Chapter 2. Regression

2.1. Introduction

Regression is a term describing the fitting of a particular function to a set of data through optimization of the parameters or constant coefficients that appear in the function. From this point of view regression is a kind of optimization, which is the subject of Chapter 8. However, many common forms of regression involve the solution of a linear set of algebraic equations. Thus, just as we separate our discussion of the solution of linear (Chapter 1) and nonlinear (Chapter 4) algebraic equations, so too do we separate our discussion of linear and nonlinear optimization. We shall call this linear optimization by the term regression and place it directly after the chapter on linear algebra because from one point of view, it is simply an application of linear algebra.

2.2. Single Variable Linear Regression

Imagine that we have a set of *n* data points (x_i, y_i) for *i* = 1 to *n*. Perhaps, we have a theory that tells us that *y* should be a linear function of *x*. We know that the equation of a line is given by

$$\hat{y} = b_1 x + b_0 \tag{2.1}$$

where b_1 is the slope and b_0 is the y-intercept of the line. The hat on $\hat{y}$ reminds us that this value of *y* comes from the model and not from data. We would like to know what the best values of the constant coefficients, b_1 and b_0. If we substitute our data points into this equation, it will not in general be satisfied due to noise in the data. Therefore, we create the equality by introducing the error in the i^{th} data point, e_i, such that

$$y_i = b_1 x_i + b_0 + e_i \tag{2.2}$$

The best-fit model will minimize the error. In particular we wish to minimize the Sum of the Square of Errors, *SSE*, defined as

$$SSE \equiv \sum_{i=1}^{n} e_i^{\,2} = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2 = \sum_{i=1}^{n}(y_i - b_1 x_i - b_0)^2 \tag{2.3}$$

In order to minimize the function SSE with respect to b_1 and b_0, we take the partial differential of SSE with respect to b_1 and b_0 and set them equal to zero. (We remember from differential calculus that the derivative of a function is zero at a minimum or maximum.)

$$\frac{\partial SSE}{\partial b_0} = -2\sum_{i=1}^{n}(y_i - b_1 x_i - b_0) = 0 \tag{2.4.a}$$

$$\frac{\partial SSE}{\partial b_1} = -2\sum_{i=1}^{n}(y_i - b_1 x_i - b_0)x_i = 0 \tag{2.4.b}$$

We then solve these two equations for b_1 and b_0. Notice that equations (2.4.a) and (2.4.b) are linear in the unknown variables, b_1 and b_0. We have already seen in Chapter 1, the solution to a set of two linear algebraic equations. In this case, we have

$$b_1 = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n}(x_i - \bar{x})^2} \tag{2.5.a}$$

$$b_0 = \frac{\sum_{i=1}^{n} y_i - b_1 \sum_{i=1}^{n} x_i}{n} = \bar{y} - b_1 \bar{x} \tag{2.5.b}$$

where $\bar{x}$ and $\bar{y}$ are the average values of the set of x and y respectively.

We note that one requires two points at a minimum to perform a single-variable linear regression. Since there are two parameters, we need two data points. One can see from equation (2.5) that if the two data points have the same value of *x*, the slope is infinite. Therefore, this method requires at least two data points at different values of the independent variable, *x*.

2.3. The Variance of the Regression Coefficients

From a statistical point of view, the regression coefficients in equation (2.5) are mean values of the slope and intercept. Whenever a mean is calculated, a variance can also be calculated. Here provided without derivation is the variance of the regression coefficients. The variance of the slope is

$$\sigma_{b_1}^2 = \frac{\sigma^2}{\sum_{i=1}^{n}\left(x_i - \bar{x}\right)^2} \tag{2.6.a}$$

The variance of the y-intercept is

$$\sigma_{b_0}^2 = \frac{\sum_{i=1}^{n} x_i^{\,2}}{n\sum_{i=1}^{n}\left(x_i - \bar{x}\right)^2}\sigma^2 \tag{2.6.b}$$

where σ^2 is the model error variance. An unbiased estimate of σ^2 is s^2 where

$$\sigma^2 \approx s^2 = \frac{SSE}{n-2} \tag{2.7}$$

Frequently, one would like a convenient metric to express the goodness of the fit. The measure of fit, *MOF*, provides a way to determine if the best-fit model is a good model. A common definition of the *MOF* is

$$MOF = \frac{SSR}{SST} = 1 - \frac{SSE}{SST} \tag{2.8}$$

where the sum of the squares of the errors, SSE , was defined in equation (2.3), and where the Sum of the Squares of the Regression, SSR , is based on the variance of the model predictions and the average values of *y*.

