MATRIX ALGEBRA FOR STATISTICS: PART 1

Matrices provide a compact notation for expressing systems of equations or variables. For instance, a linear function might be written as:

$$y = x_1 b_1 + x_2 b_2 + x_3 b_3 + \ldots + x_k b_k$$

This is really the product of a bunch of b variables and a bunch of x variables. A vector is simply a collection of variables (in a particular order). We could define a (k-dimensional) vector $\mathbf{x} = (x_1, x_2, \ldots, x_n)$, and another vector $\mathbf{b} = (b_1, b_2, \ldots, b_n)$. Again, these vectors simply represent the collections of x and a variables; the **dimension** of the vector is the number of elements in it.

We define the **product** of two vectors to be:

$$\mathbf{x} \cdot \mathbf{b} \equiv \sum_{i=1}^{k} x_i b_i = x_1 b_1 + x_2 b_2 + \ldots + x_k b_k$$

(Specifically, this is called a **dot product** or **inner product**; there exist other ways to calculate products, but we won't be using those.) If you think of $\mathbf{b}$ as "the collection of all b variables" and $\mathbf{x}$ as "the collection of all x variables", then the product $\mathbf{x} \cdot \mathbf{b}$ is "the product of each b variable with the corresponding x variable." You can calculate the (dot) product only when the two vectors have the same dimension.

Example: *Let* $\mathbf{a} = (1,2,3,4)$ *and let* $\mathbf{b} = (5,6,7,8)$. *These are both 4-dimensional vectors, so we can calculate their dot product.* $\mathbf{a} \cdot \mathbf{b} = \sum_{i=1}^{4} a_i b_i = (1 \cdot 5) + (2 \cdot 6) + (3 \cdot 7) + (4 \cdot 8)$ $= 5 + 12 + 21 + 32 = 70$.

Sometimes, we say that two vectors are **orthogonal** if their dot product equals zero. Orthogonality has two interpretations. Graphically, it means that the vectors are perpendicular. On a deeper philosophical level, it means that the vectors are unrelated.

Example: *Let* $\mathbf{c} = (0,1)$ *and let* $\mathbf{d} = (1,0)$. *Since* $\mathbf{c} \cdot \mathbf{d} = (0 \cdot 1) + (1 \cdot 0) = 0 + 0 = 0$, *the vectors are orthogonal. Graphically, we can represent* $\mathbf{c}$ *as a line from the origin to the point* $(0,1)$ *and* $\mathbf{d}$ *as a line from the origin to* $(1,0)$. *These lines are perpendicular. (On a deeper sense, they are "unrelated" because the first vector moves only along the x-axis and never changes its y-direction; the second moves only along the y-axis and doesn't change its x-direction.)*

Example: *Let* $\mathbf{e} = (1,1)$ *and let* $\mathbf{f} = (1,-1)$. *Since* $\mathbf{e} \cdot \mathbf{f} = (1 \cdot 1) + (1 \cdot -1) = 1 + (-1) = 0$, *the vectors are orthogonal. Again, we can show that these lines are perpendicular in a graph. (It's a bit hard to graph how they are unrelated; but we could create a new coordinate system for the space in which they are.) There's a moral to this exercise: two vectors can have a product of zero, even though neither of the vectors is zero.*

Finally, dot products have a statistical interpretation. Let's let x and y be two random variables, each with mean zero. We will collect a sample of size N, and we will record the value of x_i and y_i for each observation. We can then construct a vector $\mathbf{x} = (x_1, x_2, \ldots, x_N)$ and a similar vector $\mathbf{y} = (y_1, y_2, \ldots, y_N)$. When we take their dot product, we calculate:

$$\mathbf{x} \cdot \mathbf{y} = \sum_{i=1}^{N} x_i y_i = (N-1) \cdot \hat{Cov}(x, y)$$

The dot product is essentially their (empirical) covariance. Saying that the vectors $\mathbf{x}$ and $\mathbf{y}$ are orthogonal is exactly the same as saying that the variables x and y are uncorrelated.

Similarly, the dot product of a vector with itself is:

$$\mathbf{x} \cdot \mathbf{x} = \sum_{i=1}^{N} x_i^2 = (N-1) \cdot \hat{Var}(x)$$

Here's an unnecessary bit of trivia: if we graph two vectors in N-dimensional space, the angle θ between them must always satisfy:

$$\cos\theta = \frac{\mathbf{x} \cdot \mathbf{y}}{\sqrt{\mathbf{x} \cdot \mathbf{x}}\sqrt{\mathbf{y} \cdot \mathbf{y}}}$$

In the case of these random variables,

$$\cos\theta = \frac{\mathbf{x} \cdot \mathbf{y}}{\sqrt{\mathbf{x} \cdot \mathbf{x}}\sqrt{\mathbf{y} \cdot \mathbf{y}}} = \frac{(N-1)\hat{Cov}(x, y)}{\sqrt{(N-1)\hat{Var}(x)}\sqrt{(N-1)\hat{Var}(y)}} = \hat{Corr}(x, y)$$

The correlation coefficient is the cosine of the angle between the vectors! (Remember that the cosine of two rays is one if they point in exactly the same direction, zero if they are perpendicular, negative one if they point in exactly opposite directions—exactly the same as with correlations.) Coincidence? Not really, on a very deep level, but we don't have to go there.

