**实验题目：**

**线性回归**

**import** numpy **as** np  
**import** matplotlib.pyplot **as** plt  
  
**class** LinerRegressionModel(object):  
 **def** \_\_init\_\_(self, data):  
 self.data = data  
 self.x = data[:, 0]  
 self.y = data[:, 1]  
  
 **def** log(self, a, b):  
 print(**"计算出的线性回归函数为:\ny = {:.5f}x + {:.5f}"**.format(a, b))  
  
 **def** plt(self, a, b):  
 plt.plot(self.x, self.y, **'o'**, label=**'data'**, markersize=10)  
 plt.plot(self.x, a \* self.x + b, **'r'**, label=**'line'**)  
 plt.legend()  
 plt.show()  
  
 **def** least\_square\_method(self):  
 **def** calc\_ab(x, y):  
 sum\_x, sum\_y, sum\_xy, sum\_xx = 0, 0, 0, 0  
 n = len(x)  
 **for** i **in** range(0, n):  
 sum\_x += x[i]  
 sum\_y += y[i]  
 sum\_xy += x[i] \* y[i]  
 sum\_xx += x[i]\*\*2  
 a = (sum\_xy - (1/n) \* (sum\_x \* sum\_y)) / (sum\_xx - (1/n) \* sum\_x\*\*2)  
 b = sum\_y/n - a \* sum\_x/n  
 **return** a, b  
 a, b = calc\_ab(self.x, self.y)  
 self.log(a, b)  
 self.plt(a, b)  
  
  
data = np.array([[1, 2.5], [2, 3.3], [2.5, 3.8],[3, 4.5], [4, 5.7], [5, 6]])  
model = LinerRegressionModel(data)  
model.least\_square\_method()

实验结果：

