```
function z = LinearDataFittingXY()
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% Fitting a straight line with errors in x and y
%
% This Matlab-program was developed using "Matlab_R2014b" on a MacBook Pro
% running OS X 10.10.1
%
% written by T. Ihn, D-PHYS ETH Zurich, 28 Nov 2014
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%
% The data
%

% Importing the data
data = csvread('errorxydata.csv');
x = data(:,1);
y = data(:,2);

% Displaying the data as a table
disp(table(data(:,1),data(:,2),'VariableNames',{'x','y'}))

% Error bars
sigmax = 0.5;
sigmay = 0.8;

% Statistics of the data
fprintf('Statistics of the data:\n');
n = length(data);
fprintf('N = %1u\n',n);
Meanx = mean(x);
fprintf('<x> = %1.2f\n',Meanx);
Meany = mean(y);
fprintf('<y> = %1.2f\n',Meany);
Varx = var(x,1);
fprintf('Var(x) = %1.2f\n',Varx);
Vary = var(y,1);
fprintf('Var(y) = %1.2f\n',Vary);
rho = corrcoef(x,y);
rho = rho(1,2);
fprintf('rho = %1.4f\n',rho);
disp(' ');

%
% Define the -log-posterior for a and R
%
function y=posteriorpdf(parms,sx,sy,rho,n)
    a = parms(1);
    R = parms(2);
    num = R*R*a*a - 2*R*R*rho*a + (sx*sx+sy*sy)*a + R*R;
    denom =  sx*sx*R*R*a*a + sx*sx*sy*sy*a + R*R*sy*sy;
    p = num/denom;
    q = denom/a;
    y = log(R*a) + (n/2)*log(q) + (n/2)*p;
end
```

```
% Minimization of the -log-posterior
[pp,Qmin]=fminsearch(@(parms) posteriorpdf(parms,sigmax/sqrt(Varx),sigmay/sqrt(Vary),rho,↙
n),[1 1]);
aa = pp(1);
RR = pp(2);

% plot the -log-posterior
ap = linspace(0.935,1.07,128);
Rp = linspace(0.68,1.52);
Q = zeros(length(Rp),length(ap));
[X,Y] = meshgrid(ap,Rp);
for m=1:length(Rp),
    for j=1:length(ap),
        Q(m,j)=posteriorpdf([ap(j) Rp(m)],sigmax/sqrt(Varx),sigmay/sqrt(Vary),rho,n)-↙
Qmin;
    end
end
figure(2)
pcolor(X,Y,Q)
shading interp
hold on
plot(ap,RR*ones(size(ap)),'w--');
plot(aa*ones(size(Rp)),Rp,'w--');
xlabel('a')
ylabel('R')
title('-log-posterior distribution')
colorbar
[C,h]=contour(X,Y,Q,[1 2 3],'Linewidth',2,'LineColor','w');
clabel(C,h);
plot(aa,RR,'wo','MarkerFaceColor','w')
hold off

% We determine the standard error of a from the contour lines
CC = C(:,2:(C(2,1)+1)); % This are the coordinates of the relevant contour line
sap = max(CC(1,:))-aa; % the maximum in g-direction
sam = aa-min(CC(1,:)); % the minimum in g-direction

fprintf('Scaled slope Parameter:\n')
fprintf('a = %1.3f + %1.3f/-%1.3f\n\n',aa,sap,sam)

% Calculating the slope and the intercept parameters
aest = aa*sqrt(Vary/Varx);
sap1 = sap*sqrt(Vary/Varx);
sam1 = sam*sqrt(Vary/Varx);
fprintf('Estimated parameters:\n')
fprintf('a = %1.3f +%1.3f/-%1.3f\n',aest,sap1,sam1);

best = Meany - Meanx*aest;
bstd = sqrt( (sigmay*sigmay+sigmax*sigmax*aest)/n + Meanx*Meanx*sam1*sap1);
fprintf('b = %1.3f +/-%1.3f\n',best,bstd);

% Plot data with error bars and fitted line
% Uses the function errorbarxy by Qi An downloaded from MatlabCentral file exchange
figure(1)
errorbarxy(x,y,sigmax*ones(size(x)),sigmay*ones(size(x)),{'o','b','b'})
hold on
xfit=linspace(min(x),max(x));
yfit = aest*xfit + best;
```

```
plot(xfit,yfit,'r','LineWidth',2)
xlabel('x')
ylabel('y')
title('fitted data with errors in x and y')
hold off
end
```