(Just about) Everything You Need to Know About Matrix Algebra[1] to Understand OLS in Matrix Form

Part I

Patrick Egan (patrick.egan@nyu.edu)

revised and corrected version, November 30, 2009

This handout is intended as an introduction to matrix algebra on a "need to know" basis for those seeking to analyze the properties of Ordinary Least Squares (and other) estimators in matrix form. It was composed by liberally drawing from many sources, which I have attempted to appropriately credit throughout. Suggestions and comments for improvements are welcome to patrick.egan@nyu.edu.

0. Matrix Algebra—Why do we care?

In the context of estimation, we need matrix algebra to move from *bivariate* to *multivariate* analysis. When we estimate the model

$$y_i = \beta x_i + u_i$$

(assume all variables are mean-deviated, and thus no constant is estimated)—we write the formula for the least squares estimator β_{OLS} as:

$$\beta_{OLS} = \frac{\sum_{i=1}^{n} x_i y_i}{\sum_{i=1}^{n} x_i^{\,2}} = \frac{\text{cov}(x,y)}{\text{var}(x)}$$

with var(x) assumed to be non-zero. And as you know (or will soon learn), the expected value of this estimator is calculated as follows:

$$E(\beta_{OLS}) = E\left(\frac{\sum_{i=1}^{n} x_i y_i}{\sum_{i=1}^{n} x_i^{\,2}}\right) = E\left[\frac{\sum_{i=1}^{n} x_i(\beta x_i + u_i)}{\sum_{i=1}^{n} x_i^{\,2}}\right]$$

$$= E\left(\frac{\beta \sum_{i=1}^{n} x_i^{\,2}}{\sum_{i=1}^{n} x_i^{\,2}}\right) + E\left(\frac{\sum_{i=1}^{n} x_i u_i}{\sum_{i=1}^{n} x_i^{\,2}}\right)$$

[1] Also known as linear algebra.

$$E(\beta_{OLS}) = \beta + E\left(\frac{\sum_{i=1}^{n} x_i u_i}{\sum_{i=1}^{n} x_i^2} \right) \qquad (1)$$

At this point, we rely on the assumptions that (1) the x_i are fixed and (2) that $E(u_i)$ is equal to zero for all i to conclude that β_{OLS} is an unbiased estimator of β. We then look to equation (1) above to determine the extent of the bias of β_{OLS} when the data generating process fails to meet these assumptions. We can perform other derivations to characterize the sampling distribution of β_{OLS} and thus analyze the estimator's efficiency, and see how departures from the OLS assumptions—such as heteroskedasticity—affect the estimator's efficiency and the expected value of our estimate of the estimator's variance, $\text{var}(\beta_{OLS})$.

All of this is well and good, but the vast amount of quantitative work in the social scientists is undertaken with multivariate analysis, in which we are often analyzing models that look like:

$$y_i = \sum_{k=1}^{K} \beta_k x_{ik} + u_i \,,$$

…where K is the number of independent variables in the model (again, assume that all the x's and y are mean-deviated). Performing the sort of analysis we did above in the multivariate context quickly becomes so complicated that it is virtually useless. For example, consider just one step up in complexity: the trivariate model $y_i = \beta_1 x_{i1} + \beta_2 x_{i2} + u_i$, which we estimate with the equation $y = \beta_1 x_{i1} + \beta_2 x_{i2}$. The classic text by Hanushek and Jackson[2] (pp. 28-35) derives the following estimators for the two β's (I've substituted my notation for theirs):

$$\beta_{1OLS} = \frac{[\text{var}(x_2)\,\text{cov}(y, x_1) - \text{cov}(x_1, x_2)\,\text{cov}(y, x_2)]}{[\text{var}(x_1)\,\text{var}(x_2) - (\text{cov}(x_1, x_2))^2]}$$

$$\beta_{2OLS} = \frac{[\text{var}(x_1)\,\text{cov}(y, x_2) - \text{cov}(x_1, x_2)\,\text{cov}(y, x_1)]}{[\text{var}(x_1)\,\text{var}(x_2) - (\text{cov}(x_1, x_2))^2]}$$

The formulas for the variances of the two estimators in the trivariate case are similarly complicated (see Hanushek and Jackson pp. 52-54 for details). The point here is that *scalar*

[2] Eric A. Hanushek and John E. Jackson, *Statistical Methods for Social Scientists.* Academic Press, 1977.

algebra—that is, the way we've learned to write mathematical expressions since kindergarten—is a very inefficient way to describe multivariate relationships. By contrast, *matrix algebra* allows us to keep track of the relationships among lots of variables with relative ease. Look again at the formulas above: you should be able to see patterns emerge. For example, the denominators of both fractions seem to be capturing a certain property about the x's in the model: specifically, the denominator is a function of how much the x's vary individually (the product var(x_1) var(x_2)) minus how much they covary (cov(x_1 x_2)). Matrix algebra allows us to write these sorts of relationships in a succinct, easily manipulable form.

Typically, matrix algebra is presented with numbers, which is intuitive but not particularly useful for statistical analysis. I'll focus instead on presenting matrix algebra with symbols that hopefully get at the core underlying concepts. One final note: this is hard. It requires a bit of a paradigm shift in how you think about math. Give yourself time to get used to matrix algebra, and keep revisiting it over and over again.

1. The three components of matrix algebra: scalars, vectors and matrices

In matrix algebra, we generally work with three different kinds of components:

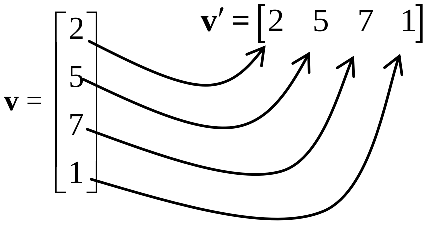
- **Scalars** are the numbers and symbols you're used to from your years of high school mathematics. They include things like the number 34, a, (a^2/8), and $5x-7y+\sqrt{3z}$. In matrix algebra, scalars represent exactly the same things that they do in scalar algebra.

- **Vectors** are columns of scalars. Here's a vector, **v**:

$$\mathbf{v} = \begin{bmatrix} 2 \\ 5 \\ 7 \\ 1 \end{bmatrix}. \qquad \text{More generally we can write: } \mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

The vector **v** has four rows and thus has the dimension (4 x 1)—for four rows by one column. The vector **x** has n rows and thus dimension (n x 1). The entries in a vector or matrix are called **elements**.

We can take a column vector (say, **v** from above) and turn it into a row vector by following a simple rule: the element in each row of the column vector becomes the element in the corresponding column of the new row vector. The new row vector is called the **transpose** of the original vector. Transposed vectors are usually denoted with the "prime" mark (e.g., $\boldsymbol{v}'$), and (less often) with the superscript *T* (e.g., $\boldsymbol{v}^{\boldsymbol{T}}$). In speech, we say "v-prime" or "v-transpose." So **v** becomes **v′** like this:


$\mathbf{v} = \begin{bmatrix} 2 \\ 5 \\ 7 \\ 1 \end{bmatrix}$
$\mathbf{v}' = \begin{bmatrix} 2 & 5 & 7 & 1 \end{bmatrix}$

and similarly, $\mathbf{x}' = \begin{bmatrix} x_1 & x_2 & \cdots & x_n \end{bmatrix}$.

The row-vector $\mathbf{v}'$ is (1 x 4), while $\mathbf{x}'$ is (1 x *n*). Finally, when we take the transpose of a transposed vector, the columns become the rows again, and thus we get the original vector back: (**x**′)′ = **x.**

- **Matrices** are composed of a series of vectors placed next to one another. Here is a matrix, **M**:

$$\mathbf{M} = \begin{bmatrix} 2 & 12 & 9 & 5 \\ 5 & -5 & 11 & 0 \\ 7 & 6 & 2 & -7 \\ 1 & 9 & 4 & 8 \end{bmatrix}.$$

The matrix **M** has four rows and four columns and thus has dimension (4 x 4). Matrices like **M** with the same number of rows and columns are **square** matrices. Notice that the vector **v** from above is the first column of this matrix.

More generally, we can write:

$$\mathbf{X} = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1c} \\ x_{21} & x_{22} & \cdots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ x_{r1} & \cdots & \cdots & x_{rc} \end{bmatrix}.$$

Notice the notation for each element in the matrix **X**: each has two subscripts. The first subscript is the element's row. The second is the element's column. **X** is a matrix of dimension (*r* x *c*).

Just as with vectors, we can take the **transpose** of a matrix by switching the rows and the columns. Try this for yourself: write a matrix on an index card, and then hold it between your thumb (on the lower right corner) and forefinger (on the upper left corner). Now flip the card over, keeping your fingers steady and letting the card pivot between your fingertips. Finally, hold up the flipped-over card to a lamp so you can see through it. You're now looking at the transpose of the original matrix.

The transpose of a matrix **X** turns the rows of **X** into the columns of the transposed matrix, which we write as **X′**. So:

$$\mathbf{M}' = \begin{bmatrix} 2 & 5 & 7 & 1 \\ 12 & -5 & 6 & 9 \\ 9 & 11 & 2 & 4 \\ 5 & 9 & -7 & 8 \end{bmatrix} \quad \text{and} \quad \mathbf{X}' = \begin{bmatrix} x_{11} & x_{21} & \cdots & x_{r1} \\ x_{12} & x_{22} & \cdots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ x_{1c} & \cdots & \cdots & x_{rc} \end{bmatrix}.$$

Notice that the transposed vector **v′** from above is now the first row of **M′**. Notice also that two elements—x_{11} and x_{rc}—are located in the same place for any matrix **X** and its transpose **X′**. Also, notice that by taking the transpose of a transposed matrix, we recover the original matrix: $(\mathbf{X}')' = \mathbf{X}$.

Notation

In print, vectors are usually represented by boldface lowercase characters (e.g. **v**, **β**), matrices with boldface uppercase characters (e.g. **M**, **Σ**). In writing, practices often differ from person to person. I tend to write vectors with a tilde underneath (e.g. $\underset{\sim}{v}$) and matrices in capital letters (e.g. M) to distinguish them from scalars, which I tend to write in lowercase.

2. Adding, subtracting and multiplying

Just like in scalar algebra, in matrix algebra we seek to combine scalars, vectors and matrices with each other through the basic operations of addition, subtraction and multiplication.[3] But unlike in scalar algebra, not all vectors and matrices may be combined with one another: they must be *conformable* in order to do so.

Addition and subtraction:

- **Conformability.** To perform addition or subtraction with two vectors or matrices, they must be of the same dimension. Thus we can never add a vector (which by definition is either of dimension *r* x 1 or 1 x *c*) to a matrix (which by definition is *r* x *c*). Similarly, we can never add a scalar to either a vector or a matrix. We can, however, add two column vectors with the same number of elements—or two matrices with the same number or rows and columns.

- **How to do it:** Easy as pie. Add the corresponding elements of each vector/matrix to produce the corresponding element in the sum. For example, here we add two (4 x 1) vectors:

$$\begin{bmatrix} 2 \\ 5 \\ 7 \\ 1 \end{bmatrix} + \begin{bmatrix} 9 \\ -5 \\ 2 \\ 12 \end{bmatrix} = \begin{bmatrix} 11 \\ 0 \\ 9 \\ 13 \end{bmatrix}$$

More generally, we write:

$$\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} + \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} (x_1 + y_1) \\ (x_2 + y_2) \\ \vdots \\ (x_n + y_n) \end{bmatrix}.$$

Here, we add two (3 x 3) matrices:

$$\begin{bmatrix} 1 & -1 & 5 \\ 2 & 0 & 10 \\ 3 & 1 & 15 \end{bmatrix} + \begin{bmatrix} 2 & -5 & 3 \\ 4 & -12 & 6 \\ 6 & -25 & 9 \end{bmatrix} = \begin{bmatrix} 3 & -6 & 8 \\ 6 & -12 & 16 \\ 9 & -24 & 24 \end{bmatrix}.$$

More generally, matrix addition works like this:

[3] We'll forego talking about matrix "division" right now. More later.

$$\mathbf{X}=\begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1c} \\ x_{21} & x_{22} & \cdots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ x_{r1} & \cdots & \cdots & x_{rc} \end{bmatrix};\ \mathbf{Y}=\begin{bmatrix} y_{11} & y_{12} & \cdots & y_{1c} \\ y_{21} & y_{22} & \cdots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ y_{r1} & \cdots & \cdots & y_{rc} \end{bmatrix};$$

$$\mathbf{X}+\mathbf{Y}=\begin{bmatrix} (x_{11}+y_{11}) & (x_{12}+y_{12}) & \cdots & (x_{1c}+y_{1c}) \\ (x_{21}+y_{21}) & (x_{22}+y_{22}) & \cdots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ (x_{r1}+y_{r1}) & \cdots & \cdots & (x_{rc}+y_{rc}) \end{bmatrix}$$

Multiplication:

- **Conformability.** Before talking about conformability for multiplication, we first note that in matrix algebra, *multiplication among vectors and matrices is not commutative*: that is, **XY** is not necessarily equal to **YX**. Therefore, *order matters* when performing multiplication with vectors and matrices.

  For (vectors/matrices) to be conformable for multiplication, the *first* vector or matrix must have the same number of *columns* as the *second* vector or matrix has *rows*. You can multiply a (5 x 12) matrix by a (12 x 35) matrix, but not the other way around. You can multiply a (1 x 4) row vector by a (4 x 6) matrix, but not vice-versa. And you can multiply a (8 x 1) column vector by a (1 x 19) row vector, but not vice-versa. As we'll see shortly, the product yielded by matrix multiplication always has the same number of *rows* as the first vector or matrix and the same number of *columns* as the second.

