Regression Analysis with SCILAB

By

Gilberto E. Urroz, Ph.D., P.E.

Distributed by

*i**nfoClearinghouse**.com*

A "zip" file containing all of the programs in this document (and other SCILAB documents at InfoClearinghouse.com) can be downloaded at the following site:

http://www.engineering.usu.edu/cee/faculty/gurro/Software_Calculators/Scilab_Docs/ScilabBookFunctions.zip

The author's SCILAB web page can be accessed at:

http://www.engineering.usu.edu/cee/faculty/gurro/Scilab.html

Please report any errors in this document to: gurro@cc.usu.edu

Regression Analysis

The idea behind regression analysis is to verify that a function

$$\hat{y} = f(\mathbf{x})$$

fits a given data set

$$\{(\mathbf{x}_1,y_1), (\mathbf{x}_2,y_2),\ldots,(\mathbf{x}_n,y_n)\}$$

after obtaining the parameters that identify function $f(\mathbf{x})$. The value $\mathbf{x}$ represents one or more independent variables. The function $f(\mathbf{x})$ can be, for example, a linear function, i.e.,

$$\hat{y} = mx+b, \text{ or } \hat{y} = b_0 + b_1x_1 + \ldots + b_kx_k),$$

a polynomial function, i.e.,

$$\hat{y} = b_0 + b_1x_1 + \ldots + b_px^p,$$

or other non-linear functions. The procedure consists in postulating a form of the function to be fitted, $\hat{y} = f(\mathbf{x})$, which will depend, in general, of a number of parameters, say $\{b_0, b_1, \ldots, b_k\}$. Then we choose a criteria to determine the values of those parameters. The most commonly used is the *least-square criteria*, by which the *sum of the squares of the errors* (*SSE*) involved in the data fitting is minimized. The error involved in fitting point i in the data set is given by

$$e_i = y_i - \hat{y}_i ,$$

thus, the quantity to be minimized is

$$SSE = \sum_{i=1}^{n} e_i^2 = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2.$$

Minimization of SSE is accomplished by taking the derivatives of *SSE* with respect to each of the parameters, $b_0, b_1, \ldots, b_k$, and setting these results to zero, i.e., $\partial(SSE)/\partial b_0 = 0, \partial(SSE)/\partial b_0 = 0, \ldots, \partial(SSE)/\partial b_k = 0$. The resulting set of equations is then solved for the values $b_0, b_1, \ldots, b_k$.

After finding the parameters by minimization of the sum of square errors (*SSE*), we can test hypotheses about those parameters under certain confidence levels to complete the regression analysis. In the following section we present the regression analysis of a simple linear regression.

Simple linear regression

Consider the data set represented in the figure below and represented by the set $\{(x_1,y_1), (x_2,y_2),\ldots,(x_n,y_n)\}$. Suppose that the equation

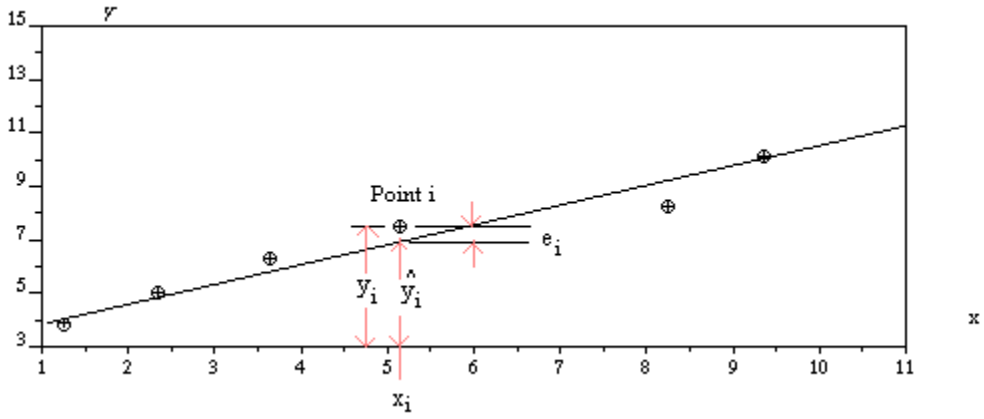
$$\hat{y} = mx+b,$$

representing a straight line in the *x*-*y* plane is used to represent the relationship between the values *x* and *y* from the set. The fitted value of *y* corresponding to point x_i is

$$\hat{y}_i = mx_i+b,$$

and the corresponding error in the fitting is

$$e_i = y_i - \hat{y}_i = y_i-(mx_i+b) = y_i-mx_i-b.$$



The sum of square errors (*SSE*) to be minimized is

$$SSE(m,b) = \sum_{i=1}^{n}(y_i - mx_i - b)^2.$$

To determine the values of *m* and *b* that minimize the sum of square errors, we use the conditions

$$\frac{\partial}{\partial a}(SSE) = 0 \qquad \frac{\partial}{\partial b}(SSE) = 0$$

from which we get the so-called *normal equations*:

$$\sum_{i=1}^{n} y_i = b \cdot n + m \cdot \sum_{i=1}^{n} x_i$$

$$\sum_{i=1}^{n} x_i \cdot y_i = b \cdot \sum_{i=1}^{n} x_i + m \cdot \sum_{i=1}^{n} x_i^2$$

This is a system of linear equations with *m* and *b* as the unknowns. In matricial form, these equations are written as

$$\begin{bmatrix} \sum_{i=1}^{n} x_i & n \\ \sum_{i=1}^{n} x_i^2 & \sum_{i=1}^{n} x_i \end{bmatrix} \begin{bmatrix} m \\ b \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{n} y_i \\ \sum_{i=1}^{n} x_i y_i \end{bmatrix}.$$

For example, consider the data set given in the following table

x	*1.2*	*2.5*	*4.3*	*8.3*	*11.6*
y	*6.05*	*11.6*	*15.8*	*21.8*	*36.8*

The following SCILAB commands will calculate the values of *m* and *b* to minimize *SSE*. A plot of the original data and the straight line fitting is also produced. The column vector *p* stores the values of *m* and *b*. Thus, for this case *m=2.6827095* and *b=3.4404811*.

```
-->x=[1.2,2.5,4.3,8.3,11.6];y=[6.05,11.6,15.8,21.8,36.8];

-->Sx=sum(x);Sx2=sum(x^2);Sy=sum(y);Sxy=sum(x.*y);n=length(x);

-->A=[Sx,n;Sx2,Sx];B=[Sy;Sxy];p=A\B

 p  =

!   2.6827095 !
!   3.4404811 !

-->deff('[y]=yh(x)','y=p(1).*x+p(2)')

-->plot2d(xf,yf,1,'011',' ',rect)

-->plot2d(x,y,-1,'011',' ',rect)

-->xtitle('Simple linear regression','x','y')
```

The value of the sum of square errors for this fitting is:

```
-->yhat=yh(x);err=y-yhat;SSE=sum(err^2)
 SSE  = 23.443412
```

To illustrate graphically the behavior of *SSE(m,b)* we use the following function *SSEPlot(mrange,brange,x,y)*, where *mrange* and *brange* are vectors with ranges of values of *m* and *b*, respectively, and *x* and *y* are the vectors with the original data.

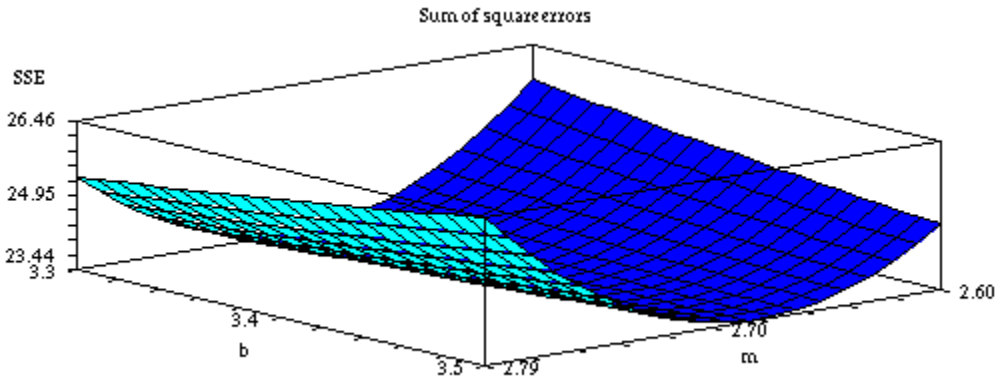
```
function [] = SSEPlot(mrange,brange,x,y)

n=length(mrange); m=length(brange);
SSE = zeros(n,m);
deff('[y]=f(x)','y=slope*x+intercept')
for i = 1:n
    for j = 1:m
        slope = mrange(i);intercept=brange(j);
        yhat  = f(x);err=y-yhat;SSE(i,j)=sum(err^2);
    end;
end;
```



```
xset('window',1);plot3d(mrange,brange,SSE,45,45,'m@b@SSE');
xtitle('Sum of square errors')
xset('window',2);contour(mrange,brange,SSE,10);
xtitle('Sum of square errors','m','b');
```

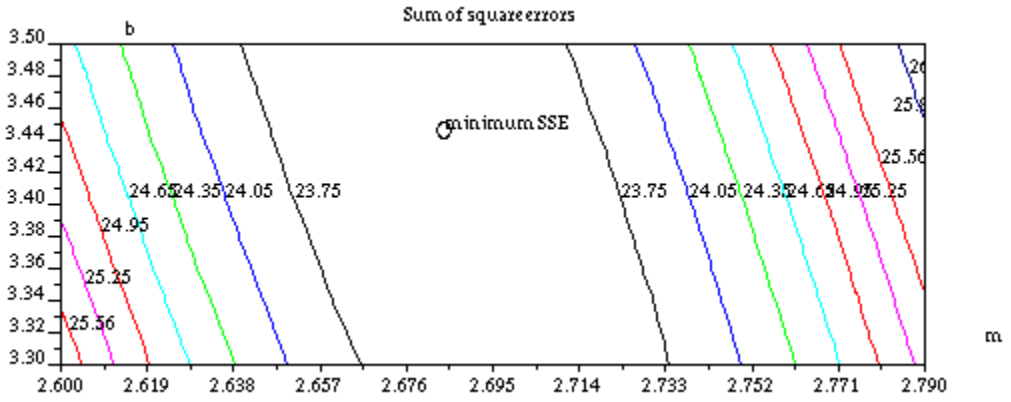
The function produces a three-dimensional plot of *SSE(m,b)* as well as a contour plot of the function. To produce the plots we use the following SCILAB commands:

```
-->mr = [2.6:0.01:2.8];br=[3.3:0.01:3.5];

-->getf('SSEPlot')

-->SSEPlot(mr,br,x,y)
```



The following two lines modify the contour plot.

```
-->plot2d([p(1)],[p(2)],-9,'011',' ',[2.600 3.30 2.790 3.50])

-->xstring(p(1)+0.002,p(2)+0.002,'minimum SSE')
```



Covariance and Correlation

The concepts of covariance and correlation were introduced in Chapter 14 in relation to bivariate random variables. For a sample of data points *(x,y),* such as the one used for the linear regression in the previous section, we define *covariance* as

$$s_{xy} = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})$$

The *sample correlation coefficient* for x,y is defined as

$$r_{xy} = \frac{s_{xy}}{s_x \cdot s_y}$$

where s_x, s_y are the standard deviations of x and y, respectively, i.e.

$$s_x^2 = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2$$

$$s_y^2 = \frac{1}{n-1}\sum_{i=1}^{n}(y_i - \bar{y})^2$$

The correlation coefficient is a measure of how well the fitting equation, i.e., ^*y* = *mx*+*b*, fits the given data. The values of r_{xy} are constrained in the interval *(-1,1).* The closer the value of r_{xy} is to *+1* or *-1*, the better the linear fitting for the given data.

Additional equations and definitions

Let's define the following quantities:

$$S_{xx} = \sum_{i=1}^{n}(x_i - \bar{x})^2 = (n-1)\cdot s_x^2 = \sum_{i=1}^{n} x_i^{\,2} - \frac{1}{n}\left(\sum_{i=1}^{n} x_i\right)^2$$

$$S_{yy} = \sum_{i=1}^{n}(y_i - \bar{y})^2 = (n-1)\cdot s_y^2 = \sum_{i=1}^{n} y_i^{\,2} - \frac{1}{n}\left(\sum_{i=1}^{n} y_i\right)^2$$

$$S_{xy} = \sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})^2 = (n-1)\cdot s_{xy} = \sum_{i=1}^{n} x_i y_i - \frac{1}{n}\left(\sum_{i=1}^{n} x_i\right)\left(\sum_{i=1}^{n} y_i\right)$$

From which it follows that the *standard deviations* of x and y, and the *covariance* of x,y are given, respectively, by

$$s_x = \sqrt{\frac{S_{xx}}{n-1}}$$

$$s_y = \sqrt{\frac{S_{yy}}{n-1}}$$

$$s_{xy} = \frac{S_{xy}}{n-1}$$

Also, the *sample correlation coefficient* is

$$r_{xy} = \frac{S_{xy}}{\sqrt{S_{xx} \cdot S_{yy}}}.$$

In terms of $\bar{x}$, $\bar{y}$, S_{xx}, S_{yy}, and S_{xy}, the solution to the normal equations is:

$$m = \frac{S_{xy}}{S_{xx}} = \frac{s_{xy}}{s_x^2}$$

$$b = \bar{y} - m\bar{x}$$

Standard error of the estimate

The function $\hat{y}_i = mx_i + b$ in a linear fitting is an approximation to the regression curve of a random variable Y on a random variable X,

$$Y = Mx + B + \varepsilon,$$

where ε is a random error. If we have a set of n data points (x_i, y_i), then we can write

$$Y_i = Mx_i + B + \varepsilon_i,$$

$i = 1,2,\ldots,n$, where Y_i are independent, normally distributed random variables with mean $(\alpha + \beta x_i)$ and common variance σ^2, and ε_i are independent, normally distributed random variables with mean zero and the common variance σ^2.

Let y_i = actual data value, $\hat{y}_i = mx_i + b$ = least-square prediction of the data. Then, the *prediction error* is:

$$e_i = y_i - \hat{y}_i = y_i - (mx_i + b).$$

The prediction error being an estimate of the regression error ε, an estimate of σ^2 is the so-called **standard error of the estimate**,

$$s_e^2 = \frac{1}{n-2}\sum_{i=1}^{n}[y_i - (mx_i + b)]^2 = \frac{SSE}{n-2} = \frac{S_{yy} - (S_{xy})^2 / S_{xx}}{n-2} = \frac{n-1}{n-2} \cdot s_y^2 \cdot (1 - r_{xy}^2)$$

A function for calculating linear regression of two variables

Using the definitions provided in this section, we can use the following user-defined function, *linreg*, to calculate the different parameters of a simple linear regression. The function returns the slope, m, and intercept, b, of the linear function, the covariance, s_{xy}, the correlation coefficient, r_{xy}, the mean and standard deviations of x and y ($\bar{x}, s_x$, $\bar{y}, s_y$), and the standard error of the estimate, s_e. The function also produces a plot of the original data and of the fitted equation. A listing of the function follows:

```
function [rxy,sxy,slope,intercept]=linreg(x,y)

n=length(x);m=length(y);
if m<>n then
   error('linreg - Vectors x and y are not of the same length.');
   abort;
end;

Sxx       = sum(x^2)-sum(x)^2/n;
Syy       = sum(y^2)-sum(y)^2/n;
Sxy       = sum(x.*y)-sum(x)*sum(y)/n;
sx        = sqrt(Sxx/(n-1));
sy        = sqrt(Syy/(n-1));
sxy       = Sxy/(n-1);
rxy       = Sxy/sqrt(Sxx*Syy);
xbar      = mean(x);
ybar      = mean(y);
slope     = Sxy/Sxx;
intercept = ybar - slope*xbar;
se        = sqrt((n-1)*sy^2*(1-rxy^2)/(n-2));

xmin   = min(x);
xmax   = max(x);
xrange = xmax-xmin;
xmin   = xmin - xrange/10;
xmax   = xmax + xrange/10;
xx     = [xmin:(xmax-xmin)/100:xmax];
deff('[y]=yhat(x)','y=slope*x+intercept');
```



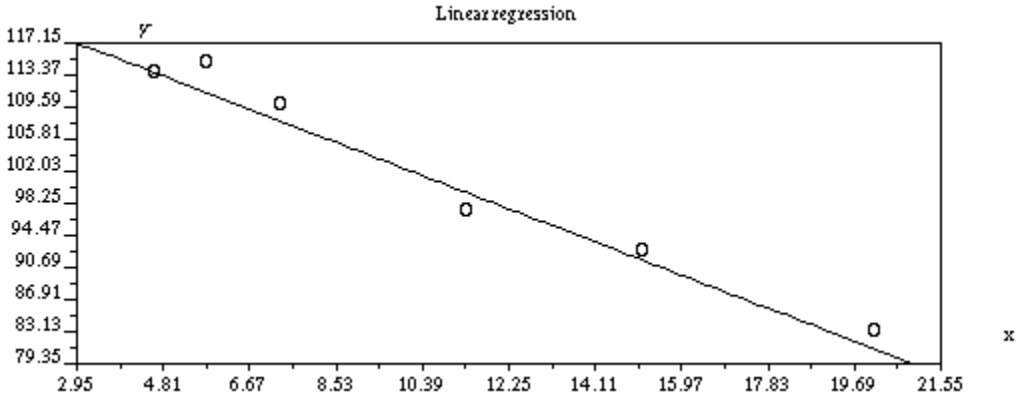
```
yy     = yhat(xx);
ymin   = min(y);
ymax   = max(y);
yrange = ymax - ymin;
ymin   = ymin - yrange/10;
ymax   = ymax + yrange/10;
rect   = [xmin ymin xmax ymax];
plot2d(xx,yy,1,'011',' ',rect);
xset('mark',-9,1);
plot2d( x, y,-9,'011',' ',rect);
xtitle('Linear regression','x','y');
```

As an example, we will use the following data set:

x	*4.5*	*5.6*	*7.2*	*11.2*	*15*	*20*
y	*113*	*114*	*109*	*96.5*	*91.9*	*82.5*

First we enter the data into vectors *x* and *y*, and then call function *linreg*.

```
-->x=[4.5,5.6,7.2,11.2,15,20];y=[113,114,109,96.5,91.9,82.5];

-->[rxy,sxy,slope,intercept] = linreg(x,y)
 xbar  =  10.583333
 ybar  = 101.15
 sx  = 6.0307269
 sy  = 12.823221
 intercept  = 123.38335
 slope  =  - 2.1007891
 sxy  =  - 76.405
 rxy  = -  .9879955
```



The correlation coefficient r_{xy} = *-0.9879955* corresponds to a decreasing linear function. The fact that the value of the correlation coefficient is close to -1 suggest a good linear fitting.

