load precomputed data

load data1 t=data1(:,1); y=data1(:,2);% + 30*randn(length(t),1); sigma=data1(:,3);

Contents

Para otros datos.

```
load dataCO2.txt
```

Se asignan variables a las columnas de los datos:

```
t = dataCO2(:,1);
y = dataCO2(:,2);
sigma = dataCO2(:,3);
N = length(t);

%use all data
% N = length(t);
```

use first 10 data

N = 10; t=t(1:N); y=y(1:N); sigma=sigma(1:N);

```
disp('displaying t,y,sigma')
[t , y , sigma]

%add an outlier
%y(4)=y(4)-300;
```

```
displaying t,y,sigma

ans =

       1959.00        315.97          0.12
       1960.00        316.91          0.12
       1961.00        317.64          0.12
       1962.00        318.45          0.12
       1963.00        318.99          0.12
       1964.00        319.62          0.12
       1965.00        320.04          0.12
```

```
1966.00         321.38          0.12
1967.00         322.16          0.12
1968.00         323.04          0.12
1969.00         324.62          0.12
1970.00         325.68          0.12
1971.00         326.32          0.12
1972.00         327.45          0.12
1973.00         329.68          0.12
1974.00         330.18          0.12
1975.00         331.08          0.12
1976.00         332.05          0.12
1977.00         333.78          0.12
1978.00         335.41          0.12
1979.00         336.78          0.12
1980.00         338.68          0.12
1981.00         340.10          0.12
1982.00         341.44          0.12
1983.00         343.03          0.12
1984.00         344.58          0.12
1985.00         346.04          0.12
1986.00         347.39          0.12
1987.00         349.16          0.12
1988.00         351.56          0.12
1989.00         353.07          0.12
1990.00         354.35          0.12
1991.00         355.57          0.12
1992.00         356.38          0.12
1993.00         357.07          0.12
1994.00         358.82          0.12
1995.00         360.80          0.12
1996.00         362.59          0.12
1997.00         363.71          0.12
1998.00         366.65          0.12
1999.00         368.33          0.12
2000.00         369.52          0.12
2001.00         371.13          0.12
2002.00         373.22          0.12
2003.00         375.77          0.12
2004.00         377.49          0.12
2005.00         379.80          0.12
2006.00         381.90          0.12
2007.00         383.77          0.12
2008.00         385.59          0.12
2009.00         387.38          0.12
2010.00         389.78          0.12
2011.00         391.57          0.12
```

build the parabolic system matrix

```
G = [ ones(N,1) , t , -1/2*t.*t ];
```

apply the weighting

```
yw = y./sigma;

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

solve for the least-squares solution

```
disp(['Least-squares solution'])
m = Gw\yw
```

```
Least-squares solution

m =

      44040.06
        -45.50
         -0.02
```

get the covariance matrix

```
ginv = inv(Gw'*Gw)*Gw';

disp(['Covariance matrix'])
covm = ginv*ginv'
```

```
Warning: Matrix is close to singular or badly scaled.
         Results may be inaccurate. RCOND = 7.244936e-22.
Covariance matrix

covm =

      96391.83        -97.12         -0.05
        -97.12          0.10          0.00
         -0.05          0.00          0.00
```

get the 1.96-sigma (95%) conf intervals

```
disp(['95% parameter confidence intervals (m-, mest, m+)'])
del = 1.96*sqrt(diag(covm));
[m-del , m , m+del]

dof = N-3;
disp(['Chi-square misfit for ',num2str(dof),' dof'])
chi2 = norm((y - G*m)./sigma)^2