- **How to do it.** Matrix multiplication consists of multiplying the rows of the first vector or matrix by the columns of the second. The simplest case of multiplication occurs when you are multiplying a row vector of dimension (1 x n) (let's call this **x′**) times a column vector of dimension (n x 1) (which we'll call **y**). Multiply the first element of the row vector (x_1) times the first element of the column vector (y_1) to get the product (x_1y_1). Now do the same with the second element of the row vector (x_2) and the second element of the column vector (y_2). Repeat until you've used all the elements of both vectors. Your last product will be (x_ny_n). Finally, sum up all of these products. This sum, $\sum_{i=1}^{n} x_i y_i$, is a scalar and is the vector product **x′y**.

$$\mathbf{x'y} = \underset{(1 \text{ x } n)}{\begin{bmatrix} x_1 & x_2 & \dots & x_n \end{bmatrix}} \underset{(n \text{ x } 1)}{\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}} = \underset{(1 \text{ x } 1)}{x_1y_1 + x_2y_2 + \dots + x_ny_n = \sum_{i=1}^{n} x_iy_i} \, .$$

To make this more concrete, consider the following numerical example:

$$\mathbf{x}' = \underset{(1 \text{ x } 4)}{\begin{bmatrix} 4 & -30 & 8 & 5 \end{bmatrix}} \quad \mathbf{y} = \underset{(4 \text{ x } 1)}{\begin{bmatrix} -1 \\ 0 \\ 4 \\ 8 \end{bmatrix}} ; \quad \underset{(1 \text{ x } 1)}{\mathbf{x'y} = (4)(-1) + (-30)(0) + (8)(4) + (5)(8) = 76.}$$

The steps we used to calculate **x′y** are the building blocks for all matrix multiplication, which consists simply of repeating these steps for each of the rows of the first vector or matrix and each of the columns of the second. Consider one move up in complexity: the case of multiplying a matrix (which we’ll call **X**) times a column vector (**y**). Conformability requires that **X** has as many columns as **y** has rows, so let’s say that **X** is of dimension (*r* x *n*) and **y** is of dimension (*n* x 1). Since **X** is simply many row vectors stacked on top of one another, we repeat the steps performed above in the **x′y** case for each row of **X**. We then stack the scalars produced by each repetition in a new column vector, **Xy**.

$$\underset{(r \text{ x } n)}{\begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ x_{r1} & \cdots & \cdots & x_{rn} \end{bmatrix}} \times \underset{(n \text{ x } 1)}{\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}} = \underset{(r \text{ x } 1)}{\begin{bmatrix} (\mathbf{Xy})_1 \\ \\ \\ \\ \end{bmatrix}} \longleftarrow \quad (\mathbf{Xy})_1 = x_{11}y_1 + x_{12}y_2 + \dots + x_{1n}y_n$$

$$\underset{(r \text{ x } n)}{\begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ x_{r1} & \cdots & \cdots & x_{rn} \end{bmatrix}} \times \underset{(n \text{ x } 1)}{\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}} = \underset{(r \text{ x } 1)}{\begin{bmatrix} (\mathbf{Xy})_1 \\ (\mathbf{Xy})_2 \\ \\ \\ \end{bmatrix}} \longleftarrow \quad (\mathbf{Xy})_2 = x_{21}y_1 + x_{22}y_2 + \dots + x_{2n}y_n$$

… (repeat *r* times) …

$$\underset{(r \text{ x } n)}{\begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ x_{r1} & \cdots & \cdots & x_{rn} \end{bmatrix}} \times \underset{(n \text{ x } 1)}{\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}} = \underset{(r \text{ x } 1)}{\begin{bmatrix} (\mathbf{Xy})_1 \\ (\mathbf{Xy})_2 \\ \vdots \\ (\mathbf{Xy})_r \end{bmatrix}} = \mathbf{Xy}. \qquad \longleftarrow \quad (\mathbf{Xy})_r = x_{r1}y_1 + x_{r2}y_2 + ... + x_{rn}y_n$$

Here’s a numerical example:

$$\underset{(2 \text{ x } 3)}{\begin{bmatrix} 1 & 3 & 5 \\ 2 & 4 & 6 \end{bmatrix}} \times \underset{(3 \text{ x } 1)}{\begin{bmatrix} 0 \\ 5 \\ 10 \end{bmatrix}} = \begin{bmatrix} (1 \text{ x } 0) + (3 \text{ x } 5) + (5 \text{ x } 10) \\ (2 \text{ x } 0) + (4 \text{ x } 5) + (6 \text{ x } 10) \end{bmatrix} = \underset{(2 \text{ x } 1)}{\begin{bmatrix} 65 \\ 80 \end{bmatrix}}$$

We use a similar strategy to calculate the product of a (1 x *n*) row vector (**x′**) and an (*n* x *c*) matrix (**Y**). In this case, **x′** has one row that we multiply by each column in **Y**. We multiply and sum each element as before, placing them left-to-right in the row vector that is **x′Y**, which has dimensions (1 x *c*).

(do this *c* times)

$$\underset{(1 \text{ x n})}{\begin{bmatrix} x_1 & x_2 & ... & x_n \end{bmatrix}} \quad \underset{(\text{n x c})}{\begin{bmatrix} y_{11} & y_{12} & \cdots & y_{1c} \\ y_{21} & y_{22} & \cdots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ y_{n1} & \cdots & \cdots & y_{nc} \end{bmatrix}} = \underset{(1 \text{ x c})}{\begin{bmatrix} \sum_{i=1}^{n} x_1 y_{i1} & \sum_{i=1}^{n} x_2 y_{i2} & ... & \sum_{i=1}^{n} x_n y_{in} \end{bmatrix}}$$

Here’s another numerical example:

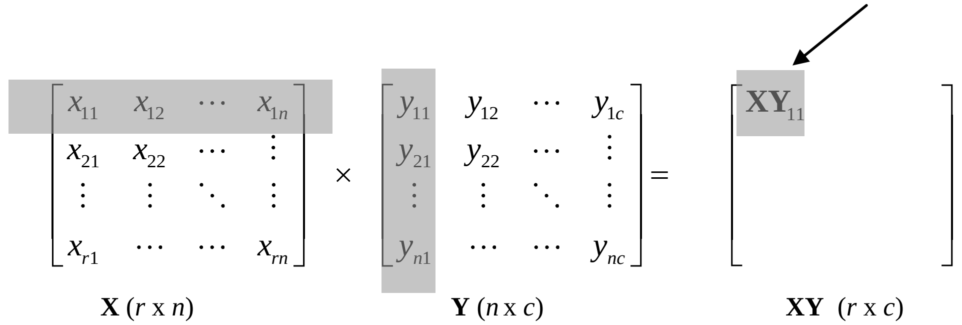
$$\begin{bmatrix}1 & 2 & 3 & 4\end{bmatrix}\begin{bmatrix}5 & 3\\ 0 & -3\\ -5 & 6\\ 10 & 9\end{bmatrix} = \begin{bmatrix}(1\text{ x }5)+(2\text{ x }0)+(3\text{ x }-5)+(4\text{ x }10) & (1\text{ x }3)+(2\text{ x }-3)+(3\text{ x }6)+(4\text{ x }9)\end{bmatrix}$$

$$= \begin{bmatrix}30 & 51\end{bmatrix}$$

(1 x 4) (4 x 2) (1 x 2)

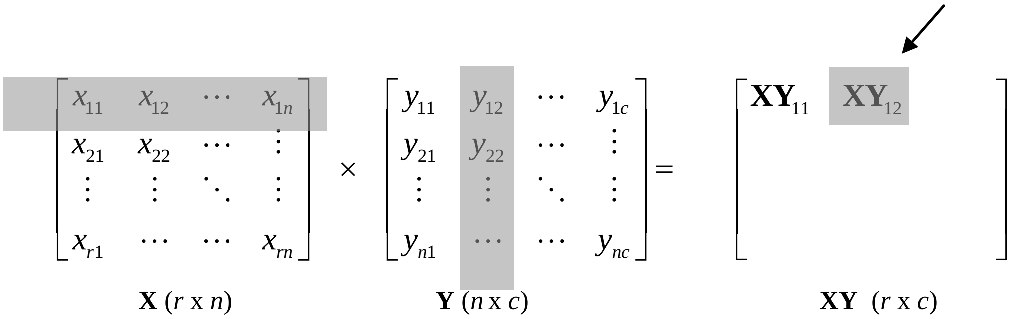
Finally, we repeat these steps even more when we multiply a matrix by a matrix. Call the first matrix **X** and label its dimensions (r x n). Similarly, call the second matrix **Y** and label its dimensions (n x c). As before, we multiply each row of **X** by each column of **Y**, summing the products. These products become the elements of the new matrix **XY**.

$$\mathbf{XY}_{11} = x_{11}y_{11} + x_{12}y_{21} + ... + x_{1n}y_{n1}$$



$\begin{bmatrix}x_{11} & x_{12} & \cdots & x_{1n}\\ x_{21} & x_{22} & \cdots & \vdots\\ \vdots & \vdots & \ddots & \vdots\\ x_{r1} & \cdots & \cdots & x_{rn}\end{bmatrix} \times \begin{bmatrix}y_{11} & y_{12} & \cdots & y_{1c}\\ y_{21} & y_{22} & \cdots & \vdots\\ \vdots & \vdots & \ddots & \vdots\\ y_{n1} & \cdots & \cdots & y_{nc}\end{bmatrix} = \begin{bmatrix}\mathbf{XY}_{11} & \\ & \end{bmatrix}$
X (r x n)
Y (n x c)
XY (r x c)

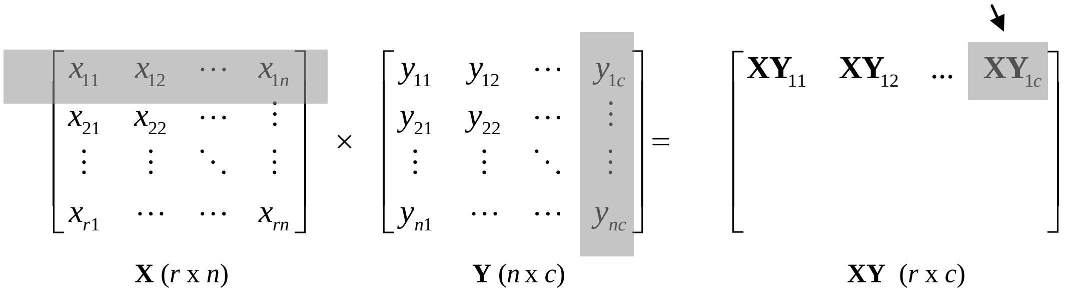
$$\mathbf{XY}_{12} = x_{11}y_{12} + x_{12}y_{22} + ... + x_{1n}y_{n2}$$



$\begin{bmatrix}x_{11} & x_{12} & \cdots & x_{1n}\\ x_{21} & x_{22} & \cdots & \vdots\\ \vdots & \vdots & \ddots & \vdots\\ x_{r1} & \cdots & \cdots & x_{rn}\end{bmatrix} \times \begin{bmatrix}y_{11} & y_{12} & \cdots & y_{1c}\\ y_{21} & y_{22} & \cdots & \vdots\\ \vdots & \vdots & \ddots & \vdots\\ y_{n1} & \cdots & \cdots & y_{nc}\end{bmatrix} = \begin{bmatrix}\mathbf{XY}_{11} & \mathbf{XY}_{12} & \\ & & \end{bmatrix}$
X (r x n)
Y (n x c)
XY (r x c)

…. repeat c times …..

$$\mathbf{XY}_{1c} = x_{11}y_{1c} + x_{12}y_{2c} + ... + x_{1n}y_{nc}$$



$\begin{bmatrix}x_{11} & x_{12} & \cdots & x_{1n}\\ x_{21} & x_{22} & \cdots & \vdots\\ \vdots & \vdots & \ddots & \vdots\\ x_{r1} & \cdots & \cdots & x_{rn}\end{bmatrix} \times \begin{bmatrix}y_{11} & y_{12} & \cdots & y_{1c}\\ y_{21} & y_{22} & \cdots & \vdots\\ \vdots & \vdots & \ddots & \vdots\\ y_{n1} & \cdots & \cdots & y_{nc}\end{bmatrix} = \begin{bmatrix}\mathbf{XY}_{11} & \mathbf{XY}_{12} & ... & \mathbf{XY}_{1c}\\ & & & \end{bmatrix}$
X (r x n)
Y (n x c)
XY (r x c)

Now, repeat these steps for each row of **X,** e.g.**:**

$$\mathbf{XY}_{21} = x_{21}y_{11} + x_{22}y_{21} + \ldots + x_{2n}y_{n1}$$

$$\begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ x_{r1} & \cdots & \cdots & x_{rn} \end{bmatrix} \times \begin{bmatrix} y_{11} & y_{12} & \cdots & y_{1c} \\ y_{21} & y_{22} & \cdots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ y_{n1} & \cdots & \cdots & y_{nc} \end{bmatrix} = \begin{bmatrix} \mathbf{XY}_{11} & \mathbf{XY}_{12} & \ldots & \mathbf{XY}_{1c} \\ \mathbf{XY}_{21} & & & \\ & & & \\ & & & \end{bmatrix}$$

X (*r* x *n*) **Y** (*n* x *c*) **XY** (*r* x *c*)

Do this *r* times to form the new matrix **XY:**

$$\begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ x_{r1} & \cdots & \cdots & x_{rn} \end{bmatrix} \times \begin{bmatrix} y_{11} & y_{12} & \cdots & y_{1c} \\ y_{21} & y_{22} & \cdots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ y_{n1} & \cdots & \cdots & y_{nc} \end{bmatrix} = \begin{bmatrix} \mathbf{XY}_{11} & \mathbf{XY}_{12} & \ldots & \mathbf{XY}_{1c} \\ \mathbf{XY}_{21} & \mathbf{XY}_{22} & \ldots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{XY}_{r1} & \ldots & \ldots & \mathbf{XY}_{rc} \end{bmatrix}$$

X (*r* x *n*) **Y** (*n* x *c*) **XY** (*r* x *c*)

We can thus write the element in the *r*'th row and *c*'th column of **XY** as:

$$\mathbf{XY}_{rc} = \sum_{i=1}^{n} x_{ri}y_{ic}$$

Here's a very simple numerical example:

$$\begin{bmatrix} 0 & 2 \\ 1 & 3 \end{bmatrix} \begin{bmatrix} 2 & 3 & 4 & 5 \\ 1 & 4 & 7 & 9 \end{bmatrix} = \begin{bmatrix} (0\times 2)+(2\times 1) & (0\times 3)+(2\times 4) & (0\times 4)+(2\times 7) & (0\times 5)+(2\times 9) \\ (1\times 2)+(3\times 1) & (1\times 3)+(3\times 4) & (1\times 4)+(3\times 7) & (1\times 5)+(3\times 9) \end{bmatrix}$$

$$= \begin{bmatrix} 2 & 8 & 14 & 18 \\ 5 & 15 & 25 & 32 \end{bmatrix}$$

(2 x 2) (2 x 4) (2 x 4)

Finally, there is one (sort of) special case to consider: that of a column vector (**x**) times a row vector (**y′**). The case is somewhat special because **x** has only one element per row and **y′** has only one element per column. Still, we do the same thing as we have all along: multiply the "rows" of **x** by the columns of **y′**. The result is a matrix, **M** = **xy′**, that has the same number of rows as **x** and the same number of columns as **y′**. The element in the *r*'th row, *c*'th column of **M** is the simple product $x_r y_c$.

(do this [*r* x *c*] times)

$$\underset{\text{(r x 1)}}{\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_r \end{bmatrix}} \underset{\text{(1 x c)}}{\begin{bmatrix} y_1 & y_2 & \cdots & y_c \end{bmatrix}} = \underset{\text{(r x c)}}{\begin{bmatrix} x_1 y_1 & x_1 y_2 & \cdots & x_1 y_c \\ x_2 y_1 & x_2 y_2 & \cdots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ x_r y_1 & \cdots & \cdots & x_r y_c \end{bmatrix}}$$

The following page sums up all of the cases we've discussed here.

