THE REGRESSION MODEL IN MATRIX FORM

We will consider the linear regression model in matrix form.

For simple linear regression, meaning one predictor, the model is

$$Y_i = \beta_0 + \beta_1 x_i + \varepsilon_i \qquad \text{for } i = 1, 2, 3, \ldots, n$$

This model includes the assumption that the ε_i 's are a sample from a population with mean zero and standard deviation σ. In most cases we also assume that this population is normally distributed.

The multiple linear regression model is

$$Y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \beta_3 x_{i3} + \ldots + \beta_K x_{iK} + \varepsilon_i \qquad \text{for } i = 1, 2, 3, \ldots, n$$

This model includes the assumption about the ε_i 's stated just above.

This requires building up our symbols into vectors. Thus

$$\underset{n \times 1}{\boldsymbol{Y}} = \begin{pmatrix} Y_1 \\ Y_2 \\ Y_3 \\ \vdots \\ Y_n \end{pmatrix}$$

captures the entire dependent variable in a single symbol. The "$n \times 1$" part of the notation is just a shape reminder. These get dropped once the context is clear.

For simple linear regression, we will capture the independent variable through this $n \times 2$ matrix:

$$\underset{n \times 2}{\boldsymbol{X}} = \begin{pmatrix} 1 & x_1 \\ 1 & x_2 \\ 1 & x_3 \\ \vdots & \vdots \\ 1 & x_n \end{pmatrix}$$

The coefficient vector will be $\underset{2 \times 1}{\boldsymbol{\beta}} = \begin{pmatrix} \beta_0 \\ \beta_1 \end{pmatrix}$ and the noise vector will be $\underset{n \times 1}{\boldsymbol{\varepsilon}} = \begin{pmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \varepsilon_3 \\ \vdots \\ \varepsilon_n \end{pmatrix}$.

The simple linear regression model is written then as $\underset{n\times 1}{\boldsymbol{Y}} = \underset{n\times 2}{\boldsymbol{X}}\ \underset{2\times 1}{\boldsymbol{\beta}} + \underset{n\times 1}{\boldsymbol{\varepsilon}}$.

The product part, meaning $\underset{n\times 2}{\boldsymbol{X}}\ \underset{2\times 1}{\boldsymbol{\beta}}$, is found through the usual rule for matrix multiplication as

$$\underset{n\times 2}{\boldsymbol{X}}\ \underset{2\times 1}{\boldsymbol{\beta}} = \begin{pmatrix} 1 & x_1 \\ 1 & x_2 \\ 1 & x_3 \\ \vdots & \vdots \\ 1 & x_n \end{pmatrix} \begin{pmatrix} \beta_0 \\ \beta_1 \end{pmatrix} = \begin{pmatrix} \beta_0 + \beta_1\, x_1 \\ \beta_0 + \beta_1\, x_2 \\ \beta_0 + \beta_1\, x_3 \\ \vdots \\ \beta_0 + \beta_1\, x_n \end{pmatrix}$$

We usually write the model without the shape reminders as $\boldsymbol{Y} = \boldsymbol{X}\,\boldsymbol{\beta} + \boldsymbol{\varepsilon}$. This is a shorthand notation for

$$\begin{pmatrix} Y_1 \\ Y_2 \\ Y_3 \\ \vdots \\ Y_n \end{pmatrix} = \begin{pmatrix} \beta_0 + \beta_1\, x_1 + \varepsilon_1 \\ \beta_0 + \beta_1\, x_2 + \varepsilon_2 \\ \beta_0 + \beta_1\, x_3 + \varepsilon_3 \\ \vdots \\ \beta_0 + \beta_1\, x_n + \varepsilon_n \end{pmatrix}$$

It is helpful that the multiple regression story with $K \geq 2$ predictors leads to the same model expression $\boldsymbol{Y} = \boldsymbol{X}\,\boldsymbol{\beta} + \boldsymbol{\varepsilon}$ (just with different shapes). As a notational convenience, let $p = K + 1$. In the multiple regression case, we have

$$\underset{n\times p}{\boldsymbol{X}} = \begin{pmatrix} 1 & x_{11} & x_{12} & \cdots & x_{1K} \\ 1 & x_{21} & x_{22} & \cdots & x_{2K} \\ 1 & x_{31} & x_{32} & \cdots & x_{3K} \\ 1 & x_{41} & x_{42} & \cdots & x_{4K} \\ 1 & x_{51} & x_{52} & \cdots & x_{5K} \\ 1 & x_{61} & x_{62} & \cdots & x_{6K} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n1} & x_{n2} & \cdots & x_{nK} \end{pmatrix} \quad \text{and} \quad \underset{p\times 1}{\boldsymbol{\beta}} = \begin{pmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \beta_3 \\ \vdots \\ \beta_K \end{pmatrix}$$

