```
% Linear fit with least square and least absolute error

slope=3; intercept=-2;
abscissa = (-5:5)'; m = length(abscissa);

WhiteNoise = 5*randn(m,1);
ordinates = slope*abscissa + intercept + WhiteNoise;

% outliers
GrossError=80;
ordinates(6)=ordinates(6)+GrossError;
ordinates(10)=ordinates(10)-GrossError;

% Plot on one figure
hold on;

% Plot data points
plot(abscissa, ordinates, 'o')

% L^1 fit for slope and intercept
e = ones(m, 1);
f = [0; 0; ones(m, 1)];
A = [[abscissa, e, -eye(m)]; [-abscissa, -e, -eye(m)]];
b = [ordinates; -ordinates];
LB = [-inf; -inf; zeros(m,1)];
X = linprog(f, A, b, [], [], LB);

% Sample data to use with fit function
a = linspace(-10, 10, 1001);
l1 = plot(a, a * X(1) + X(2));

% Linear fit with least square error, solved by setting up a linear system:
Xls = [abscissa, e] \ ordinates;
l2 = plot(a, a * Xls(1) + Xls(2), 'r-.');

% Actual plot
l3 = plot(a, a * slope + intercept, 'k--');

% Plot details
legend([l1, l2, l3], 'L^1', 'L^2', 'Actual Line')
title('L^1 and L^2 fitting of data')
```