

```
In [1]: import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
```

```
In [2]: df=pd.read_csv('heart.csv')
```

```
In [3]: df.head()
```

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

```
In [4]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 303 entries, 0 to 302
Data columns (total 14 columns):
age          303 non-null int64
sex          303 non-null int64
cp           303 non-null int64
trestbps     303 non-null int64
chol         303 non-null int64
fbs          303 non-null int64
restecg      303 non-null int64
thalach      303 non-null int64
exang        303 non-null int64
```

```
oldpeak      303 non-null float64
slope        303 non-null int64
ca           303 non-null int64
thal         303 non-null int64
target       303 non-null int64
dtypes: float64(1), int64(13)
memory usage: 33.2 KB
```

```
In [7]: youth=df[(df['age']<=40) & (df['target']==1)]
        len(youth)

        middle_age=df[(df.age>40) & (df.target==1)]
        len(middle_age)
```

```
Out[7]: 152
```

```
In [8]: df.target.value_counts()
```

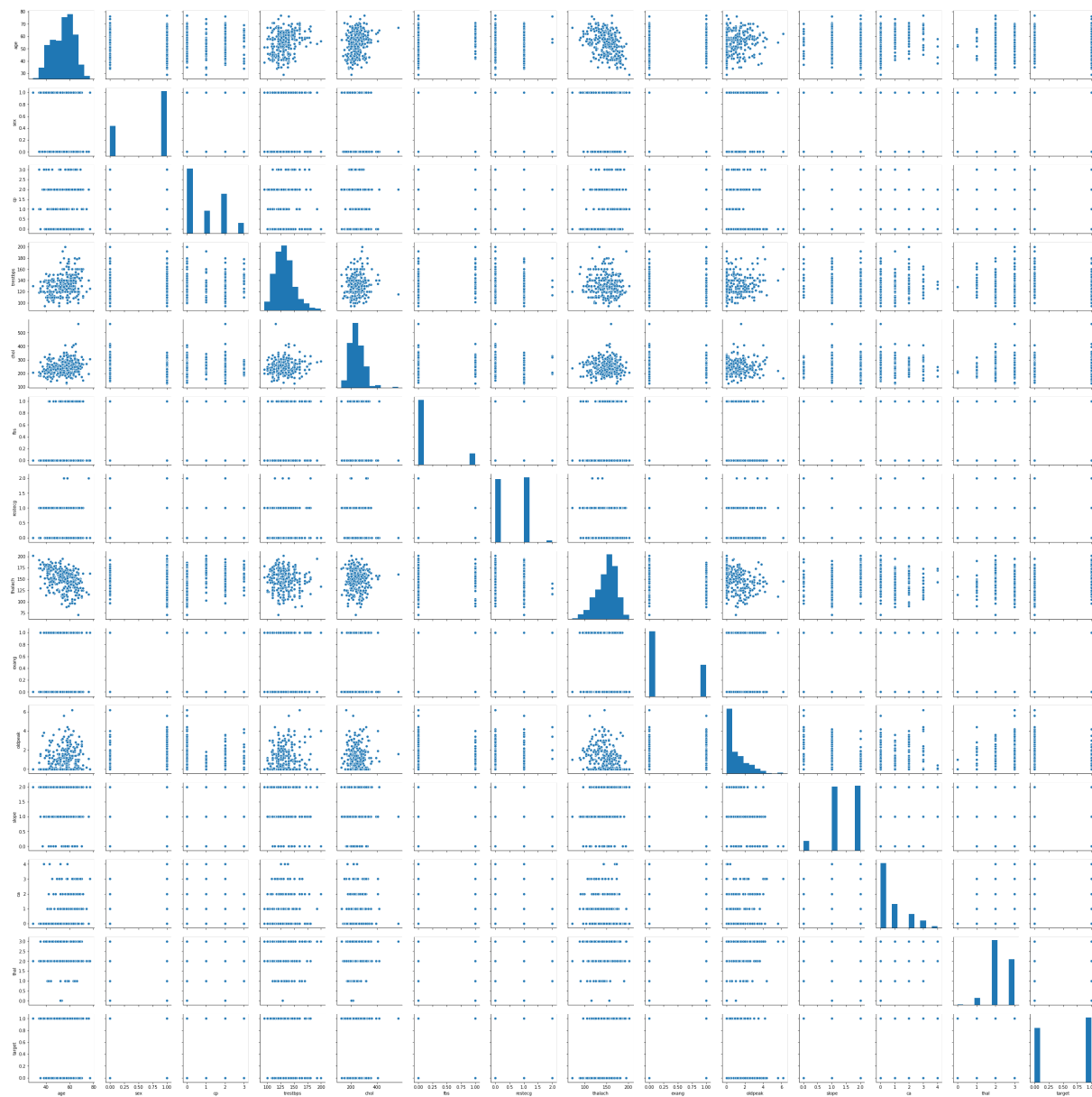
```
Out[8]: 1    165
        0    138
        Name: target, dtype: int64
```

```
In [9]: df.sex.value_counts()
```

```
Out[9]: 1    207
        0    96
        Name: sex, dtype: int64
```

