Multiple Linear Regression I:
Basic Model and Sum-of-Squares Decomposition

More General Regression Models

Two Main Generalizations:

- The ***multilinear model*** with linear dependence on $p \in \mathbb{N}$ parameters $X_1, \ldots, X_p$,

$$\mu_{Y|x_1,\ldots,x_p} = \beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p \tag{26.1}$$

  and

- The ***polynomial model*** with dependenco on a polynomial of degree p of a single parameter X,

$$\mu_{Y|x} = \beta_0 + \beta_1 x + \beta_2 x^2 + \cdots + \beta_p x^p.$$

The Polynomial Model

A random sample of size n, $(x_i, Y \mid x_i)$, $i = 1, \dots, n$ is given. We write $Y_i := Y \mid x_i$ as usual.

Goal: Find $b_0, \dots, b_p$ such that for

$$y_i = b_0 + b_1 x_i + \cdots + b_p x_i^p + e_i, \qquad i = 1, \dots, n, \tag{26.2}$$

the sum of squares error

$$\mathrm{SS_E} = \sum_{i=1}^{n} e_i^2 = \sum_{i=1}^{n} \bigl(y_i - (b_0 + b_1 x_i + \cdots + b_p x_i^p)\bigr)^2 \tag{26.3}$$

is minimized.

The Model Specification Matrix

To discuss the model

$$Y_i = Y \mid x_i = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \cdots + \beta_p x_i^p + E_i, \tag{26.4}$$

it is convenient to adopt a matrix formalism. We define

$$\mathbf{Y} = \begin{pmatrix} Y_1 \\ \vdots \\ Y_n \end{pmatrix}, \quad X = \begin{pmatrix} 1 & x_1 & x_1^2 & \dots & x_1^p \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_n & x_n^2 & \dots & x_n^p \end{pmatrix}, \quad \boldsymbol{\beta} = \begin{pmatrix} \beta_0 \\ \vdots \\ \beta_p \end{pmatrix}, \quad \mathbf{E} = \begin{pmatrix} E_1 \\ \vdots \\ E_n \end{pmatrix}.$$

Then (26.4) can be written as

$$\mathbf{Y} = X\boldsymbol{\beta} + \mathbf{E}. \tag{26.5}$$

The matrix X is called the ***model specification matrix***. We see from (26.5) that the polynomial model is a ***linear regression model***.

Polynomial Regression

Defining

$$\widehat{\beta} = \mathbf{b} := \begin{pmatrix} b_0 \\ \vdots \\ b_p \end{pmatrix} \qquad \text{and} \qquad \mathbf{e} = \begin{pmatrix} e_1 \\ \vdots \\ e_n \end{pmatrix}$$

(26.2) becomes

$$\mathbf{Y} = X\mathbf{b} + \mathbf{e},$$

where $\mathbf{b}$ is chosen to minimize the error sum of squares

$$\mathrm{SS}_{\mathrm{E}} = \langle \mathbf{Y} - X\mathbf{b}, \mathbf{Y} - X\mathbf{b} \rangle = (\mathbf{Y} - X\mathbf{b})^T (\mathbf{Y} - X\mathbf{b}). \tag{26.6}$$

Here A^T denotes the transpose of a matrix A and $\langle a, b \rangle = \sum_{i=1}^{n} a_i b_i$ is the usual scalar product of two vectors $a, b \in \mathbb{R}^n$.

Minimizing the $\mathrm{SS_E}$

Using the norm $\|a\| = \sqrt{\langle a, a\rangle}$, we write

$$\begin{aligned}\mathrm{SS_E} &= \langle \mathbf{Y} - X\mathbf{b}, \mathbf{Y} - X\mathbf{b}\rangle \\ &= \|\mathbf{Y}\|^2 - 2\langle X\mathbf{b}, \mathbf{Y}\rangle + \|X\mathbf{b}\|^2.\end{aligned}$$

The minimum of the sum-of-squares error is found from

$$\nabla_{\mathbf{b}} \mathrm{SS_E} = \begin{pmatrix} \frac{\partial \mathrm{SS_E}}{\partial b_0} \\ \vdots \\ \frac{\partial \mathrm{SS_E}}{\partial b_p} \end{pmatrix} = 0.$$