$$SSR = \sum_{i=1}^{n}\left(\hat{y}_i - \bar{y}\right)^2 \tag{2.9}$$

The Sum of the Squares of the Total variance, SST , is

$$SST = SSR + SSE = \sum_{i=1}^{n}\left(\hat{y}_i - \bar{y}\right)^2 + \sum_{i=1}^{n}\left(y_i - \hat{y}_i\right)^2 \tag{2.10}$$

An analysis of equation (2.10) makes it clear that there are two sources of variance, that captured by the regression, *SSR*, and that outside the regression, *SSE*. This *MOF* is the fraction of variance captured by the regression. It is bounded between 0 and 1. A value of *MOF* of 1 means the model fits the data perfectly. The farther the value of the MOF is below 1, the worse the fit of the model.

We note in passing that a more rigorous method for evaluating the goodness of the fit is to use the f-distribution. We can define a variable, f , by

$$\mathrm{f} = \frac{SSR}{s^2} \tag{2.11}$$

For a given level of confidence, γ, and for a given number of data points, *n*, one can determine whether the regression models the data to within that level of confidence by direct comparison with the appropriate critical value of the f statistic

$$\mathrm{f} > \mathrm{f}_{\gamma}(1, n-2) \tag{2.12}$$

where $\mathrm{f}_{\gamma}(v_1, v_2)$ is defined as the lower limit in the following constraint

$$\gamma = \int_{\mathrm{f}_{\gamma}(v_1,v_2)}^{\infty} \frac{\Gamma\left[\frac{v_1+v_2}{2}\right]\left(\frac{v_1}{v_2}\right)^{\frac{v_1}{2}}}{\Gamma\left[\frac{v_1}{2}\right]\Gamma\left[\frac{v_2}{2}\right]} \frac{\mathrm{f}^{\frac{v_1}{2}-1}}{\left(1+\frac{v_1}{v_2}f\right)^{\frac{v_1+v_2}{2}}} df \tag{2.13}$$

In equation (2.13), the gamma function, $\Gamma[z]$, is a standard integral

$$\Gamma[z] = \int_0^{\infty} t^{z-1} e^{-t} dt \tag{2.14.a}$$

For positive integers, the expression simplifies to

$$\Gamma[n] = (n-1)! \tag{2.14.b}$$

In MATLAB, the gamma function can be accessed with the `gamma(z)` command. In MATLAB, the cumulative integral of the f distribution can be accessed with the `fcdf(f,v1,v2)` command. For a given value of f , this function will return $1-\gamma$. Values of the integral in equation (2.13) as a function of γ, v_1 and v_2 are routinely available in tables of critical f values" [Walpole *et al.* 1998].

2.4. Multivariate Linear Regression

Above, we learned how to perform a regression for *y* when it is a linear function of a single variable *x*. In this section, we extend the capability to performing a regression for *y* when it is a linear function of an arbitrary number, *m*, of variables. In this case, our model has the general form

$$\hat{y}_i = b_0 + \sum_{j=1}^{m} b_j x_{i,j} \tag{2.15}$$

Note that there are now two subscripts. The subscript *j* differentiates the different independent variables and runs from 1 to *m*. Perhaps one variable is temperature and another concentration. The subscript *i* designates the individual data points, $\left(x_{i,1}, x_{i,2} \ldots x_{i,m,} y_i\right)$, and runs from 1 to *n*. The error for each data point is again defined by the equation

$$y_i = b_0 + \sum_{j=1}^{m} b_j x_{i,j} + e_i \tag{2.16}$$

The method for finding the best-fit parameters for this system is exactly analogous to what we did for the single-variable linear regression. We define, the Sum of the Squares of the Error, SSE, exactly as we did before in equation (25.6)

$$SSE \equiv \sum_{i=1}^{n} e_i^{\ 2} = \sum_{i=1}^{n} \left(y_i - \hat{y}_i\right)^2 = \sum_{i=1}^{n} \left(y_i - b_0 - \sum_{j=1}^{m} b_j x_{i,j} \right)^2 \tag{2.17}$$