Confidence intervals and hypothesis testing in linear regression

The values *m* and *b* in the linear fitting $\hat{y}_i = mx_i+b$ are approximations to the parameters *M* and *B* in the regression curve

$$Y = Mx + B + \varepsilon.$$

Therefore, we can produce confidence intervals for the parameters *M* and *B* for a confidence level α. We can also perform hypotheses testing on specific values of the parameters.

- *Confidence limits for regression coefficients*:

For the slope (M):

$$m - (t_{n-2,\alpha/2}) \cdot s_e / \sqrt{S_{xx}} < M < m + (t_{n-2,\alpha/2}) \cdot s_e / \sqrt{S_{xx}},$$

For the intercept (B):

$$b - (t_{n-2,\alpha/2}) \cdot s_e \cdot [(1/n) + \bar{x}^2 / S_{xx}]^{1/2} < B < b + (t_{n-2,\alpha/2}) \cdot s_e \cdot [(1/n) + \bar{x}^2 / S_{xx}]^{1/2},$$

where t follows the Student's t distribution with $\nu = n - 2$, degrees of freedom, and n represents the number of points in the sample.

- *Confidence interval for the mean value of Y at $x = x_0$, i.e., $mx_0 + b$*:

$$[mx_0 + b - (t_{n-2,\alpha/2}) \cdot s_e \cdot [(1/n) + (x_0 - \bar{x})^2 / S_{xx}]^{1/2} ; mx_0 + b + (t_{n-2,\alpha/2}) \cdot s_e \cdot [(1/n) + (x_0 - \bar{x})^2 / S_{xx}]^{1/2}]$$

- *Limits of prediction: confidence interval for the predicted value $Y_0 = Y(x_0)$:*

$$[mx_0 + b - (t_{n-2,\alpha/2}) \cdot s_e \cdot [1 + (1/n) + (x_0 - \bar{x})^2 / S_{xx}]^{1/2} ; mx_0 + b + (t_{n-2,\alpha/2}) \cdot s_e \cdot [1 + (1/n) + (x_0 - \bar{x})^2 / S_{xx}]^{1/2}]$$

- *Hypothesis testing on the slope, M:*

The null hypothesis, H_0: $M = M_0$, is tested against the alternative hypothesis, H_1: $M \neq M_0$. The test statistic is

$$t_0 = (m - M_0) / (s_e / \sqrt{S_{xx}}),$$

where t follows the Student's t distribution with $\nu = n - 2$, degrees of freedom, and *n* represents the number of points in the sample. The test is carried out as that of a mean value hypothesis testing, i.e., given the level of significance, α, determine the critical value of *t*, $t_{\alpha/2}$, then, reject H_0 if $t_0 > t_{\alpha/2}$ or if $t_0 < -t_{\alpha/2}$.

If you test for the value $M_0 = 0$, and it turns out you do not reject the null hypothesis, H_0: $M = 0$, then, the validity of a linear regression is in doubt. In other words, the sample data does not support the assertion that $M \neq 0$. Therefore, this is a test of the *significance of the regression model*.

- *Hypothesis testing on the intercept , B:*

The null hypothesis, H_0: $B = B_0$, is tested against the alternative hypothesis, H_1: $B \neq B_0$. The test statistic is

$$t_0 = (b - B_0) / [(1/n) + \bar{x}^2 / S_{xx}]^{1/2},$$

where *t* follows the Student's *t* distribution with $\nu = n - 2$, degrees of freedom, and *n* represents the number of points in the sample. The test is carried out as that of a mean

value hypothesis testing, i.e., given the level of significance, α, determine the critical value of t, $t_{\alpha/2}$, then, reject H_0 if $t_0 > t_{\alpha/2}$ or if $t_0 < -t_{\alpha/2}$.

A function for a comprehensive linear regression analysis

The following function, *linregtable,* produces a comprehensive analysis of linear regression returning not only the basic information produced by function *linreg*, but also including a table of data including the fitted values, the errors, and the confidence intervals for the mean value and the predicted values of the regression line. The function also returns estimated errors of the slope and intercept, and performs hypotheses testing for the cases $M = 0$ and $B = 0$.

```
function [se,rxy,sxy,slope,intercept,sy,sx,ybar,xbar]=linreg(x,y)

n=length(x);m=length(y);

if m<>n then
   error('linreg - Vectors x and y are not of the same length.');
   abort;
end;

Sxx       = sum(x^2)-sum(x)^2/n;
Syy       = sum(y^2)-sum(y)^2/n;
Sxy       = sum(x.*y)-sum(x)*sum(y)/n;
sx        = sqrt(Sxx/(n-1));
sy        = sqrt(Syy/(n-1));
sxy       = Sxy/(n-1);
rxy       = Sxy/sqrt(Sxx*Syy);
xbar      = mean(x);
ybar      = mean(y);
slope     = Sxy/Sxx;
intercept = ybar - slope*xbar;
se        = sqrt((n-1)*sy^2*(1-rxy^2)/(n-2));

xmin   = min(x);
xmax   = max(x);
xrange = xmax-xmin;
xmin   = xmin - xrange/10;
xmax   = xmax + xrange/10;
xx     = [xmin:(xmax-xmin)/100:xmax];
deff('[y]=yhat(x)','y=slope*x+intercept');
yy     = yhat(xx);
ymin   = min(y);
ymax   = max(y);
yrange = ymax - ymin;
ymin   = ymin - yrange/10;
ymax   = ymax + yrange/10;
rect   = [xmin ymin xmax ymax];
plot2d(xx,yy,1,'011',' ',rect);
xset('mark',-9,1);
plot2d( x, y,-9,'011',' ',rect);
xtitle('Linear regression','x','y');
```

An example for linear regression analysis using function *linregtable*

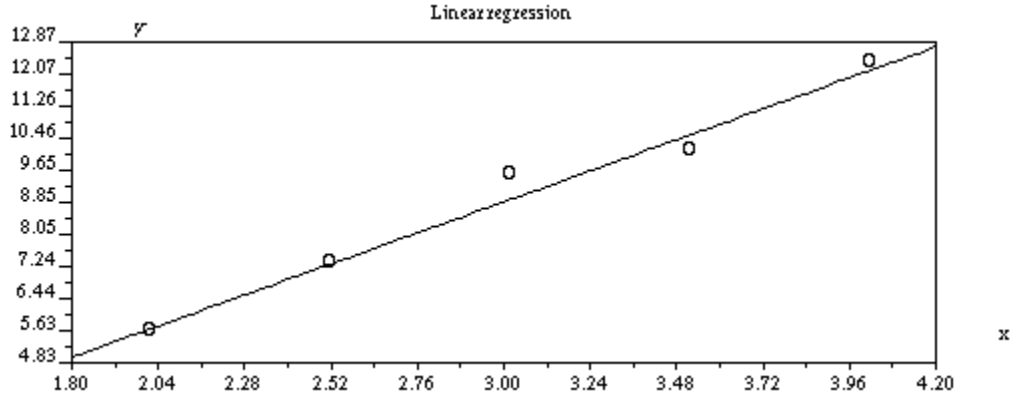
For an application of function *linregtable* consider the following (x,y) data. We use a significance level of 0.05.

x	2.0	2.5	3.0	3.5	4.0
y	5.5	7.2	9.4	10.0	12.2

The following SCILAB commands are used to load the data and perform the regression analysis:

```
-->getf('linregtable')

-->x=[2.0,2.5,3.0,3.5,4.0];y=[5.5,7.2,9.4,10.0,12.2];

-->linregtable(x,y,0.05)

Regression line:                 y = 3.24*x +  -.86
Significance level                 =  .05
Value of t_alpha/2                 = 3.18245
Confidence interval for slope      = [2.37976;4.10024]
Confidence interval for intercept  = [-3.51144;1.79144]
Covariance of x and y              = 2.025
Correlation coefficient            =  .98972
Standard error of estimate         =  .42740
Standard error of slope            =  .27031
Standard error of intercept        =  .83315
Mean values of x and y             = 3  8.86
Standard deviations of x and y     =  .79057  2.58805
Error sum of squares               =  .548

-------------------------------------------------------------------------------------
        x          y         ^y      error         C.I. mean          C.I. predicted
-------------------------------------------------------------------------------------
        2        5.5       5.62       -.12    4.56642    6.67358    3.89952    7.34048
      2.5        7.2       7.24       -.04    6.49501    7.98499    5.68918    8.79082
        3        9.4       8.86        .54    8.25172    9.46828    7.37002      10.35
      3.5         10      10.48       -.48    9.73501     11.225    8.92918    12.0308
        4       12.2       12.1         .1    11.0464    13.1536    10.3795    13.8205
-------------------------------------------------------------------------------------

Reject the null hypothesis H0:Slope = 0.
Test parameter for hypothesis testing on the slope (t) = 11.9863

Do not reject the null hypothesis H0:Intercept = 0.
Test parameter for hypothesis testing on the intercept (t) =  -.44117
```

The plot of the original data and the fitted data, also produced by function *linregtable*, is shown next:



The graph shows a good linear fitting of the data confirmed by a correlation coefficient (0.98972) very close to 1.0. The hypotheses testing indicate that the null hypothesis H_0:b = 0 cannot be rejected, i.e., a zero intercept may be substituted for the intercept of -0.86 with a 95% confidence level. On the other hand, the null hypothesis H_0:m=0 is rejected, indicating a proper linear relationship.

SCILAB function *reglin*

SCILAB provides function *reglin*, with call: *[m,b,sig] = reglin(x,y),* which returns the values of the slope, *m*, the intercept, *b*, and the standard deviation of the residual, σ, for the linear fitting $\hat{y} = mx+b$. For the data of the example above, using *reglin* we obtain:

```
--> [m,b,sig] = reglin(x,y)
sig = 0.3310589, b = -0.85, m = 3.24
```

Graphical display of multivariate data

In the next section we will present techniques of analysis for multiple linear regression in which we use fittings of the form $\hat{y} = b_0 + b_1 \cdot x_1 + b_2 \cdot x_2 + b_3 \cdot x_3 + \ldots + b_n \cdot x_n$. Before we present the details of the analysis, however, we want to introduce a simple way to visualize relationships between pairs of variables in a multivariate data set. The proposed graph is an array of plots representing the relationships between independent variables x_i and x_j, for $i \neq j$, as well as the relationship of the dependent variable *y* and each of the independent variables x_i. Function *multiplot*, which takes as input a matrix *X* whose columns are values of the independent variables, and a (row) vector *y*, which represents the dependent variable, produces such array of plots. A listing of the function *multiplot* follows next.

```
function [] = multiplot(X,y)

//Produces a matrix of plots:

//  ---x1----  x1-vs-x2   x1-vs-x3 ...   x1-vs-y
//  x2-vs-x1   ---x2---   x2-vs-x3 ...   x2-vs-y
//     .          .          .              .
//   y-vs-x1    y-vs-x2    y-vs-x3 ...    ---y---


[m n] = size(X);
nr = n+1; nc = nr;
XX = [X y'];

xset('window',1);
xset('default');
xbasc();
xset('mark',-1,1);

for i = 1:nr
    for j = 1:nc
        mtlb_subplot(nr,nc,(i-1)*nr+j);
        if i <> j then
          rect= [min(XX(:,j)) min(XX(:,i)) max(XX(:,j)) max(XX(:,i))];
           plot2d(XX(:,j),XX(:,i),-1);
```



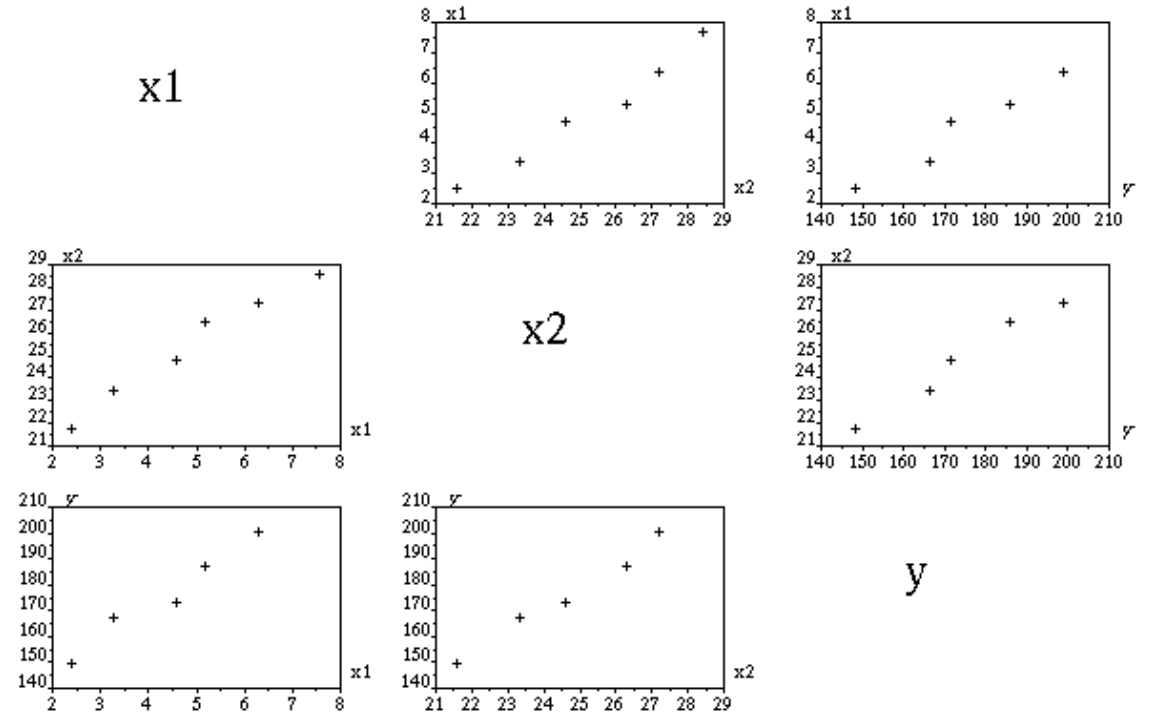
```
            if i==nr & j == nc then
               xtitle(' ','y','y');
            elseif i==nr then
               xtitle(' ','x'+string(j),'y');
            elseif j==nc then
               xtitle(' ','y','x'+string(i));
            else
               xtitle(' ','x'+string(j),'x'+string(i))
            end;
         end;
    end;
end;

xset('font',2,5);

for i = 1:nr
    for j = 1:nc
        mtlb_subplot(nr,nc,(i-1)*nr+j);
        if i==j then
           plot2d([0],[0],1,'010',' ',[0 0 10 10]);
           if i==nr & j==nc then
              xstring(3,5,'y');
           else
              xstring(3,5,'x'+string(i));
           end;
        end;
    end;
end;
```

To sub-divide the plot window into subplots, function *multiplot* uses function *mtlb_subplot*, a function that emulates Matlab®'s function *subplot* (which explains the prefix *mtlb_*). Details of this, and other functions with the *mtlb_* prefix are presented in more detail in Chapter 20.
To illustrate the use of function *multiplot* we will use the following data set:

x_1	x_2	y
2.3	*21.5*	*147.47*
3.2	*23.2*	*165.42*
4.5	*24.5*	*170.60*
5.1	*26.2*	*184.84*
6.2	*27.1*	*198.05*
7.5	*28.3*	*209.96*

