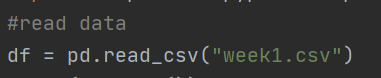
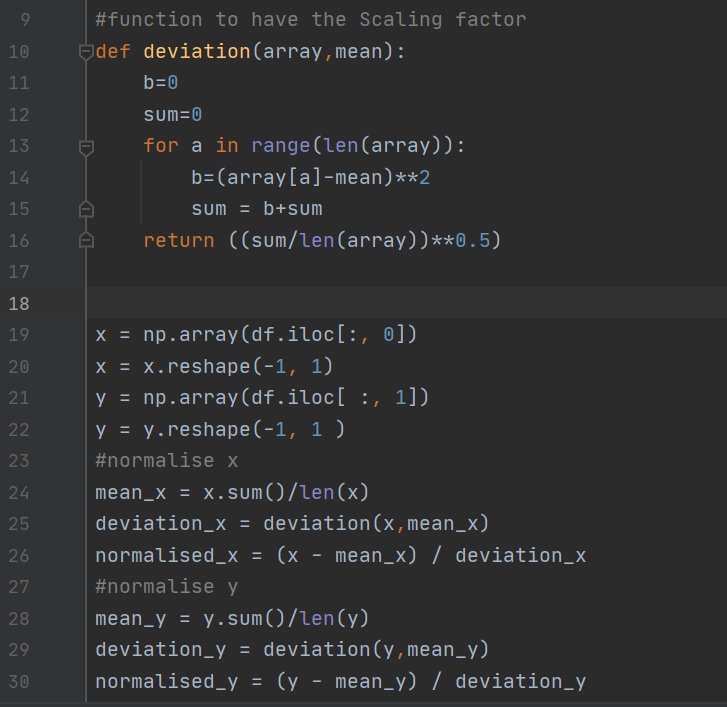
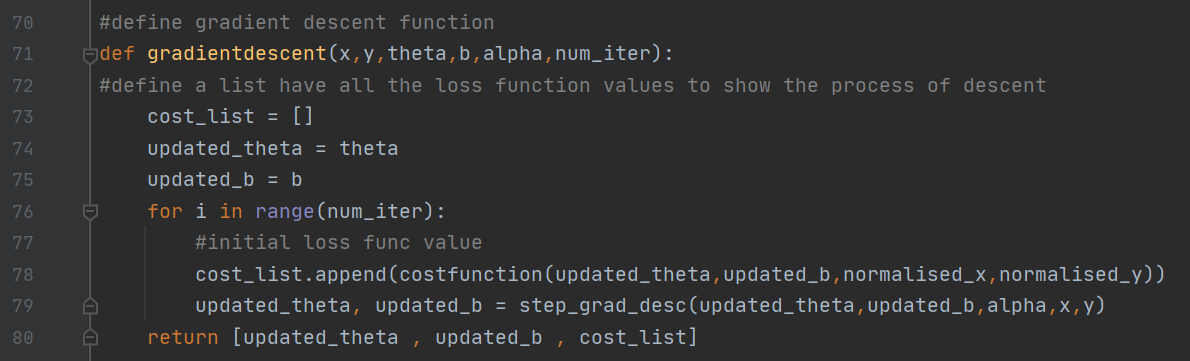
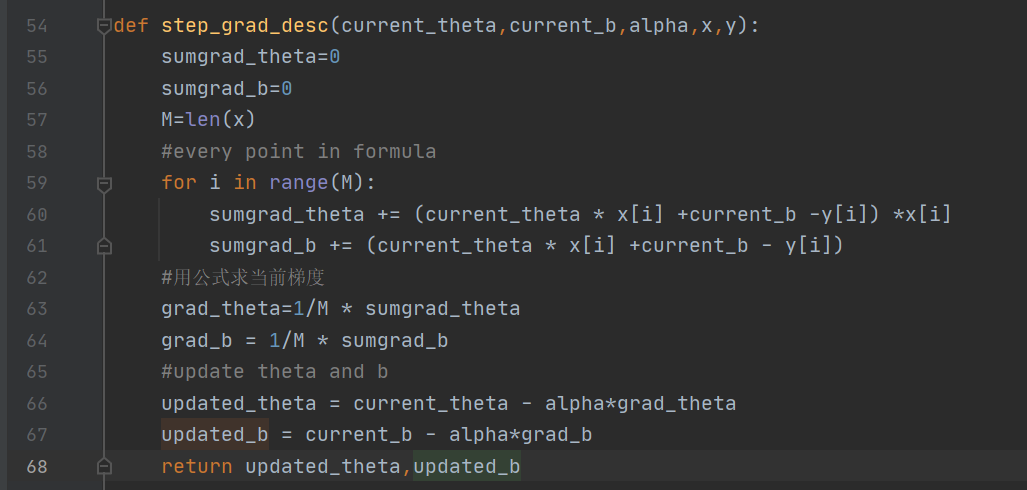
(a)