Now let's move on to matrices. As it turns out, vectors are just special cases of matrices, so there's not much point in discussing them specifically. We used vectors to express a single linear equation, and we will use matrices to present a system of linear equations, like:

$$\begin{aligned}
y_1 &= x_{11}b_1 + x_{12}b_2 + x_{13}b_3 + \ldots + x_{1k}b_k \\
y_2 &= x_{21}b_1 + x_{22}b_2 + x_{23}b_3 + \ldots + x_{2k}b_k \\
y_3 &= x_{31}b_1 + x_{32}b_2 + x_{33}b_3 + \ldots + x_{3k}b_k \\
&\vdots \\
y_n &= x_{n1}b_1 + x_{n2}b_2 + x_{n3}b_3 + \ldots + x_{nk}b_k
\end{aligned}$$

(The subscripts above are two separate numbers. The first line would be read "*y*-one equals *x*-one-one times *a*-one plus *x*-one-two...." A careful person might separate the indices with a comma, to make it clear that $x_{1,1}$ is not *x*-eleven.) Instead of this complicated system of equations, we can represent the vector $\mathbf{y} = (y_1, y_2, \ldots, y_n)$ as the product of an $n \times k$ matrix $\mathbf{X}$ with the vector $\mathbf{b} = (b_1, b_2, \ldots, b_k)$.

A **matrix A** is defined as a collection of $n \times k$ entries arranged into *n* rows and *k* columns. The entry in the *i*-th row and *j*-th column is denoted by a_{ij}:

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1k} \\ a_{21} & a_{22} & \ldots & a_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \ldots & a_{nk} \end{bmatrix}_{n \times k}$$

The elements of a matrix are scalars. A **scalar** is a real number (or a function that takes on a specific value). Tacking a set of dimensions onto the bottom right-hand corner of the matrix always makes it easier to remember the dimensions of that matrix. This is strictly optional. The dimensions are always expressed as *rows* x *columns*. An $n \times k$ matrix is different from an $k \times n$ matrix.

Incidentally, a vector is just a special kind of matrix: it is a matrix with a single column. An *n*-dimensional vector is nothing more or less than an $n \times 1$ matrix.

A spreadsheet containing data is a common example of a matrix. I might have an Excel file with my students' grades:

Student	**Exam 1**	**Exam 2**	**Exam 3**
Ann	90	85	86
Bob	78	62	73
Carl	83	86	91
Doris	92	91	90
Pat	97	98	93

Essentially, I have a 5×3 matrix of grades,

$$\mathbf{G} = \begin{bmatrix} 90 & 85 & 86 \\ 78 & 62 & 73 \\ 83 & 86 & 91 \\ 92 & 91 & 90 \\ 97 & 98 & 93 \end{bmatrix}$$

This is how we usually use matrices in econometrics: to express a collection of data. We will be applying the same formula to each observation in our empirical model (much as I would apply the same formula to calculate the final grade of each student). However, let's just leave this example matrix for now, and study basic matrix operations.

Given an $n \times k$ matrix $\mathbf{A}$ with the entries described as above, the **transpose** of $\mathbf{A}$ is the $k \times n$ matrix $\mathbf{A}'$ (sometimes written as $\mathbf{A}^T$) that results from interchanging the columns and rows of $\mathbf{A}$. That is, the *i*-th column of $\mathbf{A}$ becomes the *i*–th row of $\mathbf{A}'$; the *j*-th row of $\mathbf{A}$ becomes the *j*-th column of $\mathbf{A}'$:

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1k} \\ a_{21} & a_{22} & \dots & a_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nk} \end{bmatrix}_{n \times k} \Rightarrow \mathbf{A}' = \begin{bmatrix} a_{11} & a_{21} & \cdots & a_{n1} \\ a_{12} & a_{22} & \dots & a_{n2} \\ \vdots & \vdots & \ddots & \vdots \\ a_{1k} & a_{2k} & \dots & a_{nk} \end{bmatrix}_{k \times n}$$

Think of this like flipping the matrix on its diagonal.

Example: *With the matrix of grades above,*

$$\mathbf{G} = \begin{bmatrix} 90 & 85 & 86 \\ 78 & 62 & 73 \\ 83 & 86 & 91 \\ 92 & 91 & 90 \\ 97 & 98 & 93 \end{bmatrix} \quad \Rightarrow \quad \mathbf{G}' = \begin{bmatrix} 90 & 78 & 83 & 92 & 97 \\ 85 & 62 & 86 & 91 & 98 \\ 86 & 73 & 91 & 90 & 93 \end{bmatrix}$$

Addition of matrices is fairly straightforward, defined in this manner. Given two matrices A and B that have the same dimension $n \times k$, their sum $A + B$ is also an $n \times k$ matrix, which we obtain by adding elements in the corresponding positions:

$$A + B = \begin{bmatrix} a_{11} + b_{11} & a_{12} + b_{12} & \cdots & a_{1k} + b_{1k} \\ a_{21} + b_{21} & a_{22} + b_{22} & \dots & a_{2k} + b_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} + b_{n1} & a_{n2} + b_{n2} & \dots & a_{nk} + b_{nk} \end{bmatrix}_{n \times k}$$

Not all matrices can be added; their dimensions must be exactly the same. As with addition of scalars (that is, addition as you know it), matrix addition is both commutative and associative; that is, if $\mathbf{A}$ and $\mathbf{B}$ and $\mathbf{C}$ are matrices of the same dimension, then $(\mathbf{A} + \mathbf{B}) + \mathbf{C} = \mathbf{A} + (\mathbf{B} + \mathbf{C})$ and $\mathbf{A} + \mathbf{B} = \mathbf{B} + \mathbf{A}$.