The SCILAB commands to produce the plot array are shown next.

```
-->x1 = [2.3 3.2 4.5 5.1 6.2 7.5];

-->x2 = [21.5 23.2 24.5 26.2 27.1 28.3];

-->y  = [147.47 165.42 170.60 184.84 198.05 209.96];

-->X=[x1' x2'];

-->getf('multiplot')

-->multiplot(X,y)
```

A result like this array of plots is useful in determining some preliminary trends among the variables. For example, the plots above show strong dependency between x_1 and x_2, besides the expected dependency of y on x_1 or y on x_2. In that sense, variables x_1 and x_2 are not independent of each other. When we refer to them as the *independent variables,* the meaning is that of *variables that explain y*, which is, in turn, referred to as the *dependent variable.*

Multiple linear regression

The subject of multiple linear regression was first introduced in Chapter 5 as an example of applications of matrix operations. For multiple linear regression fitting consider a data set of the form

$\mathbf{x}_1$	$\mathbf{x}_2$	$\mathbf{x}_3$	$\mathbf{x}_4$	$\mathbf{x}_5$	y
x_{11}	x_{21}	x_{31}	...	x_{n1}	y_1
x_{12}	x_{22}	x_{32}	...	x_{n2}	y_2
x_{13}	x_{32}	x_{33}	...	x_{n3}	y_3
.	.	.		.	.
.	.	.	.	.	.
$x_{1,m-1}$	$x_{2,m-1}$	$x_{3,m-1}$	...	$x_{n,m-1}$	y_{m-1}
$x_{1,m}$	$x_{2,m}$	$x_{3,m}$	...	$x_{n,m}$	y_m

to which we fit an equation of the form

$$\hat{y} = b_0 + b_1 \cdot x_1 + b_2 \cdot x_2 + b_3 \cdot x_3 + \ldots + b_n \cdot x_n.$$

If we write the independent variables $x_1, x_2, \ldots, x_n$ into a row vector, i.e., $\mathbf{x}_i = [x_{1i}\ x_{2i}\ \ldots\ x_{ni}]$, and the coefficients $b_0\ b_1\ b_2\ \ldots\ b_n$ into a column vector $\mathbf{b} = [b_0\ \ b_1\ \ b_2\ \ b_3\ \ldots\ b_n]^T$, we can write

$$\hat{y}_i = \mathbf{x}_i \cdot \mathbf{b}.$$

If we put together the matrix, $\mathbf{X} = [\mathbf{x}_1\ \mathbf{x}_2\ \ldots\ \mathbf{x}_n]^T$, i.e.,

$$\begin{bmatrix} 1 & x_{11} & x_{21} & x_{31} & \ldots & x_{n1} \\ 1 & x_{12} & x_{22} & x_{32} & \ldots & x_{n2} \\ 1 & x_{13} & x_{32} & x_{33} & \ldots & x_{n3} \\ . & . & . & . & & . \\ . & . & . & . & . & . \\ 1 & x_{1,m} & x_{2,m} & x_{3,m} & \ldots & x_{n,m} \end{bmatrix}$$

and the vector, $\hat{\mathbf{y}} = [\hat{y}_1\ \hat{y}_2\ \ldots.\ \hat{y}_n]^T$, we can summarize the original table into the expression

$$\hat{\mathbf{y}} = \mathbf{X} \cdot \mathbf{b}.$$

The error vector is

$$\boldsymbol{e} = \boldsymbol{y} - \hat{\mathbf{y}},$$

and the sum of squared erros, SSE, is

$$SSE = \boldsymbol{e}^T \cdot \boldsymbol{e} = (\boldsymbol{y} - \hat{\mathbf{y}})^T \cdot (\boldsymbol{y} - \hat{\mathbf{y}}) = (\boldsymbol{y} - \boldsymbol{X} \cdot \boldsymbol{b})^T \cdot (\boldsymbol{y} - \boldsymbol{X} \cdot \boldsymbol{b}).$$

To minimize SSE we write $\partial(SSE)/\boldsymbol{b} = \boldsymbol{0}$. It can be shown that this results in the expression $\boldsymbol{X}^T \cdot \boldsymbol{X} \cdot \boldsymbol{b} = \boldsymbol{X}^T \cdot \boldsymbol{y}$, from which it follows that

$$\boldsymbol{b} = (\boldsymbol{X}^T \cdot \boldsymbol{X})^{-1} \cdot \boldsymbol{X}^T \cdot \boldsymbol{y}$$

An example for calculating the vector of coefficients ***b*** for a multiple linear regression was presented in Chapter 5. The example is repeated here to facilitate understanding of the procedure.

Example of multiple linear regression using matrices

use the following data to obtain the multiple linear fitting

$$y = b_0 + b_1 \cdot x_1 + b_2 \cdot x_2 + b_3 \cdot x_3,$$

x_1	x_2	x_3	y
1.20	3.10	2.00	5.70
2.50	3.10	2.50	8.20
3.50	4.50	2.50	5.00
4.00	4.50	3.00	8.20
6.00	5.00	3.50	9.50

With SCILAB you can proceed as follows:

First, enter the vectors x_1, x_2, x_3, and y, as row vectors:

```
->x1 = [1.2,2.5,3.5,4.0,6.0]
 x1  = !   1.2    2.5    3.5    4.    6. !

-->x2 = [3.1,3.1,4.5,4.5,5.0]
 x2  = !   3.1    3.1    4.5    4.5    5. !

-->x3 = [2.0,2.5,2.5,3.0,3.5]
 x3  = !   2.    2.5    2.5    3.    3.5 !

-->y = [5.7,8.2,5.0,8.2,9.5]
 y  =  !   5.7    8.2    5.    8.2    9.5 !
```

Next, we form matrix **X**:

```
-->X = [ones(5,1) x1' x2' x3']
 X  =

!   1.    1.2    3.1    2.  !
!   1.    2.5    3.1    2.5 !
!   1.    3.5    4.5    2.5 !
!   1.    4.     4.5    3.  !
!   1.    6.     5.     3.5 !
```

The vector of coefficients for the multiple linear equation is calculated as:

```
-->b =inv(X'*X)*X'*y
 b  =
! - 2.1649851 !
! -  .7144632 !
! - 1.7850398 !
!   7.0941849 !
```


Thus, the multiple-linear regression equation is:

$$y^{\wedge} = -2.1649851 - 0.7144632 \cdot x_1 - 1.7850398 \cdot x_2 + 7.0941849 \cdot x_3.$$

This function can be used to evaluate y for values of **x** given as $[x_1,x_2,x_3]$. For example, for $[x_1,x_2,x_3] = [3,4,2]$, construct a vector xx = [1,3,4,2], and multiply xx times b, to obtain y(xx):

```
-->xx = [1,3,4,2]
 xx  = !   1.    3.    4.    2. !

-->xx*b
 ans  = 2.739836
```

The fitted values of y corresponding to the values of x_1, x_2, and x_3 from the table are obtained from **y** = **X**·**b**:

```
-->X*b
 ans  =
!   5.6324056 !
!   8.2506958 !
!   5.0371769 !
!   8.2270378 !
!   9.4526839 !
```

Compare these fitted values with the original data as shown in the table below:

x_1	x_2	x_3	y	y-fitted
1.20	3.10	2.00	5.70	5.63
2.50	3.10	2.50	8.20	8.25
3.50	4.50	2.50	5.00	5.04
4.00	4.50	3.00	8.20	8.23
6.00	5.00	3.50	9.50	9.45

This procedure will be coded into a user-defined SCILAB function in an upcoming section incorporating some of the calculations for the regression analysis as shown next.

An array of plots showing the dependency of the different variables involved in the multiple linear fitting is shown in the following page. It was produced by using function *multiplot*.

```
-->multiplot(X,y);
```

(*See plot in next page*).

Covariance in multiple linear regression

In simple linear regression we defined a standard error of the estimate, s_e, as an approximation to the variance σ of the distribution of errors. The standard error of the estimate, also known as the *mean square error*, for multiple linear regression is defined as

$$s_e = MSE = SSE/(m-n-1),$$

where m is the number of data points available and n is the number of coefficients required for the multiple linear fitting.

The matrix $\boldsymbol{C} = s_e^2(\boldsymbol{X}^T\cdot\boldsymbol{X})^{-1}$ is a symmetric matrix known as the *covariance matrix*. The diagonal elements c_{ii} are the variances associated with each of the coefficients b_i, i.e.,

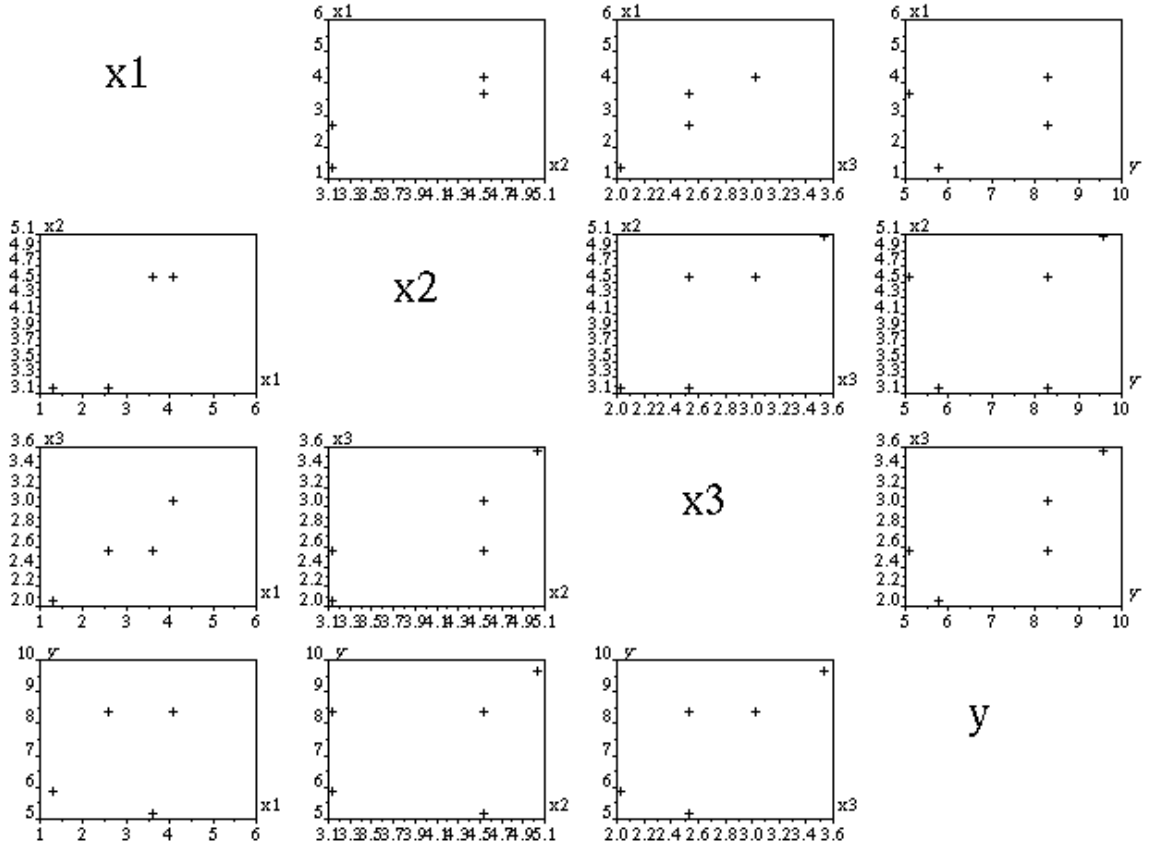
$$Var(b_i) = c_{ii},$$

while elements off the diagonal, c_{ij}, $i \neq j$, are the covariances of b_i and b_j, i.e.,

$$Cov(b_i,b_j) = c_{ij,}\ i \neq j.$$

The square root of the variances $Var(b_i)$ are referred to as the standard error of the estimate for each coefficient, i.e.,

$$s_e(b_i) = \sqrt{c_{ii}} = [Var(b_i)]^{1/2}.$$



Confidence intervals and hypotheses testing in multiple linear regression

The multiple linear relationship defined earlier, namely,

$$\hat{y} = b_0 + b_1 \cdot x_1 + b_2 \cdot x_2 + b_3 \cdot x_3 + \ldots + b_n \cdot x_n,$$

is an approximation to the multiple linear model

$$Y = \beta_0 + \beta_1 \cdot x_1 + \beta_2 \cdot x_2 + \beta_3 \cdot x_3 + \ldots + \beta_n \cdot x_n + \varepsilon,$$

where Y is a normally-distributed random variable with mean $\hat{y}$, ε is also normally-distributed with mean zero. The standard deviation of the distributions of both Y and ε is σ. An approximation to the value of σ is the standard error of the estimate for the regression, s_e, defined earlier.

- *Confidence intervals for the coefficients*

  Using a level of confidence α, we can write confidence intervals for each of the coefficients β_i in the linear model for Y, as

  $$b_i - (t_{m-n,\alpha/2}) \cdot s_e(b_i) < \beta_i < b_i + (t_{m-n,\alpha/2}) \cdot s_e(b_i),$$

  for $i=1,2,\ldots,n$, where b_i is the i-th coefficient in the linear fitting, $t_{m-n,\alpha/2}$ is the value of the Student's t variable for $\nu = m-n$ degrees of freedom corresponding to a cumulative probability of $1-\alpha/2$, and $s_e(b_i)$ is the standard error of the estimate for b_i.

- *Confidence interval for the mean value of Y at* $\boldsymbol{x} = \boldsymbol{x}_0$, *i.e.,* $\boldsymbol{x}_0^T\boldsymbol{b}$:

  $$[\boldsymbol{x}_0^T\boldsymbol{b} - (t_{m-n,\alpha/2}) \cdot [\boldsymbol{x}_0^T \cdot \boldsymbol{C} \cdot \boldsymbol{x}_0]^{1/2} ; \boldsymbol{x}_0^T\boldsymbol{b} + (t_{m-n,\alpha/2}) \cdot [\boldsymbol{x}_0^T \cdot \boldsymbol{C} \cdot \boldsymbol{x}_0]^{1/2\ 2}]$$

  where $\boldsymbol{C}$ is the covariance matrix.

- *Limits of prediction: confidence interval for the predicted value* $Y_0 = Y(\boldsymbol{x}_0)$:

  $$[\boldsymbol{x}_0^T\boldsymbol{b} - (t_{m-n,\alpha/2}) \cdot s_e \cdot [1+\boldsymbol{x}_0^T \cdot (\boldsymbol{C}/s_e) \cdot \boldsymbol{x}_0]^{1/2} ; \boldsymbol{x}_0^T\boldsymbol{b} + (t_{m-n,\alpha/2}) \cdot s_e \cdot [1+\boldsymbol{x}_0^T \cdot (\boldsymbol{C}/s_e) \cdot \boldsymbol{x}_0]^{1/2\ 2}]$$

- *Hypothesis testing on the coefficients,* β_i:

  The null hypothesis, H_0: $\beta_i = \beta_0$, is tested against the alternative hypothesis, H_1: $\beta_i \neq \beta_0$. The test statistic is

  $$t_0 = (b_i - \beta_0)/s_e(b_i)$$

  where t follows the Student's t distribution with $\nu = m-n$, degrees of freedom. The test is carried out as that of a mean value hypothesis testing, i.e., given the level of significance, α, determine the critical value of t, $t_{\alpha/2}$, then, reject H_0 if $t_0 > t_{\alpha/2}$ or if $t_0 < -t_{\alpha/2}$. Of interest, many times, is the test that a particular coefficient b_i be zero, i.e., H_0: $\beta_i = 0$.

- *Hypothesis testing for significance of regression*

  This test is aimed at determining if a linear regression indeed exists between *y* and ***x***. The null hypothesis to be tested is H_0: $\beta_1=\beta_2=\beta_3=\ldots=\beta_n=0$, against the alternative hypothesis, H_1: $\beta_i \neq 0$ *for at least one value of* i. The appropriate test to be conducted is an *F* test based on the test parameter

$$F_0 = \frac{SSR/n}{SSE/(m-n-1)} = \frac{MSR}{MSE},$$

  where $MSR = SSR/n$ is known as the *regression mean square*, $MSE = SSE/(m-n-1) = s_e^2$ is the *mean square error*, *SSE* is the sum of squared errors (defined earlier), and *SSR* is the regression sum of squares defined as

$$SSR = \sum_{i=1}^{m} (\hat{y}_i - \bar{y})^2,$$

  where $\bar{y}$ is the mean value of *y*. The parameter F_0 follows the *F* distribution with *n* degrees of freedom in the numerator and *m-n-1* degrees of freedom in the denominator.

  For a given confidence level α, we will determine the value F_α from the appropriate *F* distribution, and reject the null hypothesis H_0 if $F_0 > F_\alpha$.

- *Analysis of variance for testing significance of regression*

  Analysis of variance is the name to the general method that produces the parameters described above for the testing of significance of regression. The method is based on the so-called *analysis of variance identity*,

$$SST = SSR + SSE,$$

  where *SSE* and *SSR* were described above and SST, the *total corrected sum of squares*, is described as

$$SST = \sum_{i=1}^{m} (y_i - \bar{y})^2.$$

  The term *SSR* accounts for the variability in y_i due to the regression line, while the terms *SSE* accounts for the residual variability not incorporated in *SSR*. The term SSR has *n* degrees of freedom, i.e., the same number of coefficients in the multiple linear fitting. The term SSE has *m-n-1* degrees of freedom, while SST has *n-1* degrees of freedom.