I. row vector times column vector

(do this just once)

$$\begin{bmatrix} x_1 & x_2 & \dots & x_n \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} = \sum_{i=1}^{n} x_i y_i$$

(1 x n) (n x 1) (1 x 1)

II. matrix times column vector

(do this *r* times)

$$\begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ x_{r1} & \cdots & \cdots & x_{rn} \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{n} x_{1i} y_i \\ \sum_{i=1}^{n} x_{2i} y_i \\ \vdots \\ \sum_{i=1}^{n} x_{ri} y_i \end{bmatrix}$$

(r x n) (n x 1) (r x 1)

III. row vector times matrix

(do this *c* times)

$$\begin{bmatrix} x_1 & x_2 & \dots & x_n \end{bmatrix} \begin{bmatrix} y_{11} & y_{12} & \cdots & y_{1c} \\ y_{21} & y_{22} & \cdots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ y_{n1} & \cdots & \cdots & y_{nc} \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{n} x_1 y_{i1} & \sum_{i=1}^{n} x_2 y_{i2} & \dots & \sum_{i=1}^{n} x_n y_{in} \end{bmatrix}$$

(1 x n) (n x c) (1 x c)

IV. matrix times matrix

(do this [*r* x *c*] times)

$$\begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ x_{r1} & \cdots & \cdots & x_{rn} \end{bmatrix} \begin{bmatrix} y_{11} & y_{12} & \cdots & y_{1c} \\ y_{21} & y_{22} & \cdots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ y_{n1} & \cdots & \cdots & y_{nc} \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{n} x_{1i} y_{i1} & \sum_{i=1}^{n} x_{1i} y_{i2} & \cdots & \sum_{i=1}^{n} x_{1i} y_{ic} \\ \sum_{i=1}^{n} x_{2i} y_{i1} & \sum_{i=1}^{n} x_{2i} y_{i2} & \cdots & \sum_{i=1}^{n} x_{2i} y_{ic} \\ \vdots & \vdots & \ddots & \vdots \\ \sum_{i=1}^{n} x_{ri} y_{i1} & \sum_{i=1}^{n} x_{ri} y_{i2} & \cdots & \sum_{i=1}^{n} x_{ri} y_{ic} \end{bmatrix}$$

(r x n) (n x c) (r x c)

V. a (sort-of) special case: column vector times row vector

(do this [*r* x *c*] times)

$$\underset{(r \times 1)}{\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_r \end{bmatrix}} \quad \underset{(1 \times c)}{\begin{bmatrix} y_1 & y_2 & \cdots & y_c \end{bmatrix}} \quad = \quad \underset{(r \times c)}{\begin{bmatrix} x_1y_1 & x_1y_2 & \cdots & x_1y_c \\ x_2y_1 & x_2y_2 & \cdots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ x_ry_1 & \cdots & \cdots & x_ry_c \end{bmatrix}}$$

An observant reader will notice that we've covered every possible case of matrix multiplication: the rules of conformability rule out all other possibilities:

Category	Case	Product	Example		
vector and vector					
	column vector times row vector	matrix	**x**	**y′**	**M = xy′**
			(*n* x 1)	(1 x *n*)	(*n* x *n*)
	row vector times column vector	scalar	**x′**	**y**	**x′y**
			(1 x *n*)	(*n* x 1)	(1 x 1)
vector and matrix					
	matrix times column vector	column vector	**X**	**y**	**Xy**
			(*r* x *n*)	(*n* x 1)	(*r* x 1)
	row vector times matrix	row vector	**x′**	**Y**	**x′Y**
			(1 x *n*)	(*n* x *c*)	(1 x *c*)
matrix and matrix		matrix	**X**	**Y**	**XY**
			(*r* x *n*)	(*n* x *c*)	(*r* x *c*)

(Just about) Everything You'll Ever Need to Know About Matrix Algebra to Understand OLS in Matrix Form, Part II

revised November 30, 2009

by Patrick Egan - patrick.egan@nyu.edu

In this handout we move from the mechanics of matrix mathematics to getting just to the point where matrix algebra can help you understand regression analysis.

1. But first: multiplication with scalars

A final thing you need to know about matrix multiplication: multiplication with scalars follows its own (very simple) rule. When multiplying a matrix or vector by a scalar, multiply every element of the matrix or vector by the scalar. So if we have the vector **v:**

$\mathbf{v} = \begin{bmatrix} 2 \\ 5 \\ 7 \\ 1 \end{bmatrix}$ and multiply it by the scalar 5, we get: $5\mathbf{v} = \begin{bmatrix} 2\times5 \\ 5\times5 \\ 7\times5 \\ 1\times5 \end{bmatrix} = \begin{bmatrix} 10 \\ 25 \\ 35 \\ 5 \end{bmatrix}$.

More generally, the product of a scalar a and a matrix $\mathbf{X}$ with elements $\mathbf{X}_{rc}$ is a matrix $a\mathbf{X}$ with elements $a\mathbf{X}_{rc}$. Note that unlike the matrix multiplication discussed previously, multiplication with scalars *is* commutative: $a\mathbf{X} = \mathbf{X}a$. Because of this, we can "move scalars around" in a matrix expression:

$$a\mathbf{XY} = \mathbf{X}a\mathbf{Y} = \mathbf{XY}a$$

2. Matrix algebra facts

Here are a few facts that come in handy as you do matrix algebra:

- $\mathbf{A} + \mathbf{B} = \mathbf{B} + \mathbf{A}$
- $(\mathbf{A} + \mathbf{B})' = \mathbf{A}' + \mathbf{B}'$
- $(\mathbf{A} + \mathbf{B})\,\mathbf{C} = \mathbf{AC} + \mathbf{BC}$
- $(\mathbf{AB})' = \mathbf{B}'\mathbf{A}'$; $(\mathbf{ABC})' = \mathbf{C}'\mathbf{B}'\mathbf{A}'$; $(\mathbf{ABCD})' = \mathbf{D}'\mathbf{C}'\mathbf{B}'\mathbf{A}'$ …
- $(\mathbf{A} + \mathbf{B})'\,(\mathbf{A} + \mathbf{B}) = (\mathbf{A}' + \mathbf{B}')(\mathbf{A} + \mathbf{B}) = \mathbf{A}'\mathbf{A} + \mathbf{A}'\mathbf{B} + \mathbf{B}'\mathbf{A} + \mathbf{B}'\mathbf{B}$
- $\mathbf{a}'\mathbf{b} = \mathbf{b}'\mathbf{a}$. But remember that $\mathbf{ab}' \neq \mathbf{ba}'$.
- A final scalar fact: we can't really transpose a scalar. But we treat the transpose of a scalar as itself: $a = a'$. So if you see something that looks like $(a\mathbf{X})'$, treat it as $a\mathbf{X}'$.

3. Two special matrices

We're now ready to consider two special matrices.

Zero: simple and straightforward. The matrix $\mathbf{0}_{rxc}$ is an r x c matrix whose entries are all zero. Here's 0_{2x3}:

$$0_{2x3} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

You should probably easily be able to see that if **X** is a (r x c) matrix, then $\mathbf{0}_{rxn}$ x $\mathbf{X} = \mathbf{0}_{rxc}$ and $\mathbf{X}$ x $\mathbf{0}_{cxn} = \mathbf{0}_{rxn}$.

One: the loneliest number(s). The matrix counterpart to the scalar number 1 is a matrix that, when multiplied by a matrix **X**, returns an identical product matrix, **X**. Let's call this matrix **I** (for *identity*) and consider: what might such a matrix look like?

- First off, **I** would have to be conformable for multiplication with **X.** It would need to have the same number of columns as **X** has rows. So if **I** were (r x n) and thus **X** were (n x c), the matrix product **IX** would be (r x c).

- But for **IX** (r x c) and **X** (n x c) to be identical, they of course must be of the same dimension. Thus it must be that $r = n$, and therefore **I** must be a square matrix.

- What might the elements of **I** be? Let's say we wanted to produce an exact replica of the (2 x 3) matrix, **A**:

$$\mathbf{A} = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}$$

We know that **I** is (1) conformable with **A** (and thus has 2 columns) and (2) a square matrix (and thus has 2 rows). So we write:

$$\mathbf{I} = \begin{bmatrix} i_{11} & i_{12} \\ i_{21} & i_{22} \end{bmatrix}.$$

The goal is to create **I** such that **IA** = **A**. So we can write the equation

$$\begin{bmatrix} i_{11} & i_{12} \\ i_{21} & i_{22} \end{bmatrix} \times \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} \quad \text{or more generally,}$$

$$\begin{bmatrix} i_{11} & i_{12} \\ i_{21} & i_{22} \end{bmatrix} \times \begin{bmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} & \mathbf{A}_{13} \\ \mathbf{A}_{21} & \mathbf{A}_{22} & \mathbf{A}_{23} \end{bmatrix} = \begin{bmatrix} \mathbf{(IA)_{11}} & \mathbf{(IA)_{12}} & \mathbf{(IA)_{13}} \\ \mathbf{(IA)_{21}} & \mathbf{(IA)_{22}} & \mathbf{(IA)_{23}} \end{bmatrix}$$

Now let's consider the element $\mathbf{A}_{11} = 1$. We want the element $(\mathbf{IA})_{11}$ to be the same thing. We know from our matrix multiplication rules that $(\mathbf{IA})_{11} = (i_{11} \text{ x } \mathbf{A}_{11}) + (i_{12} \text{ x } \mathbf{A}_{21})$. We're not interested in $\mathbf{A}_{21}$ at the moment, so let's make i_{12} equal to zero. We want $(\mathbf{IA})_{11} = \mathbf{A}_{11}$, so let's make i_{11} equal to one:

$$\begin{bmatrix} 1 & 0 \\ i_{21} & i_{22} \end{bmatrix} \times \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} = \begin{bmatrix} 1 & & \\ & & \end{bmatrix}$$

Before we move on, notice that when we repeat the multiplication of the first row of **I** with the remaining columns of **A**, we get a similar—and pleasing—result: we replicate the elements of the first row of **A**.

$$\begin{bmatrix} 1 & 0 \\ i_{21} & i_{22} \end{bmatrix} \times \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} = \begin{bmatrix} 1 & 2 & 3 \\ & & \end{bmatrix}$$

So if a (1 0) first row of **I** replicated the first row of **A**, what do you think would replicate the *second* row of **A**? It must be a (0 1) row, which "ignores" the first row of **A** and returns the second row:

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \times \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}$$

The case just discussed generalizes to any matrix. The identity matrix **I** for a (r x c) matrix **X** such that **IX** = **X** is of dimension (r x r), and it has ones on its diagonal and zeroes off its diagonal. This matrix is written $\mathbf{I}_r$ or $\mathbf{I}_{rxr}$. So, e.g.,:

$$\mathbf{I}_{3x3} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

Furthermore, if **X** is (r x c), then $\mathbf{I}_r \text{ x } \mathbf{X} = \mathbf{X} = \mathbf{X} \text{ x } \mathbf{I}_c$.

Many authors write **i** to denote a *vector* that contains a column of ones. If we are working with a vector **i** of dimension (n x 1), then **ii′** is an n x n matrix whose every element is 1. (Note that **ii′** ≠ **I.)** What about **i′i**? What is that equal to? (Hint: it's not 1.)

- A helpful (if obvious) thing to know: **IA** = **AI = A**.

4. Bits and pieces[1]

Here are a few more things to know:

(0) A very useful vector is the vector **i**, which is simply a column of ones. **i** can be of any dimension. Here's what **i** looks like when it is of dimension (4 x 1):

$$\mathbf{i} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

(1) To create a column vector whose every element is some scalar, a, we write a**i**.

(2) To sum up a vector's elements, premultiply[2] it by the row-vector **i′**. So if:

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \text{ , then } \sum_{i=1}^{n} x_i = \mathbf{i}'\mathbf{x} \text{ . Remember that } \mathbf{i}'\mathbf{x} = \mathbf{x}'\mathbf{i}\,.$$

(3) To find the average value of a vector's elements, simply divide the sum of its elements by *n*: $\bar{x} = \frac{1}{n}\mathbf{i}'\mathbf{x} = \frac{1}{n}\mathbf{x}'\mathbf{i}$.

(4) To sum up the squares of a vector's elements; premultiply the vector by itself, transposed:

$$\sum_{i=1}^{n} x_i^{\,2} = \mathbf{x}'\mathbf{x}$$

Knowing these things we can now write a vector of of *n* observations of a mean-deviated variable, *x*, in matrix form. Begin by writing the mean-deviation as the difference between two vectors:

(5)
$$\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} - \begin{bmatrix} \bar{x} \\ \bar{x} \\ \vdots \\ \bar{x} \end{bmatrix} .$$

[1] This section draws very heavily from William Greene's *Econometric Analysis* (5th ed.) pp. 807-809.

[2] Did you notice I said *pre*multiply? Since matrix algebra multiplication is not commutative, we need to distinguish the order in which we multiply. So, e.g., in the matrix product AB, we are *pre*multiplying B by A, and *post*multiplying A by B.

Now we know a few things:

- We can write $\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$ as simply **x**.

- We can write the scalar $\bar{x}$ as $\frac{1}{n}\mathbf{i}'\mathbf{x}$ by (3) above.

- By (1) above, we know that if we want to create a column vector of identical scalars (e.g., a bunch of $\bar{x}$'s), we simply multiply the scalar by **i**.

So let's rewrite (5) as:

$$\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} - \begin{bmatrix} \bar{x} \\ \bar{x} \\ \vdots \\ \bar{x} \end{bmatrix} = \mathbf{x} - \mathbf{i}\left[\frac{1}{n}\mathbf{i}'\mathbf{x}\right] = \mathbf{x} - \frac{1}{n}\mathbf{i}\mathbf{i}'\mathbf{x}.$$

OK, now let's do a little matrix algebra:

$$\mathbf{x} - \frac{1}{n}\mathbf{i}\mathbf{i}'\mathbf{x} = \mathbf{I}\mathbf{x} - \frac{1}{n}\mathbf{i}\mathbf{i}'\mathbf{x} = \left[\mathbf{I} - \frac{1}{n}\mathbf{i}\mathbf{i}'\right]\mathbf{x} = \mathbf{M}^{\mathbf{0}}\mathbf{x}.$$

$\mathbf{M}^0 = \left[\mathbf{I} - \frac{1}{n}\mathbf{i}\mathbf{i}'\right]$ is a special (*n* x *n*) matrix often seen in derivations. It is very useful because when we premultiply any vector **x** by $\mathbf{M}^0$, we get back a vector whose elements are the mean-deviated values of the elements of **x**. $\mathbf{M}^0$ is an **idempotent** matrix, which means that it is its own square: $\mathbf{M}^0$ x $\mathbf{M}^0 = \mathbf{M}^0$. Most of the idempotent matrices (including, as you'll see, $\mathbf{M}^0$) are also symmetric, so $\mathbf{M}^0 = (\mathbf{M}^0)'$, and so $\mathbf{M}^0(\mathbf{M}^0)' = \mathbf{M}^0$ x $\mathbf{M}^0 = \mathbf{M}^0$.

