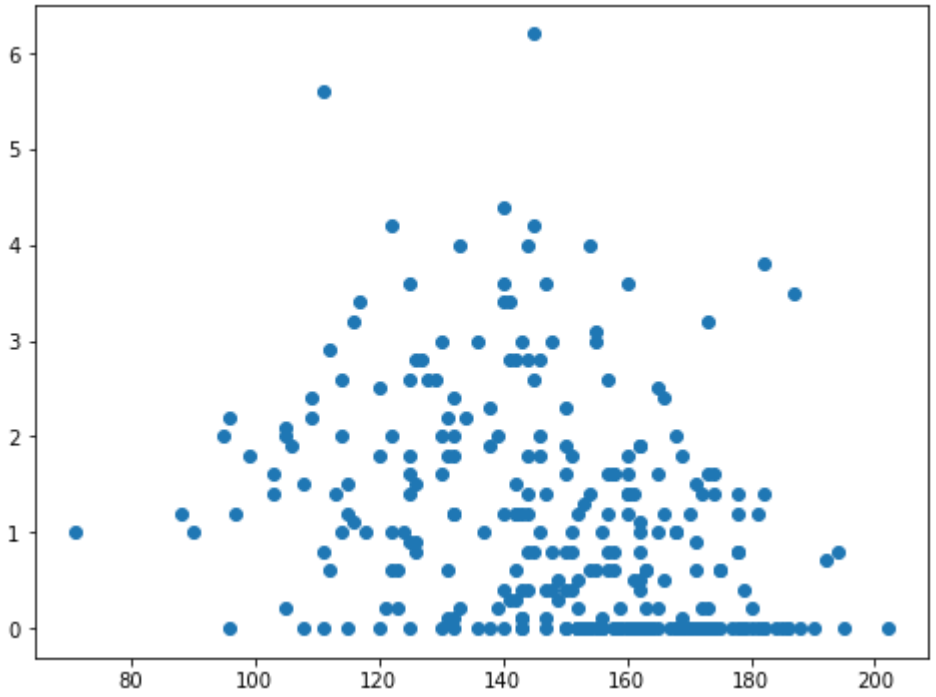


In [58]:
```
1 df.columns
```

Out[58]:
```
Index(['age', 'sex', 'cp', 'trestbps', 'chol', 'fbs', 'restecg', 'thalach',
       'exang', 'oldpeak', 'slope', 'ca', 'thal', 'target'],
      dtype='object')
```

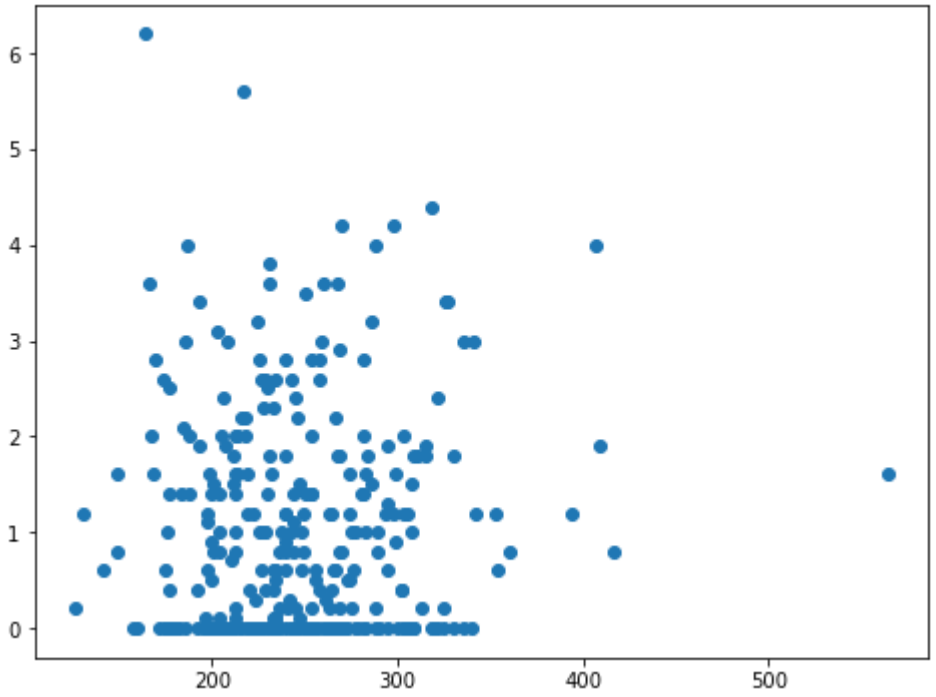
In [117]:
```
1 fig, ax = plt.subplots(figsize=(8,6))
2 ax.scatter(df['thalach'], df['oldpeak'])
```