Analysis of variance for the test of significance of regression is typically reported in a table that includes the following information:

Variation source	Sum of squares	Degrees of freedom	Mean square	F_0
Regression	*SSR*	*n*	*MSR*	*MSR/MSE*
Residual/error	*SSE*	*m-k-1*	*MSE*	
Total	*SST*	*m-1*		

Coefficient of multiple determination

The coefficient of multiple determination R^2 is defined in terms of SST, SSR, and SSE, as

$$R^2 = \frac{SSR}{SST} = 1 - \frac{SSE}{SST},$$

while the positive square root of this value is referred to as the *multiple correlation coefficient, R*. This multiple correlation coefficient, for the case of a simple linear regression, is the same as the correlation coefficient r_{xy}. Values of R^2 are restricted to the range [0,1].

Unlike the simple correlation coefficient, r_{xy}, the coefficient of multiple determination, R^2, is not a good indicator of linearity. A better indicator is the adjusted coefficient of multiple determination,

$$R^2_{adj} = 1 - \frac{SSE/(m-n-1)}{SST/(m-1)}.$$

A function for multiple linear regression analysis

The following function, *multiplelinear*, includes the calculation of the coefficients for a multiple linear regression equation, their standard error of estimates, their confidence interval, and the recommendation for rejection or not rejection for the null hypotheses $H_0: \beta_i = 0$. The function also produces a table of the values of *y*, the fitted values $\hat{y}$, the errors, and the confidence intervals for the mean linear regression Y, and for the predicted linear regression. The analysis-of-variance table is also produced by the function. Finally, the function prints the values of the standard error of the estimate, s_e, the coefficient of multiple determination, the multiple correlation coefficient, the adjusted coefficient of multiple determination, and the covariance matrix. The function returns the vector of coefficients, $\boldsymbol{b}$, the covariance matrix, $cov(\boldsymbol{x}_i, \boldsymbol{x}_j)$, and the standard error of the estimate, s_e. The arguments of the function are a matrix *XA* whose columns include the column data vectors $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n$, a column vector *y* containing the dependent variable data, and a level of confidence, *alpha* (typical values are *0.01, 0.05, 0.10*).

A listing of the function follows:

```
function [b,C] = multiplelinear(XA,y,alpha)

[m n] = size(XA);          //Size of original X matrix
X = [ones(m,1) XA];       //Augmenting matrix X
b=inv(X'*X)*X'*y;         //Coefficients of function
yh = X*b;                 //Fitted value of y
e=y-yh;                   //Errors or residuals
SSE=e'*e;                 //Sum of squared errors
MSE = SSE/(m-n-1);        //Mean square error
se = sqrt(MSE);           //Standard error of estimate
C = MSE*inv(X'*X);        //Covariance matrix
[nC mC]=size(C);
seb = [];                 //Standard errors for coefficients

for i = 1:nC
    seb = [seb; sqrt(C(i,i))];
end;
ta2 = cdft('T',m-n,1-alpha/2,alpha/2);  //t_alpha/2
sY = []; sYp = [];       //Terms involved in C.I. for Y, Ypred
for i=1:m
    sY  = [sY;  sqrt(X(i,:)*C*X(i,:)')];
    sYp = [sYp; se*sqrt(1+X(i,:)*(C/se)*X(i,:)')];
end;
CIYL = yh-sY;            //Lower limit for C.I. for mean Y
CIYU = yh+sY;            //Upper limit for C.I. for mean Y
CIYpL = yh-sYp;          //Lower limit for C.I. for predicted Y
CIYpU = yh+sYp;          //Upper limit for C.I. for predicted Y
CIbL = b-ta2*seb;        //Lower limit for C.I. for coefficients
CIbU = b+ta2*seb;        //Upper limit for C.I. for coefficients
t0b  = b./seb;           //t parameter for testing H0:b(i)=0
decision = [];           //Hypothesis testing for H0:b(i)=0
for i = 1:n+1
    if t0b(i)>ta2 | t0b(i)<-ta2 then
       decision = [decision; ' reject       '];
    else
       decision = [decision; ' do not reject'];
    end;
end;

ybar = mean(y);           //Mean value of y
SST = sum((y-ybar)^2);    //Total sum of squares
SSR = sum((yh-ybar)^2);   //Residual sum of squares

MSR = SSR/n;              //Regression mean square
MSE = SSE/(m-n-1);        //Error mean square
F0  = MSR/MSE;            //F parameter for significance of regression
Fa  = cdff('F',n,m-n-1,1-alpha,alpha);   //F_alpha

R2  = 1-SSE/SST; R = sqrt(R2);      //Coeff. of multiple regression
R2a = 1-(SSE/(m-n-1))/(SST/(m-1));  //Adj. Coeff. of multiple regression

//Printing of results
printf(' ');
printf('Multiple linear regression');
printf('==========================');
printf(' ');

printf('Table of coefficients');
printf('------------------------------------------------------------------------
--');
```



```
printf('    i          b(i)    se(b(i))        Lower        Upper             t0
H0:b(i)=0');
printf('------------------------------------------------------------------------
--');
for i = 1:n+1
    printf('%4.0f %10g %10g %10g %10g %10g '+decision(i),...
           i-1,b(i),seb(i),CIbL(i),CIbU(i),t0b(i));
end;
printf('------------------------------------------------------------------------
--');
printf('                                        t_alpha/2 = %g',ta2);
printf('------------------------------------------------------------------------
--');
printf(' ');printf(' ');

printf('Table of fitted values and errors');
printf('------------------------------------------------------------------------
----------');
printf('   i       y(i)      yh(i)       e(i)        C.I. for Y           C.I.
for Ypred');
printf('------------------------------------------------------------------------
----------');
for i = 1:m
  printf('%4.0f %10.6g %10.6g %10.6g %10.6g %10.6g %10.6g %10.6g',...
         i,y(i),yh(i),e(i),CIYL(i),CIYU(i),CIYpL(i),CIYpU(i));
end;
printf('------------------------------------------------------------------------
----------');

printf('  ');printf(' ');
printf('Analysis of variance');
printf('-----------------------------------------------------------');
printf('Source of      Sum of   Degrees of       Mean')
printf('variation      squares     freedom     square         F0');
printf('-----------------------------------------------------------');
printf('Regression   %10.6g %10.0f %10.6g %10.6g',SSR,n,MSR,F0');
printf('Residual     %10.6g %10.0f %10.6g       ',SSE,m-n-1,MSE);
printf('Total        %10.6g %10.0f              ',SST,m-1);
printf('-----------------------------------------------------------');

printf('With F0 = %g and F_alpha = %g,',F0,Fa);
if F0>Fa then
   printf('reject the null hypothesis H0:beta1=beta2=...=betan=0.');
else
   printf('do not reject the null hypothesis H0:beta1=beta2=...=betan=0.');
end;
printf('-----------------------------------------------------------');

disp(' ');
printf('Additional information');
printf('-----------------------------------------------------------');
printf('Standard error of estimate (se)               = %g',se);
printf('Coefficient of multiple determination (R^2)   = %g',R2);
printf('Multiple correlation coefficient (R)          = %g',R);
printf('Adjusted coefficient of multiple determination = %g',R2a);
printf('-----------------------------------------------------------');

printf(' ');
printf('Covariance matrix:');
disp(C);
printf(' ');
printf('-----------------------------------------------------------');
//Plots of residuals - several options
```



```
for j = 1:n
    xset('window',j);xset('mark',-9,2);xbasc(j);
    plot2d(XA(:,j),e,-9)
    xtitle('Residual plot - error vs. x'+string(j),'x'+string(j),'error');
end;
xset('window',n+1);xset('mark',-9,2);
plot2d(y,e,-9);
xtitle('Residual plot - error vs. y','y','error');
xset('window',n+2);xset('mark',-9,2);
plot2d(yh,e,-9);
xtitle('Residual plot - error vs. yh','yh','error');
```

Application of function *multiplelinear*

Consider the multiple linear regression of the form $y = b_0+b_1x_1+b_2x_2$ for the data shown in the following table:

x_1	x_2	x_3	y
1.20	*3.10*	*2.00*	*5.70*
2.50	*3.10*	*2.50*	*8.20*
3.50	*4.50*	*2.50*	*5.00*
4.00	*4.50*	*3.00*	*8.20*
6.00	*5.00*	*3.50*	*9.50*

First, we load the data:

```
-->x1=[1.2,2.5,3.5,4.0,6.0];x2=[3.1,3.1,4.5,4.5,5.0];
-->x3=[2.0,2.5,2.5,3.0,3.5];y =[5.7,8.2,5.0,8.2,9.5];
-->X=[x1' x2' x3']
 X  =
!   1.2    3.1    2.  !
!   2.5    3.1    2.5 !
!   3.5    4.5    2.5 !
!   4.     4.5    3.  !
!   6.     5.     3.5 !
-->y=y'
 y  =
!   5.7 !
!   8.2 !
!   5.  !
!   8.2 !
!   9.5 !
```

Then, we call function *multiplelinear* to obtain information on the multiple linear regression:

```
-->[b,C,se]=multiplelinear(X,y,0.1);

Multiple linear regression
==========================

Table of coefficients
-------------------------------------------------------------------------
   i       b(i)   se(b(i))      Lower      Upper         t0  H0:b(i)=0
-------------------------------------------------------------------------
   0   -2.16499    1.14458   -5.50713    1.17716   -1.89152  do not reject
   1    -.71446     .21459   -1.34105  -.0878760    -3.3295  reject
   2   -1.78504     .18141   -2.31477   -1.25531   -9.83958  reject
   3    7.09418     .49595    5.64603    8.54234    14.3043  reject
-------------------------------------------------------------------------
                                       t_alpha/2 = 2.91999
-------------------------------------------------------------------------
```


Table of fitted values and errors

i	y(i)	yh(i)	e(i)	C.I. for Y		C.I. for Ypred	
1	5.7	5.63241	.0675944	5.54921	5.7156	5.5218	5.74301
2	8.2	8.2507	-.0506958	8.15624	8.34515	8.13913	8.36226
3	5	5.03718	-.0371769	4.93663	5.13772	4.92504	5.14931
4	8.2	8.22704	-.0270378	8.12331	8.33077	8.11459	8.33949
5	9.5	9.45268	.0473161	9.3565	9.54887	9.34096	9.56441

Analysis of variance

Source of variation	Sum of squares	Degrees of freedom	Mean square	F0
Regression	14.2965	3	4.7655	414.714
Residual	.0114911	1	.0114911	
Total	14.308	4		

With F0 = 414.714 and F_alpha = 53.5932,
reject the null hypothesis H0:beta1=beta2=...=betan=0.

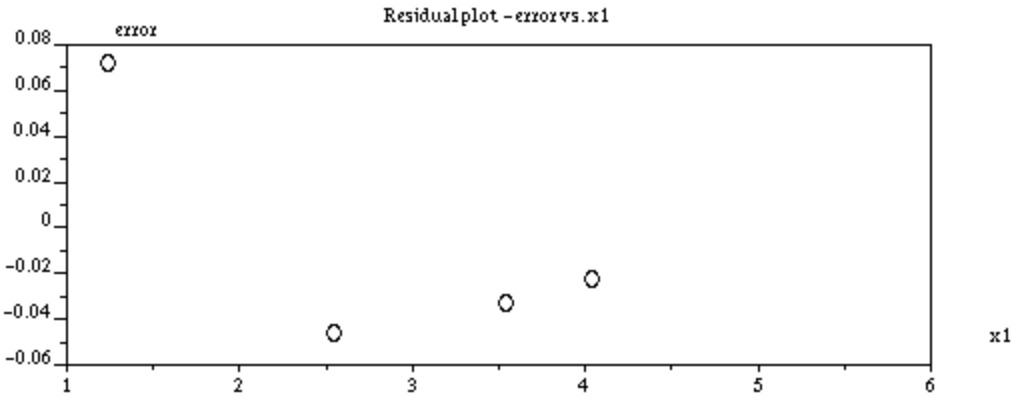
Additional information

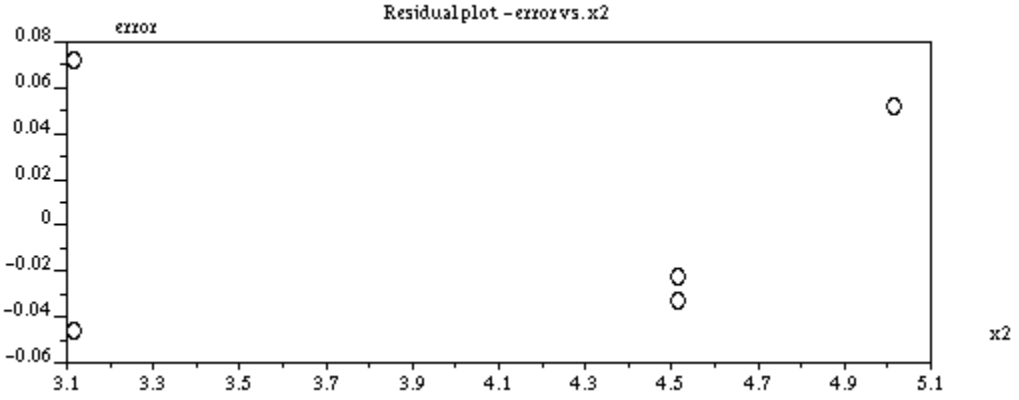
Standard error of estimate (se)	= .10720
Coefficient of multiple determination (R^2)	= .99920
Multiple correlation coefficient (R)	= .99960
Adjusted coefficient of multiple determination	= .99679

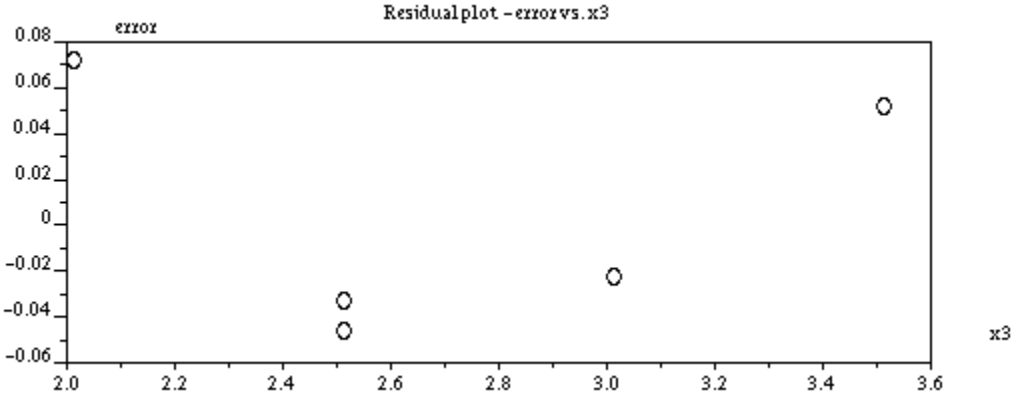
Covariance matrix:

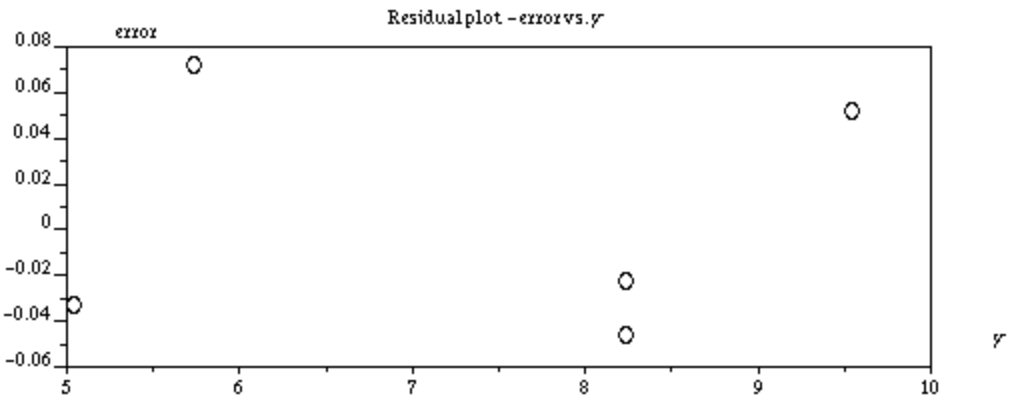
1.3100522	.2388806	- .1694459	- .5351636
.2388806	.0460470	- .0313405	- .1002469
- .1694459	- .0313405	.0329111	.0534431
- .5351636	- .1002469	.0534431	.2459639

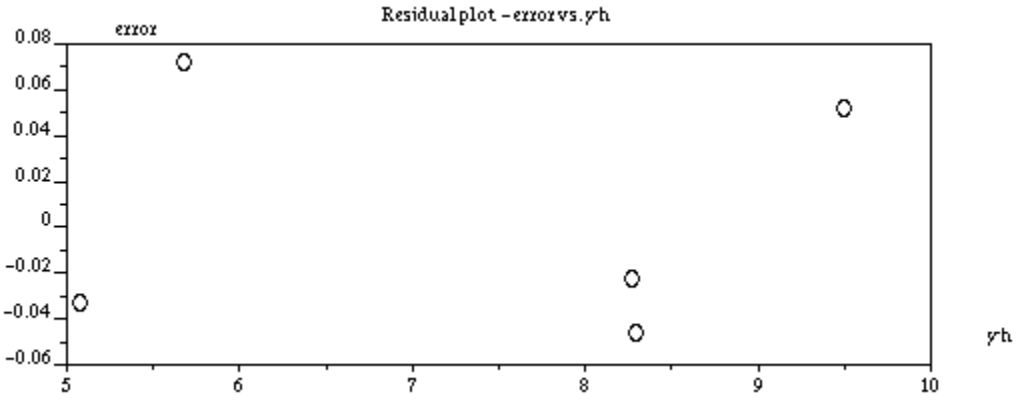
The results show that, for a confidence level $\alpha = 0.1$, the hypothesis $H_0: \beta_0 = 0$ may not be rejected, meaning that you could eliminate that term from the multiple linear fitting. On the other hand, the test for significance of regression indicates that we cannot reject the hypothesis that a linear regression does exist. Plots of the residuals against variables x_1, x_2, x_3, y, and $\hat{y}$ are shown next.