We take the partial derivatives of SSE with respect to each of the parameters in $\{b\}$ and set them equal to zero. This gives $m+1$ independent, linear algebraic equations of the form:

$$b_0 \sum_{i=1}^{n} x_{i,0} x_{i,j} + b_1 \sum_{i=1}^{n} x_{i,1} x_{i,j} + b_2 \sum_{i=1}^{n} x_{i,2} x_{i,j} + \ldots + b_m \sum_{i=1}^{n} x_{i,m} x_{i,j} = \sum_{i=1}^{n} y_i x_{i,j} \quad \text{for } i = 0 \text{ to } m \tag{2.18}$$

We have introduced a new set of constants, $x_{0,i} = 1$ in this equation to allow for compact expression of the equations. This set of equations can be written in matrix form as

$$\underline{\underline{A}}\,\underline{b} = \underline{g} \tag{2.19}$$

where the *j*,*k* element of the matrix $\underline{\underline{A}}$ is defined as

$$A_{j,k} = \sum_{i=1}^{n} x_{i,k-1} x_{i,j-1} \tag{2.20}$$

where the *j* element of the column vector of constants, $\underline{g}$, is defined as

$$g_j = \sum_{i=1}^{n} y_i x_{i,j-1} \tag{2.21}$$

The solution vector, $\underline{b}$, contains the regression parameters,

$$\underline{b}_j = \begin{bmatrix} b_0 \\ b_1 \\ \vdots \\ b_m \end{bmatrix} \tag{2.22}$$

The presence of the *k*-1 and *j*-1 subscripts is due to the fact that in our model, we numbered our regression parameters starting from 0 rather than 1, but the numbering of rows and columns in the matrix begins at 1.

To determine the variances of the parameters in multivariate linear regression, we use analogous equations as for the single-variable case. The variances of each parameter are defined as

$$\sigma_{b_j}^2 = A_{j-1,j-1}^{-1} \sigma^2 \tag{2.23}$$

where $A_{j-1,j-1}^{-1}$ is the diagonal element of the inverse of $\underline{\underline{A}}$ (which is not the inverse of the corresponding element of $\underline{\underline{A}}$). The covariances are

$$\sigma_{b_j b_k} = A_{j-1,k-1}^{-1} \sigma^2 \tag{2.24}$$

where, as in the single-variable case, an unbiased estimate of σ^2 is given by s^2, which is defined as

$$\sigma^2 \approx s^2 = \frac{SSE}{n-m-2} \tag{2.25}$$

The measure of fit for the multiple regression case is defined in exactly the same way as the single-variable regression case, given in equation (2.8).

2.5. Polynomial Regression

One might think that in a chapter on linear regression, that polynomial regression would be outside the scope of the discussion. However, the defining characteristic of this chapter is that the **parameters in the regression be linear**. The tools in this chapter can be used even if the independent variables can have any sort of nonlinear functionality so long as the regression coefficients appear in a linear form. In a polynomial, the coefficients appear in a linear manner,

$$\hat{y}_i = b_0 + \sum_{j=1}^{m} b_j x_i^{\,j} \tag{2.26}$$

We see that the form of the model in a polynomial regression is very similar to the form of our previous multivariate linear regression in equation (2.15).

$$\hat{y}_i = b_0 + \sum_{j=1}^{m} b_j x_{j,i} \tag{2.15}$$

In fact we can perform polynomial regression using the tools from multivariate linear regression if

$$x_{j,i} = x_i^{\,j} \tag{2.27}$$

The rest of the procedure for polynomial regression is then exactly the same as multivariate linear regression.

2.6 Linearization of Equations

Frequently we attempt to fit experimental data with a nonlinear theory. Often the parameterization of this data can be accomplished with linear regression because the theory is linear in the parameters and nonlinear in the independent variables. Consider a general expression for the rate of a process,

$$r = k\exp\left(-\frac{E_a}{k_B T}\right) \tag{2.28}$$

in which the rate, r, is a function of the temperature, T, and two constants, the activation energy, E_a, and the pre-exponential factor, k. (k_B is Boltzmann's constant.) Many physical processes obey this functional form including chemical reactions and diffusion.