To gain a deeper understanding, let's write out $\mathbf{M}^0$:

$$\mathbf{M}^0 = \left[\mathbf{I} - \frac{1}{n}\mathbf{ii}'\right]$$

$$\mathbf{M}^0 = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & & \vdots \\ \vdots & & \ddots & \vdots \\ 0 & \cdots & \cdots & 1 \end{bmatrix} - \frac{1}{n}\begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}\begin{bmatrix} 1 & 1 & 1 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & & \vdots \\ \vdots & & \ddots & \vdots \\ 0 & \cdots & \cdots & 1 \end{bmatrix} - \begin{bmatrix} 1/n & 1/n & \cdots & 1/n \\ 1/n & 1/n & & \vdots \\ & & \ddots & \vdots \\ 1/n & \cdots & \cdots & 1/n \end{bmatrix}$$

$$= \begin{bmatrix} 1-1/n & -1/n & \cdots & -1/n \\ -1/n & 1-1/n & & \vdots \\ \vdots & & \ddots & \vdots \\ -1/n & \cdots & \cdots & 1-1/n \end{bmatrix}$$

Do a little multiplication for yourself to see that $\mathbf{M}^0$ x $\mathbf{M}^0 = \mathbf{M}^0$.

Here's another reason why this matrix is so special. Let's say we'd like the sum of the squared deviations of the elements of **x** about their mean, $\bar{x}$. To find this, we take the vector of mean-deviated variables just discussed: $\mathbf{M^0x}$. From (4) above, we know that when we want the sum of the squares of the elements of any vector, we premultiply this vector by its transpose. So the sum of squared deviations can be written:

$$\sum_{i=1}^{n}(x_i - \bar{x})^2 = (\mathbf{M^0x})'(\mathbf{M^0x}) = \mathbf{x}'\left(\mathbf{M^0}\right)'\mathbf{M^0x}$$

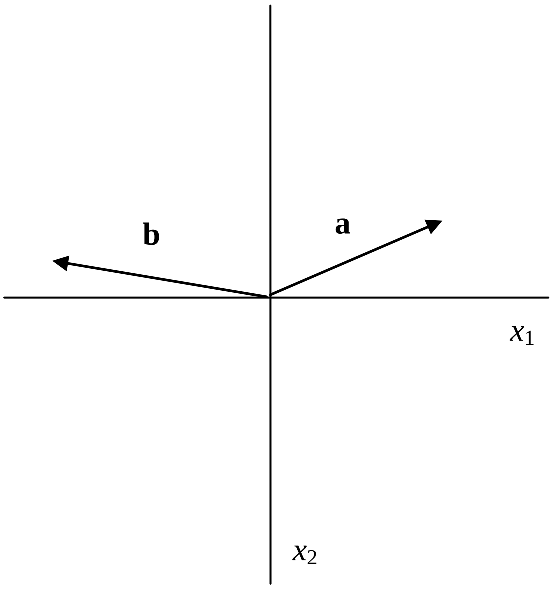
Recall that $(\mathbf{M}^0)' = \mathbf{M}^0$, and that $\mathbf{M}^0$ x $\mathbf{M}^0 = \mathbf{M}^0$. So we can write

$$\mathbf{x}'\left(\mathbf{M^0}\right)'\mathbf{M^0x} = \mathbf{x}'\mathbf{M^0x}.$$

4. Points, vectors, *n*-dimensional space, and even more about multiplication

As you probably already know, (n x 1) vectors can be used to represent points[3] in n-dimensional space. We are used to thinking of points in two dimensions on the x, y plane: for example, the point (3, 2) is located 3 units to the right and 2 units above the origin. We can represent this point with the vector $\begin{bmatrix} 3 \\ 2 \end{bmatrix}$, and convention has it that when we do so we usually talk about the x_1, x_2 plane instead of the x, y plane—don't stress out, they're the same thing.

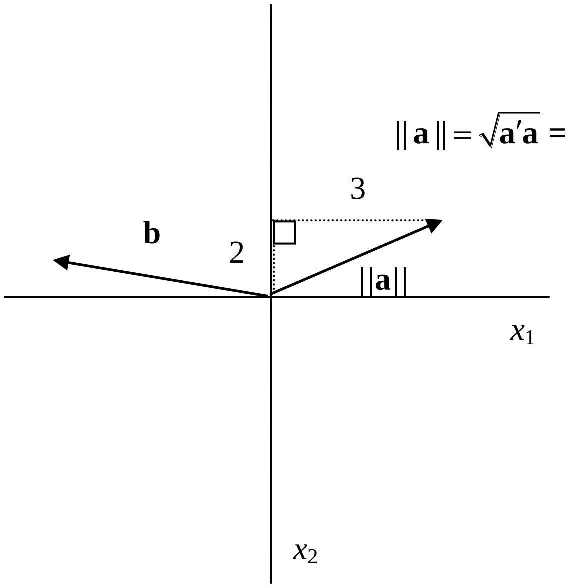
Let's discuss two column vectors, **a** and **b**, of dimension 2 x 1, where **a** = $\begin{bmatrix} 3 \\ 2 \end{bmatrix}$ and **b** = $\begin{bmatrix} -4 \\ 1 \end{bmatrix}$. It is conventional to draw these vectors as arrows pointing out of the origin that terminate at the points (3, 2) and (-4, 1) respectively:



Now that we've drawn these vectors on a graph, we might be curious of their properties. For example, what is the angle between them? And what is the distance between the terminus of **a** and the origin? One of the very cool things about matrix algebra is that it gives us useful tools to answer these questions.

Length of a vector. Let's answer the question of distance first. We refer to the distance between the terminus of a vector and the origin as the *length*—or *norm*—of the vector. A natural way to measure the length is the good old Pythagorean theorem: we draw a right triangle as so:

[3] Note that "vector" and "point" are two terms that basically describe the same concept: a location in n-dimensional space.



$$\| \mathbf{a} \| = \sqrt{\mathbf{a}'\mathbf{a}} = \sqrt{\begin{bmatrix} 3 & 2 \end{bmatrix} \begin{bmatrix} 3 \\ 2 \end{bmatrix}} = \sqrt{9+4} = \sqrt{13}$$

But instead of using scalar algebra ($a^2 + b^2 = c^2$, etc.) we've become terribly important adults and so we now use matrix algebra. We note that we can write the equivalent of the sum $a^2 + b^2$ as $\mathbf{a}'\mathbf{a}$, and so we say that the *length of the vector* $\mathbf{a}$ = $\sqrt{\mathbf{a}'\mathbf{a}}$, or in matrix notation $\| \mathbf{a} \|$.[4] As calculated above, $\| \mathbf{a} \| = \sqrt{13}$. Try finding $\| \mathbf{b} \|$ on your own.[5]

Inner product, a.k.a. dot product. The scalar $\mathbf{a}'\mathbf{a}$ is specific instance of a general notion called the **inner product**, or the **dot product**, of two vectors. The inner product of the vectors **a** and **b** is written $\mathbf{a} \cdot \mathbf{b}$ and is calculated $\mathbf{a}'\mathbf{b}$ (which as you'll remember from above is equal to $\mathbf{b}'\mathbf{a}$). The dot product of **a** and **b** above is $\begin{bmatrix} 3 & 2 \end{bmatrix} \begin{bmatrix} -4 \\ 1 \end{bmatrix} = -12 + 2 = 10.$

The inner product is contrasted with the **outer product**, which is $\mathbf{a}\mathbf{b}'$.

As it happens, the dot product $\mathbf{a} \cdot \mathbf{b}$ always equals $\cos(\theta) \| \mathbf{a} \| \| \mathbf{b} \|$, where θ is the angle between the two vectors. The cosine of a 90-degree angle equals zero, and so when **a** and **b** are at right-angles to one another, their dot product equals zero. When this is the case, we say that **a** and **b** are **orthogonal** to each other and write $\mathbf{a} \perp \mathbf{b}$.

Moving to n-dimensions. All of this is very nice, you may be saying, but why go to all the trouble? Scalar algebra seemed to do just fine for these tasks. Yes it does—in two dimensions. All of the matrix algebra above applies in *n*-dimensional space with (*n* x 1) dimensional vectors. Matrix algebra allows us to measure length, distance, and angles in the *n*-dimensional context.

So much more. There is a wealth of basic knowledge about the geometry of vectors and matrices that I am not covering here. See Greene (pp. 809-819) for a nice discussion.

[4] This is annoying, but the length is also sometimes written $| \mathbf{a} |$. Sorry about that.

[5] The distance between two points measured this way is often called *Euclidean distance*. And this definition of the length of a vector is often called either the *Euclidean norm* or the *Pythagorean norm*.

5. Linear dependence and the rank of a matrix.

Consider a matrix **M** of dimension (*r* x *c*). As noted earlier, **M** can be considered a collection of *c* column vectors laid side-by-side, which we'll call $\mathbf{m}_1$, $\mathbf{m}_2$,…$\mathbf{m}_c$. Let's now think about these column vectors (which are each of dimension *r* x 1) as vectors/arrows/points in *r*-dimensional space. These vectors are said to be **linearly independent** if and only if the only solution to the equation $a_1\mathbf{m}_1 + a_2\mathbf{m}_2 + ... + a_c\mathbf{m}_c = 0$ is the trivial solution that $a_1 = a_2 = ... = a_c = 0$. If a non-zero solution exists, the set of vectors are **linearly dependent**.

A simple example of linear dependence is the set of two vectors $\begin{bmatrix} 2 \\ 4 \end{bmatrix}$ and $\begin{bmatrix} 4 \\ 8 \end{bmatrix}$. Draw them on the x_1, x_2 plane and you'll see that they are right on top of one another—they are literally **collinear**. Two things to say about linear dependence: 1. A system of vectors can be linearly dependent even if no two pairs in the system are linearly dependent. 2. Determining linear dependence by hand is actually more difficult than it looks, and not worth it to learn in my opinion. In other words, don't try this at home: that's why we have computers.

The **rank** of a matrix **M** is the number of columns in the largest possible set of linearly independent columns of **M**. The matrix **M** can be said to be of **full rank** if its rank equals its number of columns. A matrix that has full rank is **nonsingular**.

6. The inverse of a matrix.

Why do we care about rank and nonsingularity? It has to do with the notion of matrix "division," which until now I've avoided discussing. The analogy in matrix algebra to division in scalar algebra is the notion of a reciprocal. In scalar algebra, multiplying a number *x* by its reciprocal (1/*x*) is the same as dividing *x* by *x*. In both cases, we get the product (or quotient, respectively) 1. You'll remember that we often denote the reciprocal of *x* as x^{-1}.

In matrix algebra, we can't "divide a matrix" by anything. But we can (pre- or post-) multiply a matrix **M** by its **inverse** (which we denote as $\mathbf{M}^{-1}$) to return the identity matrix **I**. That's the definition of inverse: the inverse of an matrix **M** is a matrix $\mathbf{M}^{-1}$ such that $\mathbf{MM}^{-1} = \mathbf{M}^{-1}\mathbf{M} = \mathbf{I}.$ By inspection, you can see that only square matrices have inverses, and that the dimensions of **M**, $\mathbf{M}^{-1}$, and **I** must all be the same.

This is all great, but the problem is that not all square matrices have inverses. Specifically, *an inverse exists for a matrix if and only if it is nonsingular*. Glancing up the page, you'll see this means that only matrices of full rank have inverses. And this means that matrices whose column vectors are linearly dependent do not have inverses.

Determining whether a matrix has an inverse and then computing it is tedious, and I won't discuss these topics. You are welcome to plow through Greene (pp. 821-822) to get a taste of how tedious it all is. You'll see terms like **determinant**, **cofactor**, and **adjoint**. Thankfully, (choose one: God/ Goddess/ Gaia/Santa/none of the above) has provided us computers to do this.

A few additional things you should know about inverses:

- $(\mathbf{A}^{-1})^{-1} = \mathbf{A}.$
- $(\mathbf{A}^{-1})' = (\mathbf{A}')^{-1}$.
- $(\mathbf{AB})^{-1} = \mathbf{B}^{-1}\mathbf{A}^{-1}$ provided both $\mathbf{B}^{-1}$ and $\mathbf{A}^{-1}$ exist.
- If c is a scalar, then $(c\mathbf{A})^{-1} = (1/c)\mathbf{A}^{-1}$.

7. Types of matrices you should know

- A **symmetric** matrix is a matrix that equals its transpose: a symmetric matrix $\mathbf{A} = \mathbf{A}'$. Symmetric matrices are by necessity **square** matrices. **I** (of any dimension) is a symmetric matrix. If at matrix **A** is symmetric, so is its inverse, $\mathbf{A}^{-1}$ (if it exists).

- A **diagonal** matrix has entries on its diagonal and zeroes everywhere else. Only square matrices have diagonals, so all diagonal matrices are by necessity square. **I** is a diagonal matrix.

- A **triangular** matrix has only zeroes either above or below the diagonal.

- I'll repeat this for posterity: an **idempotent** matrix is one that is equal to its square. So if a matrix **MM = M**, then **M** is idempotent. Just as in scalar algebra, we can write **MM** as $\mathbf{M}^2$. Most of the idempotent matrices we encounter are also symmetric, so $\mathbf{M} = \mathbf{M}'$, and so $\mathbf{MM}' = \mathbf{M}^2 = \mathbf{M}$.

- A related tidbit: the **trace** operator adds up all of the diagonal elements of a square matrix. So trace $\begin{bmatrix} 4 & 7 \\ 1 & 8 \end{bmatrix} = 12$, and (you'll care about this one) trace($\mathbf{uu}'$) = $\sum_{i=1}^{n} u_i^2$.

8. Linear and quadratic functions in matrix notation. Systems of equations.

If y is a linear function of x, it means that we can write the familiar equation:

$$y = a_1x_1 + a_2x_2 + ... + a_kx_k \,,$$

where the a's are coefficients attached to a total of k respective x's. We can write this more simply by considering y the function of the vector $\mathbf{x}$:

$$\underset{(1 \text{ x } 1)}{y} \quad = \quad \underset{(1 \text{ x } k)(k \text{ x } 1)}{\mathbf{a}'\mathbf{x}}$$

Now consider that we want to represent a *system* of linear equations, in which we have many observations of *y* and the a's, and would like to determine the value of the *x*'s.[6] We would write:

$$
\begin{aligned}
y_1 &= a_{11}x_1 + a_{21}x_2 + ... + a_{k1}x_k \\
y_2 &= a_{12}x_1 + a_{22}x_2 + ... + a_{k2}x_k \\
\vdots & \qquad \vdots \qquad\qquad \vdots \\
y_n &= a_{1n}x_1 + a_{2n}x_2 + ... + a_{kn}x_k
\end{aligned}
$$

But this takes up a lot of space and is difficult to analyze. How might we write this in matrix form? You guessed it:[7, 8]

$$
\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} a_{11} & a_{21} & ... & a_{k1} \\ a_{12} & a_{22} & & a_{k2} \\ \vdots & \vdots & \ddots & \vdots \\ a_{1n} & a_{2n} & ... & a_{kn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_k \end{bmatrix}
$$

$$
\underset{(n \text{ x } 1)}{\mathbf{y}} \quad = \quad \underset{(n \text{ x } k)(k \text{ x } 1)}{\mathbf{Ax}}
$$

In matrix algebra, a functional form that is one step up in complexity is often analyzed: these are equations in which *y* is what's called a *quadratic* function of the *x*'s. (We throw the word "quadratic" around a lot; we present a new use of this term here.) As with linear functions, a quadratic function has terms for every *x* in the vector **x**. But now, each of these *x*'s is *squared*. In addition, there are other terms for every possible cross-product of the *x*'s. So a quadratic function with two *x*'s—x_1 and x_2—looks like:

$$y = a_{11}x_1{}^2 + a_{22}x_2{}^2 + 2a_{12}x_1x_2$$

(Note that we are *not* talking about "quadratic equations" in the sense that we do in OLS. In fact, it is *impossible* to represent the typical OLS quadratic equation—$y = \alpha + \beta_1 x + \beta_2 x^2$—in this form. Instead, it must be represented linearly, with x^2 as an element in the vector **x**.)

