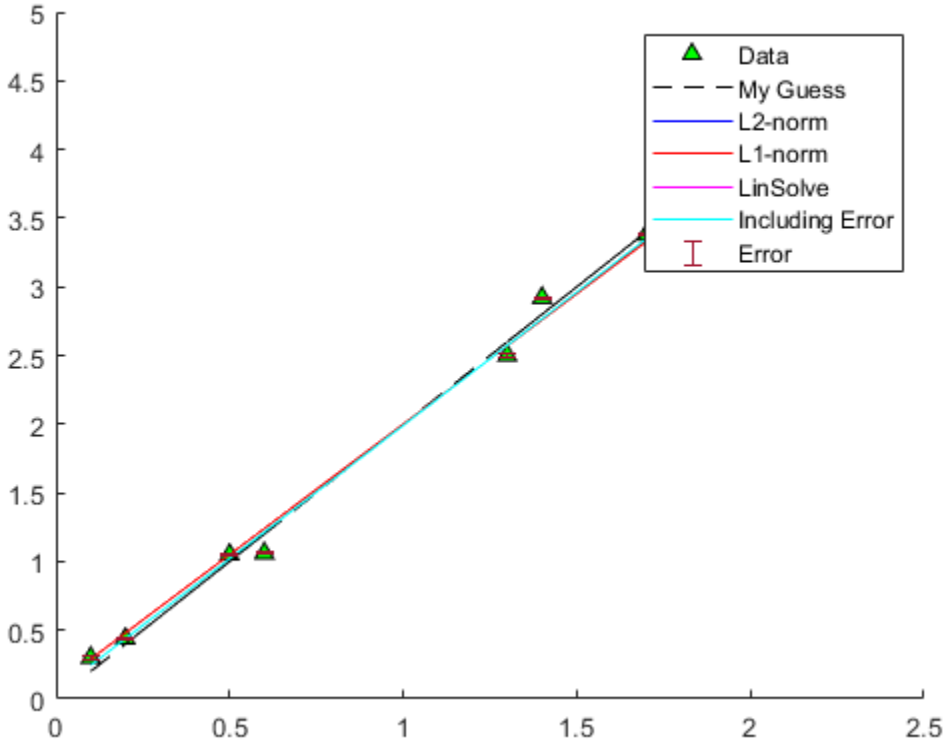
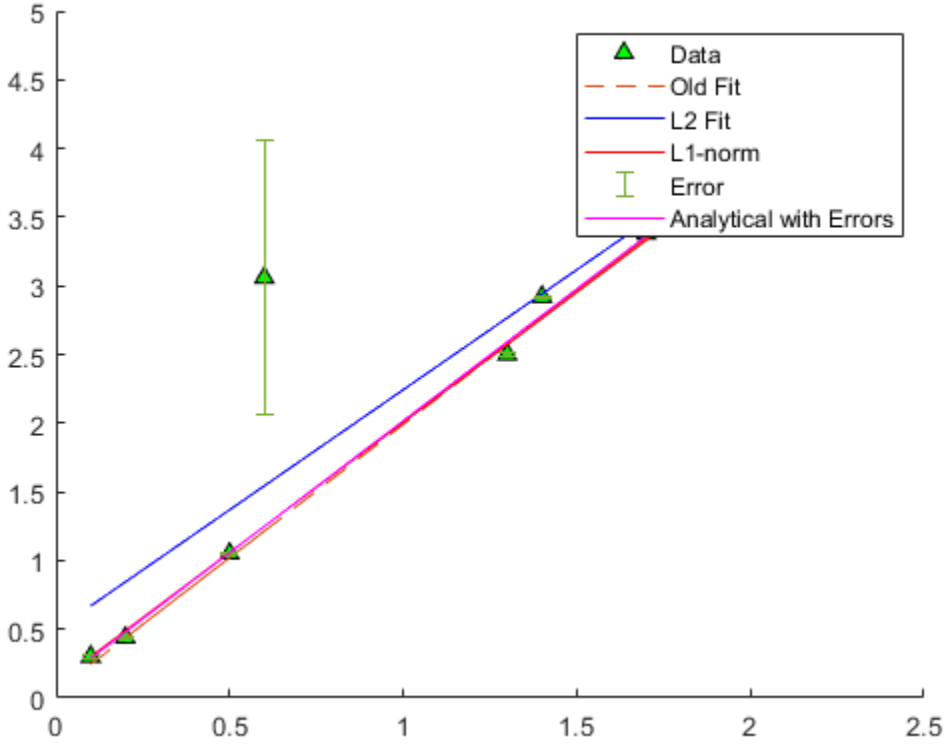
```
%Matlab Practical 9
%Zach Vig

clear;
clearvars;
close all;

%q1.1
x = [0.1; 0.6; 0.2; 0.5; 1.4; 2.3; 1.7; 1.3];
y = [0.3; 1.06; 0.44; 1.05; 2.92; 4.47; 3.38; 2.5];

%q1.2
figure(1); hold on;
plot(x,y,'k^','MarkerFaceColor','green',"DisplayName","Data");

%q1.3
% m = [slope; y-intercept]
% Am = y

%my guess :)
A = [x,ones(length(x),1)]; m0 = [2; 0];
plot(x,A*m0,"DisplayName","My Guess","Color","Black","LineStyle","--");

%q1.4
%{
    A is an 8 by 2 matrix, meaning we have 8 data points to
determine 2 parameters (slope, y-intercept). This means that the problem is
overdetermined.
%}

%q1.5
misfitL2 = @(m)sum((A*m-y) .^ 2);
m2 = fminsearch(misfitL2,m0);
plot(x,A*m2,"DisplayName","L2-norm","Color","Blue");
%slope: 1.9405, y-intercept: 0.0502

%q1.6
misfitL1 = @(m)sum(abs(A*m-y));
m1 = fminsearch(misfitL1,m0);
plot(x,A*m1,"DisplayName","L1-norm","Color","Red");
%slope: 1.900, y-intercept: 0.1000. These model values are slightly
different from the L2 model, favoring going through the endpoints points.

%q1.7
m3 = linsolve(A,y);
plot(x,A*m3,"DisplayName","LinSolve","Color","magenta");
%{
    slope: 1.9405, y-intercept: 0.0502. These results match the L2 model
better than the L1 model.
%}

%q1.8
```

```
%{
    0.01 0.00 0.00 0.00 0.00 0.00 0.00 0.00
    0.00 0.01 0.00 0.00 0.00 0.00 0.00 0.00
    0.00 0.00 0.01 0.00 0.00 0.00 0.00 0.00
    0.00 0.00 0.00 0.01 0.00 0.00 0.00 0.00
    0.00 0.00 0.00 0.00 0.01 0.00 0.00 0.00
    0.00 0.00 0.00 0.00 0.00 0.01 0.00 0.00
    0.00 0.00 0.00 0.00 0.00 0.00 0.01 0.00
    0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.01
%}

%q1.9
Cd = diag(repmat(0.01,8,1));

m_analytical = ((A' * (Cd\A)) \ A') * (Cd \ y);

plot(x,A*m_analytical,"DisplayName","Including Error","Color","cyan");
legend()
errorbar(x,y,diag(Cd),"LineStyle","none","DisplayName","Error");
%{
    slope: 1.9405, y-intercept: 0.0502. Accounting for errors did not affect
the results of the fit.
%}
hold off;

Cm = inv(A' * (Cd\A));