The detail shown here is to suggest that $\boldsymbol{X}$ is a tall, skinny matrix. We formally require $n \geq p$. In most applications, n is much, much larger than p. The ratio $\frac{n}{p}$ is often in the hundreds.

If it happens that $\frac{n}{p}$ is as small as 5, we will worry that we don't have enough data (reflected in n) to estimate the number of parameters in $\boldsymbol{\beta}$ (reflected in p).

The multiple regression model is now $\underset{n\times 1}{\boldsymbol{Y}} = \underset{n\times p}{\boldsymbol{X}}\ \underset{p\times 1}{\boldsymbol{\beta}} + \underset{n\times 1}{\boldsymbol{\varepsilon}}$, and this is a shorthand for

$$\begin{pmatrix} Y_1 \\ Y_2 \\ Y_3 \\ \vdots \\ Y_n \end{pmatrix} = \begin{pmatrix} \beta_0 + \beta_1 x_{11} + \beta_2 x_{12} + \beta_3 x_{13} + \cdots + \beta_K x_{1K} + \varepsilon_1 \\ \beta_0 + \beta_1 x_{21} + \beta_2 x_{22} + \beta_3 x_{23} + \cdots + \beta_K x_{2K} + \varepsilon_2 \\ \beta_0 + \beta_1 x_{31} + \beta_2 x_{32} + \beta_3 x_{33} + \cdots + \beta_K x_{3K} + \varepsilon_3 \\ \vdots \\ \beta_0 + \beta_1 x_{n1} + \beta_2 x_{n2} + \beta_3 x_{n3} + \cdots + \beta_K x_{nK} + \varepsilon_n \end{pmatrix}$$

The model form $\boldsymbol{Y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}$ is thus completely general.

The assumptions on the noise terms can be written as $\mathrm{E}\,\boldsymbol{\varepsilon} = \mathbf{0}$ and $\mathrm{Var}\,\boldsymbol{\varepsilon} = \sigma^2\,\mathbf{I}$. The $\mathbf{I}$ here is the $n \times n$ identity matrix. That is,

$$\mathbf{I} = \begin{pmatrix} 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \end{pmatrix}$$

The variance assumption can be written as $\mathrm{Var}\,\boldsymbol{\varepsilon} = \begin{pmatrix} \sigma^2 & 0 & 0 & \cdots & 0 \\ 0 & \sigma^2 & 0 & \cdots & 0 \\ 0 & 0 & \sigma^2 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & \sigma^2 \end{pmatrix}$. You may see this expressed as $\mathrm{Cov}(\varepsilon_i, \varepsilon_j) = \sigma^2\ \delta_{ij}$, where

$$\delta_{ij} = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{if } i \neq j \end{cases}$$

We will call $\boldsymbol{b}$ as the estimate for unknown parameter vector $\boldsymbol{\beta}$.

You will also find the notation $\hat{\boldsymbol{\beta}}$ as the estimate.

Once we get $\boldsymbol{b}$, we can compute the *fitted* vector $\hat{\boldsymbol{Y}} = \boldsymbol{X}\,\boldsymbol{b}$. This fitted value represents an ex-post guess at the expected value of $\boldsymbol{Y}$.

The estimate $\boldsymbol{b}$ is found so that the fitted vector $\hat{\boldsymbol{Y}}$ is close to the actual data vector $\boldsymbol{Y}$. Closeness is defined in the least squares sense, meaning that we want to minimize the criterion Q, where

$$Q = \sum_{i=1}^{n} \left(Y_i - \left(\boldsymbol{X}\,\boldsymbol{b} \right)_{i^{\text{th}} \text{ entry}} \right)^2$$

This can be done by differentiating this quantity $p = K + 1$ times, once with respect to b_0, once with respect to b_1, ….., and once with respect to b_K. This is routine in simple regression ($K = 1$), and it's possible with a lot of messy work in general.