A function for predicting values from a multiple regression

The following function, *mlpredict*, takes as arguments a row vector *x* containing values of the independent variables *[x_1 x_2 … x_n]* in a multiple linear regression, *b* is a column vector containing the coefficients of the linear regression, *C* is the corresponding covariance matrix, *se* is the standard error of the estimate, and *alpha* is the level of confidence. The function returns the predicted value *y* and prints the confidence intervals for the mean value *Y* and for the predicted value of *Y*.

```
function [y] = mlpredict(x,b,C,se,alpha)

nb = length(b); nx = length(x);
if nb<>nx+1 then
   error('mlpredict - Vectors x and b are of incompatible length.');
   abort;
else
   n = nx;
end;

[nC mC] = size(C);
if nC<>mC then
   error('mlpredict - Covariance matrix C must be a square matrix.');
```



```
   abort;
elseif nC<>n+1 then
   error('mlpredict - Dimensions of covariance matrix incompatible with vector
b.');
   abort;
end;

xx = [1 x];                              //augment vector x
y = xx*b;                                //calculate y
CIYL  = y - sqrt(xx*C*xx');              //Lower limit C.I. for mean Y
CIYU  = y + sqrt(xx*C*xx');              //Upper limit C.I. for mean Y
CIYpL = y - se*sqrt(1+xx*(C/se)*xx');    //Lower limit C.I. for predicted Y
CIYpU = y + se*sqrt(1+xx*(C/se)*xx');    //Upper limit C.I. for predicted Y

//Print results
printf(' ');
disp(' ',x,'For x = ');
printf('Multiple linear regression prediction is y = %g',y);
printf('Confidence interval for mean value Y       = [%g,%g]',CIYL,CIYU);
printf('Confidence interval for predicted value Y  = [%g,%g]',CIYpL,CIYpU);
```

An application of function *mlpredict*, using the values of *b*, *C*, and *se* obtained from function *multiplelinear* as shown above, is presented next:

```
-->y=mlpredict(x,b,C,se,0.1);

 For x =

!   2.    3.5    2.8 !


Multiple linear regression prediction is y = 10.0222
Confidence interval for mean value Y       = [9.732,10.3123]
Confidence interval for predicted value Y  = [9.87893,10.1654]
```

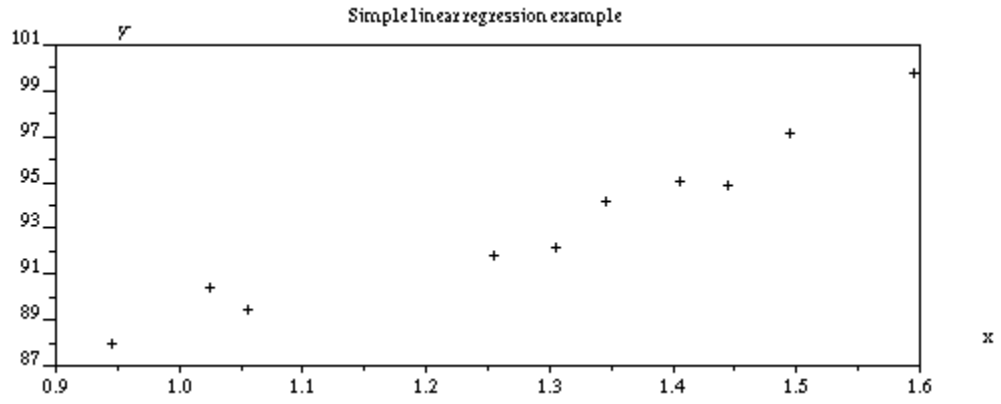
Simple linear regression using function *multiplelinear*

Function *multiplelinear* can be used to produce the regression analysis for a simple linear regression as illustrated in the following example. The data to be fitted is given in the following table:

x	*y*
1.02	*90.02*
1.05	*89.14*
1.25	*91.48*
1.34	*93.81*
1.49	*96.77*
1.44	*94.49*
0.94	*87.62*
1.30	*91.78*
1.59	*99.43*
1.40	*94.69*

A plot of the data is produced with:

```
-->plot2d(x,y,-1);xtitle('Simple linear regression example', 'x','y');
```



The regression analysis using function *multiplelinear* follows:

```
-->[b,C,se] = multiplelinear(X,Y,0.05);

Multiple linear regression
==========================

Table of coefficients
-------------------------------------------------------------------------
   i       b(i)   se(b(i))      Lower      Upper         t0  H0:b(i)=0
-------------------------------------------------------------------------
   0    72.2564    2.03849     67.645    76.8678     35.446  reject
   1    16.1206     1.5701    12.5688    19.6724    10.2672  reject
-------------------------------------------------------------------------
                                      t_alpha/2 = 2.26216
-------------------------------------------------------------------------


Table of fitted values and errors
-------------------------------------------------------------------------------
   i     y(i)      yh(i)       e(i)        C.I. for Y          C.I. for Ypred
-------------------------------------------------------------------------------
   1    90.02    88.6994    1.32059    88.1769    89.2219     87.552    89.8468
   2    89.14     89.183  -.0430274    88.6967    89.6694     88.052    90.3141
   3    91.48    92.4071    -.92714     92.081    92.7333    91.3363    93.4779
   4    93.81     93.858  -.0479932    93.5232    94.1928    92.7845    94.9315
   5    96.77    96.2761     .49392    95.8173    96.7349    95.1568    97.3953
   6    94.49    95.4701    -.98005    95.0634    95.8767    94.3715    96.5686
   7    87.62    87.4098     .21024    86.7835     88.036    86.2107    88.6089
   8    91.78    93.2132   -1.43317    92.8897    93.5366    92.1432    94.2831
   9    99.43    97.8881    1.54186     97.307    98.4692    96.7124    99.0639
  10    94.69    94.8252    -.13523    94.4535    95.1969    93.7394    95.9111
-------------------------------------------------------------------------------
```

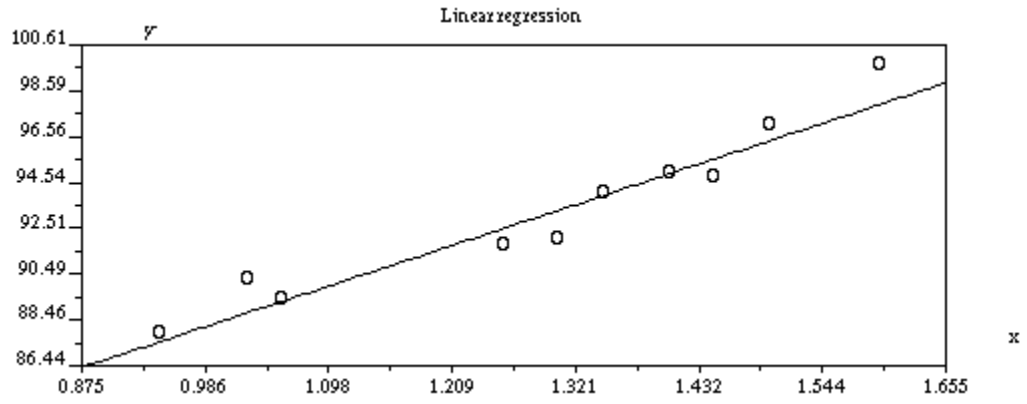


```
Analysis of variance
-----------------------------------------------------------
Source of      Sum of   Degrees of       Mean
variation      squares     freedom     square         F0
-----------------------------------------------------------
Regression      109.448          1    109.448    105.416
Residual        8.30597          8    1.03825
Total           117.754          9
-----------------------------------------------------------
With F0 = 105.416 and F_alpha = 5.31766,
reject the null hypothesis H0:beta1=beta2=...=betan=0.
-----------------------------------------------------------

Additional information
-----------------------------------------------------------
Standard error of estimate (se)                = 1.01894
Coefficient of multiple determination (R^2)    =  .92946
Multiple correlation coefficient (R)           =  .96409
Adjusted coefficient of multiple determination =  .92065
-----------------------------------------------------------

Covariance matrix:

!   4.1554489  - 3.1603934 !
! - 3.1603934    2.4652054 !

-----------------------------------------------------------
```

Compare the results with those obtained using function *linreg*:

```
-->[se,rxy,sxy,m,b,sy,sx,ybar,xbar]=linreg(x,y);

-->[m,b]
 ans  = !   16.120572    72.256427 !

-->se
 se  = 1.0189435

-->rxy
 rxy  = .9640868
```

The data fitting produced with *linreg* is shown in the next figure:



Analysis of residuals

Residuals are simply the errors between the original data values $\boldsymbol{y}$ and the fitted values $\hat{\mathbf{y}}$, i.e., the values

$$\boldsymbol{e} = \boldsymbol{y} - \hat{\mathbf{y}}.$$

Plots of the residuals against each of the independent variables, $\boldsymbol{x}_1$, $\boldsymbol{x}_2$, …, $\boldsymbol{x}_n$, or against the original data values $\boldsymbol{y}$ or the fitted values $\hat{\mathbf{y}}$ can be used to identify trends in the residuals. If the residuals are randomly distributed about zero, the plots will show no specific pattern for the residuals. Thus, residual analysis can be used to check the assumption of normal distribution of errors about a zero mean. If the assumption of normal distribution of residuals about zero does not hold, the plots of residuals may show specific trends.

For example, consider the data from the following table:

x_1	x_2	p	q	r
1.1	22.1	1524.7407	3585.7418	1558.505
2.1	23.2	1600.8101	3863.4938	1588.175
3.4	24.5	1630.6414	4205.4344	1630.79
4.2	20.4	1416.1757	3216.1131	1371.75
5.5	25.2	1681.7725	4409.4029	1658.565
6.1	23.1	1554.5774	3876.8094	1541.455
7.4	19.2	1317.4763	2975.0054	1315.83
8.1	18.2	1324.6139	2764.6509	1296.275
9.9	20.5	1446.5163	3289.158	1481.265
11.	19.1	1328.9309	2983.4153	1458.17

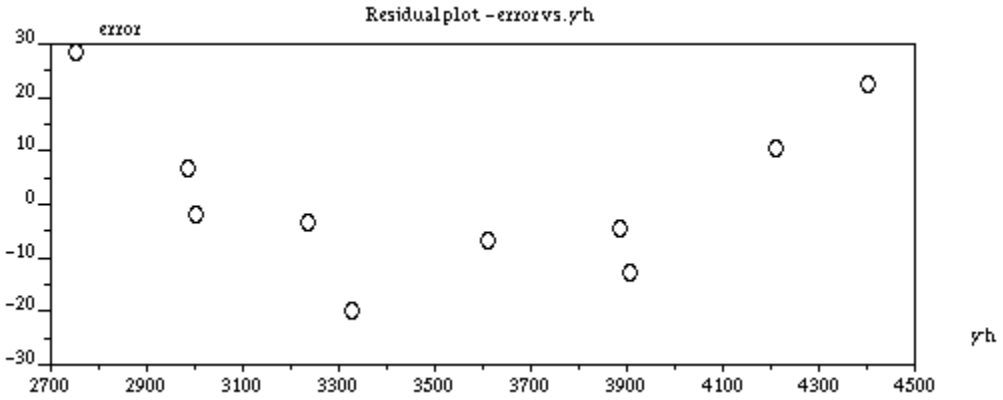
The table shows two independent variables, x_1 and x_2, and three dependent variables, i.e., $p = p(x_1,x_2)$, $q = q(x_1,x_2)$, and $r = r(x_1,x_2)$. We will try to fit a multiple linear function to the three functions, e.g., $p = b_0 + b_1x_1 + b_2x_2$, using function *multiplelinear*, with the specific purpose of checking the distribution of the errors. Thus, we will not include in this section the output for the function calls. Only the plots of residuals (or errors) against the fitted data will be presented. The SCILAB command required to produce the plots are shown next.

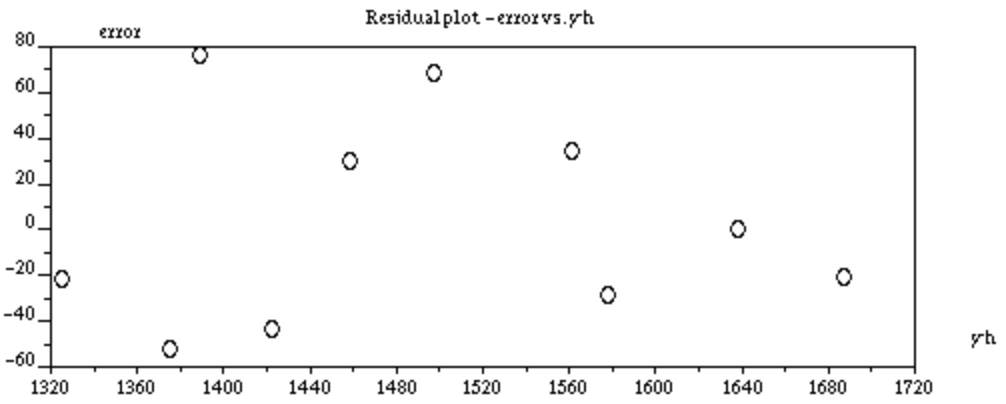
```
-->x1=[1.1 2.1 3.4 4.2 5.5 6.1 7.4 8.1 9.9 11.0];
-->x2 = [22.1 23.2 24.5 20.4 25.2 23.1 19.2 18.2 20.5 19.1];
-->p = [ 1524.7407    1600.8101    1630.6414    1416.1757    1681.7725 ...
-->      1554.5774    1317.4763    1324.6139    1446.5163    1328.9309 ];
-->q = [ 3585.7418    3863.4938    4205.4344    3216.1131    4409.4029 ...
-->      3876.8094    2975.0054    2764.6509    3289.158    2983.4153 ];
-->r = [ 1558.505    1588.175    1630.79    1371.75    1658.565    1541.455 ...
-->      1315.83     1296.275    1481.265   1458.17 ];

-->X = [x1' x2'];

-->[b,C,se] = multiplelinear(X,p',0.01);

-->[b,C,se] = multiplelinear(X,q',0.01);

-->[b,C,se] = multiplelinear(X,r',0.01);
```


The following table summarizes information regarding the fittings:

Fitting	R^2	R	R^2_{adj}	*Significance*
$p=p(x_1,x_2)$	0.97658	0.98822	0.96989	reject H_0
$q=q(x_1,x_2)$	0.99926	0.99963	0.99905	reject H_0
$r=r(x_1,x_2)$	0.8732	0.93445	0.83697	reject H_0

The three fitting show good values of the coefficients R^2, R, and R^2_{adj}, and the F test for the significance of the regression indicates rejecting the null hypothesis $H_0: \beta_1 = \beta_2 = ... = \beta_n = 0$ in all cases. In other words, a multiple linear fitting is acceptable for all cases. The residual plots, depicted below, however, indicate that only the fitting of $p=p(x_1,x_2)$ shows a random distribution about zero. The fitting of $q=q(x_1,x_2)$ shows a clear non-linear trend, while the fitting of $r=r(x_1,x_2)$ shows a funnel shape.





Scaling residuals

The errors e_i may be standardized by using the values

$$ze_i = e_i/s_e,$$

where s_e is the standard error of the estimate from the multiple linear fitting. If the errors are normally distributed then approximately 95% of them will fall within the interval (-2,+2). Any residual located far outside of this interval may indicate an *outlier* point. Outlier points are points that do not follow the general trend of the data. They may indicate an error in the data collection or simply an extremely large or low value in the data. If sufficient justification exists, outlier points may be discarded and the regression repeated with the remaining data points.

