In [2]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

讀取檔案

In [3]:
```
df = pd.read_csv("/Users/chuanyang/Downloads/regression.csv")
```

In [11]:
```
df.head()
```

Out[11]:

	X	y
0	1.394330	591
1	1.110690	539
2	0.095543	413
3	-1.591390	310
4	-0.844961	308

輸入X跟y

In [19]:
```
X = df.iloc[:, 0].values #代表row的第0到最後一個, column第0列
y = df.iloc[:, 1].values #代表row的第0到最後一個, column第1列
```

In [ ]:
```
########範例: iloc
#df.iloc[row,column]
```

In [7]:
```
example= pd.DataFrame({'A':[1,2,3],
                       'B':[4,5,6],
                       'C':[7,8,9],
                       'D':[1,3,5],
                       'E':[5,3,6],
                       'F':[7,4,3]})
print (example)
```

```
   A  B  C  D  E  F
0  1  4  7  1  5  7
1  2  5  8  3  3  4
2  3  6  9  5  6  3
```

In [9]:
```
print(example.iloc[:, :-1])
```

```
   A  B  C  D  E
0  1  4  7  1  5
1  2  5  8  3  3
2  3  6  9  5  6
```

In [10]:
```
print(example.iloc[:, :-1].values) #去掉column name, 值變成array
```

```
[[1 4 7 1 5]
 [2 5 8 3 3]
 [3 6 9 5 6]]
```

In [ ]:
```
```

查看資料的correlation

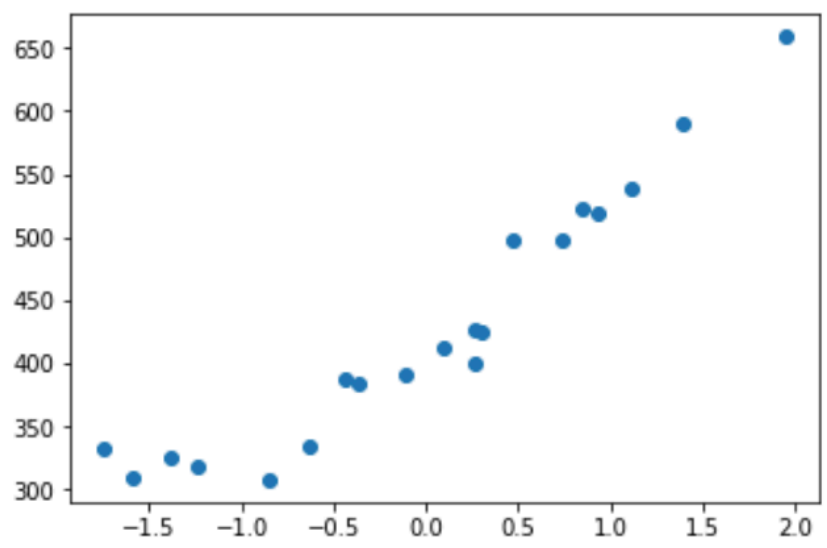
In [12]:
```
df.corr()
```

Out[12]:

	X	y
X	1.000000	0.949494
y	0.949494	1.000000

查看資料分布

In [4]:
```
# 資料分佈
plt.plot(X, y, 'o')
plt.show()
```



Gradient descent algorithm

寫出Cost Fuction ¶

def computeCost(X, y):

\\ 回傳

$$E(\theta_0, \theta_1) = \frac{1}{2}\sum_{i=1}^{m}\left(h_\theta(x^{(i)}) - y^{(i)}\right)^2$$

In [21]:
```
# y = ax + b
# hθ(x) = θ0 + θ1x
def f(X):
    return theta[0]+theta[1]*X
```

In [20]:
```
# compute cost function
def computeCost(X, y):
    return np.sum(((f(X)-y)**2)/2)
```

計算theta0,theta1

$$temp0 := \theta_0 - \eta\sum_{i=1}^{m}\left(h_\theta(x^{(i)}) - y^{(i)}\right)$$

$$temp1 := \theta_1 - \eta\sum_{i=1}^{m}\left(h_\theta(x^{(i)}) - y^{(i)}\right)x^{(i)}$$