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

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

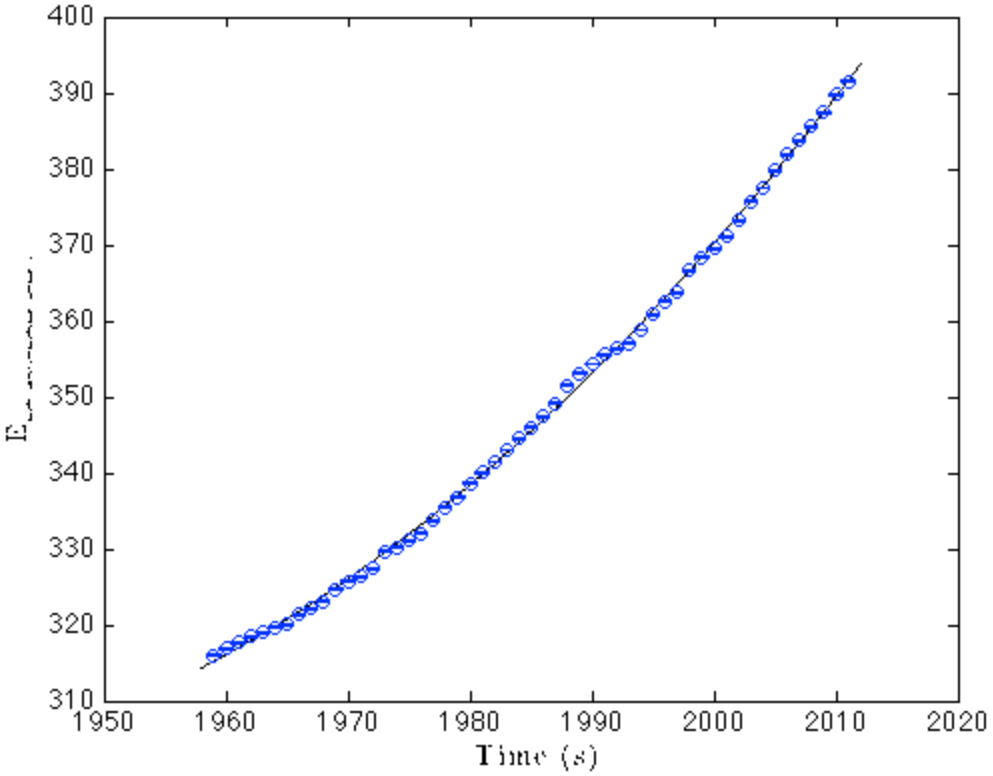
```
95% parameter confidence intervals (m-, mest, m+)

ans =

      43431.54      44040.06      44648.58
        -46.11        -45.50        -44.88
         -0.02         -0.02         -0.02

Chi-square misfit for 50 dof

chi2 =

       1584.52

chi-square p-value

p =

             0


s =

        310.47
          0.31
          0.00

correlation matrix

r =

          1.00         -1.00         -1.00
         -1.00          1.00          1.00
         -1.00          1.00          1.00
```

Plot the data and model

```
xx=min(t)-1:0.05:max(t)+1;
mm=m(1)+m(2)*xx-0.5*m(3)*xx.^2;

figure(1)
bookfonts;
plot(xx,mm,'k');
hold on
% plot(t,y,'*')
errorbar(t,y,sigma,'o');
```

```
xlabel('Time (s)','interpreter','latex');
ylabel('Elevation (m)','interpreter','latex');
disp(['displaying data and model fit'])
hold off
% print -dpdf parabfig1.pdf
```

```
displaying data and model fit
```



Monte Carlo Section

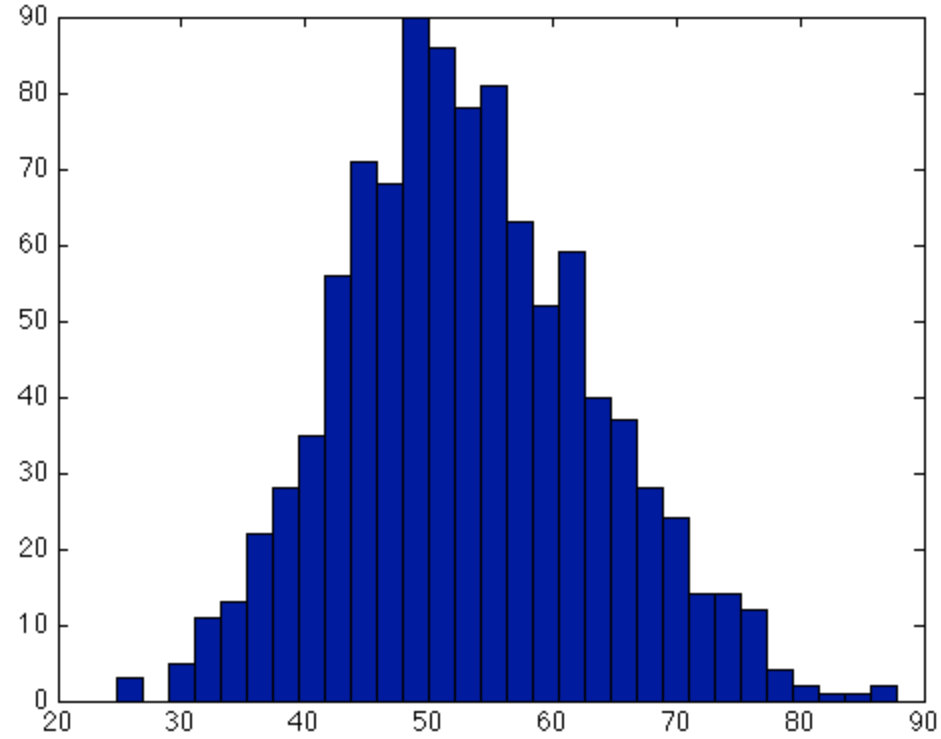
```
y0 = G*m;

for nreal = 1:1000
%
%generate a trial data set of perturbed, weighted data
%
  ytrial = y0+sigma.*randn(N,1);
  ywtrial=ytrial./sigma;
  mmc(nreal,:)=(Gw\ywtrial)';
  chimc(nreal)= norm((ytrial-y0)./sigma)^2;
end

figure(2)
bookfonts;
hist(chimc,30);
%title(['1000 Monte-Carlo Chi-square Values'])
disp(['Displaying 1000 Monte-Carlo Chi-square Values'])
```

