```
function m2 = regresionL1(x, y, sigma)

%
% The code that generated the data.
%
%beta0 = 10;
%beta1 = 100;
%beta2 = 9.8;
%for t=1:40
%x(t,1)=t;
%sigma(t,1)=8;
%y(t,1)=beta0+beta1*t-(1/2*beta2)*t^2+sigma(t,1)*randn;
%end

% set data length
N = length(x);

% build the parabolic system matrix
G = [ones(N,1) , x , -1/2*x.*x];

% apply the weighting
yw = y./sigma;

Gw = G./[sigma,sigma,sigma];

%solve for the least-squares solution
disp(['Least-squares solution'])
m2 = Gw\yw

disp(['1-norm solution'])
m1 = irls(Gw,yw,1.0e-5,1.0e-5,1,25)

m=m1;

%get the covariance matrix
ginv = inv(Gw'*Gw)*Gw';

disp(['Covariance matrix'])
covm = ginv*eye(N).^2*ginv'

% get the 1.96-sigma (95%) conf intervals
disp(['95% parameter confidence intervals on 2-norm solution'])
del = 1.96*sqrt(diag(covm));
[m-del , m2 , m+del]

disp(['Chi-square misfit'])

chi2 = norm((y - G*m2)./sigma)^2

%find the p-value for this data set
%degrees of freedom
dof = N-3;
disp(['Chi-square p-value'])
p = 1-chi2cdf(chi2,dof)

% find the parameter correlations
s=sqrt(diag(covm))
disp(['Parameter correlations'])
r = covm./(s*s')

% Plot the predicted data from the two models
% i = 1;
% for k = min(x)-1:.05:max(x)+1
%     xx(i) = k;
%     mm1(i) = m1(1) + k*m1(2) - 1/2*k^2*m1(3);
%     mm2(i) = m2(1) + k*m2(2) - 1/2*k^2*m2(3);
%     i = i + 1;
% end
```

```
% figure
% plot(xx,mm1,xx,mm2,'k');
% hold on
% errorbar(x,y,sigma,'ok');
% %title(['Data and Model'])
% xlabel('Time (s)');
% ylabel('Elevation (m)');
% %print -dpdf l1example.pdf
% hold off

%Monte Carlo Section, L1
y0 = G*m1;

nreal = 10000;

for j = 1:nreal
    %
    %generate a trial data set of perturbed, weighted data
    %
    ytrial = y0 + sigma.*randn(N,1);
    ywtrial = ytrial./sigma;

    %L1
    mmc(j,:) = irls(Gw,ywtrial,1.0e-5,1.0e-5,1,25)';

    %L2
    %mmc(j,:)=(Gw\ywtrial)';

    chimc(j) = norm((ytrial-y0)./sigma)^2;

end

% figure
% hist(chimc);
%title(['1000 Monte-Carlo Chi-square Values'])
%print -dpdf parabfig2.pdf

% figure
% subplot(1,3,1)
% hist(mmc(:,1))
% title(['m_1'])
%
% subplot(1,3,2)
% hist(mmc(:,2))
% title(['m_2'])
%
% subplot(1,3,3)
% hist(mmc(:,3))
% title(['m_3'])
%print -dpdf parabfig3.pdf

%empirical covariance estimation
covmemp = mmc - [mean(mmc(:,1))*ones(nreal,1), ...
