Statistics 580

A Review of Elementary Numerical Methods for the Solution of Linear Equations and their Applications

Introduction

Many of the problems in statistical analysis, especially in multiple regression, can be reduced to the problem of solving linear systems of equations. Thus, it is useful to study numerical methods available for the solution of such systems. We use matrix notation to represent linear systems of equations and their solutions. A system of m linear equations in n unknowns has the general form

$$\begin{array}{rcl} a_{11}x_1 + a_{12}x_2 + & \cdots & +a_{1n}x_n = b_1 \\ a_{21}x_1 + a_{22}x_2 + & \cdots & +a_{2n}x_n = b_2 \\ & \cdots & \\ & \cdots & \\ a_{m1}x_1 + a_{m2}x_2 + & \cdots & +a_{mn}x_n = b_m \end{array}$$

The coefficients $a_{ij}(i = 1, \ldots, m; \, j = 1, \ldots, n)$ and the right hand sides $b_i(i = 1, \ldots, m)$ are known numbers. The problem is to find numbers $x_j(j = 1, \ldots, n)$ such that the m equations are satisfied simultaneously.

The above system of equations can be expressed in the form

$$A\boldsymbol{x} = \boldsymbol{b} \tag{1}$$

where A is the $m \times n$ matrix $A = (a_{ij})$, known as the coefficient matrix, $\boldsymbol{b}$ is the $m \times 1$ vector $\boldsymbol{b} = (b_1, b_2, \ldots, b_m)'$ and $\mathtt{x}$ is the $n \times 1$ vector $\mathtt{x} = (x_1, x_2, \ldots, x_n)'$. We shall restrict our discussion exclusively to the case where the number of equations are equal to the number of unknowns, i.e., $m = n$. In this case the coefficient matrix A is a square matrix.

`Theorem`

The system of linear equations (1) with $m = n$ has a unique solution if and only if A is a nonsingular matrix. If A is nonsingular then the solution is given by

$$\boldsymbol{x} = A^{-1}\boldsymbol{b}$$

Consequently, in this discussion we will assume that all linear systems considered have nonsingular coefficient matrices.

`Example:` Solve the equations

$$\begin{aligned} x_1 - x_2 &= 5 \\ 3x_1 + 2x_2 &= 10 \end{aligned}$$

Rewriting the equations in matrix form, we have

$$\begin{bmatrix} 1 & -1 \\ 3 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 5 \\ 10 \end{bmatrix}$$

and thus the solution vector is given by

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 1 & -1 \\ 3 & 2 \end{bmatrix}^{-1} \begin{bmatrix} 5 \\ 10 \end{bmatrix} = \begin{bmatrix} .4 & .2 \\ -.6 & .2 \end{bmatrix} \begin{bmatrix} 5 \\ 10 \end{bmatrix} = \begin{bmatrix} 4 \\ -1 \end{bmatrix}$$

Hence the solution is $x_1 = 4$ and $x_2 = -1$.

In this example we explicitly evaluated the inverse of the coefficient matrix in order to compute the solution. In practice, however, it is not necessary nor computationally efficient to evaluate inverses to find solutions to systems of linear equations.

Solution of Linear Equations by Gauss-Jordan Elimination

A fundamental numerical procedure for the solution of systems of linear equations is called Gauss-Jordan elimination. This method consists of

i) Multiplication of each equation by a chosen constant, and

ii) Addition of multiples of one equation to another in order to make selected coefficients zero.

To illustrate the procedure, consider the system

$$4x_1 + 2x_2 = 6 \qquad (1)$$
$$2x_1 + 6x_2 = 10 \qquad (2)$$

`Step 1:` Multiply (1) by 1/4.

$$x_1 + (1/2)x_2 = 3/2 \qquad (3)$$

`Step 2:` Add -2 times (3) to (2).

$$5x_2 = 7 \tag{4}$$

In these 2 steps we have `eliminated` variable x_1 from equation (2) to obtain equation(4). At this stage we solve equation (4) to obtain $x_2 = 7/5$ and substitute back in equation (1) to obtain $x_1 = 4/5$. This last operation is known as `back substitution`.

We can express the procedure followed in steps 1 and 2 above as `elementary row` operations performed on the appended matrix $[A \mid \boldsymbol{b}]$. For this example, we shall perform these row operations on the matrix

$$\left[\begin{array}{cc|c} 4 & 2 & 6 \\ 2 & 6 & 10 \end{array}\right]$$

in one step, as follows:
Add $-2/4$ times row 1 to row 2 to obtain

$$\left[\begin{array}{cc|c} 4 & 2 & 6 \\ 0 & 5 & 7 \end{array}\right]$$

The row operations performed on the rows of $[A \mid \boldsymbol{b}]$ do not affect the solution to the system of equations. We see that by performing the above operations we have in effect transformed the matrix A to upper triangular form. Thus, the primary aim of the Gauss-Jordan elimination is to perform elementary row operations on the rows of $[A \mid \boldsymbol{b}]$ such that the matrix A is transformed into an upper triangular matrix. Once this is done, the system of equations can be solved quickly be back substitution.

An Algorithm for the Solution of Linear Equations by Gauss-Jordan Elimination

As discussed above, the Gauss-Jordan elimination is a simple enough procedure to implement in a computer. However, there is a host of improvements that can be made to make such an algorithm perform correctly in all types of situations, such as when it is not known that the coefficient matrix is nonsingular or not etc., or to improve its computational efficiency and accuracy. To describe such a complete algorithm here would defeat our purpose; that is to understand how the Gauss-Jordan elimination procedure might be implemented on a computer. Thus, we shall discuss a very straightforward algorithm with the understanding that actual programs available for the solution of linear equations using Gauss-Jordan elimination are far more complex and sophisticated in their implementation than the algorithm discussed here, although the underlying principles are the same.