Example: *Let* $\mathbf{D}$ *and* $\mathbf{E}$ *be the matrices below:*

$$\mathbf{D} = \begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 6 & 7 \end{bmatrix}, \quad \mathbf{E} = \begin{bmatrix} 1 & 0 \\ 1 & 1 \\ 0 & 1 \end{bmatrix}$$

Then their sum is the matrix:

$$\mathbf{D} + \mathbf{E} = \begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 6 & 7 \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 1 & 1 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 1+1 & 2+0 \\ 3+1 & 4+1 \\ 6+0 & 7+1 \end{bmatrix} = \begin{bmatrix} 2 & 2 \\ 4 & 5 \\ 6 & 8 \end{bmatrix},$$

Again, matrix addition probably feels very natural. Matrix subtraction is the same.

There are two types of multiplication used with matrices, and the first should also feel natural. This is called **scalar multiplication**: when we multiply an entire matrix by a constant value. If α is some scalar (just a single number), and **B** is an $n \times k$ matrix, then $\alpha\mathbf{B}$ is computed by multiplying each component of **B** by the constant α:

$$\mathbf{B} = \begin{bmatrix} b_{11} & b_{12} & \cdots & b_{1k} \\ b_{21} & b_{22} & \dots & b_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ b_{n1} & b_{n2} & \dots & b_{nk} \end{bmatrix}_{n \times k} \Rightarrow \alpha\mathbf{B} = \begin{bmatrix} \alpha b_{11} & \alpha b_{12} & \cdots & \alpha b_{1k} \\ \alpha b_{21} & \alpha b_{22} & \dots & \alpha b_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ \alpha b_{n1} & \alpha b_{n2} & \dots & \alpha b_{nk} \end{bmatrix}_{n \times k}$$

Scalar multiplication has all the familiar properties: it is distributive, commutative, and associative. That is, $\alpha(\mathbf{A} + \mathbf{B}) = \alpha\mathbf{A} + \alpha\mathbf{B}$, $(\alpha + \beta)\mathbf{A} = (\beta + \alpha)\mathbf{A}$, $\alpha(\beta\mathbf{A}) = (\alpha\beta)\mathbf{A}$, and $\alpha(\beta + \gamma)\mathbf{A} = \alpha\beta\mathbf{A} + \alpha\gamma\mathbf{A}$.

Example: *Use the matrix* **D** *from the previous example. Then* $4\mathbf{D}$ *is the matrix:*

$$4 \cdot \mathbf{D} = \begin{bmatrix} 4 \cdot 1 & 4 \cdot 2 \\ 4 \cdot 3 & 4 \cdot 4 \\ 4 \cdot 6 & 4 \cdot 7 \end{bmatrix} = \begin{bmatrix} 4 & 8 \\ 12 & 16 \\ 24 & 28 \end{bmatrix}$$

Multiplying one matrix by another matrix is more complicated. **Matrix multiplication** is only defined between an $n \times k$ matrix **A** and an $k \times m$ matrix **B**, and *the order matters*. The number of columns in the first must equal the number of rows in the second. Their product is the $n \times m$ matrix **C**, where the *ij*-th element is defined as:

$$c_{ij} = a_{i1}b_{1j} + a_{i2}b_{2j} + \ldots + a_{ik}b_{kj} + \ldots + a_{in}b_{nj}$$

In other words, we take get c_{ij} by taking the dot product of the *i*-th row of **A** and the *j*-th column of **B**:

$$\begin{bmatrix} a_{11} & \cdots & a_{1\ell} & \cdots & a_{1k} \\ \vdots & \ddots & \vdots & \iddots & \vdots \\ \langle \boldsymbol{a_{i1}} & \cdots & \boldsymbol{a_{i\ell}} & \cdots & \boldsymbol{a_{ik}} \rangle \\ \vdots & \iddots & \vdots & \ddots & \vdots \\ a_{m1} & \cdots & a_{n\ell} & \cdots & a_{nk} \end{bmatrix}_{n\times k} \begin{bmatrix} b_{11} & \cdots & \langle \boldsymbol{b_{1j}} \rangle & \cdots & b_{1m} \\ \vdots & \ddots & \vdots & \iddots & \vdots \\ b_{p1} & \cdots & \langle \boldsymbol{b_{pj}} \rangle & \cdots & b_{pm} \\ \vdots & \iddots & \vdots & \ddots & \vdots \\ b_{k1} & \cdots & \langle \boldsymbol{b_{kj}} \rangle & \cdots & b_{km} \end{bmatrix}_{k\times m} = \begin{bmatrix} c_{11} & \cdots & c_{1j} & \cdots & c_{1m} \\ \vdots & \ddots & \vdots & \iddots & \vdots \\ c_{i1} & \cdots & \langle \boldsymbol{c_{ij}} \rangle & \cdots & c_{im} \\ \vdots & \iddots & \vdots & \ddots & \vdots \\ c_{n1} & \cdots & c_{nj} & \cdots & c_{nm} \end{bmatrix}_{n\times m}$$

Notice that multiplying a row of **A** by a column of **B** is unlikely to give you the same answer as multiplying a *column* of **A** by a *row* of **B**. *Matrix multiplication is not commutative:* $\mathbf{AB} \neq \mathbf{BA}$ (except by coincidence, or when both are diagonal matrices of the same dimension). It is very, very important to keep the order right.