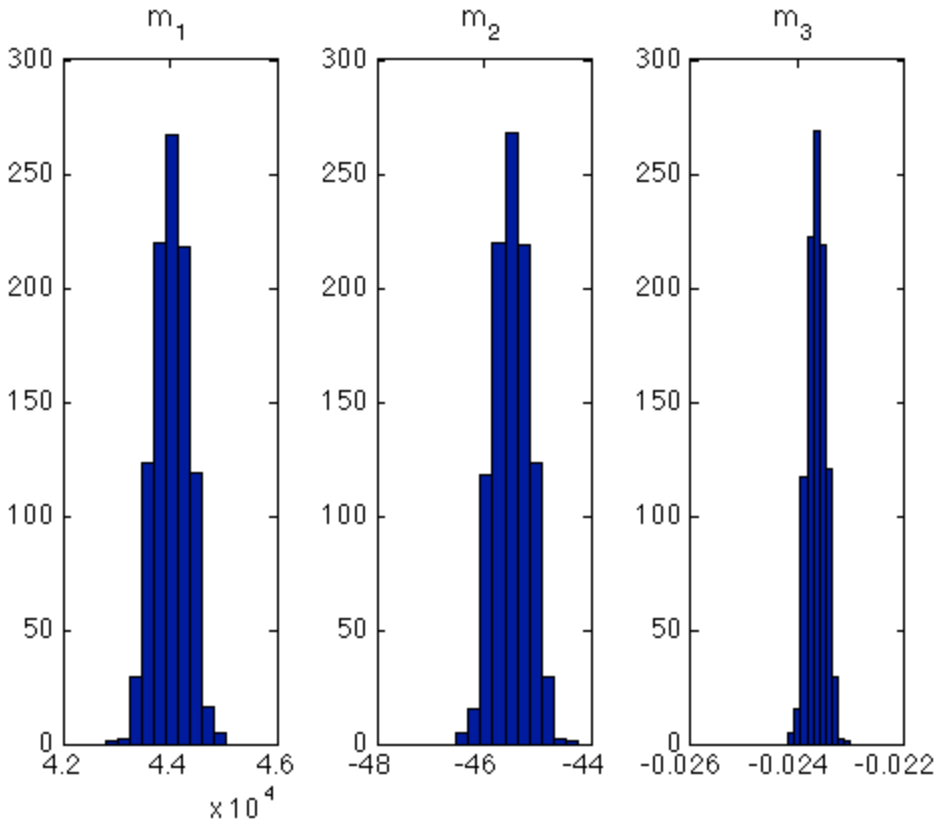
```
%print -deps parabfig2.eps

figure(3)

subplot(1,3,1)
bookfonts;
hist(mmc(:,1))
title(['m_1'])

subplot(1,3,2)
bookfonts;
hist(mmc(:,2))
title(['m_2'])

subplot(1,3,3)
bookfonts;
hist(mmc(:,3))
title(['m_3'])
%print -deps parabfig3.eps

figure(4)
subplot(1,3,1)
bookfonts;
plot(mmc(:,1),mmc(:,2),'k*')
xlabel('M_1')
ylabel('M_2')
subplot(1,3,2)
bookfonts;
plot(mmc(:,1),mmc(:,3),'k*')
xlabel('M_1')
ylabel('M_3')
subplot(1,3,3)
bookfonts;
plot(mmc(:,2),mmc(:,3),'k*')
xlabel('M_2')
ylabel('M_3')
disp(['Displaying 1000 Monte-Carlo models'])
%
% Plot the ellipses.
%
figure(5);
clf;
%
% Do the m1, m2 ellipsoid.
%
C=covm([1:2],[1:2]);
[u,lam]=eig(inv(C));
deltachisq=chi2inv(0.95,2);
delta=sqrt(deltachisq);
%generate a vector of angles from 0 to 2*pi
theta=(0:.01:2*pi)';
%calculate the x component of the ellipsoid for all angles
r=zeros(length(theta),2);
r(:,1)=(delta/sqrt(lam(1,1)))*u(1,1)*cos(theta)+(delta/sqrt(lam(2,2)))*u(1,2)*sin(th
%calculate the y component of the ellipsoid for all angles
r(:,2)=(delta/sqrt(lam(1,1)))*u(2,1)*cos(theta)+(delta/sqrt(lam(2,2)))*u(2,2)*sin(th
```