Frequently, one has rates as a function of temperature and one wishes to determine the activation barrier for the process and the rate constant, k. In its current form, this equation is nonlinear in the parameters. However, taking the natural log of both sides of the equation yields

$$\ln(r) = \ln(k) - \frac{E_a}{k_B T} \tag{2.29}$$

Lo and behold, now the equation is linear of the form

$$\hat{y} = b_1 x + b_0 \tag{2.1}$$

where $\hat{y} = \ln(r)$, $x = \frac{1}{T}$, $b_0 = \ln(k)$ and $b_1 = -\frac{E_a}{k_B}$. This transformation is so common that it is given its own name, the Arrhenius form of the rate equation. A single variable linear regression can be performed on this data, yielding b_0 and b_1. The rate constant and activation energy can be directly obtained from $k = \exp(b_0)$ and $E_a = -k_B b_1$.

2.7. Confidence Intervals

Sometimes, in addition to the mean and standard deviation of the regression coefficients, one would also like to know the uncertainty in the regression as a function of the independent variables. It is probably not surprising that the uncertainty is typically smaller in the middle of the region where data was extracted and larger at the ends. Such a feat can be accomplished through the use of confidence intervals. First, we provide formulae for confidence intervals on the regression coefficients, then we provide confidence intervals as a function of the independent variable.

Typically one is interested in a 90% (or 95% or 99% or in general *CI*%) confidence interval, which provides a lower bound and upper bound to a range within which you are *CI*% confident that the true result lies. Typically, this confidence, *CI*, is related to a parameter, α through the expression, $CI = 100(1-2\alpha)$. Thus a confidence interval of 90% corresponds to α = 5%. Under the assumption that the observations (data points) are normally and independently distributed, a confidence interval on the slope, b_1, is given by

$$P(\hat{b} - t_{\alpha,n-2}\sqrt{\sigma_b^2} < b < \hat{b} + t_{\alpha,n-2}\sqrt{\sigma_b^2}) = 1 - 2\alpha \quad (2.30.a)$$

This mathematical equation states that the probability that the slope lies between the lower and upper limit is *CI*. The confidence interval on the intercept, b_0, is given by

$$P(\hat{a} - t_{\alpha,n-2}\sqrt{\sigma_a^2} < a < \hat{a} + t_{\alpha,n-2}\sqrt{\sigma_a^2}) = 1 - 2\alpha \quad (2.30.b)$$

The variances that appear in equation (2.30) are the same as those already computed for the slope and intercept via equation (2.6). The number of data points is *n*. The critical t-statistic, $t_{\alpha,v}$, is defined as the lower limit in the following constraint

$$\alpha = \int_{t_{\alpha,v}}^{\infty} \frac{\Gamma\left[\frac{v+1}{2}\right]}{\Gamma\left[\frac{v}{2}\right]\sqrt{\pi v}}\left(1+\frac{t^2}{v}\right)^{-\frac{v+1}{2}} dt \quad (2.31)$$

Values of this integral as a function of α and *v* are routinely available in tables of critical t values" [Walpole *et al.* 1998].

The confidence interval at any arbitrary value of the independent variable can be obtained as follows. Under the same assumption that the observations are normally and independently distributed, a $CI = 100(1-2\alpha)$ confidence interval on the regression at a point, x_o, is given by

$$P(\hat{y}(x_o) - t_{\alpha,n-2}\sqrt{\sigma_{y(x_o)}^2} < y(x_o) < \hat{y}(x_o) + t_{\alpha,n-2}\sqrt{\sigma_{y(x_o)}^2}) = 1 - 2\alpha \quad (2.32)$$

where $\hat{y}(x_o) = b_0 + b_1 x_o$ and where

$$\sigma_{y(x_o)}^2 = \left[\frac{1}{n} + \frac{(x_o - \bar{x})^2}{\sum_{i=1}^{n} x_i^2 - \frac{1}{n}\left(\sum_{i=1}^{n} x_i\right)^2}\right]\sigma^2 \quad (2.33)$$

where σ^2 is the model error variance, as estimated in equation (2.7) This allows you to evaluate the confidence interval at every value of *x*, giving upper and lower confidence limits that are functions of *x*.

More on confidence intervals is available in the literature.[Montgomery & Runger, 1999].