Algorithm 1:
Let W be the $n \times (n+1)$ augmented matrix $[A \mid \boldsymbol{b}]$.

I. `Elimination Step`
For each row $k = 1, 2, \ldots, n-1$ repeat the following for each other row $i = k+1, \ldots, n$,
Set $d = -w_{ik}/w_{kk}$ and $w_{ik} = 0$
Set $w_{ij} = w_{ij} + dw_{kj}$ for $j = k+1, \ldots, n+1$

II. `Substitution Step`
Set $x_n = w_{n,n+1}/w_{nn}$
Then for $k = n-1, \ldots 1$ set
$x_k = \frac{w_{k,n+1} - \sum_{j=k+1}^{n} w_{kj} x_j}{w_{kk}}$

As an illustration, we shall use this algorithm to solve the following set of equations:

$$
\begin{aligned}
2x_1 &+ 3x_2 - x_3 = 5 \\
4x_1 &+ 4x_2 - 3x_3 = 3 \\
-2x_1 &+ 3x_2 - x_3 = 1
\end{aligned}
$$

For this example

$$
W = \begin{bmatrix} 2 & 3 & -1 & 5 \\ 4 & 4 & -3 & 3 \\ -2 & 3 & -1 & 1 \end{bmatrix} \tag{2}
$$

The elimination begins with row 1 i.e., $k = 1$. We say that row 1 is the `pivot row` and use the `pivot element` w_{11} to eliminate the elements w_{21} and w_{31}. The operations may be summarized as follows:

For $k = 1$, $w_{11} = 2$; $i = 2$; set $d = -4/2$, add $d \times$ row 1 to row 2., set $w_{21} = 0$.

$i = 3$; set $d = 2/2$, add d $\times$ row 1 to row 3, set $w_{31} = 0$.

The resultant matrix after pivoting on row 1 is:

$$\begin{bmatrix} 2 & 3 & -1 & 5 \\ 4 & 4 & -3 & 3 \\ -2 & 3 & -1 & 1 \end{bmatrix} \longrightarrow \begin{bmatrix} 2 & 3 & -1 & 5 \\ 0 & -2 & -1 & -7 \\ 0 & 6 & -2 & 6 \end{bmatrix}$$

For $k = 2, w_{22} = -2$; $i = 3; d = -6/(-2)$, add d $\times$ row 2 to row 3, set $w_{32} = 0$. The resultant matrix after pivoting on row 2 is:

$$\begin{bmatrix} 2 & 3 & -1 & 5 \\ 0 & -2 & -1 & -7 \\ 0 & 6 & -2 & 6 \end{bmatrix} \longrightarrow \begin{bmatrix} 2 & 3 & -1 & 5 \\ 0 & -2 & -1 & -7 \\ 0 & 0 & -5 & -15 \end{bmatrix}$$

We have the appended matrix in the required form since matrix A is now upper-triangular. The solution is now found by back substitution:

$$\begin{aligned} x_3 &= -15/(-5) = 3 \\ x_2 &= [-7 - (-1)(3)]/(-2) = 2 \\ x_1 &= [5 - (-1)(3) - 3(2)]/(2) = 1 \end{aligned}$$

Computing the Inverse of a Matrix Using the Gauss-Jordan Method

Using the algorithm discussed in the previous section, the solution to the system $A\mathbf{x} = \boldsymbol{b}$ could be directly computed without computing A^{-1} explicitly and forming the product $A^{-1}\boldsymbol{b}$. However, it is sometimes necessary that the inverse matrix A^{-1} is also available to the user. In this section we will discuss an algorithm that computes the inverse of a nonsingular matrix A using elementary row operations of the Gauss-Jordan type.

We begin by appending an identity matrix which is of the same size as A to the right hand side of A. Elementary row operations are performed on the resulting appended matrix $[A \mid I]$ so that matrix A is reduced to an identity matrix. The inverse matrix A^{-1} is found on the right hand side at the termination of this procedure. To illustrate the procedure consider the matrix

$$A = \begin{bmatrix} 2 & 3 & -1 \\ 4 & 4 & -3 \\ -2 & 3 & -1 \end{bmatrix} \tag{3}$$

We form the appended matrix $[A \mid I]$ and consider row operations such that the left-hand 3×3 matrix reduces to an identify matrix. We use r_1, r_2 and r_3 to denote the rows.

$$\left[\begin{array}{rrr|rrr} 2 & 3 & -1 & 1 & 0 & 0 \\ 4 & 4 & -3 & 0 & 1 & 0 \\ -2 & 3 & -1 & 0 & 0 & 1 \end{array}\right]$$

`Step 1:`

$$
\begin{array}{l}
r_1 \leftarrow (1/2)r_1 \\
r_2 \leftarrow r_2 + (-4)r_1 \\
r_3 \leftarrow r_3 + 2r_1
\end{array}
\qquad
\left[\begin{array}{rrr|rrr}
1 & 3/2 & -1/2 & 1/2 & 0 & 0 \\
0 & -2 & -1 & -2 & 1 & 0 \\
0 & 6 & -2 & 1 & 0 & 1
\end{array}\right]
$$

For these elementary operations the `pivot row` was row 1 and the `pivot element` or the `pivot` was the (1,1) element.