In practice, we are not limited to just two *x*'s: we can have as many as we want, and these equations look like this:

[6] Note that this looks different—but is actually virtually the same as—the situation we face in least squares estimation. There, we have a matrix **X** of (known) observations, and we seek a vector of β's that meets the least-squares criteria.

[7] Note that the matrix of *a*'s has the subscripts in reverse of their usual order. This is to keep things consistent with the notation used in the scalar algebraic representation of the system of equations.

[8] Just to make sure you get the point, read footnote 6 again. How do we write this equation in the OLS context?

$$y = a_{11}x_1{}^2 + a_{22}x_2{}^2 + ... + a_{kk}x_k{}^2 + a_{12}x_1x_2 + ... + a_{1k}x_1x_k + ... + a_{k-1,k}x_{k-1}x_k$$

or in summation notation as: $y = \sum_{i=1}^{k}\sum_{j=1}^{k} a_{ij}x_ix_j$.

We can write this in matrix notion as:

$$y = \begin{bmatrix} x_1 & x_2 & ... & x_k \end{bmatrix} \begin{bmatrix} a_{11} & a_{21} & ... & a_{k1} \\ a_{12} & a_{22} & & a_{k2} \\ \vdots & \vdots & \ddots & \vdots \\ a_{1k} & a_{2k} & ... & a_{kk} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_k \end{bmatrix}$$

This of course been be written as:

$$y = \mathbf{x'Ax}$$

Look at the matrix **A**: it has the coefficients of the cross-products on its off-diagonals. In practice, **A** is often symmetric because the coefficient on any cross product x_ix_j is generally the same as the coefficient on its corresponding cross-product x_jx_i.

9. Differential Calculus in Matrix Form

You can be forgiven for having a "why the heck do we care" moment as you read section 8. Here's why: many extended equations in matrix algebra have terms that are either linear or quadratic forms. Recognizing these terms allows you to more easily manipulate them—particularly when performing differential calculus in matrix algebra. Read on for more.

You'll remember the following from scalar algebra:

- If we have a function $y = ax + bz$, we find the partial derivative of y with respect to x by calculating $\frac{\partial y}{\partial x} = a$.

- The same holds true if, instead of calling the variables x and z, we call all the variables x_1, x_2, … x_k.. The earlier equation becomes $y = a_1x_1 + a_2x_2$ and the partial derivative is now written $\frac{\partial y}{\partial x_1} = a_1$.

- OK, now the fancy stuff. When you see a list of variables x_1, x_2, … x_k, you should by now be thinking: vector! We can instead say that y is a function of the vector **x**, and that we can write the general form (does this look familiar?) as:

$$y = a_1x_1 + a_2x_2 + \ldots + a_kx_k \quad = \quad \begin{bmatrix} a_1 & a_2 & \ldots & a_k \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_k \end{bmatrix},$$

where the *a*'s are coefficients attached to a total of *k* respective *x*'s. We can write this more simply by considering *y* the function of the vector **x**:

$$\underset{(1 \text{ x } 1)}{y} \quad = \quad \underset{(1 \text{ x } k)(k \text{ x } 1)}{\mathbf{a}'\mathbf{x}}$$

- With this in hand, we can define the *vector partial derivative* of *y* with respect to *x* as a vector of partial derivatives of *y* with respect to each of the elements of *x*:

$$\frac{\partial y}{\partial \mathbf{x}} = \begin{bmatrix} \dfrac{\partial y}{\partial x_1} \\ \dfrac{\partial y}{\partial x_2} \\ \vdots \\ \dfrac{\partial y}{\partial x_k} \end{bmatrix}$$

- Now we note that $\frac{\partial y}{\partial x_1} = a_1;\ \frac{\partial y}{\partial x_2} = a_2;$ etc. Stack up these *a*'s and you get the vector **a**:

$$\frac{\partial y}{\partial \mathbf{x}} = \begin{bmatrix} \dfrac{\partial y}{\partial x_1} \\ \dfrac{\partial y}{\partial x_2} \\ \vdots \\ \dfrac{\partial y}{\partial x_k} \end{bmatrix} = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_k \end{bmatrix} = \mathbf{a}$$

- Thus the vector partial derivative of *y*= **a′x** with respect to **x** is $\frac{\partial y}{\partial \mathbf{x}} = \frac{\partial(\mathbf{a}'\mathbf{x})}{\partial \mathbf{x}} = \mathbf{a}$.

- To discuss how we take the derivative of a *matrix* with respect to a vector, we return to the set of linear functions in the example above:

$$y_1 = a_{11}x_1 + a_{21}x_2 + \ldots + a_{k1}x_k$$

$$y_2 = a_{12}x_1 + a_{22}x_2 + \ldots + a_{k2}x_k$$

$$\vdots \qquad\qquad \vdots \qquad\qquad \vdots$$

$$y_n = a_{1n}x_1 + a_{2n}x_2 + \ldots + a_{kn}x_k$$

or (as before)

$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} a_{11} & a_{21} & \ldots & a_{k1} \\ a_{12} & a_{22} & & a_{k2} \\ \vdots & \vdots & \ddots & \vdots \\ a_{1n} & a_{2n} & \ldots & a_{kn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_k \end{bmatrix}$$

$$\underset{(n \text{ x } 1)}{\mathbf{y}} \quad = \quad \underset{(n \text{ x } k)(k \text{ x } 1)}{\mathbf{Ax}}$$

- We are interested in finding the derivative of **y** with respect to **x**, which is $\frac{\partial \mathbf{y}}{\partial \mathbf{x}} = \frac{\partial \mathbf{Ax}}{\partial \mathbf{x}}$. To do this, we consider one element of **y** at a time and calculate $\frac{\partial y_1}{\partial \mathbf{x}}, \frac{\partial y_2}{\partial \mathbf{x}}, \ldots \frac{\partial y_n}{\partial \mathbf{x}}$, repeating the process we used above to find the vector partial derivative of *y* with respect to **x**:

$$\frac{\partial y_1}{\partial \mathbf{x}} = \begin{bmatrix} \frac{\partial y_1}{\partial x_1} \\ \frac{\partial y_1}{\partial x_2} \\ \vdots \\ \frac{\partial y_1}{\partial x_k} \end{bmatrix} = \begin{bmatrix} a_{11} \\ a_{21} \\ \vdots \\ a_{k1} \end{bmatrix}; \quad \frac{\partial y_2}{\partial \mathbf{x}} = \begin{bmatrix} \frac{\partial y_2}{\partial x_1} \\ \frac{\partial y_2}{\partial x_2} \\ \vdots \\ \frac{\partial y_2}{\partial x_k} \end{bmatrix} = \begin{bmatrix} a_{12} \\ a_{22} \\ \vdots \\ a_{k2} \end{bmatrix}; \quad \cdots \quad \frac{\partial y_n}{\partial \mathbf{x}} = \begin{bmatrix} \frac{\partial y_n}{\partial x_1} \\ \frac{\partial y_n}{\partial x_2} \\ \vdots \\ \frac{\partial y_n}{\partial x_k} \end{bmatrix} = \begin{bmatrix} a_{1n} \\ a_{2n} \\ \vdots \\ a_{kn} \end{bmatrix} .$$

- OK, now let’s lay these vectors of *a*’s side-by-side and put them in a matrix:

$$\begin{bmatrix} a_{11} & a_{12} & \ldots & a_{1n} \\ a_{21} & a_{22} & & \vdots \\ \vdots & & \ddots & \vdots \\ a_{k1} & \ldots & \ldots & a_{kn} \end{bmatrix}$$

- But what is this matrix? It is the transpose of our original matrix, **A**: our friend, $\mathbf{A}'$.

- Thus if **y** is a set of linear functions of **x**, the partial derivative $\frac{\partial \mathbf{y}}{\partial \mathbf{x}} = \frac{\partial (\mathbf{Ax})}{\partial \mathbf{x}} = \mathbf{A}'$.

- One more thing, and I'm not going to dwell on the math.[9] In the previous section, we also discussed the case where *y* is a quadratic function of the vector **x**, written $y = \mathbf{x}'\mathbf{A}\mathbf{x}$**.** If **A** is symmetric (and it usually is in the derivations we're doing), then the partial derivative of *y* with respect to the vector **x** is $\frac{\partial y}{\partial \mathbf{x}} = \frac{\partial (\mathbf{x}'\mathbf{Ax})}{\partial \mathbf{x}} = 2\mathbf{Ax}$.

- A final helpful result is that $\frac{\partial (\mathbf{x}'\mathbf{Ax})}{\partial \mathbf{A}} = \mathbf{xx}'$.

[9] For the math, see Greene p. 839.

(Just about) Everything You’ll Ever Need to Know About Matrix Algebra to Understand OLS in Matrix Form, Part III

Patrick Egan – patrick.egan@nyu.edu

revised: November 30, 2009

1. Random vectors and matrices

- ***Definitions.*** In scalar algebra, we work with random variables. In matrix algebra, we often work with **random vectors**, which are simply vectors of random variables. Consider the random vector **e:**

$$\mathbf{e} = \begin{bmatrix} e_1 \\ e_2 \\ e_3 \\ e_4 \end{bmatrix}$$

This random vector **e** is a (4 x 1) column vector of four random variables: $e_{1,}$ $e_{2,}$ e_3, and e_4.

And—you guessed it—we also work with **random matrices**, which are matrices of random variables. They can be of any size.

- ***Expected values.*** The expected value of a random matrix **X** is a matrix whose elements are the expected values of each of the corresponding elements of **X**. The same holds true for random matrices. If we have the following (r x c) random matrix, **X**:

$$\mathbf{X} = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1c} \\ x_{21} & x_{22} & \cdots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ x_{r1} & \cdots & \cdots & x_{rc} \end{bmatrix}, \quad \text{then} \quad E(\mathbf{X}) = \begin{bmatrix} E(x_{11}) & E(x_{12}) & \cdots & E(x_{1c}) \\ E(x_{21}) & E(x_{22}) & \cdots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ E(x_{r1}) & \cdots & \cdots & E(x_{rc}) \end{bmatrix}.$$

The same thing holds true for random vectors. So:

$$E(\mathbf{e}) = \begin{bmatrix} E(e_1) \\ E(e_2) \\ E(e_3) \\ E(e_4) \end{bmatrix}$$

A vector of expected values is often referred to as a **mean vector** and may be written like this:

$$E(\mathbf{e}) = \begin{bmatrix} E(e_1) \\ E(e_2) \\ E(e_3) \\ E(e_4) \end{bmatrix} = \begin{bmatrix} \mu_1 \\ \mu_2 \\ \mu_3 \\ \mu_4 \end{bmatrix} = \boldsymbol{\mu}\,.$$

You'll recall the definition the variance of a scalar random variable: var(x) ≡ E(x – E(x))^2. The matrix equivalence of this concept is (confusingly) called the **covariance**[1] of a random vector. The covariance of a random vector **e** is written cov(**e**) and is defined as: cov(**e**) ≡ E[(**e** – E(**e**)) (**e** – E(**e**))′], or $E[(\mathbf{e}-\boldsymbol{\mu})(\mathbf{e}-\boldsymbol{\mu})']$.

Let's write this out:

$$\text{cov}(\mathbf{e}) \equiv E[(\mathbf{e}-\boldsymbol{\mu})(\mathbf{e}-\boldsymbol{\mu})']$$

$$= E\left\{ \begin{bmatrix} e_1 - E(e_1) \\ e_2 - E(e_2) \\ e_3 - E(e_3) \\ e_4 - E(e_4) \end{bmatrix} \begin{bmatrix} e_1 - E(e_1) & e_2 - E(e_2) & e_3 - E(e_3) & e_4 - E(e_4) \end{bmatrix} \right\}$$

$$= E\begin{bmatrix} [e_1 - E(e_1)]^2 & [e_1 - E(e_1)][e_2 - E(e_2)] & [e_1 - E(e_1)][e_3 - E(e_3)] & [e_1 - E(e_1)][e_4 - E(e_4)] \\ [e_2 - E(e_2)][e_1 - E(e_1)] & [e_2 - E(e_2)]^2 & [e_2 - E(e_2)][e_3 - E(e_3)] & [e_2 - E(e_2)][e_4 - E(e_4)] \\ [e_3 - E(e_3)][e_1 - E(e_1)] & [e_3 - E(e_3)][e_2 - E(e_2)] & [e_3 - E(e_3)]^2 & [e_3 - E(e_3)][e_4 - E(e_4)] \\ [e_4 - E(e_4)][e_1 - E(e_1)] & [e_4 - E(e_4)][e_2 - E(e_2)] & [e_4 - E(e_4)][e_3 - E(e_3)] & [e_4 - E(e_4)]^2 \end{bmatrix}$$

This matrix is called the **covariance matrix** of the random vector **e**, and in most circles it is also called the **variance-covariance matrix**. We often represent the covariance matrix with the bold-faced symbol **Σ** ("sigma"). **Σ** is always symmetric: **Σ = Σ′.**

Now distribute expectations over the entire matrix. On the main diagonal of cov(**e**) are the variances of $e_{1,}$ $e_{2,}$ e_3, and e_4. On the off-diagonal are the covariances of every element of **e** with every other element of **e**:

$$\boldsymbol{\Sigma} = \begin{bmatrix} \sigma_{e_1}{}^2 & \sigma_{e_1e_2} & \sigma_{e_1e_3} & \sigma_{e_1e_4} \\ \sigma_{e_2e_1} & \sigma_{e_2}{}^2 & \sigma_{e_2e_3} & \sigma_{e_2e_4} \\ \sigma_{e_3e_1} & \sigma_{e_3e_2} & \sigma_{e_3}{}^2 & \sigma_{e_3e_4} \\ \sigma_{e_4e_1} & \sigma_{e_4e_2} & \sigma_{e_4e_3} & \sigma_{e_4}{}^2 \end{bmatrix}$$

[1] Just to make things more confusing, some texts do refer to this concept as the *variance* of a random vector.