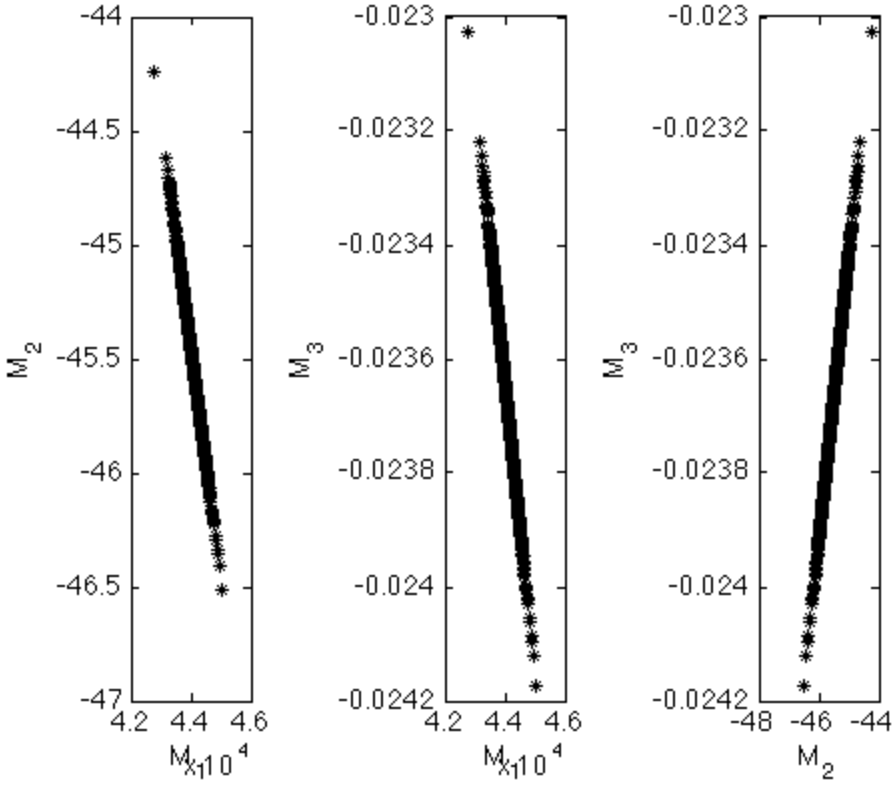
```
%plot(x,y), adding in the model parameters
subplot(1,3,1)
bookfonts;
plot(m(1)+r(:,1),m(2)+r(:,2),'k');
fill(m(1)+r(:,1),m(2)+r(:,2),'y');
%axis([-50 50 85 110]);
xlabel('m_{1}');
ylabel('m_{2}');
m1max=max(r(:,1));
m1min=min(r(:,1));
m2max=max(r(:,2));
m2min=min(r(:,2));
%
% Do the m1, m3 ellipsoid.
%
C=covm([1,3],[1,3]);
[u,lam]=eig(inv(C));
deltachisq=chi2inv(0.95,2);
delta=sqrt(deltachisq);
%calculate the x component of the ellipsoid for all angles
r(:,1)=(delta/sqrt(lam(1,1)))*u(1,1)*cos(theta)+(delta/sqrt(lam(2,2)))*u(1,2)*sin(th
%calculate the y component of the ellipsoid for all angles
r(:,2)=(delta/sqrt(lam(1,1)))*u(2,1)*cos(theta)+(delta/sqrt(lam(2,2)))*u(2,2)*sin(th
%plot(x,y), adding in the model parameters
subplot(1,3,2)
bookfonts;
plot(m(1)+r(:,1),m(3)+r(:,2),'k');
fill(m(1)+r(:,1),m(3)+r(:,2),'y');
% axis([-50 50 7 12]);
xlabel('m_{1}');
ylabel('m_{3}');
m1max=max([m1max; r(:,1)]);
m1min=min([m1min; r(:,1)]);
m3max=max(r(:,2));
m3min=min(r(:,2));

%
% Do the m1, m3 ellipsoid.
%
C=covm([2,3],[2,3]);
[u,lam]=eig(inv(C));
deltachisq=chi2inv(0.95,2);
delta=sqrt(deltachisq);
%calculate the x component of the ellipsoid for all angles
r(:,1)=(delta/sqrt(lam(1,1)))*u(1,1)*cos(theta)+(delta/sqrt(lam(2,2)))*u(1,2)*sin(th
%calculate the y component of the ellipsoid for all angles
r(:,2)=(delta/sqrt(lam(1,1)))*u(2,1)*cos(theta)+(delta/sqrt(lam(2,2)))*u(2,2)*sin(th
%plot(x,y), adding in the model parameters
subplot(1,3,3)
bookfonts;
plot(m(2)+r(:,1),m(3)+r(:,2),'k');
fill(m(2)+r(:,1),m(3)+r(:,2),'y');
% axis([80 120 7 12]);
xlabel('m_{2}');
ylabel('m_{3}');
%print -deps parabfig5.eps
```