Out[117]: <matplotlib.collections.PathCollection at 0x2b3fb659400>



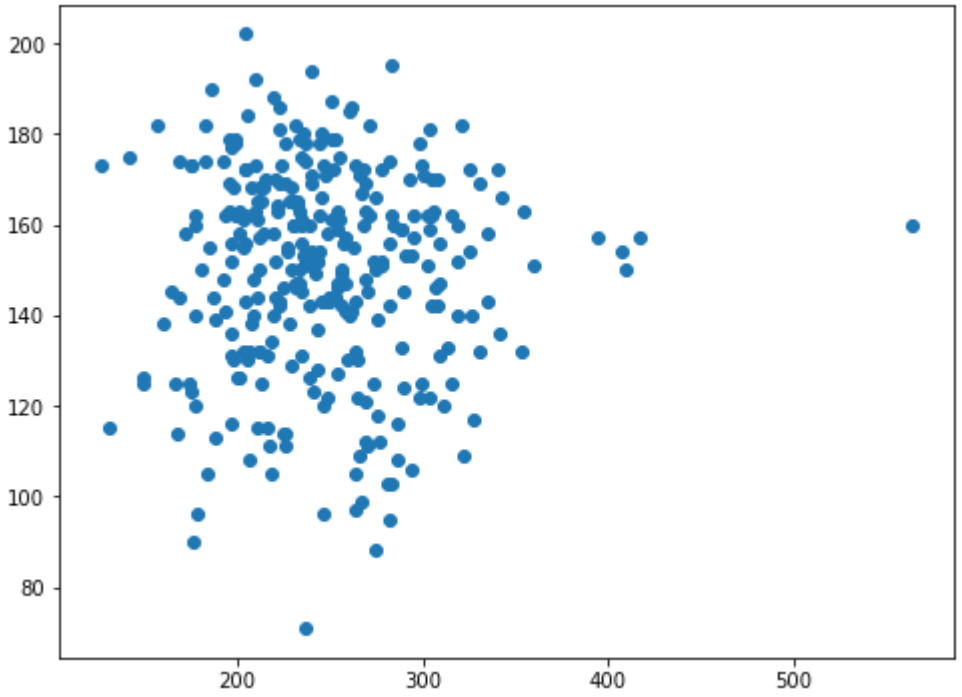
In [96]:
```
1 fig, ax = plt.subplots(figsize=(8,6))
2 ax.scatter(df['chol'], df['oldpeak'])
```

Out[96]: <matplotlib.collections.PathCollection at 0x2b3fec96070>



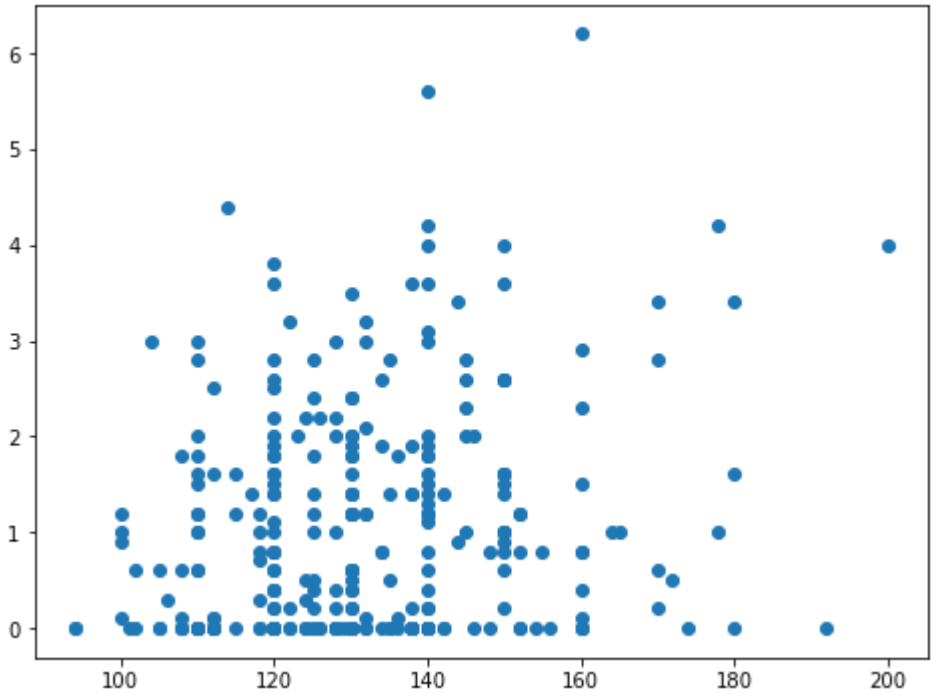
In [94]:
```
1 fig, ax = plt.subplots(figsize=(8,6))
2 ax.scatter(df['chol'], df['thalach'])
```