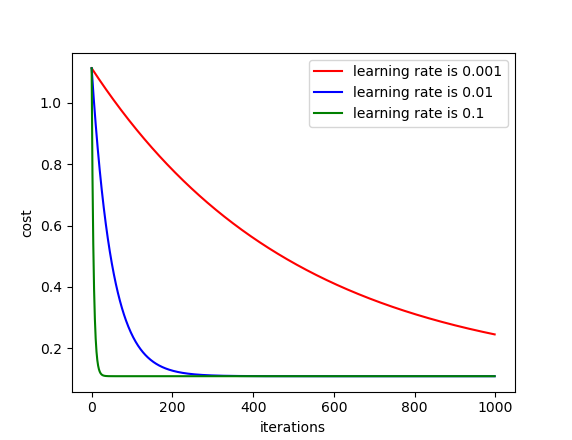
1. Read data
2. Normalize the data

First I define a function to calculate the deviation, and then I use (x-mean\_x)/deviation\_x to replace x input. (the same to y)

1. Gradient descent(2 functions).I define step\_grad\_desc(…) to update parameters for each step, then in function gradiantdescent(…) I apply step\_grad\_desc(…) for each iterations.



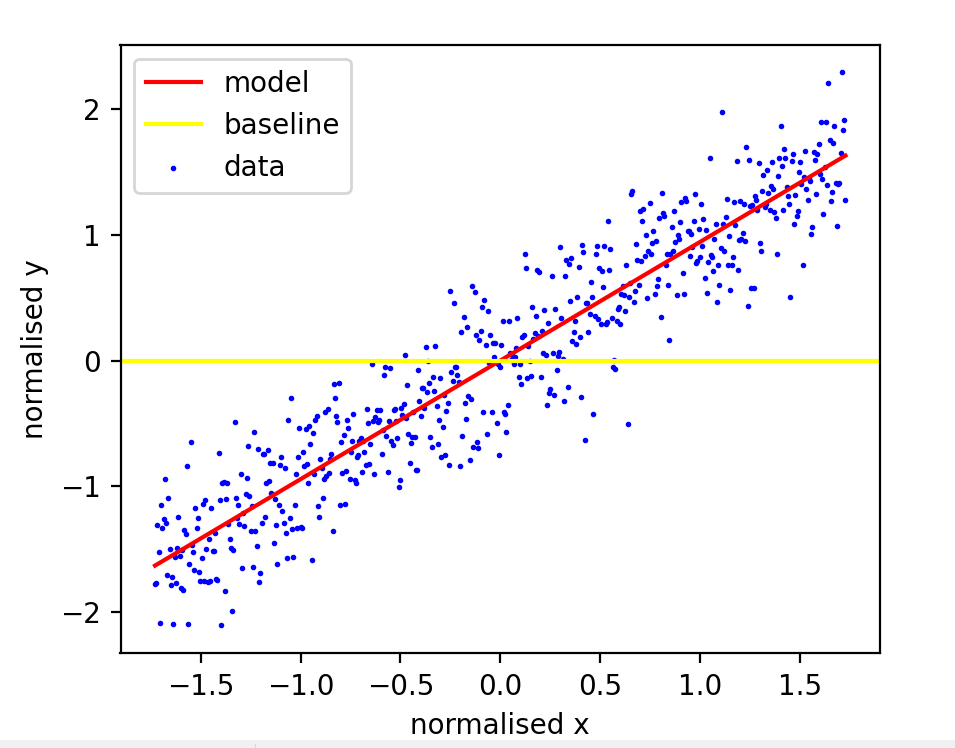
(b)

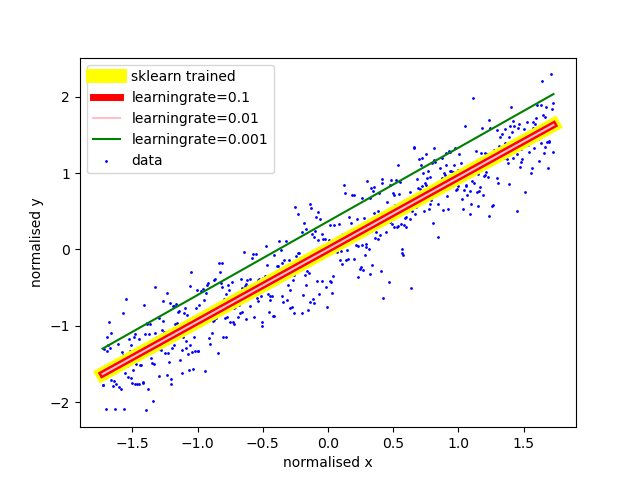
1. Learning rate varies, and the 0.001 of learning rate is too small which seems hard to converge.
2. The liner regression model is y=theta\*x + b , and the final result is :

theta = 0.9644183

b = 0.36806349

1. The cost is: 1.1126116296011173. The baseline is the average of the normalized y, I find the data is evenly distributed.



1. 

As is shown in the figure, The learning rate of 0.001 is too small and the green model is not good , the sk-learn trained is the best and together with the 0.1 and 0.01 of the learning rate. (I changed the width of the line so as to see clearly).