`Step 2:`

$$
\begin{array}{l}
r_2 \leftarrow (-1/2)r_2 \\
r_1 \leftarrow r_1 + (-3/2)r_2 \\
r_3 \leftarrow r_3 + (-6)r_2
\end{array}
\qquad
\left[\begin{array}{rrr|rrr}
1 & 0 & -5/4 & -1 & 3/4 & 0 \\
0 & 1 & 1/2 & 1 & -1/2 & 0 \\
0 & 0 & -5 & -5 & 3 & 1
\end{array}\right]
$$

`Step 3:`

$$
\begin{array}{l}
r_3 \leftarrow (-1/5)r_3 \\
r_1 \leftarrow r_1 + (5/4)r_3 \\
r_2 \leftarrow r_2 + (-1/2)r_3
\end{array}
\qquad
\left[\begin{array}{rrr|rrr}
1 & 0 & 0 & 1/4 & 0 & -1/4 \\
0 & 1 & 0 & 1/2 & -1/5 & 1/10 \\
0 & 0 & 1 & 1 & -3/5 & -1/5
\end{array}\right]
$$

Thus, the matrix on the right is A^{-1}. Notice that at each step the pivot element is changed to 1 and the remaining elements in the corresponding column are zeroed. It is more important to notice that after each pivoting, the column with the corresponding pivot element does not enter into any further computations. This means that these columns may be used to store the corresponding columns of the inverted matrix. This fact can be used to invert a matrix `in place` by using Gauss-Jordan operations. The following algorithm computes the inverse of a given matrix in place by using row operations.

Algorithm 2:
Let $A = (a_{ij})$ be the $n \times n$ matrix to be inverted. For each pivot row $k = 1, 2, \ldots, n$ repeat the following:

Set $d = a_{kk}$
Set $a_{kj} = a_{kj}/d$ for $j = 1, 2, \ldots, n$
For each other row $i = 1, \ldots,\ n(i \neq k)$,
Set $c = -a_{ik}$

Set $a_{ij} = a_{ij} + ca_{kj}$ for $j = 1, \ldots, n$
Set $a_{ik} = c/d$
Set $a_{kk} = 1/d$

As an illustration of the above algorithm, we shall use it to invert the matrix A given previously:

$$A = \begin{bmatrix} 2 & 3 & -1 \\ 4 & 4 & -3 \\ -2 & 3 & -1 \end{bmatrix}$$

Pivoting on a_{11} gives

$$\begin{array}{l} d = 2 \\ c = -4 \\ c = 2 \end{array} \begin{bmatrix} 1/2 & 3/2 & -1/2 \\ -2 & -2 & -1 \\ 1 & 6 & -2 \end{bmatrix}$$

Pivoting on a_{22} gives

$$\begin{array}{l} c = -3/2 \\ d = -2 \\ c = -6 \end{array} \begin{bmatrix} -1 & 3/4 & -5/4 \\ 1 & -1/2 & 1/2 \\ -5 & 3 & -5 \end{bmatrix}$$

Pivoting on a_{33} gives

$$\begin{array}{l} c = 5/4 \\ c = -1/2 \\ d = -5 \end{array} \begin{bmatrix} 1/4 & 0 & -1/4 \\ 1/2 & -1/5 & 1/10 \\ 1 & -3/5 & -1/5 \end{bmatrix}$$

When the Gauss-Jordan reductions are applied to each pivot row to obtain the inverse in place, a `sweep operation` on the corresponding row is said to be performed. The set of operations carried out for each row in the above algorithm define the `sweep operator` for that row.

There are `two` properties of the sweep operator which are extremely useful in stepwise regression computations. One is that it is not necessary to perform the `sweeps of the pivots in any particular order`. In the above example, the inverse could have been obtained by sweeping on the pivots in the order a_{33}, a_{11} and a_{22} respectively. The second property is that `sweep operations are reversible`. This implies that the status of a matrix that existed prior to a sweep on a particular row can be reestablished by repeating a sweep operation on the same row. Reverse sweep operations may be used repeatedly to re-establish a matrix that existed before a series of sweep operations were performed. Moreover, the reverse sweeps need not be performed in the same order the forward sweeps were originally performed.

Improved Algorithm for Solution of Linear Equations

In the previous section, a simple algorithm for implementing the Gauss-Jordan elimination technique for solution of linear equations was discussed. It was noted at the beginning of that section that a simplified version of the algorithm is presented because the additional complexity of a more sophisticated algorithm is not helpful in understanding the computer implementation of the Gauss-Jordan method. In this section a modified version of Algorithm 1 is given. The main difference is in the choice of the pivotal rows in the elimination step. Recall that in Algorithm 1 the pivotal rows are actually the n rows of the equations taken in sequence from row 1 to row n of W where $W = [A \mid \boldsymbol{b}]$. This is of course possible only if the pivotal element w_{kk} in row k is not zero for all rows, $k = 1, \ldots, n$. Unfortunately, in many linear systems of equations this condition will not be satisfied and therefore a method must be devised by which this difficulty might be overcome. Greater freedom in the choice of the pivotal row at each stage must be allowed instead of being forced to choose the n rows in succession. This greater freedom in the choice of pivotal rows is necessary not only because of the possibility of zero coefficients. Experience has shown that this freedom is also essential for improving accuracy of computations by reducing round-off errors.

Usually various strategies have been proposed for selecting the order in which the pivotal rows are selected. Once a method for selecting this order has been specified, the operations described in the elimination step of Algorithm 1 are carried out for the rows in the selected order. An $n \times 1$ vector is used to record the order in which the rows of the system of equations are used as piotal rows. This array keeps track of which rows of the set of equations (or rows of W) have already been used as pivotal rows and which rows remain as candidates for the next pivotal row to be selected. The following algorithm incorporates a pivoting strategy known as `partial pivoting`. With this strategy, the 1^{st} pivotal row to be used in the elimination step is the row with the largest first element in absolute value. After the first pivot operation, the second pivotal row is selected in the same way using the matrix resulting from the first pivot operation. This corresponds to the row with the largest second element, and so on.

