linear regression

In [2]:

```
from sklearn import datasets

data = datasets.load_boston()
```

In [15]:

```
len(data.data), len(data.feature_names), data.feature_names
```

Out[15]:

```
(506,
 13,
 array(['CRIM', 'ZN', 'INDUS', 'CHAS', 'NOX', 'RM', 'AGE', 'DIS', 'RAD',
        'TAX', 'PTRATIO', 'B', 'LSTAT'], dtype='<U7'))
```

In [8]:

```
X = data.data[:,0]
Y = data.target
```

In [10]:

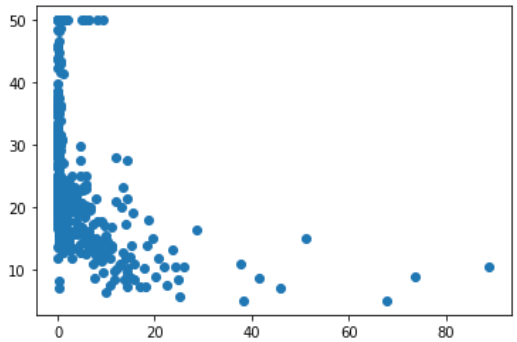
```
X.ndim, Y.ndim, X.shape, Y.shape #하나만 있어서 일치
```

Out[10]:

```
(1, 1, (506,), (506,))
```

In [13]:

```
import matplotlib.pyplot as plt
plt.scatter(X,Y)
plt.show()
```




In [17]:

```
import numpy as np

X = np.c_[np.ones(len(X)), X] #using trick
```

In [20]:

```
X.shape #to be vectorized
X[0]
```

Out[20]:

```
array([1.     , 0.00632])
```

In [21]:

```
theta = np.linalg.inv(X.T.dot(X)).dot(X.T).dot(Y)
```

In [23]:

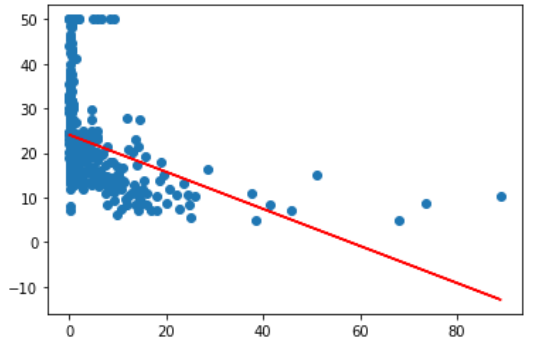
```
theta.shape
theta #감소 형태
```

Out[23]:

```
array([24.03310617, -0.41519028])
```

In [26]:

```
plt.scatter(X[:,1],Y)
plt.plot(X[:,1], X.dot(theta), "r-")
plt.show()
```

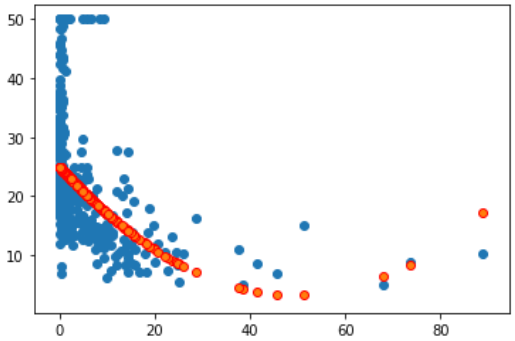



In [34]:

```
X = np.c_[X, np.power(X[:,1],2)] #multiplized
theta = np.linalg.inv(X.T.dot(X)).dot(X.T).dot(Y)
```

In [31]:

```
plt.scatter(X[:,1],Y)
plt.scatter(X[:,1], X.dot(theta), edgecolors="r")
plt.show()
```

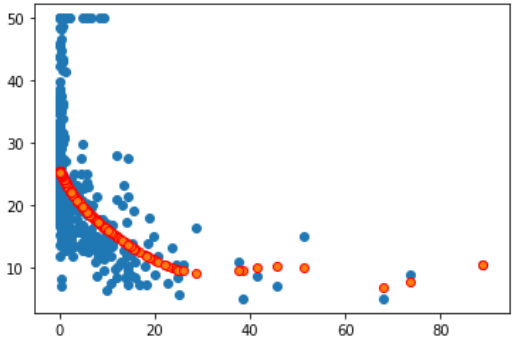



In [32]:

```
#polynomialized
X = np.c_[X, np.power(X[:,1],3), np.power(X[:,1],4), np.power(X[:,1],5), np.power(X[:,1],6), np.power(X[:,1],7), np.power(X[:,1],8)]
theta = np.linalg.inv(X.T.dot(X)).dot(X.T).dot(Y)
```

In [33]:

```
plt.scatter(X[:,1],Y)
plt.scatter(X[:,1], X.dot(theta), edgecolors="r")
plt.show()
```




In [ ]:

```
why logistic <- linear로 가능하지 않아서
```