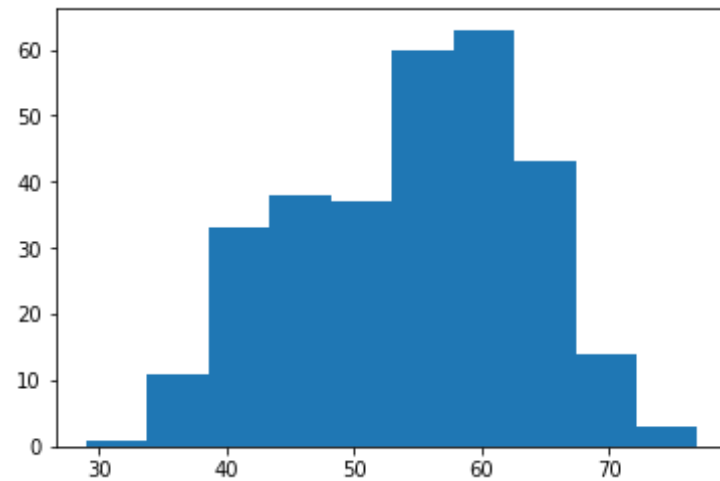
```
In [10]: sns.pairplot(data=df)
```

```
Out[10]: <seaborn.axisgrid.PairGrid at 0x14e8c3d2630>
```



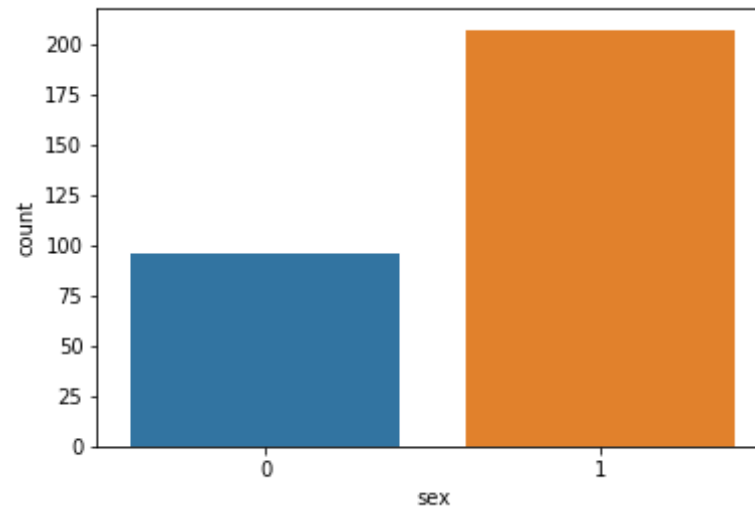
```
In [11]: df.age.hist(grid=False)
```

```
Out[11]: <matplotlib.axes._subplots.AxesSubplot at 0x14e94246550>
```