Out[94]: <matplotlib.collections.PathCollection at 0x2b3ff95a7c0>



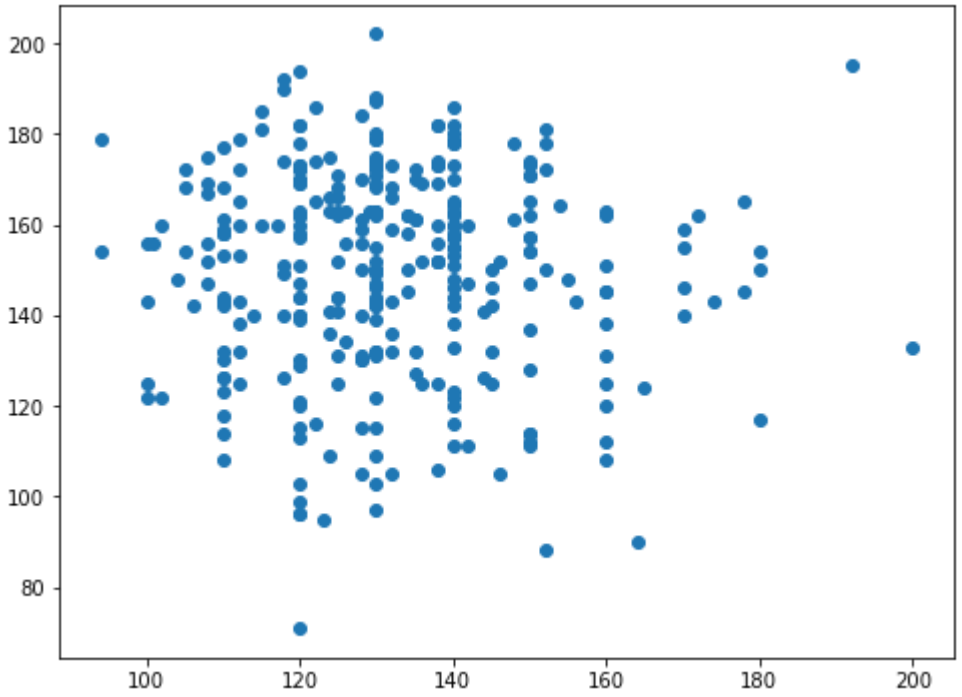
In [87]:
```
1 fig, ax = plt.subplots(figsize=(8,6))
2 ax.scatter(df['trestbps'], df['oldpeak'])
```

Out[87]: <matplotlib.collections.PathCollection at 0x2b3fb434b20>



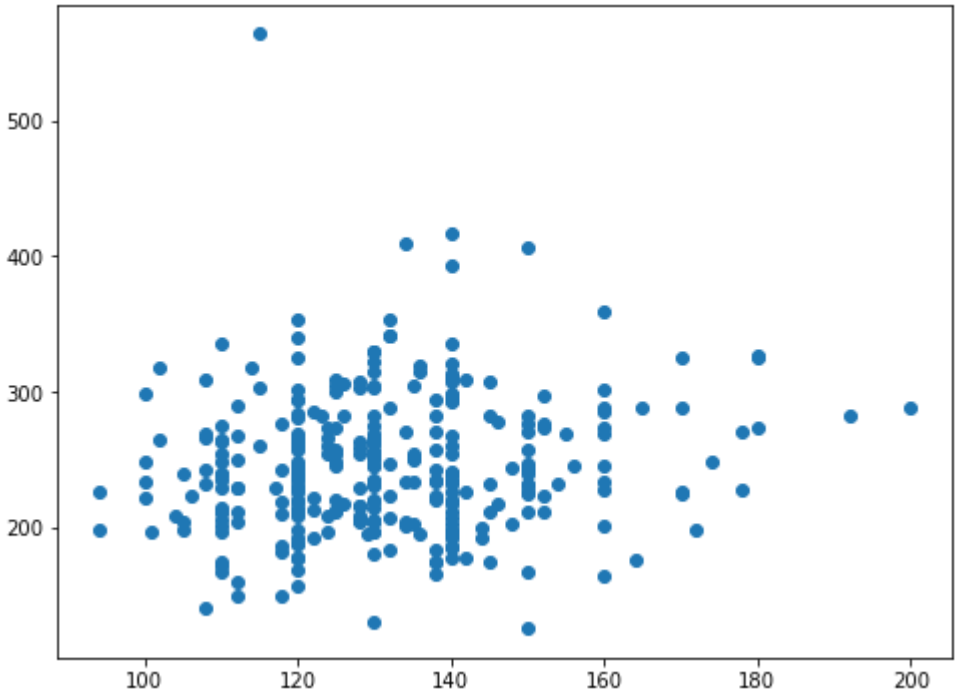
In [85]:
```
1 fig, ax = plt.subplots(figsize=(8,6))
2 ax.scatter(df['trestbps'], df['thalach'])
```