Here are two other rules to know about matrix multiplication:

$$\mathbf{AB} = 0 \nRightarrow (\mathbf{A} = 0 \text{ or } \mathbf{B} = 0) \quad \text{and:} \quad \mathbf{AB} = \mathbf{AC} \nRightarrow \mathbf{B} = \mathbf{C}$$

except in special cases. Fortunately, matrix multiplication is still associative and distributive. That is, $\mathbf{A}(\mathbf{BC}) = (\mathbf{AB})\mathbf{C}$ and $\mathbf{A}(\mathbf{B}+\mathbf{C}) = \mathbf{AB}+\mathbf{BC}$. This makes multiplication a bit easier.

Because I find it really hard to remember which column gets multiplied by which row and ends up where, I use this trick to keep everything straight when multiplying matrices. I align the two matrices **A** and **B** so that the second one is above and to the right of the first. For each row i of **A** I trace a line out to the right, and each column j of **B** a line going down, and where these intersect is where their product lies in the matrix **C**. This is like a coordinate system for the c_{ij}.

$$\begin{bmatrix} b_{11} & \cdots & \langle \boldsymbol{b_{1j}} \rangle & \cdots & b_{1p} \\ \vdots & \ddots & \vdots & \iddots & \vdots \\ b_{k1} & \cdots & \langle \boldsymbol{b_{kj}} \rangle & \cdots & b_{kp} \\ \vdots & \iddots & \vdots & \ddots & \vdots \\ b_{n1} & \cdots & \langle \boldsymbol{b_{nj}} \rangle & \cdots & b_{np} \end{bmatrix}$$

$$\begin{bmatrix} a_{11} & \cdots & a_{1k} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots & \iddots & \vdots \\ \langle \boldsymbol{a_{i1}} & \cdots & \boldsymbol{a_{ik}} & \cdots & \boldsymbol{a_{in}} \rangle \\ \vdots & \iddots & \vdots & \ddots & \vdots \\ a_{m1} & \cdots & a_{mk} & \cdots & a_{mn} \end{bmatrix} \begin{bmatrix} c_{11} & \cdots & c_{1j} & \cdots & c_{1p} \\ \vdots & \ddots & \vdots & \iddots & \vdots \\ c_{i1} & \cdots & \langle \boldsymbol{c_{ij}} \rangle & \cdots & c_{ip} \\ \vdots & \iddots & \vdots & \ddots & \vdots \\ c_{m1} & \cdots & c_{mj} & \cdots & c_{mp} \end{bmatrix}$$

I also find this trick very useful for multiplying a bunch of matrices. If we have find the product **ABD** of three matrices, Once I find $\mathbf{C} = \mathbf{AB}$ as above, all I have to do is stick the matrix **D** immediately to the right of **B**, and I have my "coordinate system" for the product of **C** and **D**.

Example: *Let* $\mathbf{F}$ *by a* 2×2 *matrix, and let* $\mathbf{G}$ *be a* 2×2 *matrix, defined below:*

$$\mathbf{F} = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}, \ \mathbf{G} = \begin{bmatrix} 1 & 0 \\ -1 & 2 \end{bmatrix}$$

Then the product $\mathbf{FG}$ *is the* 2×2 *matrix:*

$$\mathbf{FG} = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ -1 & 2 \end{bmatrix} = \begin{bmatrix} 1 \cdot 1 - 1 \cdot 2 & 0 \cdot 1 + 2 \cdot 2 \\ 1 \cdot 3 - 1 \cdot 4 & 0 \cdot 3 + 2 \cdot 4 \end{bmatrix}$$
$$= \begin{bmatrix} 1-2 & 0+2 \\ 3-4 & 0+8 \end{bmatrix} = \begin{bmatrix} -1 & 2 \\ -3 & 8 \end{bmatrix}$$

Example: *Let* $\mathbf{C}$ *by a* 2×3 *matrix, and let* $\mathbf{D}$ *be a* 3×2 *matrix, defined below:*

$$\mathbf{C} = \begin{bmatrix} 1 & 2 & 0 \\ 0 & 3 & -1 \end{bmatrix}, \ \mathbf{D} = \begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 6 & 7 \end{bmatrix}$$