It happens that Q is the squared length of the vector difference $\boldsymbol{Y} - \boldsymbol{Xb}$. This means that we can write

$$Q = \underbrace{\left(\boldsymbol{Y} - \boldsymbol{Xb} \right)'}_{1 \times n} \underbrace{\left(\boldsymbol{Y} - \boldsymbol{Xb} \right)}_{n \times 1}$$

This represents Q as a 1×1 matrix, and so we can think of Q as an ordinary number.

There are several ways to find the $\boldsymbol{b}$ that minimizes Q. The simple solution we'll show here (alas) requires knowing the answer and working backward.

Define the matrix $\underset{n \times n}{\boldsymbol{H}} = \underset{n \times p}{\boldsymbol{X}} \left(\underset{p \times n}{\boldsymbol{X}'} \underset{n \times p}{\boldsymbol{X}} \right)^{-1} \underset{p \times n}{\boldsymbol{X}'}$. We will call $\boldsymbol{H}$ as the "hat matrix," and it has some important uses. There are several technical comments about $\boldsymbol{H}$:

(1) Finding $\boldsymbol{H}$ requires the ability to get $\left(\underset{p \times n}{\boldsymbol{X}'} \underset{n \times p}{\boldsymbol{X}} \right)^{-1}$. This matrix inversion is possible if and only if $\boldsymbol{X}$ has full rank p. Things get very interesting when $\boldsymbol{X}$ *almost* has full rank p; that's a longer story for another time.

(2) The matrix $\boldsymbol{H}$ is *idempotent*. The defining condition for idempotence is this:

The matrix $\boldsymbol{C}$ is idempotent $\Leftrightarrow$ $\boldsymbol{C}\,\boldsymbol{C} = \boldsymbol{C}$.

Only square matrices can be idempotent.

Since $\boldsymbol{H}$ is square (It's $n \times n$.), it can be checked for idempotence. You will indeed find that $\boldsymbol{H}\,\boldsymbol{H} = \boldsymbol{H}$.

(3) The i^{th} diagonal entry, that in position (i, i), will be identified for later use as the i^{th} leverage value. The notation is usually h_i , but you'll also see h_{ii} .

Now write $\boldsymbol{Y}$ in the form $\boldsymbol{H}\,\boldsymbol{Y} + (\mathbf{I} - \boldsymbol{H})\,\boldsymbol{Y}$.

Now let's develop Q. This will require using the fact that $\boldsymbol{H}$ is symmetric, meaning $\boldsymbol{H}' = \boldsymbol{H}$. This will also require using the transpose of a matrix product. Specifically, the property will be $\left(\boldsymbol{X}\,\boldsymbol{b}\right)' = \boldsymbol{b}'\,\boldsymbol{X}'$.

$$Q = \left(\boldsymbol{Y} - \boldsymbol{Xb}\right)'\left(\boldsymbol{Y} - \boldsymbol{Xb}\right)$$

$$= \left(\left\{\boldsymbol{H}\,\boldsymbol{Y} + \left(\mathbf{I} - \boldsymbol{H}\right)\boldsymbol{Y}\right\} - \boldsymbol{Xb}\right)'\left(\left\{\boldsymbol{H}\,\boldsymbol{Y} + \left(\mathbf{I} - \boldsymbol{H}\right)\boldsymbol{Y}\right\} - \boldsymbol{Xb}\right)$$

$$= \left(\left\{\boldsymbol{H}\,\boldsymbol{Y} - \boldsymbol{Xb}\right\} + \left(\mathbf{I} - \boldsymbol{H}\right)\boldsymbol{Y}\right)'\left(\left\{\boldsymbol{H}\,\boldsymbol{Y} - \boldsymbol{Xb}\right\} + \left(\mathbf{I} - \boldsymbol{H}\right)\boldsymbol{Y}\right)$$