2.8. Regression Subroutines

Note that in order to present short codes, the versions of the codes given below makes three sacrifices. First, these codes contain no comments or instructions for use. Second these codes contain no error checking. For example it does not check that the user has provided sufficient data points. Third, these codes take advantage of the most succinct MATLAB commands, such as implicit for-loops, which shortens the code but may make the code difficult to understand.

Therefore, on the course website, two entirely equivalent versions of this code are provided and are titled *code.m* and *code_short.m*. The short version is presented here. The longer version, containing instructions and serving more as a learning tool, is not presented here.

Code 2.1. Single Variable Linear Regression (linreg1_short)

The single variable regression described in Section 2.2., can be accomplished using the following MATLAB code.

```
function [b,bsd,MOF] = linreg1_short(x,y);
n = max(size(x));
b = zeros(2,1);
bsd = zeros(2,1);
yhat = zeros(n,1);
xsum = sum(x);
xavg = xsum/n;
ysum = sum(y);
yavg = ysum/n;
x2sum = sum(x.*x);
x2avg = x2sum/n;
xvarsum = sum((x-xavg).*(x-xavg));
yvarsum = sum((y-yavg).*(y-yavg));
xycovarsum = sum((x-xavg).*(y-yavg));
b(2) = xycovarsum/xvarsum;
b(1) = yavg - b(2)*xavg;
yhat = b(1) + b(2)*x;
SSR = sum((yhat - yavg).^2);
SSE = sum((y - yhat).^2);
SST = SSR + SSE;
MOF = SSR/SST;
sigma = SSE/(n-2);
bsd(2) = sigma/xvarsum;
bsd(1) = x2avg*sigma/xvarsum;
bsd(1:2) = sqrt(bsd(1:2));
figure(1);
plot(x,y,'ro'), xlabel( 'x' ), ylabel ( 'y' );
hold on;
```

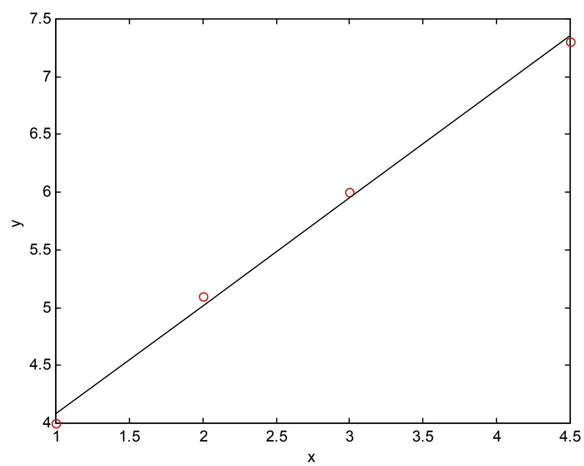

Figure 2.1. Single variable linear regression for Example 2.1.

```
plot(x,yhat,'k-');
hold off;
```

An example of using linreg1_short is given below.

```
» [b,bsd,MOF] = linreg1_short([1; 2; 3; 4.5], [4; 5.1; 6; 7.3])

b =
   3.14672897196262
   0.93457943925234

bsd =
   0.10993530680867
   0.03756962849017

MOF =

   0.99677841217186
```

In this example the mean value of the intercept is 3.147 with a standard deviation of 0.110. The mean value of the slope is 0.934 with a standard deviation of 0.038. The measure of fit is very good at 0.997.

Code 2.2. Multiariate Linear Regression (linregn_short)

The multivariate linear regression described in Section 2.4., can be accomplished using the following MATLAB code.

```
function [b,bsd,MOF] = linregn_short(m,x,y);
n = max(size(y));
mp1 = m + 1;
b = zeros(mp1,1);
bsd = zeros(mp1,1);
yhat = zeros(n,1);
g = zeros(mp1,1);
a = zeros(mp1,mp1);
xp1 = ones(n,mp1);
xp1(1:n,2:mp1) = x(1:n,1:m);
gvec = [xp1]'*y;
Amat = [xp1]'*xp1;
detA = det(Amat);
Amatinv = inv(Amat);
b = Amatinv*gvec;
dof = n - m;
yhat = xp1*b;
yavg = sum(y)/n;
SSR = sum((yhat - yavg).^2);
SSE = sum((y - yhat).^2);
SST = SSR + SSE;
MOF = SSR/SST;
s2 = SSE/dof;
```

```
for j = 1:m
   bsd(j) =sqrt(Amatinv(j,j)*s2);
end
```