Then the product $\mathbf{CD}$ *is the* 2×2 *matrix:*

$$\mathbf{CD} = \begin{bmatrix} 1 & 2 & 0 \\ 0 & 3 & -1 \end{bmatrix} \begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 6 & 7 \end{bmatrix} = \begin{bmatrix} 1 \cdot 1 + 2 \cdot 3 + 0 \cdot 6 & 1 \cdot 2 + 2 \cdot 4 + 0 \cdot 7 \\ 0 \cdot 1 + 3 \cdot 3 - 1 \cdot 6 & 0 \cdot 2 + 3 \cdot 4 - 1 \cdot 7 \end{bmatrix}$$
$$= \begin{bmatrix} 1+6+0 & 2+8+0 \\ 0+9-6 & 0+12-7 \end{bmatrix} = \begin{bmatrix} 7 & 10 \\ 15 & 5 \end{bmatrix}$$

Now let's talk about some names for special types of matrices. A **square matrix** is one that has the same number of rows as columns; that is, an $n \times n$ matrix. A **diagonal matrix** is a square matrix that has the entry $a_{ij} = 0$ for all $i \neq j$ (in other words, zero everywhere except for the diagonal). For example,

$$\mathbf{A} = \begin{bmatrix} 13 & 0 & 0 & 0 \\ 0 & -7 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 52 \end{bmatrix}$$

is a diagonal matrix. A **symmetric** matrix is one that is the same as its transpose, $\mathbf{A} = \mathbf{A}'$. **Idempotent** matrices ares one that are the same when multiplied by themselves, $\mathbf{A}^2 = \mathbf{AA} = \mathbf{A}$.

The $n \times n$ **identity matrix** (denoted by $\mathbf{I}$ or $\mathbf{I}_n$) is a diagonal matrix with ones on the diagonal (and zeros everywhere else):

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

This has the property that for any $n \times k$ matrix **A**, the product $\mathbf{AI}_k$ equals **A**. In matrix multiplication, it is the analogue of the number one in simple multiplication. (In fact, you could take the position that simple multiplication is just matrix multiplication using 1×1 matrices; the 1×1 identity matrix is just [1].)

We will use the identity matrix to define the matrix equivalent of division. However, we never "divide" matrices; we always "multiply by the inverse". With normal numbers, the "inverse of a" is defined as the number a^{-1} such that $a \cdot a^{-1} = 1 = a^{-1}a$. Most square matrices (but not all) are invertible, and given an $n \times n$ matrix **A**, its **inverse** is the matrix $\mathbf{A}^{-1}$ with the property that:

$$\mathbf{AA}^{-1} = \mathbf{I_n} = \mathbf{A}^{-1}\mathbf{A}$$

Computing inverses of matrices is a major pain in the ass most of the time. Fortunately, we usually only do this in theory; we let Stata calculate it for us the rest of the time. However, you should know that the inverse is *not* obtained by inverting each individual component of the matrix.

Counterexample and Example: *Let* **F** *by a* 2×2 *matrix, and let* **H** *be a* 2×2 *matrix, defined below:*

$$\mathbf{F} = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}, \quad \mathbf{H} = \begin{bmatrix} 1/1 & 1/2 \\ 1/3 & 1/4 \end{bmatrix}$$

H *is not the inverse of* **F***, since the product is not the identity matrix:*

$$\mathbf{FH} = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}\begin{bmatrix} 1/1 & 1/2 \\ 1/3 & 1/4 \end{bmatrix} = \begin{bmatrix} 1/1+2/3 & 1/2+2/4 \\ 3/1+4/3 & 3/2+4/4 \end{bmatrix}$$
$$= \begin{bmatrix} 5/3 & 4/4 \\ 13/3 & 10/4 \end{bmatrix} \neq \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

If we want to compute the inverse of **F***, it is some* 2×2 *matrix of the form:*

$$\mathbf{F}^{-1} = \begin{bmatrix} w & x \\ y & z \end{bmatrix}$$

where we will treat w, x, y, and z as unknowns. This matrix $\mathbf{F}^{-1}$ *has the property that:*

$$\mathbf{F}\mathbf{F}^{-1} = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}\begin{bmatrix} w & x \\ y & z \end{bmatrix} = \begin{bmatrix} 1\cdot w + 2\cdot y & 1\cdot x + 2\cdot z \\ 3\cdot w + 4\cdot y & 3\cdot x + 4\cdot z \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

This gives us a system of four equations in four unknowns:

$$\begin{aligned} 1\cdot w + 2\cdot y &= 1 \\ 1\cdot x + 2\cdot z &= 0 \\ 3\cdot w + 4\cdot y &= 0 \\ 3\cdot x + 4\cdot z &= 1 \end{aligned}$$

We can solve this by iterated substitution. The second equation tells us that $x = -2\cdot z$. *We can plug this into the last equation and get* $3\cdot(-2z) + 4z = 1$, *so* $-2z = 1$, *and* $z = -1/2$. *This means that* $x = -2z = -2(-1/2) = 1$. *Next, we observe from the third equation that* $y = -3/4\,w$. *Plugging this into the first equation, we have* $1w + 2(-3/4\,w) = 1$, *so* $w - 3/2\,w = 1$, $-1/2\,w = 1$, *so* $w = -2$. *This means that* $y = 3/2$. *Putting this altogether, the inverse of* $\mathbf{F}$ *must be:*

$$\mathbf{F}^{-1} = \begin{bmatrix} -2 & 1 \\ 3/2 & -1/2 \end{bmatrix}$$

We can verify this by taking the product,

$$\begin{aligned} \mathbf{F}\mathbf{F}^{-1} &= \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}\begin{bmatrix} -2 & 1 \\ 3/2 & -1/2 \end{bmatrix} = \begin{bmatrix} 1\cdot(-2) + 2\cdot(3/2) & 1\cdot 1 + 2(-1/2) \\ 3\cdot(-2) + 4\cdot(3/2) & 3\cdot 1 + 4(-1/2) \end{bmatrix} \\ &= \begin{bmatrix} -2+3 & 1+1 \\ -6+6 & 3-2 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \end{aligned}$$