Hence, we need to solve

$$-2\nabla_b \langle X\mathbf{b}, \mathbf{Y}\rangle + \nabla_b \langle X\mathbf{b}, X\mathbf{b}\rangle = 0.$$

Minimizing the SS_E

We use that

$$\langle X\mathbf{b}, \mathbf{Y}\rangle = \langle \mathbf{b}, X^T\mathbf{Y}\rangle = \sum_{i=0}^{p} b_i (X^T\mathbf{Y})_{i+1}$$

to see

$$\frac{\partial}{\partial b_k}\langle X\mathbf{b}, \mathbf{Y}\rangle = (X^T\mathbf{Y})_{k+1}.$$

and hence

$$\nabla_b \langle Xb, Y\rangle = \begin{pmatrix} \frac{\partial\langle X\mathbf{b},\mathbf{Y}\rangle}{\partial b_0} \\ \vdots \\ \frac{\partial\langle Xb,Y\rangle}{\partial b_p} \end{pmatrix} = \begin{pmatrix} (X^T\mathbf{Y})_1 \\ \vdots \\ (X^T\mathbf{Y})_{p+1} \end{pmatrix} = X^T\mathbf{Y}.$$

It is also not difficult to show that

$$\nabla_{\mathbf{b}}\langle X\mathbf{b}, X\mathbf{b}\rangle = 2X^TX\mathbf{b}.$$

The Regression Coefficients

It follows that the stationary point is given by

$$(X^T X)\mathbf{b} = X^T \mathbf{Y}.$$

Since the entries of X are numerical, $X^T X$ will almost surely be invertible.

Then the regression coefficients are given by

$$\mathbf{b} = (X^T X)^{-1} X^T \mathbf{Y}.$$

Of course, this formulation can also be used for simple linear regression; this is just the case $p = 1$.

Since the values in X are numerical, practical calculations are best done using a computer.

A Polynomial Model

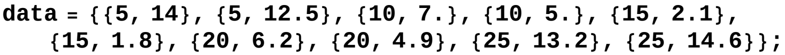
26.1. Example. A study is conducted to develop an equation by which the unit cost of producing a new drug (Y) can be predicted based on the number of units produced (X). The proposed model is

$$\mu_{Y|x} = \beta_0 + \beta_1 x + \beta_2 x^2.$$

The following data are available:

x	5	5	10	10	15	15	20	20	25	25
y	14.0	12.5	7.0	5.0	2.1	1.8	6.2	4.9	13.2	14.6

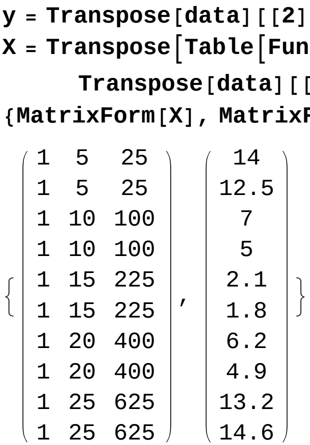
We will use Mathematica for our calculations. We first enter the data as a list of pairs:



```
data = {{5, 14}, {5, 12.5}, {10, 7.}, {10, 5.}, {15, 2.1},
   {15, 1.8}, {20, 6.2}, {20, 4.9}, {25, 13.2}, {25, 14.6}};
```

A Polynomial Model

We construct the model specification matrix X and the response vector y:



```
y = Transpose[data][[2]];
X = Transpose[Table[Function[x, x^k] /@
      Transpose[data][[1]], {k, 0, 2}]];
{MatrixForm[X], MatrixForm[y]}
```

$$\left\{\begin{pmatrix} 1 & 5 & 25 \\ 1 & 5 & 25 \\ 1 & 10 & 100 \\ 1 & 10 & 100 \\ 1 & 15 & 225 \\ 1 & 15 & 225 \\ 1 & 20 & 400 \\ 1 & 20 & 400 \\ 1 & 25 & 625 \\ 1 & 25 & 625 \end{pmatrix}, \begin{pmatrix} 14 \\ 12.5 \\ 7 \\ 5 \\ 2.1 \\ 1.8 \\ 6.2 \\ 4.9 \\ 13.2 \\ 14.6 \end{pmatrix}\right\}$$