Another way to scale residuals is to use the so-called *studentized residuals* defined as

$$r_i = \frac{e_i}{s_e\sqrt{1-h_{ii}}},$$

$i = 1,2,\ldots,m$, where h_{ii} are the diagonal elements in the 'hat' matrix, H, defined as

$$\mathbf{H} = \mathbf{X}\cdot(\mathbf{X}^T\cdot\mathbf{X})^{-1}\cdot\mathbf{X}.$$

This matrix relates the observed values $\boldsymbol{y}$ to the fitted values $\hat{\mathbf{y}}$, i.e., $\hat{\mathbf{y}} = \boldsymbol{H}\cdot\boldsymbol{y}$. Thus, the name 'hat' matrix. Using the values h_{ii} we can write the *studentized standard error* of the i-th residual as

$$s_e(e_i) = s_e\cdot(1-h_{ii})^{1/2}.$$

Influential observations

Sometimes in a simple or multiple linear fitting there will be one or more points whose effects on the regression are unusually influential. Typically, these so-called *influential observations* correspond to outlier points that tend to "drag" the fitting in one or other direction. To determine whether or not a point is influential we computed the *Cook's distance* defined as

$$d_i = \frac{e_i^2 h_{ii}}{(n+1)s_e^2(1+h_{ii})^2} = \frac{ze_i^2 h_{ii}}{(n+1)(1+h_{ii})^2} = \frac{r_i^2 h_{ii}}{(n+1)(1-h_{ii})},$$

$i=1,2,\ldots,m$. A value of $d_i > 1$ may indicate an influential point.

A function for residual analysis

The following function, *residuals*, produces a comprehensive residual analysis including the standardized and studentized residuals, studentized standard error of estimates for the residuals, and the corresponding Cooks' distances. The function takes as input a matrix *X* whose columns represent the independent variables, i.e., $X = [\mathbf{x}_1\ \mathbf{x}_2\ \cdots\ \mathbf{x}_n]$, and the column vector *y* containing the dependent variable. The function produces a table of results, plots of the residuals and Cook distances, and returns the values $s_e(e_i)$, ze_i, r_i, and d_i. A listing of the function follows:

```
function [e,stderr,ze,r,d] = residuals(XA,y)

//Multiple linear regression - residual analysis
//e      = residuals
//stderr = standard errors of residuals
//ze     = standardized residuals
//r      = studentized residuals
//d      = Cook's distances

[m n] = size(XA);           //Size of original X matrix
X = [ones(m,1) XA];         //Augmenting matrix X
H = X*inv(X'*X)*X';         //H ('hat') matrix
yh = H*y;                   //Fitted value of y
e=y-yh;                     //Errors or residuals
SSE=e'*e;                   //Sum of squared errors
MSE = SSE/(m-n-1);          //Mean square error
se = sqrt(MSE);             //Standard error of estimate
[nh mh] = size(H);          //Size of matrix H
h=[];                       //Vector h
for i=1:nh
    h=[h;H(i,i)];
end;

see = se*(1-h).^2;           //standard errors for residuals
ze   = e/se;                 //standardized residuals
r    = e./see;               //studentized residuals
d   = r.*r.*h./(1-h)/(n+1);  //Cook's distances

//Printing of results

printf(' ');
```



```
printf('Residual Analysis for Multiple Linear Regression');
printf('------------------------------------------------------------------------
----------');
printf('   i       y(i)      yh(i)       e(i)   se(e(i))      ze(i)       r(i)
d(i)');
printf('------------------------------------------------------------------------
----------');

for i = 1:m
  printf('%4.0f %10.6g %10.6g %10.6g %10.6g %10.6g %10.6g %10.6g',...
         i,y(i),yh(i),e(i),see(i),ze(i),r(i),d(i));
end;

printf('------------------------------------------------------------------------
----------');
printf('Standard error of estimate (se)                    = %g',se);
printf('------------------------------------------------------------------------
----------');

printf(' ');

//Plots of residuals - several options
xset('window',1);xset('mark',-9,2);

plot2d(yh,e,-9);
xtitle('Residual plot - residual vs. y','y','e');

xset('window',2);xset('mark',-9,2);
plot2d(yh,ze,-9);
xtitle('Residual plot - standardized residual vs. yh','yh','ze');

xset('window',3);xset('mark',-9,2);
plot2d(yh,r,-9);
xtitle('Residual plot - studentized residual vs. yh','yh','r');

xset('window',4);xset('mark',-9,2);
plot2d(yh,d,-9);
xtitle('Cook distance plot','yh','d');
```

Applications of function *residuals*

Applications of function *residuals* to the multiple linear fittings $p = p(x_1,x_2)$, $q = q(x_1,x_2)$, and $r = r(x_1,x_2)$, follow. First, we load function *residuals:*

```
-->getf('residuals')
```

Residual analysis for $p = p(x_1,x_2)$

```
-->residuals(X,p');
```



```
Residual Analysis for Multiple Linear Regression
-------------------------------------------------------------------------------
   i       y(i)      yh(i)       e(i)   se(e(i))    ze(i)       r(i)       d(i)
-------------------------------------------------------------------------------
   1    1524.74    1521.69    3.04682    8.02336    .13006    .37974   .0340687
   2    1600.81    1578.51    22.3035    13.2116    .95204   1.68817   .31503
   3    1630.64    1645.41   -14.7688    12.6407   -.63042  -1.16835   .16442
   4    1416.18    1424.51   -8.33603     13.589   -.35583   -.61344   .0392612
   5    1681.77    1678.61    3.16424    6.70004    .13507    .47227   .0646745
   6    1554.58    1565.08   -10.5036    15.6687   -.44835   -.67035   .0333676
   7    1317.48    1353.87   -36.3932    14.7089  -1.55347  -2.47422   .53468
   8    1324.61    1298.97    25.6415    10.9047   1.09453   2.35141   .85834
   9    1446.52    1418.35    28.1663    11.9558   1.2023    2.35587   .73966
  10    1328.93    1341.25   -12.3207    9.18077   -.52592  -1.34202   .35865
-------------------------------------------------------------------------------
Standard error of estimate (se)                  = 23.427
-------------------------------------------------------------------------------
```

Notice that most of the standardized residuals are within the interval (-2,2), and all of the values d_i are less than one. This residual analysis, thus, does not reveal any outliers. Similar results are obtained from the following table corresponding to the fitting $q = q(x_1,x_2)$.

Residual analysis for $q = q(x_1,x_2)$

```
-->residuals(X,q');
```

```
Residual Analysis for Multiple Linear Regression
-------------------------------------------------------------------------------
 i       y(i)      yh(i)       e(i)   se(e(i))      ze(i)       r(i)       d(i)
-------------------------------------------------------------------------------
 1    3585.74    3594.19   -8.45215    5.89407    -.49112   -1.43401     .48583
 2    3863.49    3869.77   -6.27968    9.70543    -.36489    -.64703   .0462771
 3    4205.43    4196.81    8.62144    9.28605     .50096     .92843     .10383
 4    3216.11    3221.54   -5.43054    9.98268    -.31555    -.54400   .0308755
 5     4409.4    4388.96    20.4452    4.92194      1.188    4.15388    5.00333
 6    3876.81    3891.61   -14.8006    11.5105    -.86001   -1.28584     .12277
 7    2975.01    2970.09    4.91247    10.8054     .28545     .45463   .0180525
 8    2764.65    2738.01    26.6417    8.01076    1.54806    3.32574    1.71704
 9    3289.16    3310.89   -21.7285    8.78291   -1.26257   -2.47396     .81567
10    2983.42    2987.34   -3.92933    6.74432    -.22832    -.58261   .0675956
-------------------------------------------------------------------------------
Standard error of estimate (se)                   = 17.2098
-------------------------------------------------------------------------------
```


Residual analysis for r = r(x_1,x_2)

```
-->residuals(X,r');

Residual Analysis for Multiple Linear Regression
-------------------------------------------------------------------------------
 i       y(i)      yh(i)       e(i)   se(e(i))      ze(i)       r(i)       d(i)
-------------------------------------------------------------------------------
 1    1558.51    1493.88     64.627    17.7424     1.2475    3.64252    3.13458
 2    1588.18    1558.26    29.9105    29.2154     .57737    1.02379     .11586
 3    1630.79    1634.99   -4.20322     27.953  -.0811352    -.15037  .00272348
 4    1371.75    1419.35   -47.6025      30.05    -.91888   -1.58411     .26181
 5    1658.57    1683.84   -25.2722    14.8161    -.48783   -1.70573     .84366
 6    1541.46    1574.41   -32.9551    34.6489    -.63614    -.95112   .0671713
 7    1315.83    1372.19   -56.3571    32.5265   -1.08787   -1.73265     .26221
 8    1296.27    1322.31   -26.0318    24.1141    -.50249   -1.07953     .18091
 9    1481.27    1455.37    25.8962    26.4384     .49988     .97949     .12786
10    1458.17    1386.18    71.9883    20.3018     1.3896     3.5459    2.50387
-------------------------------------------------------------------------------
Standard error of estimate (se)                   = 51.8051
-------------------------------------------------------------------------------
```

The table corresponding to the fitting $q=q(x_1,x_2)$ shows two residuals, e_1 and e_{10} whose Cook's distance is larger than one. These two points, even if their standardized residuals are in the interval (-2,2), may be considered outliers. The residual analysis eliminating these two suspected outliers is shown next.

Residual analysis for r = r(x_1,x_2) eliminating suspected outliers

To eliminate the outliers we modify matrix *X* and vector *r* as follows:

```
-->XX = X(2:9,:)
 XX  =


!   2.1    23.2 !
!   3.4    24.5 !
!   4.2    20.4 !
!   5.5    25.2 !
!   6.1    23.1 !
!   7.4    19.2 !
!   8.1    18.2 !
!   9.9    20.5 !

-->rr = r(2:9)
 rr  =

!   1588.175    1630.79    1371.75    1658.565    1541.455    1315.83
1296.275    1481.265 !



-->residuals(XX,rr');
```



```
Residual Analysis for Multiple Linear Regression
-------------------------------------------------------------------------------
 i       y(i)      yh(i)       e(i)   se(e(i))      ze(i)       r(i)       d(i)
-------------------------------------------------------------------------------
 1    1588.18    1542.72     45.459    10.9925    1.27186    4.13546    4.57875
 2    1630.79    1627.12    3.67448    17.1966     .10280     .21367  .00672191
 3    1371.75    1393.76    -22.008    14.0957    -.61574   -1.56133     .48136
 4    1658.57    1681.94   -23.3752    9.70312    -.65399   -2.40904    1.77832
 5    1541.46    1563.69   -22.2351    22.8787    -.62210    -.97187   .0786800
 6    1315.83    1345.33   -29.4969    18.9967    -.82527   -1.55274     .29870
 7    1296.27    1291.79    4.48228     12.618     .12541     .35523   .0287305
 8    1481.27    1437.77    43.4995    8.21833    1.21703    5.29299    10.1365
-------------------------------------------------------------------------------
Standard error of estimate (se)                   = 35.7423
-------------------------------------------------------------------------------
```

Even after eliminating the two influential observations we find that the remaining e_1 and e_8 are influential in the reduced data set. We can check the analysis of residuals eliminating these two influential observations as shown next:

```
-->XXX = XX(2:7), rrr = rr(2:7)
 XXX  =

!   3.4 !
!   4.2 !
!   5.5 !
!   6.1 !
!   7.4 !
!   8.1 !


 rrr  =

!   1630.79    1371.75    1658.565    1541.455    1315.83    1296.275 !

-->residuals(XXX,rrr');

Residual Analysis for Multiple Linear Regression
-------------------------------------------------------------------------------
 i       y(i)      yh(i)       e(i)   se(e(i))      ze(i)       r(i)       d(i)
-------------------------------------------------------------------------------
 1  1630.79   1601.8   28.9923   33.2701   .20573   .87142   .40176
 2    1371.75    1557.26   -185.509    65.1625   -1.31635   -2.84688     1.9071
 3    1658.57    1484.88     173.68    96.7163    1.23241    1.79577     .33395
 4    1541.46    1451.48    89.9739    96.4308     .63844     .93304   .0909297
 5    1315.83    1379.11   -63.2765     63.918    -.44900    -.98996     .23759
 6    1296.27    1340.14   -43.8605    35.9466    -.31123   -1.22016     .72952
-------------------------------------------------------------------------------
Standard error of estimate (se)                   = 140.927
-------------------------------------------------------------------------------
```

Even after eliminating the two points e_1 and e_8 from the reduced data set, another influential point is identified, e_2. We may continue eliminating influential points, at the risk of running out of data, or try a different data fitting.

Multiple linear regression with function *datafit*

SCILAB provides function *datafit*, introduced in Chapter 8, which can be used to determine the coefficients of a fitting function through a least-square criteria. Function *datafit* is used for fitting data to a model by defining an error function $e = G(b,x)$ where b is a column vector of m rows representing the parameters of the model, and x is a column vector of n rows representing the variables involved in the model. Function *datafit* finds a solution to the set of k equations $e_i = G(b,x_i) = 0$.

The simplest call to function *datafit* is

[p,err]=datafit(G,X,b0)

where *G* is the name of the error function *G(b,x)*, *X* is a matrix whose rows consists of the different vectors of variables, i.e., $X = [x_1; x_2; \ldots; x_n]$, and *b0* is a column vector representing initial guesses of the parameters *b* sought.

Function *datafit* can be used to determine the coefficients of a multiple linear fitting as illustrated in the example below. The data to be fit is given in the following table:

x_1	x_2	x_3	x_4	y
25	24	91	100	240
31	21	90	95	236
45	24	88	110	290
60	25	87	88	274
65	25	91	94	301
72	26	94	99	316
80	25	87	97	300
84	25	86	96	296
75	24	88	110	267
60	25	91	105	276
50	25	90	100	288
38	23	89	98	261

First, we load the data and prepare the matrix XX for the application of function *datafit*.

```
-->XY=...
-->[25 24     91     100    240
-->31  21     90     95     236
-->45  24     88     110    290
-->60  25     87     88     274
-->65  25     91     94     301
-->72  26     94     99     316
-->80  25     87     97     300
-->84  25     86     96     296
-->75  24     88     110    267
-->60  25     91     105    276
-->50  25     90     100    288
-->38  23     89     98     261];

-->XX=XY';
```


Next, we define the error function to be minimized and call function *datafit*:

```
-->deff('[e]=G(b,z)',...
--> 'e=b(1)+b(2)*z(1)+b(3)*z(2)+b(4)*z(3)+b(5)*z(4)-z(5)')

-->[b,er]=datafit(G,XX,b0)
 er  = 1699.0093
 b  =
! - 102.71289 !
!    .6053697 !
!   8.9236567 !
!   1.4374508 !
!    .0136086 !
```

You can check these results using function *multiplelinear* as follows:

```
-->size(XY)
 ans  =
!   12.    5. !

-->X=XY(1:12,1:4);y=XY(1:12,5);   //Prepare matrix X and vector y
-->[bb,C,se]=multiplelinear(X,y,0.10);  //Call function multiplelinear

Multiple linear regression
==========================

Table of coefficients
-------------------------------------------------------------------------
   i       b(i)   se(b(i))      Lower      Upper         t0  H0:b(i)=0
-------------------------------------------------------------------------
   0   -102.713    207.859   -489.237     283.81    -.49415  do not reject
   1     .60537     .36890  -.0806111    1.29135    1.64103  do not reject
   2    8.92364    5.30052    -.93293    18.7802    1.68354  do not reject
   3    1.43746    2.39162   -3.00988    5.88479     .60104  do not reject
   4   .0136093     .73382   -1.35097    1.37819   .0185458  do not reject
-------------------------------------------------------------------------
                                       t_alpha/2 = 1.85955
-------------------------------------------------------------------------

Table of fitted values and errors
-------------------------------------------------------------------------------
   i     y(i)      yh(i)       e(i)        C.I. for Y          C.I. for Ypred
-------------------------------------------------------------------------------
   1      240    258.758    -18.758     248.31    269.206    214.676     302.84
   2      236    234.114    1.88623     219.86    248.367    175.738     292.49
   3      290    266.689    23.3109    255.836    277.542    221.107    312.271
   4      274    282.956   -8.95646    271.601    294.312    235.506    330.407
   5      301    291.815    9.18521    284.131    299.498     257.72    325.909
   6      316    309.356    6.64355    296.747    321.966    257.204    361.509
   7      300    295.186    4.81365     287.17    303.203    259.916    330.457
   8      296    296.157    -.15677    286.462    305.851    254.842    337.472
   9      267     284.85   -17.8502    273.464    296.236    237.285    332.415
  10      276    288.938   -12.9376     281.73    296.145    256.503    321.372
  11      288    281.378    6.62157    274.517     288.24    250.134    312.622
  12      261    254.802    6.19798     247.76    261.844     222.94    286.664
-------------------------------------------------------------------------------
```



```
Analysis of variance
-------------------------------------------------------
Source of      Sum of   Degrees of       Mean
variation      squares     freedom     square         F0
-------------------------------------------------------
Regression      4957.24          4    1239.31    5.10602
Residual        1699.01          7    242.716
Total           6656.25         11
-------------------------------------------------------
With F0 = 5.10602 and F_alpha = 2.96053,
reject the null hypothesis H0:beta1=beta2=...=betan=0.
-------------------------------------------------------


Additional information
-----------------------------------------------------------
Standard error of estimate (se)                = 15.5793
Coefficient of multiple determination (R^2)    =  .74475
Multiple correlation coefficient (R)           =  .86299
Adjusted coefficient of multiple determination =  .59889
-----------------------------------------------------------

Covariance matrix:

!   43205.302  - 10.472107  - 142.32333  - 389.41606  - 43.653591 !
! - 10.472107     .1360850  - 1.4244286     .4289197  -  .0095819 !
! - 142.32333  - 1.4244286    28.095536  - 5.3633513     .1923068 !
! - 389.41606     .4289197  - 5.3633513    5.7198487  -  .1563728 !
! - 43.653591  -  .0095819     .1923068  -  .1563728     .5384939 !

-----------------------------------------------------------
```

Notice that the error, *e*, returned by *datafit* is the sum of squared errors, *SSE*, returned by function *multiplelinear*. The detailed regression analysis provided by *multiplelinear* indicates that the hypothesis for significance of regression is to be rejected, i.e., the linear model is not necessarily the best for this data set. Also, the coefficient of multiple regression and its adjusted value are relatively small.