    mean(mmc(:,2))*ones(nreal,1),mean(mmc(:,3))*ones(nreal,1)];
covmemp = (covmemp'*covmemp)/nreal

% figure
% subplot(1,3,1)
% plot(mmc(:,1),mmc(:,2),'b*')
% hold on
% xlabel('M_1')
% ylabel('M_2')
%
% subplot(1,3,2)
% plot(mmc(:,1),mmc(:,3),'b*')
% hold on
% xlabel('M_1')
```

```
% ylabel('M_3')
%
% subplot(1,3,3)
% plot(mmc(:,2),mmc(:,3),'b*')
% xlabel('M_2')
% ylabel('M_3')
% hold on
%
% %plot the error ellipsoid
% %get the sub-axes of the ellipsoid for plotting
% n1 = [0;0;1];
% n2 = [0;1;0];
% n3 = [1;0;0];
% vp1 = (eye(3) - n1*n1')*covmemp*(eye(3)-n1*n1');
% vp2 = (eye(3) - n2*n2')*covmemp*(eye(3)-n2*n2');
% vp3 = (eye(3) - n3*n3')*covmemp*(eye(3)-n3*n3');
%
% vp1 = vp1(1:2,1:2);
% vp2 = [vp2(1,1),vp2(1,3);vp2(3,1),vp2(3,3)];
% vp3 = vp3(2:3,2:3);
%
% [u1,lam1] = eig(vp1);
% [u2,lam2] = eig(vp2);
% [u3,lam3] = eig(vp3);
%
% l1 = sqrt(diag(lam1));
% l2 = sqrt(diag(lam2));
% l3 = sqrt(diag(lam3));
% dtheta = 0.01;
% theta = 0:dtheta:2*pi;
% for i = 1:length(theta)
%     v1 = 2.79*(l1(1)*u1(:,1)*cos(theta(i))+l1(2)*u1(:,2)*sin(theta(i)));
%     v2 = 2.79*(l2(1)*u2(:,1)*cos(theta(i))+l2(2)*u2(:,2)*sin(theta(i)));
%     v3 = 2.79*(l3(1)*u3(:,1)*cos(theta(i))+l3(2)*u3(:,2)*sin(theta(i)));
%     x1(i) = m(1) + v1(1);
%     y1(i) = m(2) + v1(2);
%     x2(i) = m(1) + v2(1);
%     z2(i) = m(3) + v2(2);
%     y3(i) = m(2) + v3(1);
%     z3(i) = m(3) + v3(2);
% end
%
% subplot(1,3,1)
% plot(x1,y1,'k')
% hold off
%
% subplot(1,3,2)
% plot(x2,z2,'k')
% hold off
%
% subplot(1,3,3)
% plot(y3,z3,'k')
% hold off

%check confidence intervals

[u, lam] = eig(covm);

%rotate model estimates into the principal coordinate system
mmcrot = mmc*u;
%subtract the (rotated) parameter means
mmcrotmean = mean(mmcrot);
mmcrot = mmcrot- ...
    [mmcrotmean(1)*ones(nreal,1), mmcrotmean(2)*ones(nreal,1), ...
    mmcrotmean(3)*ones(nreal,1)];

%counting of model estimates within the ellipsoid
count = 0;
```

```
for i=1:nreal
    if ((mmcrot(i,1)^2/lam(1,1) + mmcrot(i,2)^2/lam(2,2) + ...
            mmcrot(i,3)^2/lam(3,3)) <= 3.35^2)
        count=count+1;
    end
end
disp(['confidence interval inclusion:'])
count/nreal

%generate confidence ellipsoids to plot on top of the model distribution
% theta=(0:.01:2*pi);
% elrot1 = 2.79*[lam(1,1)^0.5*cos(theta)',lam(2,2)^0.5*sin(theta)'];
% elrot2 = 2.79*[lam(1,1)^0.5*cos(theta)',lam(3,3)^0.5*sin(theta)'];
% elrot3 = 2.79*[lam(2,2)^0.5*cos(theta)',lam(3,3)^0.5*sin(theta)'];

% figure
% subplot(1,3,1)
% plot(mmcrot(:,1),mmcrot(:,2),'*',elrot1(:,1),elrot1(:,2),'k')
% xlabel('M_1')
% ylabel('M_2')
%
% subplot(1,3,2)
% plot(mmcrot(:,1),mmcrot(:,3),'*',elrot2(:,1),elrot2(:,2),'k')
% xlabel('M_1')
% ylabel('M_3')
%
% subplot(1,3,3)
% plot(mmcrot(:,2),mmcrot(:,3),'*',elrot3(:,1),elrot3(:,2),'k')
% xlabel('M_2')
% ylabel('M_3')
```