A Polynomial Model

Then **b** is given by

```
b = Inverse[Transpose[X].X].Transpose[X].y;
MatrixForm[b]
```

$$\begin{pmatrix} 27.3 \\ -3.313 \\ 0.111 \end{pmatrix}$$

Thus we obtain

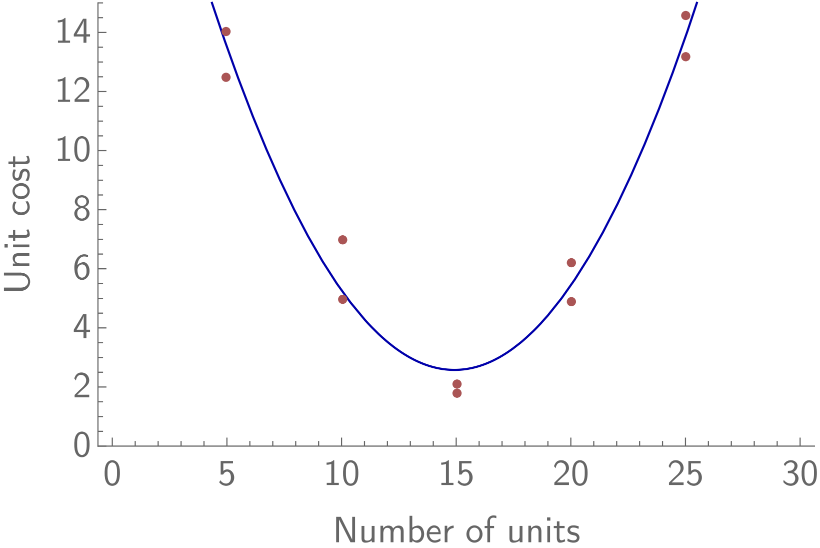
$$\hat{\mu}_{Y|x} = 27.3 - 3.313 \cdot x + 0.111 \cdot x^2.$$

The same result can also be found by using **NonLinearModelFit**:

```
model = NonlinearModelFit[data, b0 + b1 x + b2 x^2,
    {b0, b1, b2}, x];
model["BestFit"]
```

$27.3 - 3.313\,x + 0.111\,x^2$

A Polynomial Model



A Polynomial Model

We can also use **LinearModelFit** with a given model specification matrix X (called a ***design matrix*** in Mathematica) and the response vector $\mathbf{y}$:

```
model = LinearModelFit[{X, y}];
model["BestFit"]

27.3 #1 - 3.313 #2 + 0.111 #3
```

The output is a “pure function” with three arguments. To obtain the desired expression, we need to insert the appropriate monomials:

```
Evaluate[model["BestFit"]] &[1, x, x^2]

27.3 - 3.313 x + 0.111 x^2
```

The Multilinear Model

In the mulilinear model, we assume that Y depends on several factors $x_1, \dots, x_p$,

$$Y \mid x = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_p x_p + E.$$

We take a random sample $(x_{1i}, x_{2i}, \dots, x_{pi}; Y \mid x_{1i}, x_{2i}, \dots, x_{pi})$, $i = 1, \dots, n$, writing $Y_i = Y \mid x_{1i}, x_{2i}, \dots, x_{pi}$ as usual.

We select $b_0, \dots, b_p$ such that for

$$y_i = b_0 + b_1 x_{1i} + \cdots + b_p x_{pi} + e_i, \qquad i = 1, \dots, n, \tag{26.7}$$

the sum of squares error

$$\mathrm{SS_E} = \sum_{i=1}^{n} e_i^2 = \sum_{i=1}^{n} \big(y_i - (b_0 + b_1 x_{1i} + \cdots + b_p x_{pi})\big)^2 \tag{26.8}$$

is minimized.