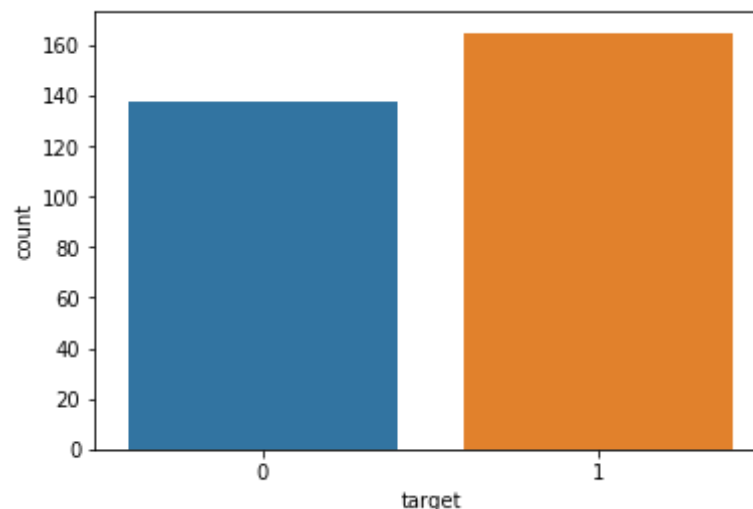
```
In [12]: sns.countplot('sex', data=df)
```

```
Out[12]: <matplotlib.axes._subplots.AxesSubplot at 0x14e94e2f400>
```



```
In [13]: sns.countplot('target', data=df)
```

Out[13]: <matplotlib.axes._subplots.AxesSubplot at 0x14e94850320>



In [14]: `df.columns`

Out[14]: Index(['age', 'sex', 'cp', 'trestbps', 'chol', 'fbs', 'restecg', 'thalach',
 'exang', 'oldpeak', 'slope', 'ca', 'thal', 'target'],
 dtype='object')

In [15]: `X=df[['age', 'sex', 'cp', 'trestbps', 'chol', 'fbs', 'restecg', 'thalach',
 'exang', 'oldpeak', 'slope', 'ca', 'thal']]`

In [16]: `y=df['target']`

In [17]: `from sklearn.cross_validation import train_test_split`

C:\Users\Dell\Anaconda3\lib\site-packages\sklearn\cross_validation.py:41: DeprecationWarning: This module was deprecated in version 0.18 in favor of the model_selection module into which all the refactored classes and functions are moved. Also note that the interface of the new CV iterators are different from that of this module. This module will be removed in a future version.

```
ved in 0.20.  
"This module will be removed in 0.20.", DeprecationWarning)
```

```
In [18]: X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.4  
, random_state=101)
```

```
In [19]: from sklearn.linear_model import LogisticRegression
```

```
In [20]: lm=LogisticRegression()
```

```
In [21]: lm.fit(X_train,y_train)
```

```
Out[21]: LogisticRegression(C=1.0, class_weight=None, dual=False, fit_intercept=  
True,  
            intercept_scaling=1, max_iter=100, multi_class='ovr', n_jobs=  
1,  
            penalty='l2', random_state=None, solver='liblinear', tol=0.00  
01,  
            verbose=0, warm_start=False)
```

```
In [22]: from sklearn.metrics import classification_report
```

```
In [23]: from sklearn.metrics import confusion_matrix
```

```
In [24]: predictions=lm.predict(X_test)
```

```
In [25]: print(classification_report(y_test,predictions))
```

	precision	recall	f1-score	support
0	0.92	0.77	0.84	60
1	0.81	0.94	0.87	62
avg / total	0.86	0.85	0.85	122

```
In [26]: print(confusion_matrix(y_test,predictions))
```

```
[[46 14]
 [ 4 58]]
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```