$$\begin{aligned} &= \left\{\boldsymbol{H}\,\boldsymbol{Y} - \boldsymbol{Xb}\right\}'\left\{\boldsymbol{H}\,\boldsymbol{Y} - \boldsymbol{Xb}\right\} \\ &+ \left\{\boldsymbol{H}\,\boldsymbol{Y} - \boldsymbol{Xb}\right\}'\left(\mathbf{I} - \boldsymbol{H}\right)\boldsymbol{Y} \\ &+ \left(\left(\mathbf{I} - \boldsymbol{H}\right)\boldsymbol{Y}\right)' \left\{\boldsymbol{H}\,\boldsymbol{Y} - \boldsymbol{Xb}\right\} \\ &+ \left(\left(\mathbf{I} - \boldsymbol{H}\right)\boldsymbol{Y}\right)' \left(\mathbf{I} - \boldsymbol{H}\right)\boldsymbol{Y} \end{aligned}$$

The second and third summands above are zero, as a consequence of $\left(\mathbf{I} - \boldsymbol{H}\right)\boldsymbol{X} = \boldsymbol{X} - \boldsymbol{H}\,\boldsymbol{X} = \boldsymbol{X} - \boldsymbol{X}\left(\boldsymbol{X}'\,\boldsymbol{X}\right)^{-1}\boldsymbol{X}'\,\boldsymbol{X} = \boldsymbol{X} - \boldsymbol{X} = \mathbf{0}$.

$$= \left\{\boldsymbol{H}\,\boldsymbol{Y} - \boldsymbol{Xb}\right\}'\left\{\boldsymbol{H}\,\boldsymbol{Y} - \boldsymbol{Xb}\right\} + \left(\left(\mathbf{I} - \boldsymbol{H}\right)\boldsymbol{Y}\right)' \left(\mathbf{I} - \boldsymbol{H}\right)\boldsymbol{Y}$$

If this is to be minimized over choices of $\boldsymbol{b}$, then the minimization can only be done with regard to the first summand $\left\{\boldsymbol{H}\,\boldsymbol{Y} - \boldsymbol{Xb}\right\}'\left\{\boldsymbol{H}\,\boldsymbol{Y} - \boldsymbol{Xb}\right\}$. It is possible to make the vector $\boldsymbol{H}\,\boldsymbol{Y} - \boldsymbol{Xb}$ equal to $\mathbf{0}$ by selecting $\boldsymbol{b} = \left(\boldsymbol{X}'\,\boldsymbol{X}\right)^{-1}\boldsymbol{X}'\boldsymbol{Y}$. This is very easy to see, as $\boldsymbol{H} = \boldsymbol{X}\left(\boldsymbol{X}'\,\boldsymbol{X}\right)^{-1}\boldsymbol{X}'$.

This $\boldsymbol{b} = \left(\boldsymbol{X}'\,\boldsymbol{X}\right)^{-1}\boldsymbol{X}'\boldsymbol{Y}$ is known as the *least squares* estimate of $\boldsymbol{\beta}$.

For the simple linear regression case $K = 1$, the estimate $\boldsymbol{b} = \begin{pmatrix} b_0 \\ b_1 \end{pmatrix}$ and be found with relative ease. The slope estimate is $b_1 = \dfrac{S_{xy}}{S_{xx}}$, where $S_{xy} = \sum_{i=1}^{n} (x_i - \overline{x})(Y_i - \overline{Y}) = \sum_{i=1}^{n} x_i Y_i - n\,\overline{x}\,\overline{Y}$ and where $S_{xx} = \sum_{i=1}^{n} (x_i - \overline{x})^2 = \sum_{i=1}^{n} x_i^2 - n\,(\overline{x})^2$.

For the multiple regression case $K \geq 2$, the calculation involves the inversion of the $p \times p$ matrix $\boldsymbol{X'X}$. This task is best left to computer software.

There is a computational trick, called "mean-centering," that converts the problem to a simpler one of inverting a $K \times K$ matrix.

The matrix notation will allow the proof of two very helpful facts:

* E $\boldsymbol{b} = \boldsymbol{\beta}$. This means that $\boldsymbol{b}$ is an unbiased estimate of $\boldsymbol{\beta}$. This is a good thing, but there are circumstances in which biased estimates will work a little bit better.

* Var $\boldsymbol{b} = \sigma^2 \left(\boldsymbol{X'}\,\boldsymbol{X}\right)^{-1}$. This identifies the variances and covariances of the estimated coefficients. It's critical to note that the separate entries of $\boldsymbol{b}$ are not statistically independent.