corr_coef = Cm(1,2)/(Cm(1,1)^0.5*Cm(2,2)^0.5);

%q2.1
y(2) = 3.06;
misfitL2 = @(m)sum((A*m-y) .^ 2);
m1_outlier = fminsearch(misfitL2,m0);
figure(2); hold on;
plot(x,y,'k^','MarkerFaceColor','green','DisplayName','Data');
plot(x,A*m_analytical,'DisplayName',"Old Fit","LineStyle","--");
plot(x,A*m1_outlier,'DisplayName','L2 Fit','Color','blue');
legend();
%{
    slope: 1.7482, y-intercept: 0.4949
%}

%q2.2
%{
    By including the outlier, the slope has decreased and the y-intercept
has increased. This occured because the fit is now trying to take the
outlier into account.
%}

%2.3
misfitL1 = @(m)sum(abs(A*m-y));
m2_outlier = fminsearch(misfitL1,m0);
plot(x,A*m2_outlier,"DisplayName","L1-norm","Color","Red");
% slope: 1.8955, y-intercept: 0.1104
```

```
%q2.4
%{
    Using the L1 norm, which tends to ignore outliers, the model is now much
closer to the answer in part 1.
%}

%q2.5
%{
    The L1 and L2 norm lead to dramatically different answers because the
model misfit scales as the square of the difference in the L2 norm but only
as the absolute value in the L1 norm. This causes the L2 norm to be much
more sensitive to outliers, one of which we now have in our data.
%}

%q2.6
Cd(2,2) = 1;
errorbar(x,y,diag(Cd),"LineStyle","none","DisplayName","Error");
m_exact_outlier = ((A' * (Cd\A)) \ A') * (Cd \ y);
plot(x,A*m_exact_outlier,"DisplayName","Analytical with
Errors","Color","magenta");
hold off;
%slope: 1.9207, y-intercept: 0.0962

%{
    Now that the outlier point has a large amount of error, the L2 misfit
function is able to ignore its contribution to the model, since it scales as
the inverse of the square of the error on each data point. The model is now
close to the L2 misfit of part 1 and the L1 misfit of part 2 and deviates
from the L2 mistfit that did not include the error on the outlier.
%}
```


Data
My Guess
L2-norm
L1-norm
LinSolve
Including Error
Error
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5
5


Data
Old Fit
L2 Fit
L1-norm
Error
Analytical with Errors
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5
5

Published with MATLAB® R2023b