An example of using linregn_short is given below.

```
   » [b,bsd,MOF] = linregn_short(2,[1 0; 2 5; 3 10; 4 20],[1.9; 13; 24; 45])

   b =
      0.79999999999995
      1.11666666666656
      1.98666666666671

   bsd =
      0.08660254037844
      0.06972166887784
                     0

   MOF =
      0.99999835603242
```

In this example the mean value of the intercept is 0.80 with a standard deviation of 0.087. The mean value of the coefficient for variable 1 is 1.117 with a standard deviation of 0.070. The mean value of the coefficient for variable 2 is 1.987 with a standard deviation of 0.0. The measure of fit is essentially perfect at 1.0.

Code 2.3. Multiariate Linear Regression (polyreg_short)

The polynomial regression described in Section 2.5., can be accomplished using the following MATLAB code.

```
function [b,bsd,MOF] = polyreg(m,x,y);
n = max(size(y));
mp1 = m + 1;
b = zeros(mp1,1);
bsd = zeros(mp1,1);
yhat = zeros(n,1);
g = zeros(mp1,1);
a = zeros(mp1,mp1);
xp1 = ones(n,mp1);
for i = 2:1:mp1
   xp1(1:n,i) = x(1:n).^(i-1);
end
gvec = [xp1]'*y;
Amat = [xp1]'*xp1;
detA = det(Amat);
Amatinv = inv(Amat);
b = Amatinv*gvec;
dof = n - m;
yhat = xp1*b;
yavg = sum(y)/n;
```

```
SSE = 0.0;
SSR = 0.0;
for i = 1:n
   SSR = SSR + (yhat(i) - yavg)^2;
   SSE = SSE + (y(i) - yhat(i))^2;
end
SST = SSR + SSE;
MOF = SSR/SST;
%
s2 = SSE/(dof);
for j = 1:m
   bsd(j) =sqrt(Amatinv(j,j)*s2);
end
figure(1);
plot(x,y,'ro'), xlabel( 'x' ), ylabel ( 'y' );
hold on;
plot(x,yhat,'k-');
hold off;
```

An example of using polyreg_short is given below.

```
   » [b,bsd,MOF] = polyreg(2,[1; 2; 3; 4],[4; 11; 22; 36])

   b =
      0.24999999999977
      1.94999999999982
      1.74999999999997


   bsd =
      0.44017042154147
      0.40155946010522
                     0

   MOF =
      0.99991449337324
```

In this example the mean value of the intercept is 0.25 with a standard deviation of 0.44. The mean value of the linear coefficient is 1.95 with a standard deviation of 0.040. The mean value of the quadratic coefficient is 1.75 with a standard deviation of 0.0. The measure of fit is essentially perfect at 1.0.

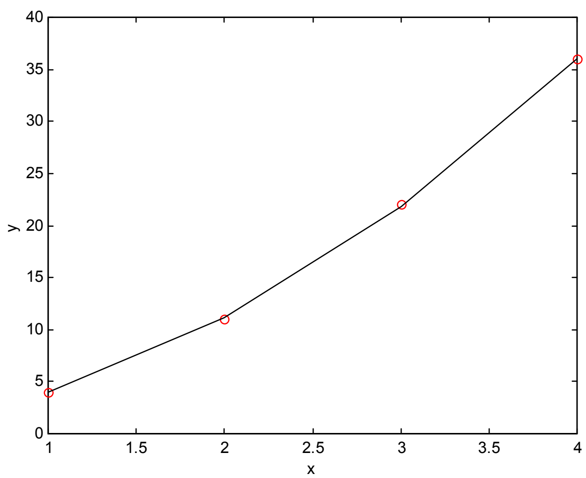

Figure 2.2. Polynomial regression of order 2 for Example 2.3.