Note: Function *datafit* can be used to fit linear and non-linear functions. Details of the application of function *datafit* were presented in Chapter 8.

Polynomial data fitting

A function for polynomial data fitting was developed in Chapter 5 to illustrate the use of matrix operations. In this section, we present the analysis of polynomial data fitting as a special case of a multilinear regression. A polynomial fitting is provided by a function of the form

$$\hat{y} = b_0 + b_1 z + b_2 z^2 + \ldots + b_n z^n.$$

This is equivalent to a multiple linear fitting if we take $x_1 = z$, $x_2 = z^2$, …, $x_n = z^n$. Thus, given a data set $\{(z_1,y_1), (z_2,y_2), \ldots, (z_m,y_m)\}$, we can use function *multiplelinear* to obtain the coefficients $[b_0,b_1,b_2,\ldots,b_n]$ of the polynomial fitting. As an example, consider the fitting of a cubic polynomial to the data given in the following table:

z	*2.10*	*3.20*	*4.50*	*6.80*	*13.50*	*18.40*	*21.00*
y	*13.41*	*46.48*	*95.39*	*380.88*	*2451.55*	*5120.46*	*8619.14*

The SCILAB instructions for producing the fitting are shown next. First, we load the data:

```
-->z=[2.10,3.20,4.50,6.80,13.50,18.40,21.00];

-->y=[13.41,46.48,95.39,380.88,2451.55,5120.46,8619.14];
```

Next, we prepare the vectors for the multiple linear regression and call the appropriate function:

```
-->x1 = z; x2 = z^2; x3 = z^3; X = [x1' x2' x3']; yy = y';


-->[b,C,se]=multiplelinear(X,yy,0.01)

Multiple linear regression
==========================

Table of coefficients
-------------------------------------------------------------------------
   i       b(i)   se(b(i))      Lower      Upper         t0  H0:b(i)=0
-------------------------------------------------------------------------
   0   -467.699    664.835   -3528.66    2593.26    -.70348  do not reject
   1    223.301    262.938   -987.289    1433.89     .84925  do not reject
   2   -23.3898    26.1662   -143.861    97.0817    -.89390  do not reject
   3    1.56949     .74616   -1.86592    5.00491    2.10341  do not reject
-------------------------------------------------------------------------
                                      t_alpha/2 = 4.60409
-------------------------------------------------------------------------

Table of fitted values and errors
-------------------------------------------------------------------------------
   i     y(i)      yh(i)       e(i)        C.I. for Y          C.I. for Ypred
-------------------------------------------------------------------------------
   1    13.41   -87.3819    100.792   -351.859    177.095   -4793.08    4618.32
   2    46.48      58.78      -12.3   -115.377    232.937   -3048.97    3166.53
   3    95.39    206.529   -111.139    25.4059    387.653   -3024.28    3437.34
   4   380.88    462.697   -81.8173    221.167    704.228   -3836.65    4762.04
   5  2451.55     2145.6     305.95     1889.3     2401.9   -2415.28    6706.48
   6  5120.46    5499.33   -378.865     5272.7    5725.95     1463.9    9534.75
   7  8619.14    8441.76     177.38    8143.74    8739.78    3141.72    13741.8
-------------------------------------------------------------------------------


Analysis of variance
-----------------------------------------------------
Source of      Sum of   Degrees of       Mean
variation      squares     freedom     square         F0
-----------------------------------------------------
Regression   6.64055e+07          3 2.21352e+07    222.864
Residual         297964          3    99321.4
Total        6.67034e+07          6
-----------------------------------------------------
With F0 = 222.864 and F_alpha = 29.4567,
reject the null hypothesis H0:beta1=beta2=...=betan=0.
-----------------------------------------------------
```



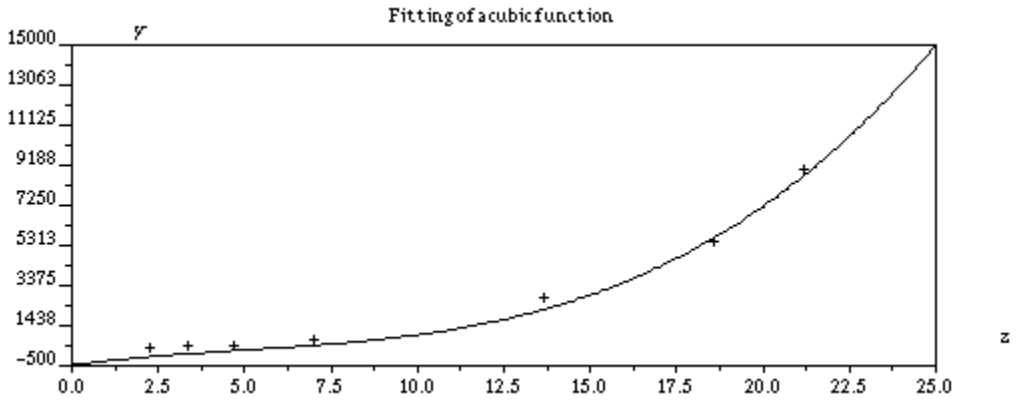
```
Additional information
-----------------------------------------------------------
Standard error of estimate (se)                = 315.153
Coefficient of multiple determination (R^2)    =  .99553
Multiple correlation coefficient (R)           =  .99776
Adjusted coefficient of multiple determination =  .99107
-----------------------------------------------------------

Covariance matrix:

!   442005.47  - 166781.09    15518.351  - 412.44179 !
! - 166781.09    69136.156  - 6728.9534    183.73635 !
!   15518.351  - 6728.9534    684.66847  - 19.297326 !
! - 412.44179    183.73635  - 19.297326     .5567618 !

-----------------------------------------------------------
```

With the vector of coefficients found above, we define the cubic function:

```
-->deff('[y]=yf(z)','y=b(1)+b(2)*z+b(3)*z^2+b(4)*z^3')
```

Then, we produce the fitted data and plot it together with the original data:

```
-->zz=[0:0.1:25];yz=yf(zz);

-->rect = [0 -500 25 15000]; //Based on min. & max. values of z and y

-->xset('mark',-1,2)

-->plot2d(zz,yz,1,'011',' ',rect)

-->plot2d(z,y,-1,'011',' ',rect)

-->xtitle('Fitting of a cubic function','z','y')
```



Alternatively, we can use function *datafit* to obtain the coefficients of the fitting as follows:

```
-->XX = [z;y]

 XX  =

!   2.1      3.2      4.5      6.8       13.5       18.4       21.      !
!   13.41    46.48    95.39    380.88    2451.55    5120.46    8619.14 !

-->b0=ones(4,1)
 b0  =

!   1. !
!   1. !
!   1. !
!   1. !

-->[b,er]=datafit(G,XX,b0)
 er  =

    297964.09
 b  =

! - 467.6971  !
!   223.30004 !
! - 23.389786 !
!   1.5694905 !
```


Exercises

[1]. To analyze the dependency of the mean annual flood, *Q(cfs)*, on the drainage area, *A(mi^2)* for a given region, data from six experimental watersheds is collected. The data is summarized in the table below:

A(mi^2)	*16.58*	*3.23*	*16.8*	*42.91*	*8.35*	*6.04*
Q(cfs)	*455*	*105*	*465*	*1000*	*290*	*157*

(a) Use function *linreg* to perform a simple linear regression analysis on these data. The purpose is to obtain a relationship of the form $Q = mA + b$. (b) Determine the covariance of A and B, (c)the correlation coefficient, and (d) the standard error of the estimate.

[2] For the data of problem [1] use function *linregtable* to perform the linear fitting. (a) What is the decision regarding the hypothesis that the slope of the linear fitting may be zero at a level of confidence of 0.05? (b) What is the decision regarding the hypothesis that the intercept of the linear fitting may be zero at a level of confidence of 0.05?

[3] For the data of problem [1] use function *multiplelinear* to produce the data fitting. (a) What is the decision regarding the hypothesis that the slope of the linear fitting may be zero at a level of confidence of 0.05? (b) What is the decision regarding the hypothesis that the intercept of the linear fitting may be zero at a level of confidence of 0.05? (b) What is the decision regarding the hypothesis that the linear fitting may not apply at all?

[4]. The data shown in the table below represents the monthly precipitation, *P(in)*, in a particular month, and the corresponding runoff, *R(in)*, out of a specific hydrological basin for the period 1960-1969.

Year	*1960*	*1961*	*1962*	*1963*	*1964*	*1965*	*1966*	*1967*	*1968*	*1969*
P (in)	*1.95*	*10.8*	*3.22*	*4.51*	*6.71*	*1.18*	*4.82*	*6.38*	*5.97*	*4.64*
R (in)	*0.46*	*2.85*	*0.99*	*1.4*	*1.98*	*0.45*	*1.31*	*2.22*	*1.36*	*1.21*

(a) Use function *linreg* to perform a simple linear regression analysis on these data. The purpose is to obtain a relationship of the form $R = mP + b$. (b) Determine the covariance of P and R, (c)the correlation coefficient, and (d) the standard error of the estimate.

[5] For the data of problem [1] use function *linregtable* to perform the linear fitting. (a) What is the decision regarding the hypothesis that the slope of the linear fitting may be zero at a level of confidence of 0.05? (b) What is the decision regarding the hypothesis that the intercept of the linear fitting may be zero at a level of confidence of 0.05?

[6] For the data of problem [1] use function *multiplelinear* to produce the data fitting. (a) What is the decision regarding the hypothesis that the slope of the linear fitting may be zero at a level of confidence of 0.05? (b) What is the decision regarding the hypothesis that the intercept of the linear fitting may be zero at a level of confidence of 0.05? (b) What is the decision regarding the hypothesis that the linear fitting may not apply at all?

[7]. The following table shows data indicating the monthly precipitation during the month of February, P_f, and during the month of March, P_m, as well as the runoff during the month of March, R_m, for a specific watershed during the period 1935-1958.

Year	Pm	Pf	Rm
1935	9.74	4.11	6.15
1936	6.01	3.33	4.93
1937	1.30	5.08	1.42
1938	4.80	2.41	3.60
1939	4.15	9.64	3.54
1940	5.94	4.04	2.26
1941	2.99	0.73	0.81
1942	5.11	3.41	2.68
1943	7.06	3.89	4.68
1944	6.38	8.68	5.18
1945	1.92	6.83	2.91
1946	2.82	5.21	2.84
1947	2.51	1.78	2.02
1948	5.07	8.39	3.27
1949	4.63	3.25	3.05
1950	4.24	5.62	2.59
1951	6.38	8.56	4.66
1952	7.01	1.96	5.40
1953	4.15	5.57	2.60
1954	4.91	2.48	2.52
1955	8.18	5.72	6.09
1956	5.85	10.19	4.58
1957	2.14	5.66	2.02
1958	3.06	3.04	2.59

a. Use function *multiplot* to show the dependence of the many variables.
b. Use function *multiplelinear* to check the multiple linear fitting $R_m = b_0 + b_1P_m + b_2P_f$.
c. For a level of confidence of 0.05, what are the decisions regarding the hypotheses that each of the coefficients may be zero?
d. What is the decision regarding the hypothesis that the linear fitting may not apply at all for the same level of confidence?
e. What value of runoff for the month of March is predicted if the precipitation in the month of March is 6.2 in, and that of the month of February is 3.2 in?
f. What are the confidence intervals for the mean value and the prediction for the data of question (e) at a confidence level 0.05?

[8]. In the analysis of runoff produced by precipitation into a watershed, often we are required to estimate a parameter known as the time of concentration (t_c) which determines the time to the peak of the hydrograph produced by the watershed. It is assumed that the time of concentration is a function of a characteristic watershed length (L), of a characteristic watershed slope (S), and of a parameter known as the runoff curve number *(CN)*. Runoff curve numbers are numbers used by the U.S. Soil Conservation Service in the estimation of runoff from watersheds. Runoff curve numbers are typically functions of the location of the watershed and of its soil and vegetation covers. The following table shows values of the time of concentration, t_c*(hr)*, the watershed length, *L(ft)*, the watershed slope, *S(%)*, and the runoff curve number *(CN)* for 5 experimental watersheds.

t_c(hr)	0.2	0.2	0.2	0.3	0.3
L (ft)	800	1200	2100	2000	1500
S (%)	2	3	4	6	1
CN	75	84	88	70	85

(a) Use function *multiplot* to show the interdependence of the various variables in the table. (b) Assuming that a multiple-linear fitting can be used to explain the dependence of t_c on *L*, *S*, and *CN*, use function *multiplelinear* to determine the coefficients of the fitting. (c) For a level of confidence of 0.01, what are the decisions regarding the hypotheses that each of the coefficients may be zero? (d) What is the decision regarding the hypothesis that the linear fitting may not apply at all for the same level of confidence? (e) What value of the time of concentration is predicted for *L = 1750 ft, S = 5%,* and *CN = 80*. (f) What are the confidence intervals for the mean value and the prediction for the data of question (e) at a confidence level 0.05?

[9]. The data in the table below shows the peak discharge, q_p*(cfs)*, the rainfall intensity, *i(in/hr)*, and the drainage area, *A(acres)*, for rainfall events in six different watersheds.

q_p(cfs)	23	45	44	64	68	62
i(in/hr)	3.2	4.6	5.1	3.8	6.1	7.4
A(acres)	12	21	18	32	24	16

(a) Use function *multiplot* to show the interdependence of the various variables in the table. (b) Assuming that a multiple-linear fitting can be used to explain the dependence of q_p on *i*, and *A*, use function *multiplelinear* to determine the coefficients of the fitting. (c) For a level of confidence of 0.1, what are the decisions regarding the hypotheses that each of the coefficients may be zero? (d) What is the decision regarding the hypothesis that the linear fitting may not apply at all for the same level of confidence? (e) What value of the time of concentration is predicted for *i = 5.6 in/hr* and *A = 25 acres*. (f) What are the confidence intervals for the mean value and the prediction for the data of question (e) at a confidence level 0.10?

[10]. Measurements performed across a pipeline diameter produce the following table of velocities, *v(fps)*, as function of the radial distance, *r(in)*, measured from the pipe centerline.

r(in)	V(fps)	r(in)	V(fps)	r(in)	V(fps)	r(in)	V(fps)
2.7	57.73	1.3	64.40	0.9	65.50	2.5	58.90
2.6	58.30	1.2	64.80	1.1	64.80	2.6	58.40
2.5	59.25	1.1	64.80	1.3	64.10	2.7	57.50
2.4	59.70	1.0	65.20	1.4	63.70	2.8	57.00
2.3	59.80	0.9	65.50	1.5	63.40	2.9	56.60
2.2	60.60	0.8	65.50	1.6	63.00	3.0	55.90
2.1	61.20	0.7	65.90	1.7	62.70	3.1	54.80
2.0	61.50	0.5	66.30	1.8	62.50	3.2	54.20
1.9	62.20	0.3	66.40	1.9	62.10	3.3	53.20
1.8	62.70	0.1	66.50	2.0	61.25	3.4	52.35
1.7	62.90	0.1	66.50	2.1	61.20	3.5	50.80
1.6	63.05	0.3	66.30	2.2	60.55	3.6	49.50
1.5	63.65	0.5	66.00	2.3	60.00	3.7	47.70
1.4	64.10	0.7	65.70	2.4	59.40	3.8	44.45

(a) Plot the velocity profile indicated in the table. Notice that the values of *r* start at 2.7 in and decrease to 0.1in, just to increase again from 0.1 in to 3.8 in. What these values of r represent are velocity measurements at both sides of the centerline along a diameter of the pipe. To produce an accurate plot of the velocity distribution, take the values of r from 2.7 in down to 0.1 in as negative, and the remaining values as positive. (b) Using SCILAB function *datafit,* fit a logarithmic function of the form $v = b_0 + b_1 \, ln(y)$ to the data, where y = R-r, and R is the pipe radius measured as R = 3.958 in.