Algorithm 3:
Let W be the $n \times (n+1)$ appended matrix $[A \mid \boldsymbol{b}]$ and ℓ be the $n \times 1$ vector with $\ell_i = i = 1, \ldots, n$.

I. `Elimination Step`

For each row $k = 1, 2, \ldots n-1$ repeat the following:
Find j between k and n for which

$$\left|w_{\ell_j k}\right| \quad \geq \quad |w_{\ell_k k}|$$

Exchange the contents of ℓ_k and ℓ_j.
For $i = k+1, \ldots n$ repeat the following:
Set $\quad d = w_{\ell_i k}/w_{\ell_k k}$ and $w_{\ell_i k} = 0$
Set $\quad w_{\ell_i j} = w_{\ell_i j} + d w_{\ell_k j}$, for $j = k+1, \ldots, n+1$

II. `Substitution Step`
Set $x_n = w_{\ell_n, n+1}/w_{\ell_n n}$
Then, for $k = n-1, n-2, \ldots, 1$ set

$$x_k = \frac{w_{\ell_k, n+1} - \sum_{j=k+1}^{n} w_{\ell_k j} x_j}{w_{\ell_k k}}$$

This algorithm is illustrated by applying it to the example used to illustrate Algorithm 2 where

$$W = \begin{bmatrix} 2 & 3 & -1 & 5 \\ 4 & 4 & -3 & 3 \\ -2 & 3 & -1 & 1 \end{bmatrix}$$

Note that initially, $\ell = (1, 2, 3)'$.
`For` $k = 1:$
Compute $\mid w_{i1} \mid$ for i = 1, 2,3 which are: 2, 4, and 2. So row 2 which has the largest value is selected as the 1^{st} pivotal row. The resulting matrix is:

$$\begin{bmatrix} 0 & 1 & 1/2 & 7/2 \\ 4 & 4 & -3 & 3 \\ 0 & 5 & -5/2 & 5/2 \end{bmatrix}$$

and $\ell = (2, 1, 3)'$.

`For` $k = 2:$
Compute $\mid w_{i2} \mid$ for i = 1, 3 which are: 1 and 5. So row 3 becomes the next pivot row. The resulting matrix is:

$$\begin{bmatrix} 0 & 0 & 1 & 3 \\ 4 & 4 & -3 & 3 \\ 0 & 5 & -5/2 & 5/2 \end{bmatrix}$$

and $\ell = (2, 3, 1)'$.
The elimination is now complete. The solution is found by back substitution, using the vector ℓ to determine the order to be used in the substitution step.

$$\begin{aligned} x_3 &= 3/1 = 3 \\ x_2 &= [5/2 - (-5/2)3]/5 = 2 \\ x_1 &= [3 - (-3)3 - 4(2)]/4 = 1 \end{aligned}$$

Regression Computations Using the Sweep Operator

A basic computational problem in regression analysis is to obtain the solution to the normal equations $X'X\boldsymbol{\beta} = X'\boldsymbol{y}$ where X is a an $n \times p$ matrix of predictors, $\boldsymbol{\beta}$ is a $p \times 1$ vector of unknown parameters, and $\boldsymbol{y}$ is an $n \times 1$ response vector. The sweep operator discussed previously is one of the available methods to obtain the solution. If the sweep operator is applied to the first p rows of the appended matrix, denoted by A

$$A = \begin{bmatrix} X'X & X'\boldsymbol{y} \\ \boldsymbol{y}'X & \boldsymbol{y}'\boldsymbol{y} \end{bmatrix}$$

it can be shown that the resulting matrix, denoted by A^*, will be of the form

$$A^* = \begin{bmatrix} (X'X)^{-1} & (X'X)^{-1}X'\boldsymbol{y} \\ -\boldsymbol{y}'X(X'X)^{-1} & \boldsymbol{y}'\boldsymbol{y} - \boldsymbol{y}'X(X'X)^{-1}X'\boldsymbol{y} \end{bmatrix}$$

which also may be expressed in the form

$$A^* = \begin{bmatrix} (X'X)^{-1} & \hat{\boldsymbol{\beta}} \\ -\hat{\boldsymbol{\beta}}' & (n-p-1)s^2 \end{bmatrix}$$

recalling that $\hat{\boldsymbol{\beta}} = (X'X)^{-1}X'\boldsymbol{y}$ and $s^2 = (\boldsymbol{y}'\boldsymbol{y} - \boldsymbol{\beta}'X'\boldsymbol{y})/(n-p-1)$.

Thus we can obtain the estimate of $\boldsymbol{\beta}$, the residual SS and the inverse of $X'X$ in a single computational operation. Note that X and $\boldsymbol{y}$ used here are assumed to be in the centered form that is the column means have been subtracted from the data in each column and thus the matrix $X'X$ is the Sums of Squares and Cross Products(SSCP) matrix. The appended matrix on which sweep operations are initiated is usually denoted as A. In a computer implementation of a procedure which uses sweep operations, it is of interest to be able to construct the matrix A directly without storing the observations. This can be easily accomplished by appending $\boldsymbol{y}$ as a column vector on the right hand side of X and applying an algorithm such as that described in the Appendix to this appended matrix. The required matrix is obtained because of the fact that

$$(X \quad \boldsymbol{y})'(X \quad \boldsymbol{y}) = \begin{bmatrix} X'X & X'\boldsymbol{y} \\ \boldsymbol{y}'X & \boldsymbol{y}'\boldsymbol{y} \end{bmatrix} .$$

Thus, the matrix A is, in fact, the SSCP matrix of a data matrix that included $\boldsymbol{y}$ as its last column. Notice that A is of size $(p+1) \times (p+1)$ where p is the number of independent variables and that $a_{p+1,p+1}$ is the Total(corrected) SS. Many of the statistical quantities needed in a multiple regression analysis can be extracted from the matrix A^* as shown above. The availability of $(X'X)^{-1}$ enables us to compute statistics such as standard errors of the regression estimates, t-tests and confidence intervals. Since the computations are done using the centered form, $\hat{\beta}_0$ and s.e.$(\hat{\beta}_0)$ need to be computed separately using simple expressions.