Again, computing the inverse of a matrix is a pain—and the 2×2 case is the easiest that it comes! (More generally, with an $n\times n$ matrix you have n^2 equations in n^2 unknowns, so it rapidly gets complicated.) We generally deal with matrix inverses only in theory, so it's important to know some theoretical properties of inverses. I'll add some rules for transposes as well, since they mirror the others:

$$(\mathbf{A}^{-1})^{-1} = \mathbf{A} \qquad (\mathbf{AB})^{-1} = \mathbf{B}^{-1}\mathbf{A}^{-1} \qquad (\mathbf{A}')^{-1} = (\mathbf{A}^{-1})'$$

$$(\mathbf{A}')' = \mathbf{A} \qquad (\mathbf{AB})' = \mathbf{B}'\mathbf{A}' \qquad (\mathbf{A}+\mathbf{B})' = \mathbf{A}' + \mathbf{B}'$$

Note that the order of multiplication changes when passing the transpose or inverse through parentheses. Also, the rule $(\mathbf{AB})^{-1} = \mathbf{B}^{-1}\mathbf{A}^{-1}$ works only when each matrix is a square matrix (otherwise, they don't have individual inverses—but their product might be a square matrix, so it might still have an inverse).

As I mentioned before, not all square matrices are invertible. (The same is true of regular numbers: zero has no inverse.) A square matrix that has no inverse is called a **singular** matrix. Let me give you one example. The 2×2 matrix

$$\mathbf{J} = \begin{bmatrix} 2 & 0 \\ 2 & 0 \end{bmatrix}$$

is not invertible. If it did have an inverse, it would be some matrix of the form:

$$\mathbf{J}^{-1} = \begin{bmatrix} w & x \\ y & z \end{bmatrix}$$

with the property that:

$$\mathbf{J}\mathbf{J}^{-1} = \mathbf{I}_2$$

(That's just the definition of an inverse.) That would mean that:

$$\begin{bmatrix} 2 & 0 \\ 2 & 0 \end{bmatrix}\begin{bmatrix} w & x \\ y & z \end{bmatrix} = \begin{bmatrix} 2w+0y & 2x+0z \\ 2w+0y & 2x+0z \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

This gives the system of equations:

$$\begin{aligned} 2w+0y &= 1 \\ 2w+0y &= 0 \\ 2x+0z &= 0 \\ 2x+0z &= 1 \end{aligned}$$

This cannot possibly have a solution. Look at the first two equations: they are the same on the left-hand side, but equal zero in the first equation and one in the second. For these to be satisfied, we would have to have $1 = 2w+0y = 0$, so $1 = 0$. That's just not possible. $\mathbf{J}$ cannot possibly have an inverse, so it is "singular".

Here are some rules for identifying whether a matrix is singular:

1. *If all of the elements in one row (or column) of the matrix are zero, then the matrix has no inverse.*
2. *If two rows (or two columns) are identical, then the matrix has no inverse.*
3. *If two rows (or two columns) are proportional, then the matrix has no inverse.*
4. *If one row (or one column) can be written as a linear function of some other rows (or of some other columns), then the matrix has no inverse.*

These essentially exhaust all possibilities. We can use the matrix $\mathbf{J}$ as an example of both the first and second cases. The second column of $\mathbf{J}$ is all zeros, so this indicates that the matrix has no inverse. It is also the case that the first and second rows are duplicates, so this also tells that the matrix is not invertible.

As an example of the third case, look at the matrix:

$$\mathbf{K} = \begin{bmatrix} 1 & 2 & 3 \\ 0 & -1 & 1 \\ 4 & 8 & 12 \end{bmatrix}$$

The third row is proportional to the first: you can obtain the third row by multiplying the first by four ($1 \cdot 4 = 4$, $2 \cdot 4 = 8$, $3 \cdot 4 = 12$). This indicates that this matrix will not have an inverse.

Finally, let's look at an example of the fourth case. The matrix:

$$\mathbf{L} = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 2 & 3 \\ 1 & 5 & 7 \end{bmatrix}$$

cannot have an inverse, since the third row can be calculated as a linear function of the other two: Third row = First row + $2 \cdot$(Second row). These rules will be relevant later when we work with data.

