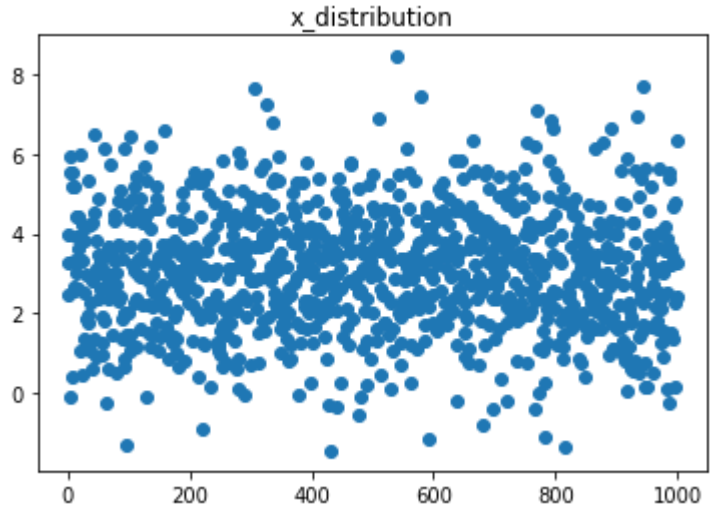
In [13]:
```
import numpy as np
import matplotlib.pyplot as plt
```

In [14]:
```
w1 = 1.3
x = np.random.normal(3,1.5, 1000)
e = np.random.normal(0.5, 0.3, 1000)
y_actual = w1*x + e
```

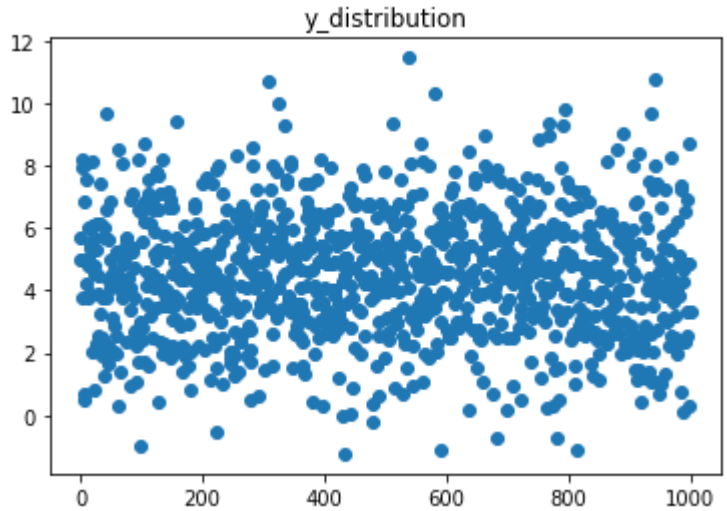
In [17]:
```
plt.scatter(np.linspace(0,1000,1000), x ); plt.title('x_distribution')
```

Out[17]:
```
Text(0.5, 1.0, 'x_distribution')
```



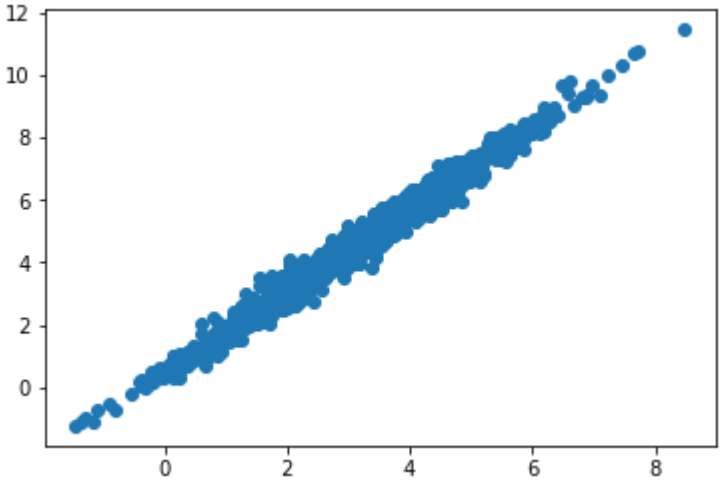
In [19]:
```
plt.scatter(np.linspace(0,1000,1000), y_actual ); plt.title('y_distribution')
```

Out[19]:
```
Text(0.5, 1.0, 'y_distribution')
```



In [21]:
```
plt.scatter(x,y_actual)
```

Out[21]:
```
<matplotlib.collections.PathCollection at 0x201ff31fa58>
```



In [ ]:
```
need to shuffle the data
```

In [31]:
```
np.random.randint(0,5)
```

Out[31]:
```
4
```

In [59]:
```
for i in range(10):
    print(x[i])
```

```
3.2859818436174937
3.954437000471674
2.485555424015841
5.541975990435575
5.95037723356632
-0.11928259508457728
5.204611850575539
0.38165702422444525
5.5171561670179585
2.585431712766103
```

In [63]:
```
m = 0.1
c = 0.2
lr = 0.01
n = len(y_actual)
cost_list = []
parameter_list = []
n_epochs = 1000

for i in range(n_epochs):
    cost = 0
    d_m = 0
    d_c = 0
    parameter_list.append((m,c))
    for i in range(n):
        index = np.random.randint(0,n)
        y_pred = m*x[index] + c

        d_m += -2/n * (x[index]*(y_actual[index] - y_pred))
        d_c += -2/n * (y_actual[index] - y_pred)


        cost += -(1/(2*n)) * (np.power(y_pred - y_actual[index], 2))


        m = m - d_m*lr
        c = c- d_c*lr
    cost_list.append(cost)
```

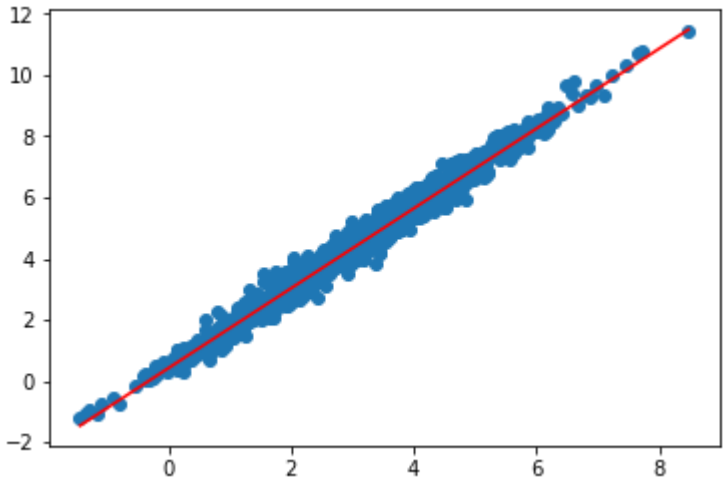
In [64]:
```
pos = cost_list.index(min(cost_list, key = abs))
m = parameter_list[pos][0]
c = parameter_list[pos][1]
cost = cost_list[pos]
```

In [65]:
```
m,c, cost
```

Out[65]:
```
(1.3047949259674168, 0.4200696931559787, -0.04073190323156033)
```

In [66]:
```
Y_pred = m*x + c

plt.scatter(x, y_actual)
plt.plot([min(x), max(x)], [min(Y_pred), max(Y_pred)], color='red')  # regress
ion line
plt.show()
```



SGD Issues

- local Minima
- slow covnvergence