To illustrate this method, we constructed matrix A corresponding to the example appearing in the Appendix:

$$A = \begin{bmatrix} 7154.41408 & -262.29985 & -571.12719 \\ -262.29985 & 218.55999 & 63.31596 \\ -571.17719 & 63.31596 & 33.81562 \end{bmatrix}$$

Recall that for this data $n = 25$ and $p = 2$. Sweep operations were therefore performed on the first two rows of A and the resulting matrix A^* is:

$$A^* = \begin{bmatrix} .146207{\bullet}10^{-3} & .175467{\bullet}10^{-3} & -.072393 \\ .175467{\bullet}10^{-3} & .467588{\bullet}10^{-2} & .202815 \\ .072393 & -.202815 & 9.65287 \end{bmatrix}$$

Using the elements of A an A^* and the means $\bar{x}_1$, $\bar{x}_2$ and $\bar{y}$, the computations needed to fit the regression model may now be performed.

Fitting Subsets of Independent Variables Using the Sweep Operator

As we have seen in the previous example, fitting a multiple regression model which includes all p independent variables $x_1, x_2, \ldots, x_p$ (and an intercept) can be accomplished by sweeping on the first p rows of A. These sweeps may be performed in any sequence because of the property of the sweep operator described earlier.

Sometimes one is interested in fitting a submodel which involves only some of the p independent variables. This may be achieved by sweeping on those rows of A which correspond to the variables which are to be included in the model of interest in any order. It is also possible to add a variable or delete a variable from a model by sweeping on the row of A^* corresponding to the variable of interest, where A^* represents the current status of matrix A (i.e., the matrix resulting from performing sweeps necessary to fit the current model).

To illustrate the procedures discussed in the previous paragraph, we start with the example of 2 independent variables used above. The matrix A obtained in this case is

$$A = \begin{bmatrix} 7154.4140 & -262.29985 & -571.12719 \\ -262.29985 & 218.55999 & 63.31596 \\ -571.12719 & 63.31596 & 63.81562 \end{bmatrix}$$

Recall also that $\bar{x}_1 = 52.5999$, $\bar{x}_2 = 20.24$, and that $\bar{y} = 9.42399$. Note that the first row (column) of A corresponds to the variable x_1, the second row (column) to x_2 and the last row (column) to y. The various models that can be fit from this data are:

$$\begin{array}{ll} y = \beta_0 + \beta_1 x_1 + e & \text{(model 1)} \\ y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + e & \text{(model 2)} \\ y = \beta_0 + \beta_2 x_2 + e & \text{(model 3)} \end{array}$$

Suppose that a sweep is performed on the 1^{st} row of A and that the resulting matrix, denoted by $A^{(1)}$ is

$$A^{(1)} = \begin{bmatrix} .13977{\bullet}10^{-3} & -.0366627 & -.0798286 \\ .036627 & 208.943 & 42.2769 \\ .0798286 & 42.3769 & 18.2233 \end{bmatrix}.$$

From this, we can fit model 1 to this data, by ignoring the row which corresponds to variable x_2, i.e., row 2. Then it is clear that $\hat{\beta}_1 = -.0798286$, and hence that

$$\hat{\beta}_0 = \bar{y} - \hat{\beta}_1 \bar{x}_1 = 9.423999 - (-0798286)(52,5999) = 13.62297$$

Further, it is easy to compute the Regression SS and thus the analysis of variance because

$$\begin{aligned}\text{Total SS} &= 63.81562 \text{ with 24 d.f., and} \\ \text{Residual SS} &= 18.2233 \text{ with 23 d.f.}\end{aligned}$$

Hence, $s^2 = 18.2233/23 = 0.7923$. The $(X'X)^{-1}$ matrix is the 1×1 matrix $0.13977{\bullet}10^{-3}$ which gives us the standard error of β_1 as

$$\text{s.e.}(\hat{\beta}_1) = (0.13977{\bullet}10^{-3})^{1/2}(0.7923)^{1/2} = 0.010524$$

and the standard error of $\hat{\beta}_0$ as

$$\begin{aligned}\text{s.e.}(\hat{\beta}_0) &= \left[\frac{1}{n} + \underline{\bar{x}}'(X'X)^{-1}\underline{\bar{x}}\right]^{1/2} s \\ &= \left[\frac{1}{25} + 52.5999(0.13977{\bullet}10^{-3})52.5999\right]^{1/2} (0.7923)^{1/2} \\ &= 0.5815\end{aligned}$$

The fitted model in this case is given by $\hat{y} = 13.62297 - 0.0798286x_1$. Now we sweep on the 2nd row of $A^{(1)}$ and obtain $A^{(1,2)}$ where

$$A^{(1,2)} = \begin{bmatrix} .146207{\bullet}10^{-3} & .175467{\bullet}10^{-3} & -.0723929 \\ .175467{\bullet}10^{-3} & .478599{\bullet}10^{-2} & .202815 \\ .0723929 & -.202815 & 9.62862 \end{bmatrix}$$

Note that $A^{(1,2)}$ is the same as A^* computed in the previous sections. Using $A^{(1,2)}$ we can fit model 2 as was done in the last section. To fit model 3, $y = \beta_0 + \beta_2 x_2 + e$ we need to sweep on row 1 of $A^{(1,2)}$ to remove variable x_1 from model 2. The resulting matrix, say $A^{(2)}$ could also have been obtained by sweeping A on row 2 in the first place. $A^{(2)}$ is given by

$$A^{(2)} = \begin{bmatrix} 6839.62 & 1.20013 & -495.14 \\ -1.20013 & .45754{\bullet}10^{-2} & .289696 \\ -495.14 & -.289696 & 45.4732 \end{bmatrix}$$

The computations necessary to complete fitting model 3 using $A^{(2)}$ are left as an exercise.