The Multilinear Model

In fact, the situation is identical to the polynomial model if the model determination matrix X is replaced by

$$X = \begin{pmatrix} 1 & x_{11} & \dots & x_{p1} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_{1n} & \dots & x_{pn} \end{pmatrix}.$$

We again have

$$\mathbf{Y} = X\boldsymbol{\beta} + \mathbf{E}.$$

and estimate β by minimizing

$$\mathrm{SS}_\mathrm{E} = (\mathbf{Y} - X\mathbf{b})^T(\mathbf{Y} - X\mathbf{b}). \tag{26.9}$$

All following calculations remain unchanged and we obtain

$$\mathbf{b} = (X^TX)^{-1}X^T\mathbf{Y}.$$

A Multilinear Model

26.2. Example. An equation is to be developed from which we can predict the gasoline mileage of an automobile based on its weight and temperature at the time of operation. The model being estimated is

$$\mu_{Y|x} = \beta_0 + \beta_1 x_1 + \beta_2 x_2.$$

These data are available:

Car number	1	2	3	4	5	6	7	8	9	10
Weight in tons (x_1)	1.35	1.90	1.70	1.80	1.30	2.05	1.60	1.80	1.85	1.40
Temp. in $^\circ$F (x_2)	90	30	80	40	35	45	50	60	65	30
Miles/Gallon (y)	17.9	16.5	16.4	16.8	18.8	15.5	17.5	16.4	15.9	18.3

We enter the data as follows:

```
rowdata :=
 {{1.35, 1.90, 1.70, 1.80, 1.30, 2.05, 1.60, 1.80, 1.85, 1.40},
  {90, 30, 80, 40, 35, 45, 50, 60, 65, 30},
  {17.9, 16.5, 16.4, 16.8, 18.8, 15.5, 17.5, 16.4, 15.9, 18.3}}
```

A Multilinear Model

We construct the specification matrix and response vector before obtaining the model parameters:

```
X = Transpose[{Table[1, {i, 1, Length[rowdata[[1]]]}],
     rowdata[[1]], rowdata[[2]]}];
y = rowdata[[3]];
{MatrixForm[X], MatrixForm[y]}
```

```
  ⎛ 1  1.35  90 ⎞   ⎛ 17.9 ⎞
  ⎜ 1   1.9  30 ⎟   ⎜ 16.5 ⎟
  ⎜ 1   1.7  80 ⎟   ⎜ 16.4 ⎟
  ⎜ 1   1.8  40 ⎟   ⎜ 16.8 ⎟
{ ⎜ 1   1.3  35 ⎟ , ⎜ 18.8 ⎟ }
  ⎜ 1  2.05  45 ⎟   ⎜ 15.5 ⎟
  ⎜ 1   1.6  50 ⎟   ⎜ 17.5 ⎟
  ⎜ 1   1.8  60 ⎟   ⎜ 16.4 ⎟
  ⎜ 1  1.85  65 ⎟   ⎜ 15.9 ⎟
  ⎝ 1   1.4  30 ⎠   ⎝ 18.3 ⎠
```

A Multilinear Model

```
b = Inverse[Transpose[X].X].Transpose[X].y; MatrixForm[b]
```

$$\begin{pmatrix} 24.7489 \\ -4.15933 \\ -0.014895 \end{pmatrix}$$

We could also have used **LinearModelFit** based on the model specification matrix,

```
model = LinearModelFit[{X, y}];
Evaluate[model["BestFit"]] &[1, x1, x2]
```

$$24.7489 - 4.15933\,x_1 - 0.014895\,x_2$$

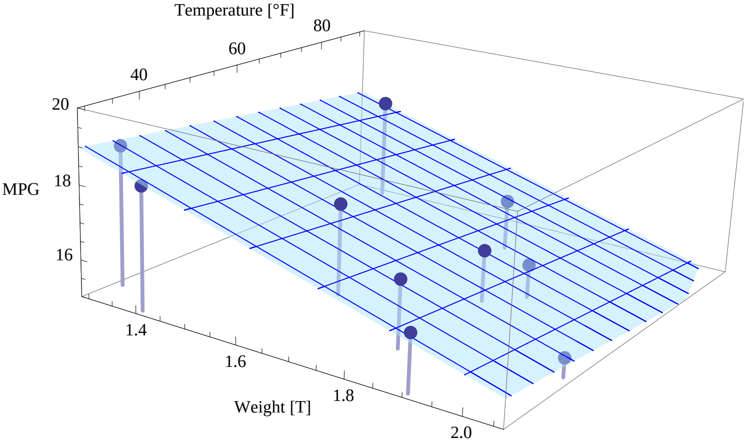
or entered the data directly in the form of a list of triples (x_1, x_2, y),

```
data = Transpose[{a1, a2, y}];
model = LinearModelFit[data, {x1, x2}, {x1, x2}];
model["BestFit"]
```

$$24.7489 - 4.15933\,x_1 - 0.014895\,x_2$$

A Multilinear Model

The model gives a regression plane:



Error Analysis: Total Variation

Let us now analyze the sources of variation in our models. The total variation is given by

$$\mathrm{SS_T} = \sum_{i=1}^{n} (Y_i - \overline{Y})^2$$

It is convenient to express this in matrix notation: we define the $n \times n$ matrix

$$P := \frac{1}{n}\begin{pmatrix} 1 & 1 & \dots & 1 \\ \vdots & \vdots & & \vdots \\ 1 & 1 & \dots & 1 \end{pmatrix}.$$

Then it is easy to see that

$$P\mathbf{Y} = \begin{pmatrix} \overline{Y} \\ \vdots \\ \overline{Y} \end{pmatrix}.$$

Error Analysis: The P Projection

This allows us to write

$$\mathrm{SS_T} = \langle (\mathbb{1}_n - P)\mathbf{Y}, (\mathbb{1}_n - P)\mathbf{Y} \rangle$$

where $\mathbb{1}_n$ is the $n \times n$ unit matrix. We further remark that

$$P^2 = P, \qquad \text{and} \qquad P^T = P. \tag{26.10}$$

A matrix with the properties (26.10) is said to be an ***orthogonal projection***. We can easily check that (26.10) implies

$$(\mathbb{1}_n - P)^2 = \mathbb{1}_n - P, \qquad \text{and} \qquad (\mathbb{1}_n - P)^T = \mathbb{1}_n - P.$$

Then

$$\mathrm{SS_T} = \langle \mathbf{Y}, (\mathbb{1}_n - P)^T(\mathbb{1}_n - P)\mathbf{Y} \rangle = \langle \mathbf{Y}, (\mathbb{1}_n - P)\mathbf{Y} \rangle \tag{26.11}$$

Such an expression is called a ***quadratic form*** in $\mathbf{Y}$.

The Hat Matrix

Recall that our both the polynomial and the multilinear models were based on writing the n responses in the form

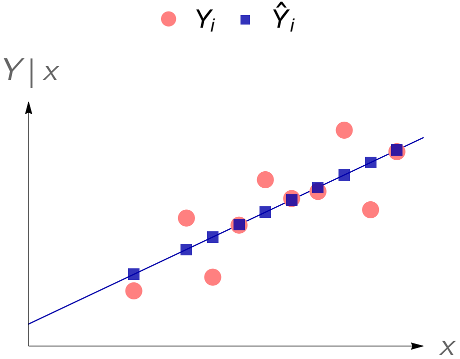
$$\mathbf{Y} = X\mathbf{b} + \mathbf{e}$$

where $\mathbf{e}$ is the least-squares vector of residuals and $\mathbf{Y} = (Y_1, \dots, Y_n)^T$ is the vector of the responses.

The vector

$$\widehat{\mathbf{Y}} := X\mathbf{b}$$

then represents the predicted values of the responses, i.e., the points $\widehat{Y}_i$ lying on the regression curve at x_i.



The Hat Matrix

Since $\mathbf{b} = (X^T X)^{-1} X^T \mathbf{Y}$ we may write

$$\widehat{\mathbf{Y}} = H\mathbf{Y}, \qquad\qquad H := X(X^T X)^{-1} X^T,$$

where the $n \times n$ matrix H is called the ***hat matrix***. It associates to each measured response Y_i the predicted response $\widehat{Y}_i$.