[11]. The tables below show measurements of stagnation pressure on an air jet in a test set up. The values of *x* represent distances from the jet outlet, where the values of *r* represent distances from the jet centerline measured radially outwards. The stagnation pressure values are used to calculate air velocities at the locations indicated by using $v = (2gh)^{1/2}$, where *g* is the acceleration of gravity. To make the units consistent, we recommend that you transform the data to *feet* by using *1 in = (1/12) ft*, and *1 cm = 0.0328 ft*, and calculate the velocities using $g = 32.2 \; ft/s^2$.

Along centerline		*Across jet at x = 12 in*			
x(in)	h(cm)	r(in)	h(cm)	r(in)	h(cm)
-0.25	20.58	8.50	0.00	-0.50	6.08
0.00	20.38	5.00	0.00	-1.00	5.06
0.50	20.16	4.50	0.15	-1.50	3.97
1.00	20.16	4.20	0.20	-2.00	2.73
3.00	20.16	3.70	0.47	-2.50	1.78
5.00	19.6	3.50	0.62	-3.00	1.11
6.00	18.25	3.09	0.94	-3.50	0.70
6.50	17.11	2.83	1.19	-4.20	0.25

8.00	*13.52*	*2.55*	*1.62*	*-4.50*	*0.17*
10.00	*9.28*	*2.00*	*2.62*	*-4.80*	*0.12*
15.00	*4.14*	*1.50*	*3.91*	*-5.00*	*0.07*
20.00	*2.23*	*1.00*	*5.12*	*-5.30*	*0.02*
		0.50	*6.07*	*-5.50*	*0.00*
		0.00	*6.51*		

(a) Convert the data columns to feet and calculate the velocities corresponding to the different values of h. (b) Plot the centerline velocities against the distance x and fit an equation of the form

$$v(x) = b_0/(b_1 + b_2 x)$$

to the data resulting from the first table. (c) Plot the velocity v at section $x = 12$ *in* against the radial distance $|r|$, and fit an equation of the form

$$v(r) = b_0 \cdot exp(-b_1 x^2)$$

to the data resulting from the second table.

[12]. For relatively low pressures, Henry's law relates the vapor pressure of a gas, P, to the molar fraction of the gas in mixture, x. The law is stated as $P = kx$, where k is known as *Henry's constant*. To determine Henry's constant in practice we use data of P against x and fit a straight line, i.e., $P = b_0 + b_1 x$. If the value of b_0 can be taken as zero, then, $b_1 = k$.

Given the pressure-molar fraction data shown in the next table, use function *linreg* to determine Henry's constant for the solubility of the following elements or compouns in water at temperature indicated:

Sulfur dioxide, 23°C		*Carbon monoxide, 19°C*		*Hydrogen, 23°C*	
P(mmHg)	*x(10³)*	*P(mmHg)*	*x(10³)*	*P(mmHg)*	*x(10³)*
5	*0.3263*	*900*	*2.417*	*1100*	*1.861*
10	*0.5709*	*1000*	*2.685*	*2000*	*3.382*
50	*2.329*	*2000*	*5.354*	*3000*	*5.067*
100	*4.213*	*3000*	*8.000*	*4000*	*6.729*
200	*7.448*	*4000*	*10.630*	*6000*	*9.841*
300	*10.2*	*5000*	*13.230*	*8000*	*12.560*
		6000	*15.800*		
		7000	*18.280*		
		8000	*20.670*		

[13]. In the analysis of liquid mixtures it is often necessary to determine parameters known as *activity coefficients*. For the mixture of two liquids the *van Laar equations* can be used to determine the activity coefficients, γ_1 and γ_2, in terms of the molecular fractions x_1 and x_2:

$$\ln\gamma_1 = \frac{A}{\left(1+\frac{A\cdot x_1}{B\cdot x_2}\right)^2}, \quad \ln\gamma_2 = \frac{A}{\left(1+\frac{B\cdot x_1}{A\cdot x_2}\right)^2}.$$

The molecular fractions are related by

$$x_1 + x_2 = 1.$$

The table below shows the activity coefficients for a liquid mixture as functions of the molecular fraction x_1. Use these data and SCILAB function *datafit* to obtain the values of the van Laar coefficients *A* and *B* for the data.

x_1	γ_1	γ_2
0.1	4.90	1.05
0.2	2.90	1.20
0.3	1.95	1.30
0.4	1.52	1.50
0.5	1.30	1.70
0.6	1.20	2.00
0.7	1.10	2.25
0.8	1.04	2.60
0.9	1.01	2.95

[14]. An alternative relationship between the activity coefficients, γ_1 and γ_2, in terms of the molecular fractions x_1 and x_2 are the Margules' equations:

$$ln\ \gamma_1 = (2B\text{-}A)x_2^2 + 2(A\text{-}B)x_2^3$$

$$ln\ \gamma_2 = (2A\text{-}B)x_1^2 + 2(B\text{-}A)x_2^3.$$

Using the data of problem [13] and SCILAB function *datafit*, determine the coefficients A and B of the Margule's equations.

[15]. Infiltration into soil is typically modeled using Horton's equation

$$f = f_c + (f_0\text{-}f_c)e^{-kt},$$

where f is the infiltration rate, f_c is the infiltration rate at saturation, f_0 is the initial infiltration rate, t is time, and k is a constant that depends primarily on the type of soil and vegetation of the area of interest.

The following table shows measurements of the infiltration rate, f, as function of time, t, for a specific storm in a watershed.

t(hr)	f(cm/hr)	t(hr)	f(cm/hr)
1	3.9	14	1.43
2	3.4	16	1.36
3	3.1	18	1.31
4	2.7	20	1.28
5	2.5	22	1.25
6	2.3	24	1.23
8	2	26	1.22
10	1.8	28	1.2
12	1.54	30	1.2

(a) Use SCILAB's function *datafit* to obtain the parameters f_0, f_c, and k for the Horton's equation.
(b) Plot the original data and the fitted data in the same set of axes.

[16]. The following data represents different properties of granite samples taken at the locations indicated by the coordinates *x(mi)* and *y(mi)* on a specific site. The properties listed in the table are as follows: x_1 = percentage of quartz in the sample, x_2 = color index (a percentage), x_3 = percentage of total feldspar, w = specific gravity (water = 1.0).

x_1	x_2	x_3	w	y	x
21.3	5.5	73.0	2.63	0.920	6.090
38.9	2.7	57.4	2.64	1.150	3.625
26.1	11.1	62.6	2.64	1.160	6.750
29.3	6.0	63.6	2.63	1.300	3.010
24.5	6.6	69.1	2.64	1.400	7.405
30.9	3.3	65.1	2.61	1.590	8.630
27.9	1.9	69.1	2.63	1.750	4.220
22.8	1.2	76.0	2.63	1.820	2.420
20.1	5.6	74.1	2.65	1.830	8.840
16.4	21.3	61.7	2.69	1.855	10.920
15.0	18.9	65.6	2.67	2.010	14.225
0.6	35.9	62.5	2.83	2.040	10.605
18.4	16.6	64.9	2.70	2.050	8.320
19.5	14.2	65.4	2.68	2.210	8.060
34.4	4.6	60.7	2.62	2.270	2.730
26.9	8.6	63.6	2.63	2.530	3.500
28.7	5.5	65.8	2.61	2.620	7.445
28.5	3.9	67.8	2.62	3.025	5.060
38.4	3.0	57.6	2.61	3.060	5.420
28.1	12.9	59.0	2.63	3.070	12.550
37.4	3.5	57.6	2.63	3.120	12.130

0.9	*22.9*	*74.4*	*2.78*	*3.400*	*15.400*
8.8	*34.9*	*55.4*	*2.76*	*3.520*	*9.910*
16.2	*5.5*	*77.6*	*2.63*	*3.610*	*11.520*
2.2	*28.4*	*69.3*	*2.74*	*4.220*	*16.400*
29.1	*5.1*	*65.7*	*2.64*	*4.250*	*11.430*
24.9	*6.9*	*67.8*	*2.70*	*4.940*	*5.910*
39.6	*3.6*	*56.6*	*2.63*	*5.040*	*1.840*
17.1	*11.3*	*70.9*	*2.71*	*5.060*	*11.760*
0.0	*47.8*	*52.2*	*2.84*	*5.090*	*16.430*
19.9	*11.6*	*67.2*	*2.68*	*5.240*	*11.330*
1.2	*34.8*	*64.0*	*2.84*	*5.320*	*8.780*
13.2	*18.8*	*67.4*	*2.74*	*5.320*	*13.730*
13.7	*21.2*	*64.0*	*2.74*	*5.330*	*12.450*
26.1	*2.3*	*71.2*	*2.61*	*5.350*	*1.430*
19.9	*4.1*	*76.0*	*2.63*	*5.610*	*4.150*
4.9	*18.8*	*74.30*	*2.77*	*5.850*	*13.840*
15.5	*12.2*	*69.70*	*2.72*	*6.460*	*11.660*
0.0	*39.7*	*60.20*	*2.83*	*6.590*	*14.640*
4.5	*30.5*	*63.90*	*2.77*	*7.260*	*12.810*
0.0	*63.8*	*35.20*	*2.92*	*7.420*	*16.610*
4.0	*24.1*	*71.80*	*2.77*	*7.910*	*14.650*
23.4	*12.4*	*63.10*	*2.79*	*8.470*	*13.330*
29.5	*9.8*	*60.40*	*2.69*	*8.740*	*15.770*

(a) Use function *multiplot* to show the interdependence of the various variables in the table. (b) Assuming that a multiple-linear fitting can be used to explain the dependence of w on x_1, x_2, and x_3, use function *multiplelinear* to determine the coefficients of the fitting. (c) For a level of confidence of 0.1, what are the decisions regarding the hypotheses that each of the coefficients may be zero? (d) What is the decision regarding the hypothesis that the linear fitting may not apply at all for the same level of confidence? (e) What value of the time of concentration is predicted for $x_1 = 17$, $x_2 = 25$, and $x_3 = 55$. (f) What are the confidence intervals for the mean value and the prediction for the data of question (e) at a confidence level 0.10?

REFERENCES (for all SCILAB documents at InfoClearinghouse.com)

Abramowitz, M. and I.A. Stegun (editors), 1965,"*Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*," Dover Publications, Inc., New York.

Arora, J.S., 1985, "*Introduction to Optimum Design*," Class notes, The University of Iowa, Iowa City, Iowa.

Asian Institute of Technology, 1969, "*Hydraulic Laboratory Manual*," AIT - Bangkok, Thailand.

Berge, P., Y. Pomeau, and C. Vidal, 1984,"*Order within chaos - Towards a deterministic approach to turbulence*," John Wiley & Sons, New York.

Bras, R.L. and I. Rodriguez-Iturbe, 1985,"*Random Functions and Hydrology*," Addison-Wesley Publishing Company, Reading, Massachussetts.

Brogan, W.L., 1974,"*Modern Control Theory*," QPI series, Quantum Publisher Incorporated, New York.

Browne, M., 1999, "*Schaum's Outline of Theory and Problems of Physics for Engineering and Science*," Schaum's outlines, McGraw-Hill, New York.

Farlow, Stanley J., 1982, "*Partial Differential Equations for Scientists and Engineers*," Dover Publications Inc., New York.

Friedman, B., 1956 (reissued 1990), "*Principles and Techniques of Applied Mathematics*," Dover Publications Inc., New York.

Gomez, C. (editor), 1999, "*Engineering and Scientific Computing with Scilab*," Birkhäuser, Boston.

Gullberg, J., 1997, "*Mathematics - From the Birth of Numbers*," W. W. Norton & Company, New York.

Harman, T.L., J. Dabney, and N. Richert, 2000, "*Advanced Engineering Mathematics with MATLAB® - Second edition*," Brooks/Cole - Thompson Learning, Australia.

Harris, J.W., and H. Stocker, 1998, "*Handbook of Mathematics and Computational Science*," Springer, New York.

Hsu, H.P., 1984, "*Applied Fourier Analysis*," Harcourt Brace Jovanovich College Outline Series, Harcourt Brace Jovanovich, Publishers, San Diego.

Journel, A.G., 1989, "*Fundamentals of Geostatistics in Five Lessons*," Short Course Presented at the 28th International Geological Congress, Washington, D.C., American Geophysical Union, Washington, D.C.

Julien, P.Y., 1998,"*Erosion and Sedimentation*," Cambridge University Press, Cambridge CB2 2RU, U.K.

Keener, J.P., 1988, "*Principles of Applied Mathematics - Transformation and Approximation*," Addison-Wesley Publishing Company, Redwood City, California.

Kitanidis, P.K., 1997,"*Introduction to Geostatistics - Applications in Hydogeology*," Cambridge University Press, Cambridge CB2 2RU, U.K.

Koch, G.S., Jr., and R. F. Link, 1971, "*Statistical Analysis of Geological Data - Volumes I and II*," Dover Publications, Inc., New York.

Korn, G.A. and T.M. Korn, 1968, "*Mathematical Handbook for Scientists and Engineers*," Dover Publications, Inc., New York.

Kottegoda, N. T., and R. Rosso, 1997, "*Probability, Statistics, and Reliability for Civil and Environmental Engineers*," The Mc-Graw Hill Companies, Inc., New York.

Kreysig, E., 1983, "*Advanced Engineering Mathematics - Fifth Edition*," John Wiley & Sons, New York.

Lindfield, G. and J. Penny, 2000, "*Numerical Methods Using Matlab®,"* Prentice Hall, Upper Saddle River, New Jersey.

Magrab, E.B., S. Azarm, B. Balachandran, J. Duncan, K. Herold, and G. Walsh, 2000, "*An Engineer's Guide to MATLAB®*", Prentice Hall, Upper Saddle River, N.J., U.S.A.

McCuen, R.H., 1989,"*Hydrologic Analysis and Design - second edition*," Prentice Hall, Upper Saddle River, New Jersey.

Middleton, G.V., 2000, "*Data Analysis in the Earth Sciences Using Matlab®,"* Prentice Hall, Upper Saddle River, New Jersey.

Montgomery, D.C., G.C. Runger, and N.F. Hubele, 1998, "*Engineering Statistics*," John Wiley & Sons, Inc.

Newland, D.E., 1993, "*An Introduction to Random Vibrations, Spectral & Wavelet Analysis - Third Edition*," Longman Scientific and Technical, New York.

Nicols, G., 1995, "*Introduction to Nonlinear Science*," Cambridge University Press, Cambridge CB2 2RU, U.K.

Parker, T.S. and L.O. Chua, , "*Practical Numerical Algorithms for Chaotic Systems*," 1989, Springer-Verlag, New York.

Peitgen, H-O. and D. Saupe (editors), 1988, "*The Science of Fractal Images*," Springer-Verlag, New York.

Peitgen, H-O., H. Jürgens, and D. Saupe, 1992, "*Chaos and Fractals - New Frontiers of Science*," Springer-Verlag, New York.

Press, W.H., B.P. Flannery, S.A. Teukolsky, and W.T. Vetterling, 1989, "*Numerical Recipes - The Art of Scientific Computing (FORTRAN version)*," Cambridge University Press, Cambridge CB2 2RU, U.K.

Raghunath, H.M., 1985, "*Hydrology - Principles, Analysis and Design*," Wiley Eastern Limited, New Delhi, India.

Recktenwald, G., 2000, "*Numerical Methods with Matlab - Implementation and Application*," Prentice Hall, Upper Saddle River, N.J., U.S.A.

Rothenberg, R.I., 1991, "*Probability and Statistics*," Harcourt Brace Jovanovich College Outline Series, Harcourt Brace Jovanovich, Publishers, San Diego, CA.

Sagan, H., 1961,"*Boundary and Eigenvalue Problems in Mathematical Physics*," Dover Publications, Inc., New York.

Spanos, A., 1999,"*Probability Theory and Statistical Inference - Econometric Modeling with Observational Data*," Cambridge University Press, Cambridge CB2 2RU, U.K.

Spiegel, M. R., 1971 (second printing, 1999), "*Schaum's Outline of Theory and Problems of Advanced Mathematics for Engineers and Scientists*," Schaum's Outline Series, McGraw-Hill, New York.

Tanis, E.A., 1987, "*Statistics II - Estimation and Tests of Hypotheses*," Harcourt Brace Jovanovich College Outline Series, Harcourt Brace Jovanovich, Publishers, Fort Worth, TX.

Tinker, M. and R. Lambourne, 2000, "*Further Mathematics for the Physical Sciences*," John Wiley & Sons, LTD., Chichester, U.K.

Tolstov, G.P., 1962, "*Fourier Series*," (Translated from the Russian by R. A. Silverman), Dover Publications, New York.

Tveito, A. and R. Winther, 1998, "*Introduction to Partial Differential Equations - A Computational Approach*," Texts in Applied Mathematics 29, Springer, New York.

Urroz, G., 2000, "*Science and Engineering Mathematics with the HP 49 G - Volumes I & II*", www.greatunpublished.com, Charleston, S.C.

Urroz, G., 2001, "*Applied Engineering Mathematics with Maple*", www.greatunpublished.com, Charleston, S.C.

Winnick, J., , "*Chemical Engineering Thermodynamics - An Introduction to Thermodynamics for Undergraduate Engineering Students*," John Wiley & Sons, Inc., New York.