Stepwise Regression Using the Sweep Operator

When the number of independent variables that has to be considered is large, an automated stepwise regression procedure is very useful for model building. The techniques described here use the sweep operator to perform a stepwise regression using the sums of squares and cross products (SSCP) matrix. Suppose that we denote the raw sums of squares matrix by $X_R'X_R$. If a sweep on the first row is performed, the sums of squares and cross products (SSCP) matrix, which we denote by $X'X$ is obtained in the lower right hand corner as shown below:

$$X_R'X_R \overset{\text{sweep}(1)}{\longrightarrow} \begin{bmatrix} n & \bar{x}_1 & \cdots & \bar{x}_p \\ -\bar{x}_1 & & & \\ \cdots & & X'X & \\ -\bar{x}_p & & & \end{bmatrix}$$

where $\bar{x}_i$, $i = 1, \ldots, p$ are the means of the p independent variables. Hence, using the $X'X$ matrix (i.e., the $X'X$ matrix computed from the centered X matrix) as the starting point for stepwise regression computations, is equivalent to starting with the intercept term β_0 in the regression model already fitted. To define the computations associated with the procedure, its assumed that we have already performed sweep operations on the first k rows of $X'X$. That is, we assume that k variables out of the p independent variables $x_1, x_2, \ldots, x_p$ have already been entered and are in the current regression model in addition to the intercept term. Without los of generality we also assume that the k variables in the model are the first k variables i.e., $x_1, x_2, \ldots, x_k$. Consider the appended matrix

$$A = \begin{bmatrix} X'X & X'\boldsymbol{y} \\ \boldsymbol{y}'X & \boldsymbol{y}'\boldsymbol{y} \end{bmatrix}$$

and denote the resulting matrix after sweeping on the first k rows of A by A^* where

$$A^* = \begin{bmatrix} X_1'X_1^{-1} & \hat{\boldsymbol{\beta}}_1 \\ & \\ -\hat{\beta}_1' & \text{SSE} \end{bmatrix}$$

The elements of A and A^* and the means of the variables $x_1, x_2, \ldots, x_p$ and y are useful for computing statistical quantities necessary in carrying out a stepwise regression procedure (and fitting regression models, in general).

The following is a list (not complete) of possible computations which are of interest.

1. (a) Residual Sum of Squares = SSE with n–k–1 d.f. $= a^*_{p+1,p+1}$,
   (b) Regression Sum of Squares $= (a_{p+1,p+1} - \text{SSE})$ with k d.f.

2. Square of the multiple correlation coefficient, R^2

$$R^2 = (\boldsymbol{y}'\boldsymbol{y} - \text{SSE})/\boldsymbol{y}'\boldsymbol{y} = (a_{p+1,p+1} - \text{SSE})/a_{p+1,p+1}$$

.

3. For each variable $x_j,\ j = 1, \ldots, k$ which are in the current regression function, the following:

   (a) The least squares estimate of the regression coefficient β_j is $\hat{\beta}_j = a^*_{j,p+1}$.

   (b) Standard Error of each $\hat{\beta}_j$ is given by s_j, where

$$s_j = \left[\frac{\text{SSE}}{n-k-1} a^*_{jj}\right]^{1/2}$$

   (c) F-to-remove x_j from the current model is given by F_j, where $F_j = (\hat{\beta}_j/s_j)^2$. Notice that this is identical to the square of the t-statistic appropriate for testing $H_0 :\ \beta_j = 0$ vs. $H_a :\ \beta_j \neq 0$.

4. The least squares estimate of the intercept, β_0 and its standard error are

$$\begin{aligned} \hat{\beta}_0 &= \bar{y} - \Sigma_{j=1}^{k} \hat{\beta}_j \bar{x}_j, \text{S.E.}(\hat{\beta}_0) = g^{1/2} s \text{ where } g = \frac{1}{n} + \underline{\bar{x}}'(X'X)^{-1}\underline{\bar{x}}\,. \\ \bar{y} &= \left[\Sigma_{i=1}^{n} y_i\right]/n,\ \bar{x}_j = \left[\Sigma_{i=1}^{n} x_{ji}\right]/n \ \text{ for } j = 1, \ldots, k \text{ and} \\ \underline{\bar{x}} &= (\bar{x}_1, \bar{x}_2, \ldots, \bar{x}_k)'. \end{aligned}$$

5. For each variable $x_j, j = k+1, \ldots, p$ not in the current regression model, the following:

   (a) Partial correlation of each x_j, $j = k+1, \ldots, p$ with y, the dependent variable adjusted for the variables already in the equation i.e., $x_1, \ldots, x_k$, is given by r_j where

$$r_j = \frac{a^*_{j,p+1}}{(a^*_{jj}\, a_{p+1,p+1})^{1/2}}$$

(b) F-to-enter x_j, $j = k+1, \ldots, p$ into the current regression model is gien by F_j where

$$F_j = \frac{(n-k-2)r_j^2}{1-r_j^2}$$

6. Prediction function for a given set of x values is given by

$$\bar{y} = \hat{\beta}_0 + \Sigma_{j=1}^{k} \hat{\beta}_j x_j \,.$$

In the above discussion we assumed that the variable $x_1, x_2, \ldots, x_k$ were selected sequentially to be entered into the model. However, in actual practice the variables are not entered into the model in a sequential fashion. Rather the variable to be entered at each stage is determined by considering the F-to-enter statistics for variables not in the model, which are computed at each stage including the initial stage. To illustrate this procedure, let us consider an example.