Remember the very helpful formula $\operatorname{var}(x) = E(x^2) - \mu^2$? We have a very similar formula in matrix algebra: $\mathbf{\Sigma} = E(\mathbf{xx'}) \text{-} \boldsymbol{\mu\mu'}$. (You now know enough to be able to do the math to see how this works.)

Sometimes people like to refer to the **correlation matrix**, rather than the covariance matrix. You'll recall that the definition for the correlation of two scalar random variables *x, y* is:

$$\operatorname{corr}(x, y) = \frac{\operatorname{cov}(x, y)}{\sqrt{\operatorname{var}(x)\operatorname{var}(y)}} .$$

Well the correlation matrix of a vector of random variables **e** is a matrix in which each element of **Σ** (e.g., $\sigma_{e_r e_c}$) is divided by the corresponding standard deviations of the two elements of **e** ($\sigma_{e_r}\sigma_{e_c}$). Doing this yields a matrix with a series of ones on its diagonal:

$$\operatorname{corr}(\mathbf{e}) = \begin{bmatrix} 1 & \rho_{e_1e_2} & \rho_{e_1e_3} & \rho_{e_1e_4} \\ \rho_{e_2e_1} & 1 & \rho_{e_2e_3} & \rho_{e_2e_4} \\ \rho_{e_3e_1} & \rho_{e_3e_2} & 1 & \rho_{e_3e_4} \\ \rho_{e_4e_1} & \rho_{e_4e_2} & \rho_{e_4e_3} & 1 \end{bmatrix} .$$

Recalling that:

> σ ("sigma") is used to represent covariances and standard deviations;
> σ^2 ("sigma-squared") is used to represent variances; and
> ρ ("rho") is used to represent correlations.

We can thus rewrite the formula for the correlation of *x* and *y* as: $\rho_{xy} = \dfrac{\sigma_{xy}}{\sqrt{\sigma_x{}^2\sigma_y{}^2}} = \dfrac{\sigma_{xy}}{\sigma_x\sigma_y}$.

- ***The math of expectations in matrix algebra.*** A few properties to know regarding expected values in matrix algebra include:

  Given that **A, b** are non-random and **y**, **X** are random, then:

  (1a) E(**b′y**) **= b′**E(**y**) **= b′μ**

  (2a) cov(**b′y**) = **b′ Σ b** (what scalar formula does this remind you of?)

  Here are corresponding formulas for matrices:

  (1b) E(**Ay**) = **A μ.**

  (2b) cov(**Ay**) = **A Σ A′** (note very important difference from (2a).)

(3) $E(\mathbf{Ay} + \mathbf{b}) = \mathbf{A}E(\mathbf{y}) + \mathbf{b} = \mathbf{A}\boldsymbol{\mu} + \mathbf{b}$

(4) $E(\mathbf{AXB}) = \mathbf{A}E(\mathbf{X})\mathbf{B}$

2. The multivariate Normal distribution

Consider an (n x 1) vector of random variables, **y**. As discussed above, call E(**y**) the mean vector of **y** = **μ**, and define cov(**y**) = $E[(\mathbf{y}-\boldsymbol{\mu})(\mathbf{y}-\boldsymbol{\mu})']$ = **Σ.**

In many cases in econometrics, we can consider all the elements of **y** – the random variables $y_1, y_2 \ldots y_n$—as having a **multivariate Normal distribution**. There are a few things to know about this:

- If **y** is a multivariate Normal random vector, we write **y** ~ N(**μ, Σ)**, where **μ** is a mean vector of the expected values of each of the elements of **y**, and **Σ** is the variance-covariance matrix of each of the elements of **y.**

- If **y** ~ N(**μ, Σ)**, then each element of **y** is (univariate) Normally distributed.

- The probability density function for a multivariate Normal distribution looks a lot like the scalar version. You don't need to know the formula right now. But you do need to know that the probability density, *f*(**y**), for any possible realization of the vector **y** can be recovered just by knowing **μ** and **Σ.** The multivariate Normal distribution is thus a very succinct way to describe the joint distribution of lots of random variables.

- If **y** ~ N(**μ, Σ)**, then any two elements of **y** (call them y_i and y_j) are independent if and only if they are uncorrelated, i.e. if $\sigma_{ij} = \rho_{ij}i = 0$.

    - If all the elements of **y** are mean deviated and also i.i.d., so each element y ~N(0, σ_y^2), then **Σ** simplifies to $\sigma_y^2\mathbf{I}$.

- Linear combinations of multivariate Normal random variables are themselves Normally distributed. If **y** ~ N(**μ, Σ)**, then:

    - $\mathbf{Ay} + \mathbf{b} \sim N(\mathbf{A}\boldsymbol{\mu} + \mathbf{b}, \mathbf{A}\boldsymbol{\Sigma}\mathbf{A}')$

    - $\text{cov}(\mathbf{Ay}, \mathbf{By}) = \mathbf{A}\boldsymbol{\Sigma}\mathbf{B}'$.

- There is a **multivariate Central Limit Theorem** that applies to vector random variables. Consider the random vector **x** of dimension (k x 1) with mean **μ** and variance-covariance matrix **Σ**. Now imagine taking a random sample of size n of each of the k elements of **x**, and arranging each of these samples into (k x 1) vectors themselves. Call the first sample $\mathbf{x}_1$, the second sample $\mathbf{x}_2$, all the way to $\mathbf{x}_n$.

Now define the vector $\bar{\mathbf{x}}$ as a (*k* x 1) vector whose *k*'th element is the sample mean of the *k*'th elements of the vectors $\mathbf{x}_{1,}\, \mathbf{x}_{2,\ldots}\, \mathbf{x}_{n.}$ Thus $\bar{\mathbf{x}} = \frac{1}{n}\sum_{i=1}^{n}\mathbf{x}_i$. Now, "standardize" this vector by subtracting its mean $\boldsymbol{\mu}$ and multiplying it by the square root of *n*.

We now have a standardized vector we can write as $\sqrt{n}(\bar{\mathbf{x}}-\boldsymbol{\mu})$. Under assumptions similar to those incorporated in the scalar Central Limit Theorem, this vector $\sqrt{n}(\bar{\mathbf{x}}-\boldsymbol{\mu})$ becomes distributed multivariate Normal with mean zero and variance $\boldsymbol{\Sigma}$ as *n* approaches infinity. That is, $\sqrt{n}(\bar{\mathbf{x}}-\boldsymbol{\mu})\xrightarrow{d} N(\mathbf{0}_k, \boldsymbol{\Sigma})$, where $\xrightarrow{d}$ means "is asymptotically distributed as."

(Just about) Everything You'll Ever Need to Know About Matrix Algebra to Understand OLS in Matrix Form,[1] Part IV

Patrick Egan – patrick.egan@nyu.edu

December 2010

1. OLS in Matrix Form

By now, you've seen the classical linear model written in several ways.

Writing the model for a generic individual, *i*:

We often want to write the population model for a generic individual, *i*, in the population. When we do this, we are discussing just one (abstract) case, not the entire population. We do so with the subscript *i*:

Scalar algebra:

$$y_i = \beta_0 + \beta_1 x_{1i} + \beta_2 x_{2i} + \ldots + \beta_k x_{ki} + u_i \qquad (1)$$

or in summation notation,

$$y_i = \beta_0 + \sum_{k=1}^{K} \beta_k x_{ki} + u_i \qquad (2)$$

Just review for yourself that you understand this: we are talking about a typical case, y_i, that is the function of a constant term β_0 plus a total of *K* *x*'s, each of which has a coefficient, labeled $\beta_1 \ldots \beta_k$.

Matrix algebra:

But of course we like to use the tools of matrix algebra to simplify things, and so we note that this equation can be written in terms of the product of two vectors:

$$y_i = \begin{bmatrix} 1 & x_{1i} & x_{2i} & \ldots & x_{Ki} \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \vdots \\ \beta_K \end{bmatrix} + u_i \qquad (3)$$

[1] Portions of this handout are inspired liberally by Greene's *Econometric Analysis* (page references are from 5th edition), Stock and Watson's *Introduction to Econometrics*, and/or Hanushek and Jackson's classic, *Statistical Methods for Social Scientists*.

We then write this more succinctly as: [2]

$$y_i = \mathbf{x}_i{}'\boldsymbol{\beta} + u_i \tag{4}$$

Note that in this equation, y_i and u_i are still scalars, but now we've condensed the x's and β's into one matrix algebra term $\mathbf{x}_i{}'\boldsymbol{\beta}$.

At this point, you should be able to see that equations (1) through (4) all say the *exact same thing*.

<u>Writing the estimated equation for a generic individual, *i*:</u>

We, of course, do not know either u_i or $\boldsymbol{\beta}$. We instead estimate the equation

$$y_i = \mathbf{x}_i{}'\hat{\boldsymbol{\beta}} + \hat{u}_i, \text{ where } \hat{u}_i = y_i - \hat{y}_i.$$

<u>Writing the estimated equation for the entire dataset of *N* observations:</u>

Finally, in a dataset of N observations, we have a total of N different y_i's, x_i's, and $\hat{u}_i$'s, and thus a system of N equations that looks like this:

$$\begin{aligned} y_1 &= \mathbf{x}_1{}'\hat{\boldsymbol{\beta}} + \hat{u}_1 \\ y_2 &= \mathbf{x}_2{}'\hat{\boldsymbol{\beta}} + \hat{u}_2 \\ &\vdots \\ y_N &= \mathbf{x}_N{}'\hat{\boldsymbol{\beta}} + \hat{u}_N \end{aligned}$$

Well we of course want to use matrix notation to condense this system. We note that the y's can be rewritten as a N x 1 vector, $\mathbf{y}$. The same holds true for the $\hat{u}$'s: they become the N x 1 vector $\hat{\mathbf{u}}$. Finally, we stack the row vectors $\mathbf{x}_1{}'$ through $\mathbf{x}_N{}'$ in a matrix that is of dimension N x K:

$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix} = \begin{bmatrix} 1 & x_{11} & x_{21} & \cdots & x_{K1} \\ 1 & x_{12} & x_{22} & & x_{K2} \\ 1 & \vdots & & \ddots & \\ 1 & x_{1N} & & & x_{KN} \end{bmatrix} \begin{bmatrix} \hat{\beta}_0 \\ \hat{\beta}_1 \\ \vdots \\ \hat{\beta}_K \end{bmatrix} + \begin{bmatrix} \hat{u}_1 \\ \hat{u}_2 \\ \vdots \\ \hat{u}_N \end{bmatrix}$$

$$\mathbf{y} \quad = \quad \mathbf{X}\hat{\boldsymbol{\beta}} \quad + \quad \hat{\mathbf{u}}$$

[2] Sometimes people like to write $y_i = \boldsymbol{\beta}'\mathbf{x}_i + \hat{u}_i$ instead. See for yourself that this is the exact same thing.

1.5 Mean deviation for convenience

From here on out, we'll transform our data in a way that has no substantive import but makes our derivations a little simpler: we will assume that *y* and all the *x*'s are *mean-deviated*. By construction, this makes $\hat{\beta}_0 = 0$. (Recall in the bivariate case that $\hat{\beta}_0 = \bar{y} - \hat{\beta}_1\bar{x}$; the means of the mean-deviated values of *x* and *y* are of course zero, so $\hat{\beta}_0 = 0$.) So in the following derivations, no estimate of β_0 is generated, the matrix **X** is of dimension *N* x *K* and the vector **β** is of dimension *K* x 1.

2. Deriving the OLS estimator

As you know, in the equation $\mathbf{y} = \mathbf{X}\hat{\boldsymbol{\beta}} + \hat{\mathbf{u}}$, the vector $\hat{\mathbf{u}}$ contains what's known as the "residuals." An even more intuitive term for them might actually be "mistakes." Because if we use $\mathbf{y} = \mathbf{X}\hat{\boldsymbol{\beta}} + \hat{\mathbf{u}}$ to fit a (multivariate) line to a set of data points, we produce predictions about **y** (which we call $\hat{\mathbf{y}}$, or "y-hat") from the equation:

$$\hat{\mathbf{y}} = \mathbf{X}\hat{\boldsymbol{\beta}} \qquad \text{whereas (from above):} \quad \mathbf{y} = \mathbf{X}\hat{\boldsymbol{\beta}} + \hat{\mathbf{u}}$$

and therefore we can calculate $\hat{\mathbf{u}}$ as:

$$\hat{\mathbf{u}} = \mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}} \qquad \text{or} \qquad \hat{\mathbf{u}} = \mathbf{y} - \hat{\mathbf{y}} \tag{5}$$

So each of the *n* rows in the vector $\hat{\mathbf{u}}$ is the "mistake" made by the estimated equation for the observation *n*—specifically, how far "off" it was in predicting each of the corresponding entries in the vector **y**. A natural criterion for a line that best fits the data is to make these mistakes as small as possible. And for various statistical and theoretical reasons, we therefore use the least squares criterion—that is, *minimizing the sum of squared mistakes/residuals*—to derive the OLS estimator.

But what is the sum of squared residuals? By page 4 of handout II, we know that if we want to sum up the squares of a vector's elements, we pre-multiply it by its transpose. Thus the sum of squared residuals is thus simply:

$$\text{SSR} = \hat{\mathbf{u}}'\hat{\mathbf{u}}.$$

Now comes the fun part. We will use matrix calculus to figure out the value of the vector **b** that minimizes $\hat{\mathbf{u}}'\hat{\mathbf{u}}$. First we use equation (5) to make the following substitution for **u**:

$$\hat{\mathbf{u}}'\hat{\mathbf{u}} = (\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}})'(\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}})$$

Now look at handout II, page 1. Just like in scalar algebra, we can use the good old "FOIL" method to multiply this out:

$$\begin{aligned}\hat{\mathbf{u}}'\hat{\mathbf{u}} &= (\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}})'(\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}}) \\ &= (\mathbf{y}' - \hat{\boldsymbol{\beta}}'\mathbf{X}')(\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}}) \\ &= \mathbf{y}'\mathbf{y} - \mathbf{y}'\mathbf{X}\hat{\boldsymbol{\beta}} - \hat{\boldsymbol{\beta}}'\mathbf{X}'\mathbf{y} + \hat{\boldsymbol{\beta}}'\mathbf{X}'\mathbf{X}\hat{\boldsymbol{\beta}}\end{aligned}$$

So far, so good, right? Now let's look at the final line of the equation above. For the indicated subtraction to be possible, every term must be comformable for subtraction, which means they must each be of the same dimension. A quick glance at $\mathbf{y}'\mathbf{y}$ and you know it is a scalar. So every every other term of this expression must be a scalar as well.