Like P, the hat matrix is an orthogonal projection: we can check that

$$HX = X, \qquad\qquad H^T = H, \qquad\qquad H^2 = H.$$

Therefore, so is $\mathbb{1}_n - H$ and we have

$$(\mathbb{1}_n - H)X = 0, \quad (\mathbb{1}_n - H)^T = \mathbb{1}_n - H, \quad (\mathbb{1}_n - H)^2 = \mathbb{1}_n - H. \quad (26.12)$$

Error Analysis: Sum of Squares Error

We may therefore write the error sum of squares as

$$\begin{aligned}\mathrm{SS_E} &= \langle \mathbf{Y} - X\mathbf{b}, \mathbf{Y} - X\mathbf{b} \rangle \\ &= \langle (\mathbb{1}_n - H)\mathbf{Y}, (\mathbb{1}_n - H)\mathbf{Y} \rangle \\ &= \langle \mathbf{Y}, (\mathbb{1}_n - H)^T (\mathbb{1}_n - H)\mathbf{Y} \rangle \\ &= \langle \mathbf{Y}, (\mathbb{1}_n - H)\mathbf{Y} \rangle.\end{aligned}$$

We may therefore write out a sums of squares decomposition very easily:

$$\begin{aligned}\mathrm{SS_T} &= \langle \mathbf{Y}, (\mathbb{1}_n - P)\mathbf{Y} \rangle \\ &= \underbrace{\langle \mathbf{Y}, (\mathbb{1}_n - H)\mathbf{Y} \rangle}_{=\mathrm{SS_E}} + \underbrace{\langle \mathbf{Y}, (H - P)\mathbf{Y} \rangle}_{=:\mathrm{SS_R}}.\end{aligned}$$

Fundamental Sum-of-Squares Decomposition

We hence have the decomposition

$$\mathrm{SS_T} = \mathrm{SS_R} + \mathrm{SS_E} \tag{26.13}$$

where

(i) $\mathrm{SS_T}$ represents the total variation of the response variable Y,

(ii) $\mathrm{SS_R}$ (called the ***regression sum of squares***) represents the variation of the response predicted by the regression model and

(iii) $\mathrm{SS_E}$ represents the deviation of the response from the model.

Analogously to (25.2), the ***coefficient of multiple determination***,

$$R^2 = \frac{\mathrm{SS_R}}{\mathrm{SS_T}}$$

gives the proportion of the response variation in Y explained by the model.

Fundamental Sum-of-Squares Error Decomposition

26.3. Remark. It can be shown that

$$\mathrm{SS_R} = \langle \mathbf{Y}, (H-P)\mathbf{Y} \rangle = \langle (H-P)\mathbf{Y}, (H-P)\mathbf{Y} \rangle.$$

Then the equation

$$\mathrm{SS_T} = \mathrm{SS_R} + \mathrm{SS_E} \tag{26.14}$$

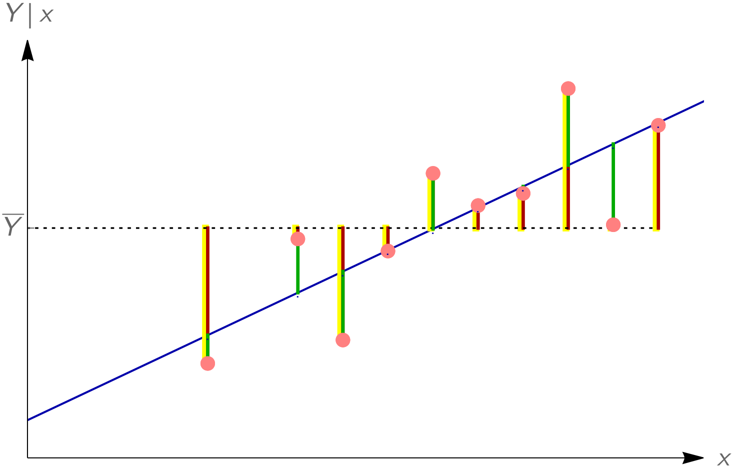
may be expressed as

$$\sum_{i=1}^{n} (Y_i - \overline{Y})^2 = \sum_{i=1}^{n} (\widehat{Y}_i - \overline{Y})^2 + \sum_{i=1}^{n} (Y_i - \widehat{Y}_i)^2$$

with $\widehat{\mathbf{Y}} = X\mathbf{b}$. Proving this inequality using only elementary algebraic manipulations is a daunting task.

Fundamental Sum-of-Squares Error Decomposition

$$\sum(\text{yellow lengths})^2 = \sum(\text{green lengths})^2 + \sum(\text{red lengths})^2$$


$Y\,|\,x$
$\overline{Y}$
x