$$\theta_0 := temp0$$

$$\theta_1 := temp1$$

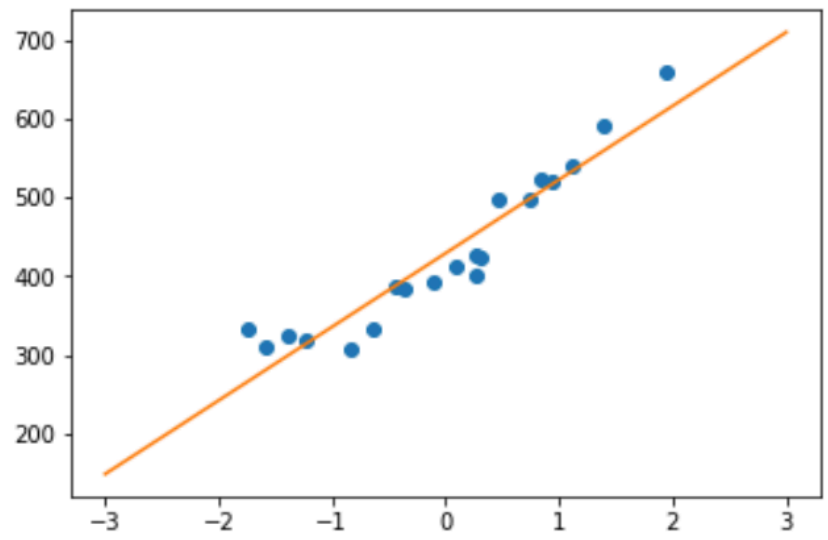
In [22]:
```
# main funtion
eta = 0.01 #learning rate
count = 200 #計算200次
theta = [0,0] #初始theta0,theta1
temp = [0,0]
error = []
# update
for i in range(1,count+1,1):
    temp[0] = theta[0] - eta*np.sum(f(X)-y)
    temp[1] = theta[1] - eta*np.sum((f(X)-y)*X)
    theta = temp
    error.append(computeCost(X,y))
    log='{}次 : theta0 = {:.3f},theta1 = {:.3f}'
    print(log.format(i,theta[0],theta[1]))
```

```
1次 : theta0 = 85.830,theta1 = 18.696
2次 : theta0 = 154.494,theta1 = 33.652
3次 : theta0 = 209.425,theta1 = 45.618
4次 : theta0 = 253.370,theta1 = 55.190
5次 : theta0 = 288.526,theta1 = 62.848
6次 : theta0 = 316.651,theta1 = 68.974
7次 : theta0 = 339.151,theta1 = 73.875
8次 : theta0 = 357.151,theta1 = 77.796
9次 : theta0 = 371.550,theta1 = 80.932
10次 : theta0 = 383.070,theta1 = 83.442
11次 : theta0 = 392.286,theta1 = 85.449
12次 : theta0 = 399.659,theta1 = 87.055
13次 : theta0 = 405.557,theta1 = 88.340
14次 : theta0 = 410.276,theta1 = 89.368
15次 : theta0 = 414.051,theta1 = 90.190
16次 : theta0 = 417.071,theta1 = 90.848
17次 : theta0 = 419.486,theta1 = 91.374
18次 : theta0 = 421.419,theta1 = 91.795
19次 : theta0 = 422.965,theta1 = 92.132
20次 : theta0 = 424.202,theta1 = 92.401
21次 : theta0 = 425.192,theta1 = 92.617
22次 : theta0 = 425.983,theta1 = 92.789
23次 : theta0 = 426.617,theta1 = 92.927
24次 : theta0 = 427.123,theta1 = 93.037
25次 : theta0 = 427.529,theta1 = 93.126
26次 : theta0 = 427.853,theta1 = 93.196
27次 : theta0 = 428.112,theta1 = 93.253
28次 : theta0 = 428.320,theta1 = 93.298
29次 : theta0 = 428.486,theta1 = 93.334
30次 : theta0 = 428.619,theta1 = 93.363
31次 : theta0 = 428.725,theta1 = 93.386
32次 : theta0 = 428.810,theta1 = 93.405
33次 : theta0 = 428.878,theta1 = 93.420
34次 : theta0 = 428.932,theta1 = 93.431
35次 : theta0 = 428.976,theta1 = 93.441
36次 : theta0 = 429.011,theta1 = 93.448
⋮
181次 : theta0 = 429.150,theta1 = 93.479
182次 : theta0 = 429.150,theta1 = 93.479
183次 : theta0 = 429.150,theta1 = 93.479
184次 : theta0 = 429.150,theta1 = 93.479
185次 : theta0 = 429.150,theta1 = 93.479
186次 : theta0 = 429.150,theta1 = 93.479
187次 : theta0 = 429.150,theta1 = 93.479
188次 : theta0 = 429.150,theta1 = 93.479
189次 : theta0 = 429.150,theta1 = 93.479
190次 : theta0 = 429.150,theta1 = 93.479
191次 : theta0 = 429.150,theta1 = 93.479
192次 : theta0 = 429.150,theta1 = 93.479
193次 : theta0 = 429.150,theta1 = 93.479
194次 : theta0 = 429.150,theta1 = 93.479
195次 : theta0 = 429.150,theta1 = 93.479
196次 : theta0 = 429.150,theta1 = 93.479
197次 : theta0 = 429.150,theta1 = 93.479
198次 : theta0 = 429.150,theta1 = 93.479
199次 : theta0 = 429.150,theta1 = 93.479
200次 : theta0 = 429.150,theta1 = 93.479
```

In [25]:
```
theta[0],theta[1]
```

Out[25]: (429.15005982643174, 93.47879974527054)

In [8]:
```
# regression line & data
t = np.linspace(-3,3,100)
plt.plot(X, y ,'o')
plt.plot(t, f(t))
plt.show()
```



In [14]:
```
# iteration loss
t = np.linspace(1, 202, 201)
plt.plot(error, 'r-')
plt.xlabel('iteration')
plt.ylabel('cost')
plt.show()
```