An Example

In this example the significance level for both entry and removal of variables will be set at .1. Consider the following set of data.

x_1	x_2	x_3	x_4	y
7.0	26.0	6.0	60.0	78.5
1.0	29.0	15.0	52.2	74.3
11.0	56.0	8.0	20.0	104.3
11.0	31.0	8.0	47.0	87.6
7.0	52.0	6.0	33.0	95.9
11.0	55.0	9.0	22.0	109.2
3.0	71.0	17.0	6.0	102.7
1.0	31.0	22.0	44.0	72.5
2.0	54.0	18.0	22.0	93.1
21.0	47.0	4.0	26.0	115.9
1.0	40.0	23.0	34.0	83.8
11.0	66.0	9.0	12.0	113.3
10.0	68.0	8.0	12.0	109.4

For this data

$$\bar{y} = 95.423 \quad \bar{x}_1 = 7.4615 \quad \bar{x}_2 = 48.1538 \quad \bar{x}_3 = 11.7692 \quad \bar{x}_4 = 30.0 \,.$$

In each of the following steps we'll consider the results of sweep operations carried out on the following tableau:

$$A = \left[\begin{array}{cc} X'X & X'\boldsymbol{y} \\ \boldsymbol{y}'X & \boldsymbol{y}'\boldsymbol{y} \end{array} \right]$$

where $X'X$ is the SSCP matrix. In fact A could be constructed directly by appending $\boldsymbol{y}$ as the last column of data. Efficient methods for constructing the matrix A for regression data were discussed previously.

The initial data tableau for this problem is:

415.231	251.077	–372.615	–290	775.962
215.077	2905.69	–166.538	–3041	2292.95
–372.615	–166.538	493.308	38	–618.231
–290	–3041	38	3362	–2481.7
775.962	2292.95	–618.231	–2481.7	2715.76

`Step 0:` Step No. $k = 0$, No. of Variables in the Model $\ell = 1$

`Variables in the model`

Var	β_j	s_j	F-to-remove
x_0	95.423		

`Variables not in the model`

Var	r_j	F-to-enter
x_1	.73072	12.603
x_2	.81625	21.960
x_3	–.53467	4.403
x_4	–.82131	22.799

Variable x_4 has the largest F-to-enter value which is significant at 10% since $F_{.1}(1, n-\ell-1) = F_{.1}(1, 11) = 3.23$.

`Step 1` Enter x_4: Sweep row 4
Step No. $k = 1$, No. of Variables in the Model $\ell = 2$
The resulting tableau is:

390.216	–11.2342	–369.338	0.0862582	561.895
–11.2342	155.044	–132.167	0.904521	48.2038
–369.338	–132.167	491.878	–0.0113028	–590.181
–0.0862582	–0.904521	0.0113028	0.000297442	–0.738162
561.895	48.2038	–590.181	0.738162	883.867

Variables to remove - none

Variable x_1 has the largest F-to-enter value which is significant at 10% since $F_{.1}(1, n-\ell-1) = F_{.1}(1, 10) = 3.29$.

Variables in the model

Var	β_j	s_j	F-to-remove
x_0	117.57		
x_4	–.7382	.1546	22.799

Variables not in the model

Var	r_j	F-to-enter
x_1	.95677	108.22
x_2	.13021	.172
x_3	–.89508	40.295

`Step 2` Enter x_1:Sweep row 1
Step No. $k = 2$, No. of Variables in the Model $\ell = 3$
The resulting tableau is:

0.00256268	–0.0287877	–0.946495	0.000221052	1.43996
0.0287897	154.72	–142.8	0.907004	64.3806
0.946495	–142.8	142.302	0.0703402	–58.3499
0.000221052	–0.907004	–0.0703402	0.00031651	–0.613954
–1.43996	64.3806	–58.3499	0.613954	74.7621

Variables in the model

Var	β_j	s_j	F-to-remove
x_0	103.097		
x_4	–.614		159.32
x_1	1.44		108.23

Variables not in the model

Var	r_j	F-to-enter
x_2	.59861	5.026
x_3	–.56571	4.236

Variable to remove: Variable x_1 has the smallest F-to-remove value which is significant at 10%. Not removed.

Variable to enter: Variable x_2 has the largest F-to-enter value which is significant at 10% since $\mathrm{F}_{.1}(1, 9) = 3.36$.

`Step 3` Enter x_2:Sweep row 2
$k = 3$. Step No. $k = 3$, No. of Variables in the Model $\ell = 4$
The resulting tableau is:

0.00256804	0.000186076	–0.973067	0.000489824	1.45194
0.000186076	0.00646328	–0.922955	0.00586223	0.41611
0.973067	0.922955	10.504	0.907465	1.07046
0.000389824	0.00586223	–0.907465	0.00563357	–0.23654
–1.45194	–0.41611	1.07046	0.23654	47.9727

`Variables in the model`

Var	β_j	s_j	F-to-remove
x_0	71.6482		
x_4	$-.2365$	1.863	
x_1	1.4519		154.0
x_2	.4161		5.03

`Variables not in the model`

Var	r_j	F-to-enter
x_3	.0477	.0182

Variable to remove: Variable x_4 has the smallest F-to-remove value which is not significant at 10% since $F.1(1,9) = 3.36$. Remove x_4 from model.

