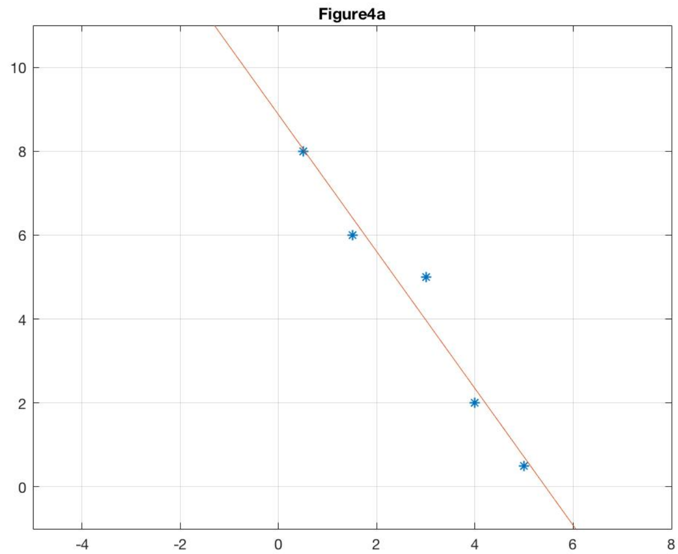
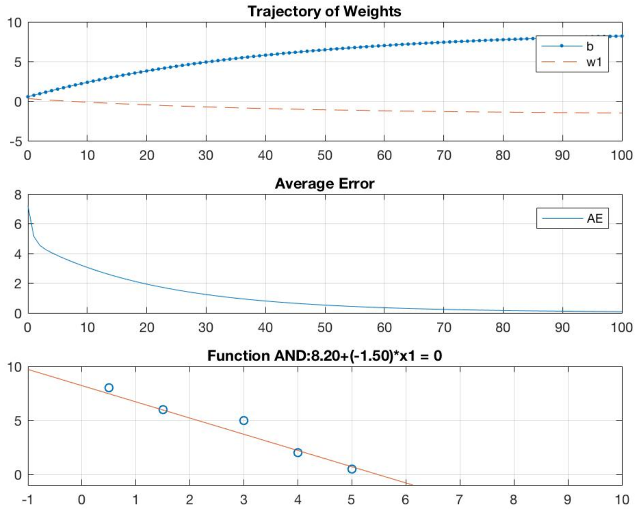



b)

The LMS algorithm is implemented by the following code based on MATLAB:

```
N = 100;
X = [1 0.5;1 1.5;1 3;1 4;1 5];
d = [8.0;6;5;2;0.5];
w = [0.4;0.6];
lr = 0.02;
for n = 0 : N
    for i = 1:length(d)
        e = d(i) - X(i,:) * w;
        e1(n+1) = e*e/2;
        w = w + e * lr * X(i,:)';
        w1(n+1) = w(1);
        w2(n+1) = w(2);
    end
end
```

After 100 epochs, the weight vector is w0 =$[8.2044, -1.5009]^T$ and the fitting result y =-1.5009x +8.2044 is shown in figure below with the trajectories of the weights and Average of the cost $e^2/2$. The weights and error do have a converging tendency, but with finite time there will always exist a small errors.




c)

It is clearly from figures that the fitting results determined by LMS algorithm is very close to that obtained from LLS method, which demonstrates the correctness of LMS algorithm. Additionally, the LMS algorithm requires a long training procedure to converge, while the computation load would will be high when applying LLS method to high-dimensional data.

d)

When the learning rate is set as $\eta = 0.2$, the weights of the fitting function as well as the estimation error will diverge as shown in figure below. This is because the LMS algorithm only valid when the Taylor approximation is effective, which is guaranteed by a fairly small learning rate.