Why do we care? Because if each of these terms is a scalar, then each of these terms equals its transpose. And so $\mathbf{y}'\mathbf{X}\hat{\boldsymbol{\beta}} = \left(\hat{\boldsymbol{\beta}}'\mathbf{X}'\mathbf{y}\right)'$, and thus we can rewrite the final expression as:

$$\hat{\mathbf{u}}'\hat{\mathbf{u}} = \mathbf{y}'\mathbf{y} - 2\mathbf{y}'\mathbf{X}\hat{\boldsymbol{\beta}} + \hat{\boldsymbol{\beta}}'\mathbf{X}'\mathbf{X}\hat{\boldsymbol{\beta}}\,.$$

OK, now recall from calculus that the first step in finding the minimum and/or maximum of a function is by taking its first derivative and setting it equal to zero. In this case, $\hat{\mathbf{u}}'\hat{\mathbf{u}}$ is a function of $\hat{\boldsymbol{\beta}}$, and we want to find the value of $\hat{\boldsymbol{\beta}}$ that minimizes $\hat{\mathbf{u}}'\hat{\mathbf{u}}$. We are thus interested in finding the value of $\hat{\boldsymbol{\beta}}$ that satisfies

$$\frac{\partial(\hat{\mathbf{u}}'\hat{\mathbf{u}})}{\partial\hat{\boldsymbol{\beta}}} = \mathbf{0}\ .$$

So let's calculate $\frac{\partial(\hat{\mathbf{u}}'\hat{\mathbf{u}})}{\partial\hat{\boldsymbol{\beta}}}$, term by term:

$$\frac{\partial(\hat{\mathbf{u}}'\hat{\mathbf{u}})}{\partial\hat{\boldsymbol{\beta}}} = \frac{\partial(\mathbf{y}'\mathbf{y})}{\partial\hat{\boldsymbol{\beta}}} - \frac{\partial(2\mathbf{y}'\mathbf{X}\hat{\boldsymbol{\beta}})}{\partial\hat{\boldsymbol{\beta}}} + \frac{\partial(\hat{\boldsymbol{\beta}}'\mathbf{X}'\mathbf{X}\hat{\boldsymbol{\beta}})}{\partial\hat{\boldsymbol{\beta}}}$$

• Since the first term ($\mathbf{y}'\mathbf{y}$) doesn't contain $\hat{\boldsymbol{\beta}}$, its derivative with respect to $\hat{\boldsymbol{\beta}}$ is equal to zero.

• Take a close look at the second term, $\frac{\partial(2\mathbf{y}'\mathbf{X}\hat{\boldsymbol{\beta}})}{\partial\hat{\boldsymbol{\beta}}}$. It is the derivative of a row vector (2$\mathbf{y}'\mathbf{X}$) times a column vector ($\hat{\boldsymbol{\beta}}$) with respect to the same column vector ($\hat{\boldsymbol{\beta}}$). Thus 2$\mathbf{y}'\mathbf{X}\hat{\boldsymbol{\beta}}$ is a *linear function* of $\hat{\boldsymbol{\beta}}$ (Handout II, page 13). So this case corresponds to that on the handout of finding $\frac{\partial(\mathbf{a}'\mathbf{x})}{\partial\mathbf{x}}$, with $\mathbf{a}'$ = ($\mathbf{y}'\mathbf{X}$) and $\mathbf{x} = \hat{\boldsymbol{\beta}}$. Since $\frac{\partial(\mathbf{a}'\mathbf{x})}{\partial\mathbf{x}} = \mathbf{a}$ (i.e., the transpose of $\mathbf{a}'$), then $\frac{\partial(2\mathbf{y}'\mathbf{X}\hat{\boldsymbol{\beta}})}{\partial\hat{\boldsymbol{\beta}}} = (2\mathbf{y}'\mathbf{X})' = 2\mathbf{X}'\mathbf{y}$.

• The final term $\hat{\boldsymbol{\beta}}'\mathbf{X}'\mathbf{X}\hat{\boldsymbol{\beta}}$ also has a correspondence to terms we've learned before. It is a row vector ($\hat{\boldsymbol{\beta}}'$) times a square, symmetric matrix ($\mathbf{X}'\mathbf{X}$) of dimension (K x K), times the same vector as a column, $\hat{\boldsymbol{\beta}}$. The term $\hat{\boldsymbol{\beta}}'\mathbf{X}'\mathbf{X}\hat{\boldsymbol{\beta}}$ is thus a *quadratic function* of the vector $\hat{\boldsymbol{\beta}}$ (see handout II, p 15), and its derivative is $2\mathbf{X}'\mathbf{X}\hat{\boldsymbol{\beta}}$.

Bringing it all together, we have:

$$\frac{\partial(\hat{\mathbf{u}}'\hat{\mathbf{u}})}{\partial\hat{\boldsymbol{\beta}}} = -2\mathbf{X}'\mathbf{y} + 2\mathbf{X}'\mathbf{X}\hat{\boldsymbol{\beta}} = \mathbf{0}$$

Now let's solve for $\hat{\boldsymbol{\beta}}$:

$$\mathbf{X}'\mathbf{X}\hat{\boldsymbol{\beta}} = \mathbf{X}'\mathbf{y}$$

To get $\hat{\boldsymbol{\beta}}$ by itself, we need to "divide" both sides of the equation by the matrix **X′X**. In matrix algebra, this is of course done by premultiplying both sides by the inverse of **X′X**:

$$(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{X}\hat{\boldsymbol{\beta}} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y}$$

$$\mathbf{I}\hat{\boldsymbol{\beta}} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y}$$

$$\hat{\boldsymbol{\beta}} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y}$$

Therefore $(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y}$ is the value of $\hat{\boldsymbol{\beta}}$ that minimizes $\hat{\mathbf{u}}'\hat{\mathbf{u}}$ and it is thus our OLS estimator. You will note that this conclusion requires that the inverse of **X′X** exists. This assumption is equivalent to the assumption that **X** is of full column rank, i.e. rank(**X**) = K… which is equivalent to the assumption that there is no perfect multicollinearity among the x's…which will be one of the assumptions we make when we move from description to inference.[3]

Before we move on, it's worth asking: what exactly is $(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y}$?

We first note that $(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y}$ is a K x 1 column vector. Now let's look at it from right to left:

- $\mathbf{X}'\mathbf{y}$, the "numerator," is a (K x 1) column vector. Its k'th row is a scalar that is the vector product of the k'th column of **X** (i.e., every observation of the variable x_k) times the corresponding observation of **y**, summed up:

[3] Those of you who remember more calculus will recall that to be sure we have a minimum and not a maximum, we need to evaluate the sign of the *second* derivative of $\hat{\mathbf{u}}'\hat{\mathbf{u}}$. We have to do this in matrix algebra, too—it's called seeing if the second derivative of $\hat{\mathbf{u}}'\hat{\mathbf{u}}$ with respect to $\hat{\boldsymbol{\beta}}$ is a "positive definite" matrix. I won't go into details about it here, but the assumption that **X** has full column rank establishes that this matrix is indeed positive definite and that the solution for $\hat{\boldsymbol{\beta}}$ is unique (details in Greene, p. 21).

$(\mathbf{X'y})_k = x_{k1}y_1 + x_{k2}y_2 + ... + x_{kN}y_N = \sum_{i=1}^{N} x_{ki}y_i$. This quantity captures how x_k and y covary with one another.

- Now let's look at the "denominator," $(\mathbf{X'X})^{-1}$, a K x K matrix. It's hard to describe this precisely without getting into a lot of math, but the k'th row of this inverse matrix captures how x_k covaries with all of the x's in the model. As x_k covaries less with the other x's, the entries on the k'th row of $(\mathbf{X'X})^{-1}$ get smaller. As x_k covaries more with the other x's, the entries on the k'th row of $(\mathbf{X'X})^{-1}$ get bigger.

- Therefore, when we calculate the expression $(\mathbf{X'X})^{-1}\mathbf{X'y}$ we are in a sense calculating a total of K "ratios," each of which corresponds to one of the x's in the model. For any of these x's—for example, x_k—this ratio compares (a) the extent to which x_k and y covary with one another; and (b) the extent to which x_k covaries with all the x's in the model. If (a) is relatively larger than (b), the k'th entry in $\hat{\boldsymbol{\beta}}$ —our estimate of β_k—will be bigger. But as (b) gets larger relative to (a), our estimate of β_k will be smaller. Note the conceptual similarity to the bivariate estimate, $\hat{\beta}$ = cov(x,y) / var(x).

3. Showing that the OLS estimator is unbiased

Having derived the value of $\hat{\boldsymbol{\beta}}$ that minimizes the squares of the mistakes made by the equation $\hat{\mathbf{y}} = \mathbf{X}\hat{\boldsymbol{\beta}}$ in predicting **y**, we now proceed to evaluate whether $\hat{\boldsymbol{\beta}}$ is indeed an unbiased estimator of the vector of population parameters, $\boldsymbol{\beta}$. We do this by seeing if E($\hat{\boldsymbol{\beta}}$) = $\boldsymbol{\beta}$**.**

We start by making the assumption that the DGP is linear in parameters:

Assumption 1: the population model can be written $\mathbf{y} = \mathbf{X}\ \boldsymbol{\beta} + \mathbf{u}$.

This allows us to make the substitution of this population model $\mathbf{X}\boldsymbol{\beta}$ + **u** for **y** in our derived equation for $\hat{\boldsymbol{\beta}}$:

$$\begin{aligned}
\hat{\boldsymbol{\beta}} &= (\mathbf{X'X})^{-1}\mathbf{X'y} \\
&= (\mathbf{X'X})^{-1}\mathbf{X'}(\mathbf{X}\boldsymbol{\beta} + \mathbf{u}) \\
&= (\mathbf{X'X})^{-1}\mathbf{X'X}\boldsymbol{\beta} + (\mathbf{X'X})^{-1}\mathbf{X'u} \\
&= \mathbf{I}\boldsymbol{\beta} + (\mathbf{X'X})^{-1}\mathbf{X'u} \\
&= \boldsymbol{\beta} + (\mathbf{X'X})^{-1}\mathbf{X'u}
\end{aligned}$$

In order to take expectations, here we make a second assumption:

Assumption 2: We have a random sample of n observations of the vector $[y, x_1 \ldots x_K]'$, making them i.i.d.

This allows us to treat the x's and y's as repeated realizations of the same DGP and thus take expectations over all observations:

$$\begin{aligned} E(\hat{\boldsymbol{\beta}}) &= E[\boldsymbol{\beta} + (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{u}] \\ &= E[\boldsymbol{\beta}] + E[(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{u}] \qquad (6) \\ &= \boldsymbol{\beta} + E[(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{u}] \end{aligned}$$

Since we need $(\mathbf{X}'\mathbf{X})^{-1}$ to exist, we need the following assumption, so we might as well make it here:

Assumption 3: There is no perfect multicollinearity among the x's, so rank(**X**) = K.

So what to do with the term $E[(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{u}]$? Here, we need the generally most audacious assumption, which is that

Assumption 4: $E(\mathbf{u} \mid \mathbf{X}) = \mathbf{0}$.

This assumption allows us to treat **X** as fixed—an unrealistic assumption. But it has the same consequences as assuming that **X** is generated by a mechanism unrelated to **u**, which is not necessarily unrealistic. And of course the big assumption we're making here is that our model has not omitted any factor that is related to both **X** and **y** and thus shows up in **u**.

With this assumption under our belt, we can rewrite (6) and state triumphantly that:

$$\begin{aligned} E(\hat{\boldsymbol{\beta}}) &= \boldsymbol{\beta} + E[(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{u}] \\ &= \boldsymbol{\beta} + (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'E(\mathbf{u} \mid \mathbf{X}) \\ &= \boldsymbol{\beta} + (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{0} \\ &= \boldsymbol{\beta} \end{aligned}$$

Some very fun exercises:

- Show (using matrix notation, of course) that the sum of the residuals/mistakes is, by construction, equal to zero. (Remember this is different from the sum of squared residuals, and it is also different from the sum of the errors u_i.)

- Show that the covariances of the residuals/mistakes with each *x* are, by construction, equal to zero.

4. The sampling distribution of $\hat{\boldsymbol\beta}$

Having established that the OLS estimator $\hat{\boldsymbol\beta}$ is an unbiased estimate of **β**, we are now interested in $\hat{\boldsymbol\beta}$'s sampling distribution—in particular, its mean vector **μ** and its variance-covariance matrix **Σ**. If we are able to fully define this distribution, we can then conduct hypothesis tests about how close any particular estimate $\hat{\boldsymbol\beta}$ is likely to be to a hypothesized population value $\boldsymbol\beta_0$. Because E($\hat{\boldsymbol\beta}$) = **β**, we know that the mean vector **μ** = E($\hat{\boldsymbol\beta}$) = **β.** Let's now examine **Σ**=cov($\hat{\boldsymbol\beta}$), the variance-covariance matrix of $\hat{\boldsymbol\beta}$ **.**

From handout III, you'll remember that, by the definition of the covariance of a vector, we calculate:

$$\boldsymbol\Sigma_{\hat{\boldsymbol\beta}} = \mathrm{cov}(\hat{\boldsymbol\beta}) = E\{[\hat{\boldsymbol\beta} - E(\hat{\boldsymbol\beta})][[\hat{\boldsymbol\beta} - E(\hat{\boldsymbol\beta})]'\}$$

But because E($\hat{\boldsymbol\beta}$) is simply **β**, we can write:

$$\mathrm{cov}(\hat{\boldsymbol\beta}) = E\{[\hat{\boldsymbol\beta} - \boldsymbol\beta][[\hat{\boldsymbol\beta} - \boldsymbol\beta]'\}.$$

Because $\hat{\boldsymbol\beta}$ and **β** are *K* x 1 vectors, cov($\hat{\boldsymbol\beta}$) is a *K* x *K* matrix. We would like to write this matrix in terms of the vector of errors **u**, so we recall (from page 5) above that $\hat{\boldsymbol\beta} = \boldsymbol\beta + (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{u}$ and thus $\hat{\boldsymbol\beta} - \boldsymbol\beta = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{u}$. We substitute this vector into the equation above to obtain:

$$\mathrm{cov}(\hat{\boldsymbol\beta}) = E\{[(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{u}][[(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{u}]'\} \qquad \text{(substituting)}$$

$$= E[(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{u}\mathbf{u}'\mathbf{X}((\mathbf{X}'\mathbf{X})^{-1})'] \qquad \text{(distributing the transpose)}$$

$$= E[(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{u}\mathbf{u}'\mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}] \qquad \text{(recalling that the transpose of an inverse is equal to the inverse of the transpose, and } (\mathbf{X'X})' = (\mathbf{X'X})$$

$$= (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'E[\mathbf{u}\mathbf{u}']\mathbf{X}(\mathbf{X}'\mathbf{X})^{-1} \qquad \text{(treating } \mathbf{X} \text{ as fixed and so E(}\mathbf{X}) = \mathbf{X})$$

But what is E[**uu′**]? It is a different variance-covariance matrix—the one associated with the random vector **u**—and it looks like this:

$$\mathbf{E[uu']} = \mathbf{\Sigma}_u = \begin{bmatrix} \sigma_{u_1}{}^2 & \sigma_{u_1u_2} & \cdots & \sigma_{u_1u_N} \\ \sigma_{u_2u_1} & \sigma_{u_2}{}^2 & & \vdots \\ \vdots & & \ddots & \vdots \\ \sigma_{u_Nu_1} & \cdots & \cdots & \sigma_{u_N}{}^2 \end{bmatrix} \qquad (7)$$

We now proceed to make two additional assumptions that allow us to dramatically simplify $\mathbf{\Sigma}_u$.