`Step 4` Remove x_4:Sweep row 4 Step No. $k = 2$, No. of Variables in the Model $\ell = 3$

The resulting tableau is:

0.00254107	–0.00021957	–0.910274	–0.0691966	1.46831
–0.00021957	0.000363125	0.0213409	–1.04059	0.66225
0.910274	–0.0213409	156.68	–161.082	39.1727
0.0691966	1.04059	–161.082	177.507	–41.9876
–1.46831	–0.66225	39.1727	–41.9876	57.9045

`Variables in the model`

Var	β_j	s_j	F-to-remove
x_0	52.5774		
x_1	1.4683		146.52
x_2	.6623		208.58

`Variables not in the model`

Var	r_j	F-to-enter
x_4	–.41415	1.8633
x_3	.41126	1.8321

Variable to remove: Variable x_1 has the smallest F-to-remove value which is significant at 10%. Not removed.

Variable to enter: Variable x_4 has the largest F-to-enter value which is significant at 10%. Not entered.

So this model is the final model obtained through stepwise regression with the significance level set at 10% for both entry and removal of variables.

Source	d.f.	SS	MS	F
Regression	2	2657.8555	1328.928	229.50
Residual	10	57.9045	5.790	

$R^2 = 0.98$

Thus the model selected by the stepwise procedure is:

$$y = 52.5774 + 1.4683x_1 + 0.6623x_2 \,.$$

Appendix

Computing the $X'X$ matrix

In this section we describe a simple straightforward algorithm for computing the $X'X$ matrix. The rows of X which consist of the centered data, are input a row at a time and are used to update the $X'X$ matrix. Thus storage of the X matrix is not necessary unless it is needed for some other purpose. Let the X matrix be

$$X = \begin{bmatrix} x_{11} & x_{21} & \cdots & x_{p1} \\ x_{12} & x_{22} & \cdots & x_{p2} \\ \vdots & \vdots & \vdots & & \vdots \\ x_{1n} & x_{2n} & & x_{pn} \end{bmatrix}_{n\times p} = \begin{bmatrix} \boldsymbol{x}_1' \\ \boldsymbol{x}_2' \\ \vdots \\ \boldsymbol{x}_n' \end{bmatrix}$$

Hence the $p \times p\, X'X$ matrix can be expressed as

$$X'X = [\boldsymbol{x}_1, \boldsymbol{x}_2 \ldots, \boldsymbol{x}_n] \begin{bmatrix} \boldsymbol{x}_1' \\ \boldsymbol{x}_2' \\ \vdots \\ \boldsymbol{x}_n' \end{bmatrix} = \boldsymbol{x}_1\boldsymbol{x}_1' + \boldsymbol{x}_2\boldsymbol{x}_2' + \cdots + \boldsymbol{x}_n\boldsymbol{x}_n' \,,$$

i.e., as a sum of n, $p \times p$ matrices, each dependent only on a single row of data. Thus an algorithm for building $X'X$ can be given as follows:

`Algorithm:`

1. Initialize a $p \times p$ array A to zero and set i = 1.
2. Repeat until i = n.
   (a) Read i^{th} row of data $\boldsymbol{x}_i$.
   (b) Set $A \leftarrow A + \boldsymbol{x}_i\, \boldsymbol{x}_i'$ and $i = i + 1$.
3. Return $X'\,X$ in A.

In practice, instead of using centered data to build the $X'X$ matrix an updating algorithm such as Youngs-Cramer algorithms is incorporated in to the above algorithm so that the corrected $X'X$ matrix (or the SSCP matrix) is obtained efficiently and accurately without storing the data matrix. In addition, a column representing y is appended to the X matrix so that computations associated with $X'\boldsymbol{y}$ and $\boldsymbol{y}'\boldsymbol{y}$ are also accomplished automatically.

Modified Youngs-Cramer Algorithm for constructing the centered $X'X$ matrix

Initialize a $(p+1) \times (p+1)$ array xpx to zero and a $(p+1) \times 1$ array xm to the first data vector (including a value for y).

`Algorithm:`

1. $xpx \leftarrow 0$
2. $xm \leftarrow$ read data vector
3. for $k = 2$ to n do
   (a) $xm \leftarrow$ read data vector
   (b) for $i = 1$ to $(p+1)$ do
      i. $xm_i \leftarrow xm_i + x_i$
      ii. for $i = 1$ to $(p+1)$ do
      $xpx_{i,j} \leftarrow xpx_{i,j} + (kx_i - xm_i)(kx_j - xm_j)/k(k-1)$
      iii. end
4. $xm \leftarrow xm/n$
5. return $xpx,\ xm$

Data used in the subset regression example:

Obs.	x_1	x_2	y
1	35.3	20	10.98
2	29.7	20	11.13
3	30.8	23	12.51
4	58.8	20	8.40
5	61.4	21	9.27
6	71.3	22	8.73
7	74.4	11	6.36
8	76.7	23	8.50
9	70.7	21	7.82
10	57.5	20	9.14
11	46.4	20	8.24
12	28.9	21	12.19
13	28.1	21	11.88
14	39.1	19	9.57
15	46.8	23	10.94
16	48.5	20	9.58
17	59.3	22	10.09
18	70.0	22	8.11
19	70.0	11	6.83
20	74.5	23	8.88
21	72.1	20	7.68
22	58.1	21	8.47
23	44.6	20	8.86
24	33.4	20	10.36
25	28.6	22	11.08

y = pounds of steam used per month
x_1 = average atmospheric temperature in $^\circ F$
x_2 = number of operating days in the month