Out[85]: <matplotlib.collections.PathCollection at 0x2b3fb278d90>



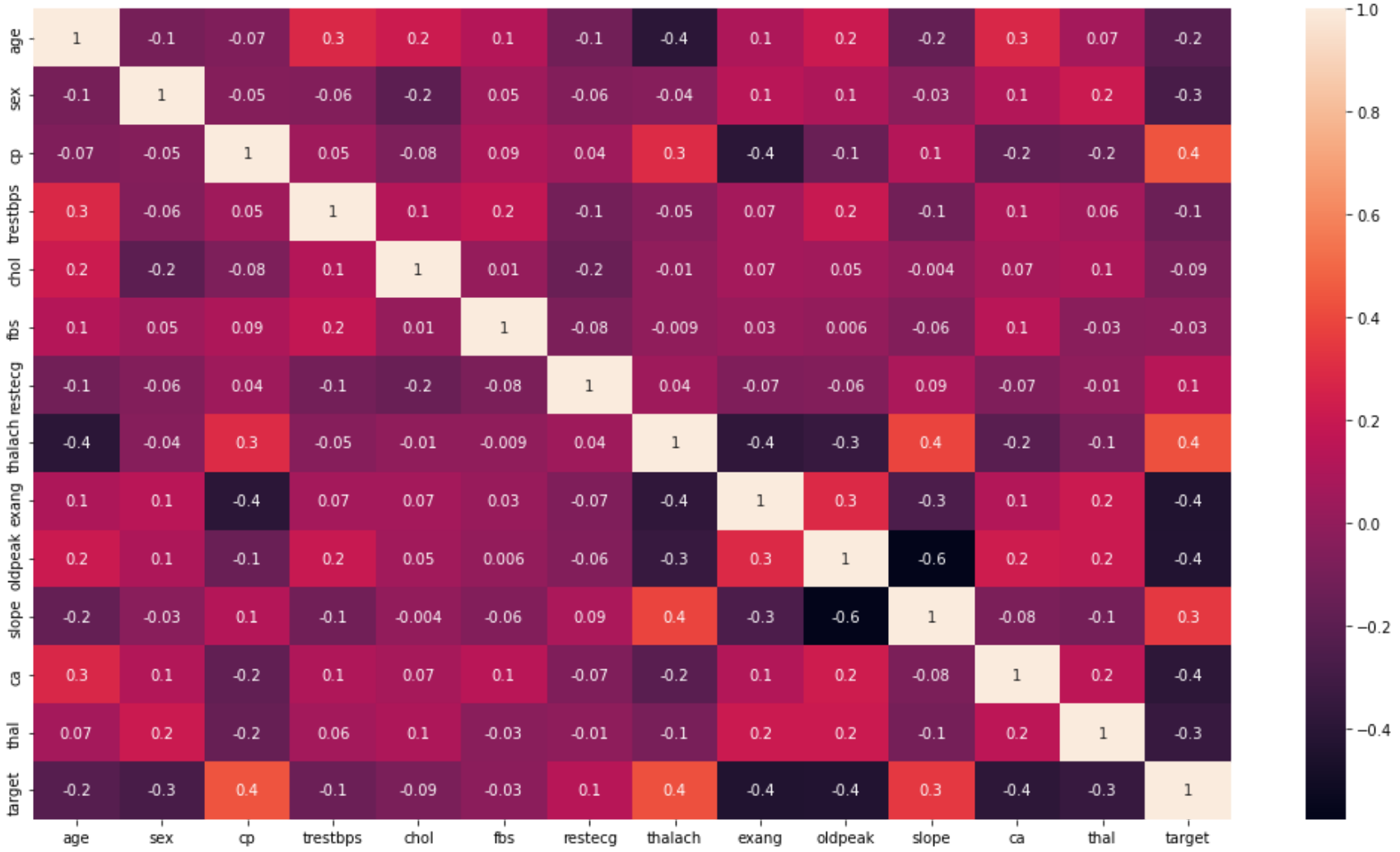
In [82]:
```
1 fig, ax = plt.subplots(figsize=(8,6))
2 ax.scatter(df['trestbps'], df['chol'])
```

