**Hw1**

In our last meeting, we studied how to generate a linear fit using a very simple data which I also gave below one more time;

x=0:1:7;

y=[0,20,60,68,77,110,100,130];

We used a simple y=mx+c line mode to fit this data and we performed this operation using polyfit . Then we saw how to use mldivide (\) instead of polyfit.

Now your next step is to demonstrate your skill to be able to fit the same data this time using a quadratic line shape : y=ax^2+bx+c . However, I am not only looking forward to see the usage of polyfit but also the usage of mldivide (\) for this process.

%Şeyma kerküklü 090200124

clear;

clc;

format short g;

format compact;

x=0:1:7;

y=[0,20,60,68,77,110,100,130];

dy=sqrt(y);

figure(1); hold on;

errorbar(x,y,dy,'ro');

%error bar function will plot the data points and overlay error

%bars that extend from each data point to show the range of uncertainty.

[nfit,others]=polyfit(x,y,2);

%since y=ax^2+bx+c is quadratic line shape we used 2nd degreeof polynomial

yfit=polyval(nfit,x);

%polyval function returns the corresponding y-values for the polynomial fit.

plot(x,yfit,'-b');

disp("each value is the coefficients of the function")

for i=1:3

disp([i , nfit(i)]) %coefficients of y=ax^2+bx+c

end

A = [x'.^2, x', ones(length(x),1)]; % for mldivide we create A matrix

coefficients=A\y'; %the result of the mldivide will give coefficient matrix

coeffs=coefficients';

disp("coefficents of mldivide")

for i=1:3

disp([i , coefficients(i)]) %coefficients of y=ax^2+bx+c

end

disp("mldivide and polyfit technique gives same results")

yfit2=coeffs(1)\*x.^2+coeffs(2)\*x+coeffs(3); %fitted y values

plot(x,yfit2,'-g');