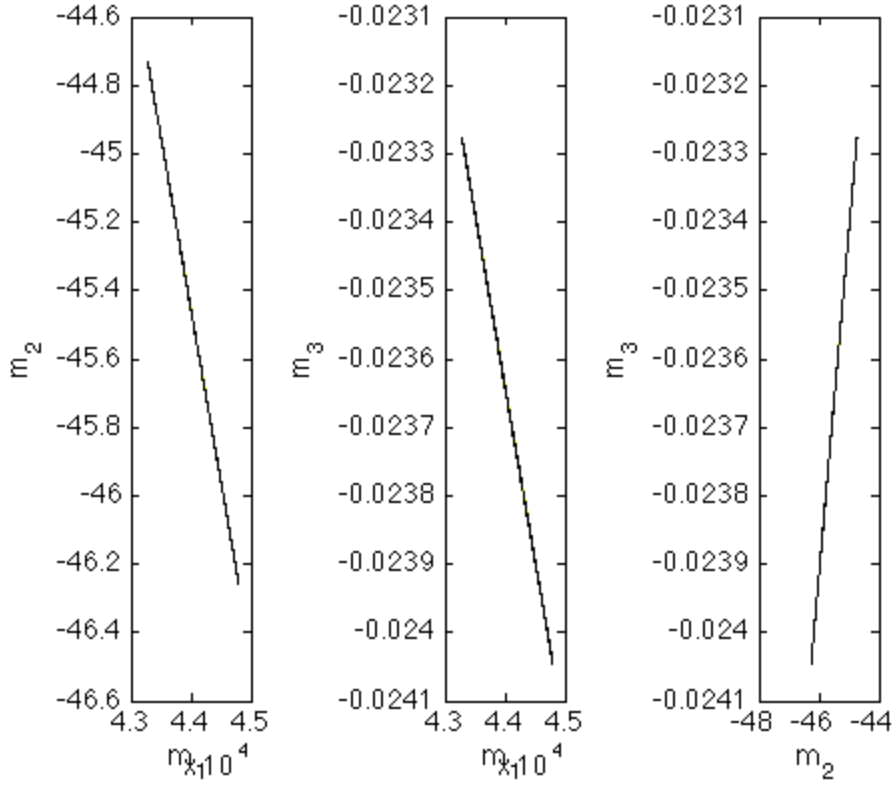
```
m2max=max([m2max; r(:,1)]);
m2min=min([m2min; r(:,1)]);
m3max=max([m3max; r(:,2)]);
m3min=min([m3min; r(:,2)]);

covm
[u,lam]=eig(inv(covm))
```

```
Displaying 1000 Monte-Carlo Chi-square Values
Displaying 1000 Monte-Carlo models
Warning: Matrix is close to singular or badly scaled.
         Results may be inaccurate. RCOND = 1.222998e-17.

covm =

      96391.83        -97.12         -0.05
        -97.12          0.10          0.00
         -0.05          0.00          0.00

Warning: Matrix is close to singular or badly scaled.
         Results may be inaccurate. RCOND = 7.244938e-22.

u =

         -1.00         -0.00         -0.00
          0.00         -1.00         -0.00
          0.00         -0.00          1.00


lam =

          0.00             0             0
             0     860881.83             0
             0             0 14290599501071938.00
```


90
80
70
60
50
40
30
20
10
0
20
30
40
50
60
70
80
90


m1
m2
m3
300
250
200
150
100
50
0
4.2
4.4
4.6
x 10^4
-48
-46
-44
-0.026
-0.024
-0.022





Published with MATLAB® 7.12