Out[82]: <matplotlib.collections.PathCollection at 0x2b3fe963700>



In [133]:
```
1 import seaborn as sns
2
3 plt.figure(figsize=(18,10))
4 sns.heatmap(df.corr(), annot= True, fmt='.1g')
```

Out[133]: <AxesSubplot:>



In [35]:
```
df[['age','sex','cp','trestbps','chol','fbs','restecg','thalach','exang',
    'oldpeak','slope','ca','thal','target']].corr()['target'][:]

corr_with_target = df.corr()
corr_with_target['target'].sort_values(ascending= False)
```

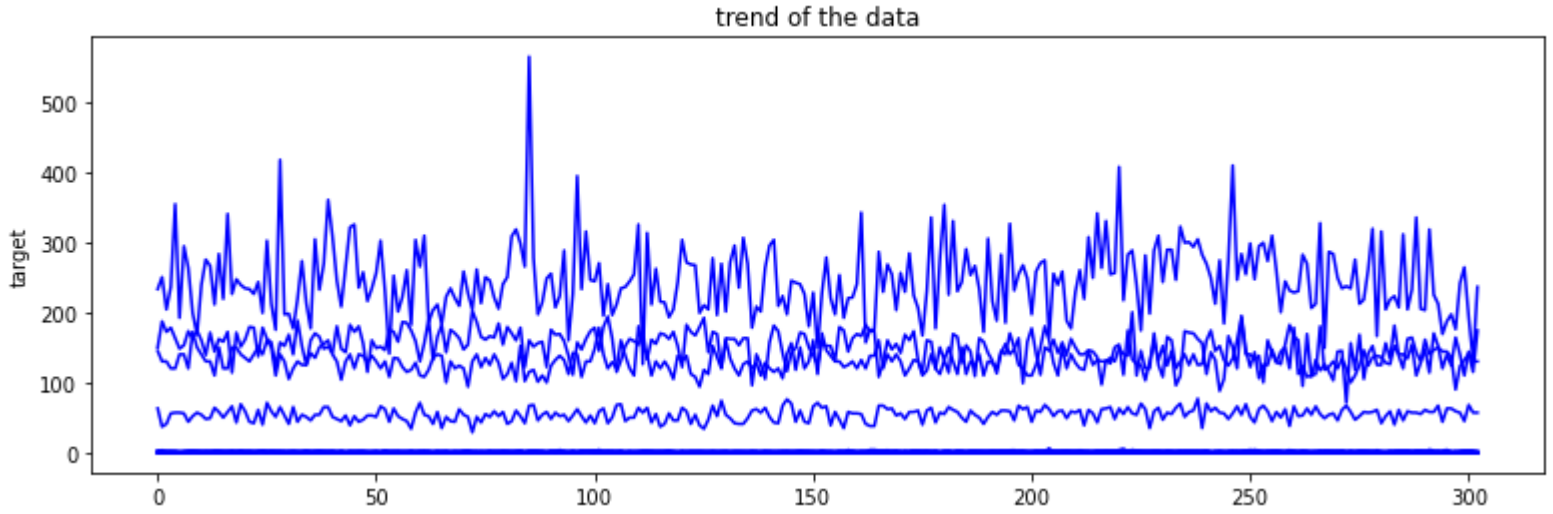
Out[35]:
```
target      1.000000
cp          0.433798
thalach     0.421741
slope       0.345877
restecg     0.137230
fbs        -0.028046
chol       -0.085239
trestbps   -0.144931
age        -0.225439
sex        -0.280937
thal       -0.344029
ca         -0.391724
oldpeak    -0.430696
exang      -0.436757
Name: target, dtype: float64
```

In [134]:
```
df.skew()
```

Out[134]:
```
age        -0.202463
sex        -0.791335
cp          0.484732
trestbps    0.713768
chol        1.143401
fbs         1.986652
restecg     0.162522
thalach    -0.537410
exang       0.742532
oldpeak     1.269720
slope      -0.508316
ca          1.310422
thal       -0.476722
target     -0.179821
dtype: float64
```

In [135]:
```
# trend of the data

plt.figure(figsize=(13,4))
plt.plot(df, color= 'blue')
plt.title('trend of the data')
plt.ylabel('target')
```

Out[135]: Text(0, 0.5, 'target')



In [ ]:
```
1
```

In [ ]:
```
1
```

In [ ]:
```
1
```

In [ ]:
```
1
```