Now that we have all the basic terminology, let's return to the example with grades. The original system of equations that I presented,

$$\begin{aligned} y_1 &= x_{11}b_1 + x_{12}b_2 + x_{13}b_3 + \ldots + x_{1k}b_k \\ y_2 &= x_{21}b_1 + x_{22}b_2 + x_{23}b_3 + \ldots + x_{2k}b_k \\ y_3 &= x_{31}b_1 + x_{32}b_2 + x_{33}b_3 + \ldots + x_{3k}b_k \\ &\vdots \\ y_n &= x_{n1}b_1 + x_{n2}b_2 + x_{n3}b_3 + \ldots + x_{nk}b_k \end{aligned}$$

Can be expressed easily through matrix multiplication. On the left-hand side, we have a bunch of y variables. It would be easy to construct an $n \times 1$ matrix (vector, really) $\mathbf{Y} = [y_1 \quad y_2 \quad \ldots \quad y_n]'$. On the right hand side, we have a bunch of x variables, and each is multiplied by the same series of b variables. Let's place all of the b variables into a $k \times 1$ matrix (vector): $\mathbf{b} = [b_1 \quad b_2 \quad \ldots \quad b_k]'$. Finally, let's arrange the x into a matrix:

$$\mathbf{X} = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1k} \\ x_{21} & x_{22} & & x_{2k} \\ \vdots & & \ddots & \\ x_{n1} & x_{n2} & \cdots & x_{nk} \end{bmatrix}$$

Then the expression:

$$\mathbf{Y} = \mathbf{Xb}$$

means precisely the same thing as the system of equations. It's also much more compact, and I feel that it distills the most important idea: the *y*s are equal to the *x*es times the *b*s.

Let's return to the matrix of grades. I have placed all the grades of my students into a 5×3 matrix,

$$\mathbf{G} = \begin{bmatrix} Exam_{Ann1} & Exam_{Ann2} & Exam_{Ann3} \\ Exam_{Bob1} & Exam_{Bob2} & Exam_{Bob3} \\ Exam_{Carl1} & Exam_{Carl2} & Exam_{Carl3} \\ Exam_{Doris1} & Exam_{Doris2} & Exam_{Doris3} \\ Exam_{Pat1} & Exam_{Pat2} & Exam_{Pat3} \end{bmatrix} = \begin{bmatrix} 90 & 85 & 86 \\ 78 & 62 & 73 \\ 83 & 86 & 91 \\ 92 & 91 & 90 \\ 97 & 98 & 93 \end{bmatrix}$$

The number in the *i*-th row and *j*-th column of this matrix represents the score of student *i* on exam *j*. I want to calculate the final average of each student using the formula:

$$Average_i = 0.3 \cdot Exam_{1i} + 0.3 \cdot Exam_{2i} + 0.4 \cdot Exam_{3i}$$

We could construct a 3×1 vector of weights:

$$\mathbf{w} = \begin{bmatrix} 0.3 \\ 0.3 \\ 0.4 \end{bmatrix}$$

Then we compute,

$$\mathbf{A} = \mathbf{Gw}$$

Since this is the product of a 5×3 matrix and a 3×1 matrix, the product is a 5×1 matrix. It contains a value of $Average_i$ for each person, and it is exactly:

$$\mathbf{A} = \begin{bmatrix} Average_{Ann} \\ Average_{Bob} \\ Average_{Carl} \\ Average_{Doris} \\ Average_{Pat} \end{bmatrix} = \begin{bmatrix} 0.3 \cdot Exam_{Ann1} + 0.3 \cdot Exam_{Ann2} + 0.4 \cdot Exam_{Ann3} \\ 0.3 \cdot Exam_{Bob1} + 0.3 \cdot Exam_{Bob2} + 0.4 \cdot Exam_{Bob3} \\ 0.3 \cdot Exam_{Carl1} + 0.3 \cdot Exam_{Carl2} + 0.4 \cdot Exam_{Carl3} \\ 0.3 \cdot Exam_{Doris1} + 0.3 \cdot Exam_{Doris2} + 0.4 \cdot Exam_{Doris3} \\ 0.3 \cdot Exam_{Pat1} + 0.3 \cdot Exam_{Pat2} + 0.4 \cdot Exam_{Pat3} \end{bmatrix} = \mathbf{Gw}$$

The matrix form "$\mathbf{A} = \mathbf{Gw}$" is a much more compact way to express this system of equations.

You can also use matrices to solve other problems. Suppose that you observed the exam grades for a few students in the class, and you observed their final averages. However, you don't know what weights I used to make these calculations, so you want to infer those from the data.

Student	**Exam 1**	**Exam 2**	**Exam 3**	**Average**
Edmund	$Exam_{Ed1}$	$Exam_{Ed2}$	$Exam_{Ed3}$	$Average_{Ed}$
Frances	$Exam_{Fran1}$	$Exam_{Fran2}$	$Exam_{Fran3}$	$Average_{Fran}$
George	$Exam_{George1}$	$Exam_{George2}$	$Exam_{George3}$	$Average_{George}$

You know that a used a formula of the form:

$$Average_i = w_1 \cdot Exam_{i1} + w_2 \cdot Exam_{i2} + w_3 \cdot Exam_{i3}$$

but you don't know the values of W_k. You can create three matrices,

$$\mathbf{A} = \begin{bmatrix} Aver_{Ed} \\ Aver_{Fran} \\ Aver_{George} \end{bmatrix}, \ \mathbf{G} = \begin{bmatrix} Exam_{Ed1} & Exam_{Ed2} & Exam_{Ed3} \\ Exam_{Fran1} & Exam_{Fran2} & Exam_{Fran3} \\ Exam_{George1} & Exam_{George2} & Exam_{George3} \end{bmatrix}, \ \mathbf{w} = \begin{bmatrix} w_1 \\ w_2 \\ w_3 \end{bmatrix}$$