Code 2.4. Single Variable Linear Regression with Confidence Intervals (linreg1ci_short)

The single variable regression described in Section 2.2. with confidence intervals described in Section 2.7., can be accomplished using the following MATLAB code.

```
function [b,bsd,bcilo,bcihi,MOF] = linreg1ci(x,y,CI);
n = max(size(x));
b = zeros(2,1);
bsd = zeros(2,1);
bcilo = zeros(2,1);
bcihi = zeros(2,1);
yhat = zeros(n,1);
xsum = sum(x);
xavg = xsum/n;
ysum = sum(y);
yavg = ysum/n;
x2sum = sum(x.*x);
x2avg = x2sum/n;
xvarsum = sum((x-xavg).*(x-xavg));
yvarsum = sum((y-yavg).*(y-yavg));
xycovarsum = sum((x-xavg).*(y-yavg));
b(2) = xycovarsum/xvarsum;
b(1) = yavg - b(2)*xavg;
yhat = b(1) + b(2)*x;
SSR = sum((yhat - yavg).^2);
SSE = sum((y - yhat).^2);
SST = SSR + SSE;
MOF = SSR/SST;
sigma = SSE/(n-2);
bsd(2) = sigma/xvarsum;
bsd(1) = x2avg*sigma/xvarsum;
bsd(1:2) = sqrt(bsd(1:2));
v = n-2;
alpha = (1 - CI/100)/2;
%  need critical value of t statistic here!
cipalpha = CI/100.0 + alpha;
talpha = icdf('t',cipalpha,v);
bcilo(1) = b(1) - talpha*bsd(1);
bcihi(1) = b(1) + talpha*bsd(1);
bcilo(2) = b(2) - talpha*bsd(2);
bcihi(2) = b(2) + talpha*bsd(2);
xmin = min(x);
xmax = max(x);
nx = 20;
dx = (xmax - xmin)/nx;
xci = zeros(nx,1);
ycilo = zeros(nx,1);
ycihi = zeros(nx,1);
for i = 1:1:nx+1
  xci(i) = xmin + (i-1)*dx;
end
for i = 1:1:nx+1
   thing = 1/n + (xci(i) - xavg)^2/(x2sum - xsum*xsum/n);
   sig = sqrt(thing*sigma);
   ycilo(i) = b(1) + b(2)*xci(i) - talpha*sig;
   ycihi(i) = b(1) + b(2)*xci(i) + talpha*sig;
```

```
end
figure(1);
plot(x,y,'ro'), xlabel( 'x' ), ylabel ( 'y' );
hold on;
plot(x,yhat,'k-');
hold on;
plot(xci,ycilo,'g--');
hold on;
plot(xci,ycihi,'g--');
hold off;
```

An example of using linregci_short is given below.

```
   » [b,bsd,bcilo,bcihi,MOF] =
linreg1ci_short([1;2;3;4;5;6],[4.2;5.1;6.1;6.7;8.2;8.9],95)

   b =
      3.19333333333333
      0.95428571428571

   bsd =
      0.18429617295288
      0.04732288856688

   bcilo =
      2.16164335714309
      0.68937218408834

   bcihi =
      4.22502330952358
      1.21919924448308

   MOF =
      0.99025920227246
```

In this example, the mean value of the slope is 0.954. There is a 95% probability that the true slope lies between 0.689 and 1.219. The mean value of the intercept is 03.193. There is a 95% probability that the true slope lies between 2.161 and 4.225.

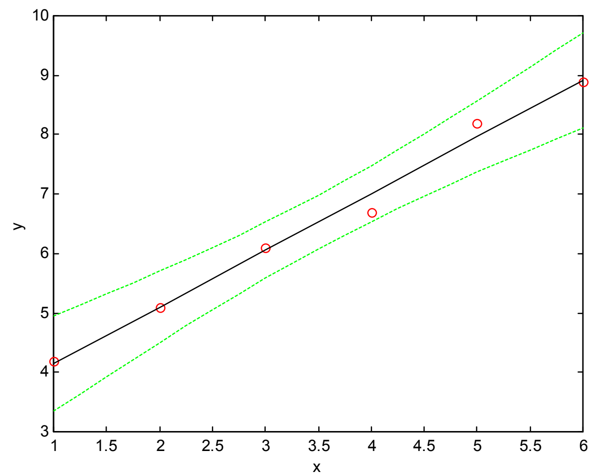

Figure 2.3. Single variable linear regression with 95% confidence intervals for Example 2.4.

2.9. Problems

Problems are located on course website.