Assumption 5:
Homoskedasticity of errors: var(u_1) = var(u_2) =…= var(u_N) =σ_u^2 for all *i*.
Non-autocorrelation of errors: cov(u_i, u_j) = 0 for all *i* ≠ *j*.

First, we assume that the variances of the errors—that is, the dispersion of the "mistakes" made by the model in predicting **y**—is the same for every case in the population. This is the homoskedasticity assumption. Second, we assume that all of the off-diagonal entries in $\mathbf{\Sigma}_u$ are zero. This is the non-autocorrelation assumption: our expectation is that none of the errors covaries with any other.

The statements in Assumption 5 together allows us to vastly simplify our expression for $\mathbf{\Sigma}_u$ in equation (7). The off-diagonals all become zero by assumption 4. The entries on the main diagonal become all the same—$\sigma_u{}^2$—by assumption 5. And so $\mathbf{\Sigma}_u$ now looks like this:

$$\mathbf{\Sigma}_u = \begin{bmatrix} \sigma_u{}^2 & 0 & \cdots & 0 \\ 0 & \sigma_u{}^2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma_u{}^2 \end{bmatrix}$$

Much nicer. But we can write this even more simply by noting that $\mathbf{\Sigma}_u$ is the product of a scalar—the variance $\sigma_u{}^2$—and an identity matrix of dimension *N*. So therefore $\mathbf{\Sigma}_u = \sigma_u{}^2\mathbf{I}_N$.

(I note here that sometimes this matrix is denoted $\mathbf{\Omega}$.) Now, we can return to the expression at the top of page 7 and substitute:

$$\begin{aligned} \text{cov}(\hat{\boldsymbol{\beta}}) &= \mathbf{(X'X)^{-1}X'}E[\mathbf{uu'}]\mathbf{X(X'X)^{-1}} \\ &= \mathbf{(X'X)^{-1}X'} \begin{bmatrix} \sigma_u{}^2 & 0 & \cdots & 0 \\ 0 & \sigma_u{}^2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma_u{}^2 \end{bmatrix} \mathbf{X(X'X)^{-1}} \\ &= \mathbf{(X'X)^{-1}X'}\sigma_u{}^2\mathbf{I}_N\mathbf{X(X'X)^{-1}} \end{aligned}$$

Both $\sigma_u{}^2$, a scalar and $\mathbf{I}_N$, the identity matrix, have commutative properties in matrix multiplication. So the expression for $\boldsymbol{\Sigma}_u$ becomes:

$$
\begin{aligned}
\text{cov}(\hat{\boldsymbol{\beta}}) &= \sigma_u{}^2\mathbf{I}_N\,\mathbf{(X'X)^{-1}\,X'X(X'X)^{-1}} && \text{(moving } \sigma_\varepsilon{}^2\mathbf{I}_N \text{ to front of the expression)} \\
&= \sigma_u{}^2\mathbf{I}_N\mathbf{I}_N\,\mathbf{(X'X)^{-1}} && \text{(because } \mathbf{(X'X)^{-1}\,X'X} \text{ equals the identity matrix)} \\
&= \sigma_u{}^2\,\mathbf{(X'X)^{-1}}
\end{aligned}
$$

And so the OLS estimator, the random vector $\hat{\boldsymbol{\beta}}$, is distributed with mean vector $\boldsymbol{\beta}$ and variance-covariance matrix $\sigma_u{}^2\mathbf{(X'X)^{-1}}$. Consider for a moment what this means about the distribution of any element of $\hat{\boldsymbol{\beta}}$—say, $\hat{\beta}_1$. To find its mean, we look to the mean vector $\boldsymbol{\mu} = E\left(\hat{\boldsymbol{\beta}}\right)$. To find its variance, we look to the variance-covariance matrix $\boldsymbol{\Sigma}_{\hat{\boldsymbol{\beta}}}$:

We find the mean of $\hat{\beta}_1$, which is E($\hat{\beta}_1$), here:

$$
\boldsymbol{\mu} = \begin{bmatrix} \boxed{E(\hat{\beta}_1)} \\ \vdots \\ E(\hat{\beta}_K) \end{bmatrix} = \begin{bmatrix} \beta_1 \\ \vdots \\ \beta_K \end{bmatrix} = \begin{bmatrix} \boxed{\mathbf{(X'X)^{-1}_1(X')_1 y}} \\ \\ \mathbf{(X'X)^{-1}_K(X')}_K\,\mathbf{y} \end{bmatrix}
$$

We find the variance of $\hat{\beta}_1$, which is E[$\hat{\beta}_1$ – E($\hat{\beta}_1$)]^2, here:

$$
\text{cov}(\hat{\boldsymbol{\beta}}) = \begin{bmatrix} \boxed{E[(\hat{\beta}_1 - E(\hat{\beta}_1)]^2} & E\{[(\hat{\beta}_1 - E(\hat{\beta}_1)][(\hat{\beta}_2 - E(\hat{\beta}_2)] & \cdots & E\{[(\hat{\beta}_1 - E(\hat{\beta}_1)][(\hat{\beta}_K - E(\hat{\beta}_K)] \\ \vdots & E[(\hat{\beta}_2 - E(\hat{\beta}_2)]^2 & & \vdots \\ E\{[(\hat{\beta}_K - E(\hat{\beta}_K)][(\hat{\beta}_1 - E(\hat{\beta}_1)] & \ldots & \ldots & E[(\hat{\beta}_K - E(\hat{\beta}_K)]^2 \end{bmatrix}
$$

$$
= \boldsymbol{\Sigma}_{\hat{\boldsymbol{\beta}}} = \begin{bmatrix} \sigma_{\hat{\beta}_1}{}^2 & \sigma_{\hat{\beta}_1\hat{\beta}_2} & \cdots & \sigma_{\hat{\beta}_1\hat{\beta}_K} \\ \sigma_{\hat{\beta}_2\hat{\beta}_1} & \sigma_{\hat{\beta}_2}{}^2 & & \\ \vdots & & \ddots & \\ \sigma_{\hat{\beta}_K\hat{\beta}_1} & & & \sigma_{\hat{\beta}_K}{}^2 \end{bmatrix} = \sigma_u{}^2\,\mathbf{(X'X)^{-1}}
$$

$$
= \begin{bmatrix} \boxed{\sigma_u{}^2\mathbf{(X'X)}^{-1}{}_{11}} & \sigma_u{}^2\mathbf{(X'X)}^{-1}{}_{12} & \cdots & \sigma_u{}^2\mathbf{(X'X)}^{-1}{}_{1K} \\ \sigma_u{}^2\mathbf{(X'X)}^{-1}{}_{21} & \sigma_u{}^2\mathbf{(X'X)}^{-1}{}_{22} & & \vdots \\ \vdots & & \ddots & \\ \sigma_u{}^2\mathbf{(X'X)}^{-1}{}_{K1} & & \cdots & \sigma_u{}^2\mathbf{(X'X)}^{-1}{}_{KK} \end{bmatrix}
$$

Thus $\hat{\beta}_1$ is distributed with mean β_1 and variance $\sigma_u{}^2\mathbf{(X'X)}^{-1}{}_{11}$. You may be asking yourself, what is this mysterious thing $\mathbf{(X'X)}^{-1}{}_{11}$? Here's a more straightforward way to think about it: as shown in Greene (p. 57), the *k*'th diagonal element of $\mathbf{(X'X)}^{-1}$, denoted $\mathbf{(X'X)}^{-1}{}_{\mathrm{kk}}$, can be calculated as $\mathbf{(X'X)}^{-1}{}_{\mathrm{kk}} = \dfrac{1}{(1-R_k^2)SST_{x_k}}$ and thus:

$$\mathrm{var}(\hat{\beta}_k) = \sigma_u{}^2\mathbf{(X'X)}^{-1}{}_{\mathrm{kk}} = \frac{\sigma_u{}^2}{(1-R_k^2)SST_{x_k}},$$

where R_k^2 is the R^2 statistic that would be obtained in a regression of x_k on all the other variables in the model, and SST_{x_k} is the total sum of squares in our sample of x_k. Note the three factors that influence the precision of our estimate of β_k: (1) the fit of the regression (which would result in a lower $\sigma_u{}^2$); (2) the extent to which x_k is multicollinear with the other *x*'s (which would result in a high R_k^2) and (3) the variance of x_k. Note how similar this is to the formula for the variance of our estimator in the bivariate case.

5. Large-sample properties of the OLS estimator

In practice, we never actually know the population parameter $\sigma_u{}^2$ and thus we never actually know var($\hat{\boldsymbol{\beta}}$). So we estimate it from our data instead. We use the unbiased[4] estimator:

$$\sigma_u{}^2 = \frac{\hat{\mathbf{u}}'\hat{\mathbf{u}}}{N-\boxed{K-1}} = \frac{\sum_{i=1}^{N}\hat{u}_i{}^2}{N-K-1}$$

Here, we no longer assume mean deviation of the x's and y's, and so we estimate a constant. We are therefore estimating K+1 parameters, leaving us with N-K-1 d.f.

to estimate $\sigma_u{}^2$. The formula for $\sigma_u{}^2$ should look familiar: its square root is the standard error of the estimate (the SEE), a.k.a. the root mean squared error (the root MSE). You'll recall that the SEE is the typical amount by which the prediction equation $\hat{\mathbf{y}} = \mathbf{X}\hat{\boldsymbol{\beta}}$ is "off" from the actual values **y**.

We return to the variance-covariance matrix of $\hat{\boldsymbol{\beta}}$, $\boldsymbol{\Sigma}_{\hat{\boldsymbol{\beta}}} = \sigma_u{}^2\mathbf{(X'X)^{-1}}$ and substitute $\sigma_u{}^2$ to get

$$\boldsymbol{\Sigma}_{\hat{\boldsymbol{\beta}}} = \sigma_u{}^2\mathbf{(X'X)^{-1}}.$$

This helps us with the final step in fully defining the sampling distribution of the random vector $\hat{\boldsymbol{\beta}}$. In large enough samples (thanks to the multivariate Central Limit Theorem), $\hat{\boldsymbol{\beta}}$'s distribution

[4] I will spare you the details of the proof of this unbiasedness. For a nice proof, see Hanushek and Jackson p. 138.

approaches the multivariate Normal distribution with mean **β** and variance $\sigma_u^{\,2}(\mathbf{X}'\mathbf{X})^{-1}$, i.e. $\hat{\boldsymbol{\beta}}$ $\xrightarrow{d}$ *MVN*($\boldsymbol{\beta}$, $\sigma_u^{\,2}(\mathbf{X}'\mathbf{X})^{-1}$). (The symbol $\xrightarrow{d}$ means "distributed asymptotically as.") This means that if we standardize any element of $\hat{\boldsymbol{\beta}}$ (let's say $\hat{\beta}_k$) by subtracting its hypothesized mean (β_{k0}) and dividing by its variance, its distribution approaches the standard Normal distribution as *N* gets large:

$$\frac{\hat{\beta}_k - \beta_{k0}}{\sqrt{\sigma_u^2(\mathbf{X}'\mathbf{X})^{-1}_{kk}}} \xrightarrow{d} N(0,1)$$

One last step: we are estimating $\sigma_u^{\,2}$ with $\sigma_u^{\,2}$, and thus calculating the statistic $\dfrac{\hat{\beta}_k - \beta_{k0}}{\sqrt{\hat{\sigma}_u^2(\mathbf{X}'\mathbf{X})^{-1}_{kk}}}$. The distribution of this statistic is not standard Normal. Instead, it is distributed according to the Student's *t* distribution with *N*-*K*-1 degrees of freedom:

$$\frac{\hat{\beta}_k - \beta_{k0}}{\sqrt{\hat{\sigma}_u^2(\mathbf{X}'\mathbf{X})^{-1}_{kk}}} \sim t_{N-K-1}$$

In practice, we are often interested in testing the null hypothesis H$_0$: $\beta_{k0} = 0$. In this case, the test statistic becomes:

$$\frac{\hat{\beta}_k}{\sqrt{\hat{\sigma}_u^2(\mathbf{X}'\mathbf{X})^{-1}_{kk}}} \sim t_{N-K-1}$$

And when you look at typical regression output, you will usually see two numbers arranged like this:

12.3***
(.45)

These numbers correspond to $\hat{\beta}_k$ and the estimated standard error of $\hat{\beta}_k$. If you divide the first number by the second, you get the *t*-statistic whose formula is shown above. The asterisks correspond to the *p*-value associated with the *t*-statistic.

6. Small-sample properties of the OLS estimator

In samples that are so small that it is difficult to argue that the multivariate Central Limit Theorem really applies to the sampling distribution of $\hat{\boldsymbol{\beta}}$, analysts tend to "fudge" it by making the additional assumption that the population errors u_i are not only homoskedastic and non-autocorrelated—but that they're also distributed Normal.[5] With this assumption in place, we do

[5] I'm not kidding. Greene writes that this assumption is "crucial" (p.42) or "useful" (p. 50) in defining the sampling distribution of **b** (which of course it is), but he doesn't justify the assumption. Hanushek and Jackson simply say the assumption is "needed" (p. 122). Stock and Watson are more forthright: the

not need to rely on a large sample to invoke the multivariate Central Limit Theorem. The formula for the *t*-statistic above thus holds in small samples if we invoke the assumption of Normality of the errors.

7. *Bon voyage*

You now much of the matrix tools and notation at your disposal to read relatively high-tech work about regression analysis in matrix form. A few tips to always remember:

- Unfortunately, it seems like just about every author uses different notation. As a bonus, they'll invoke the assumptions listed here in a slightly different order or with slightly different language. Be alert for these sorts of changes.
- Remember that while we use matrix algebra for calculations, we're usually ultimately interested in talking about results in terms of scalar quantities. If you're puzzled over what a result means in matrix notation, write out the vectors and matrices with their scalar elements. Think about where the numbers you see in regression output actually come from.
- Comfort with summation notation is often crucial for understanding matrix notation. Get it down cold.
- A few key terms—including $\hat{\mathbf{u}}'\hat{\mathbf{u}}$, $\hat{\mathbf{u}}\hat{\mathbf{u}}'$, $\mathbf{X}'\mathbf{X}$, and (the behemoth) $(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'E[\mathbf{u}\mathbf{u}']\mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}$ are the building blocks of OLS, and (later) GLS and WLS. Get to know these terms and learn to recognize them.
- Later in your studies, you'll focus somewhat obsessively on the variance-covariance matrix of the errors, $E[\mathbf{u}\mathbf{u}']$. Become accustomed to this matrix and learn how departures from the homoskedasticity and non-autocorrelation assumptions affect it.

assumption "may not hold in practice" but it can "enhance our understanding of the OLS estimator." (p. 575-76.)