The relationship between exam scores and the final average can be summarized as:

$$\mathbf{A} = \mathbf{Gw}$$

How would you solve for $\mathbf{w}$? This would be simple algebra if we were dealing with normal numbers: divide both sides by **G**. The only difference is that with matrices, we don't divide; we multiply by the inverse. Since **G** is a square matrix, an inverse usually exists:

$$\mathbf{G}^{-1}\mathbf{A} = \mathbf{G}^{-1}\mathbf{Gw}$$

It must be the case that $\mathbf{G}^{-1}\mathbf{G} = \mathbf{I}$, the identity matrix. Anything multiplied by the identity matrix is the same thing again, so $\mathbf{Iw} = \mathbf{w}$. We have a formula for calculating the weights:

$$\mathbf{w} = \mathbf{G}^{-1}\mathbf{A}$$

This formula will always work. And that's essentially what we're always doing in econometrics: trying to guess some "weights" that are attached to variables in determining some outcome. It's not quite as easy as this example, since the equivalent of the **G** matrix is not square (and so it can't be inverted), but it's essentially the same thing.

MATRIX ALGEBRA FOR STATISTICS: PART 2

In econometrics and economics, it's common to have a function that depends on a bunch of variables. (For example, your utility depends on your consumption of a number of different goods.) We could write these functions as:

$$y = f(x_1, x_2, \ldots, x_k)$$

or we could create an k-dimensional vector $\mathbf{x} = [x_1, x_2, \ldots, x_k]'$ to convey the same idea:

$$y = f(\mathbf{x})$$

Regardless, a function of many variables has a bunch of partial derivatives: $\partial f/\partial x_1$, $\partial f/\partial x_2$, ..., $\partial f/\partial x_k$. It is simply to create a $1 \times k$ matrix to contain all of these partial derivatives. We'll call it $\partial f/\partial \mathbf{x}$:

$$\partial f/\partial \mathbf{x} = [\partial f/\partial x_1 \quad \partial f/\partial x_2 \quad \cdots \quad \partial f/\partial x_k]$$

This matrix of first derivatives is sometimes called a **Jacobian matrix.** By convention, we always construct this to be a $1 \times k$ matrix, not an $k \times 1$. If we wanted it to be arranged the other way, we would write either $(\partial f/\partial \mathbf{x})'$ or $\partial f/\partial \mathbf{x}'$. It is worth knowing that:

$$(\partial f/\partial \mathbf{x})' = \partial f/\partial \mathbf{x}'$$

Anyhow, when we have a system of equations,

$$\begin{aligned} y_1 &= f_1(x_1, x_2, \ldots, x_k) \\ y_2 &= f_2(x_1, x_2, \ldots, x_k) \\ &\vdots \\ y_n &= f_n(x_1, x_2, \ldots, x_k) \end{aligned}$$

we can represent this using a vector $\mathbf{y} = [y_1, y_2, \ldots, y_n]'$, another vector $\mathbf{x} = [x_1, x_2, \ldots, x_k]'$, and a **vector-valued function** $\mathbf{f}(\cdot) = [f_1(\cdot), f_2(\cdot), \ldots, f_n(\cdot)]'$:

$$\mathbf{y} = \mathbf{f}(\mathbf{x})$$

Again, that represents precisely the same thing as the system of equations. When we take the matrix of first derivatives, we write $\partial \mathbf{f}/\partial \mathbf{x}$ to represent:

$$\partial \mathbf{f}/\partial \mathbf{x} = \begin{bmatrix} \partial f_1/\partial x_1 & \partial f_1/\partial x_2 & \cdots & \partial f_1/\partial x_k \\ \partial f_2/\partial x_1 & \partial f_2/\partial x_2 & & \partial f_2/\partial x_k \\ \vdots & & \ddots & \\ \partial f_n/\partial x_1 & \partial f_n/\partial x_2 & & \partial f_n/\partial x_k \end{bmatrix}$$

This is an $n \times k$ matrix. As is often the case with matrices, it might be best to think of this abstractly: this $\partial \mathbf{f}/\partial \mathbf{x}$ is simply a matrix that contains all of the derivatives of each of the *f* functions with respect to each of the *x* variables.

So now, let's work on taking some derivatives of some matrices. We need to learn only two special cases: linear functions and quadratic functions. Let's start with the simple system:

$$y_1 = 2 \cdot x_1 + 3 \cdot x_2$$
$$y_2 = 4 \cdot x_1 + 5 \cdot x_2$$

In each case, *y* is a linear function of the *x* variables. We know that we can write this system of equations in matrix form as:

$$\mathbf{y} = \mathbf{A}\mathbf{x}$$

where $\mathbf{y} = [y_1 \quad y_2]'$, $\mathbf{x} = [x_1 \quad x_2]'$, and

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

As a note, all systems of linear equations (regardless of how many *y*s you have, or how many *x*es you have) can be written in the form $\mathbf{y} = \mathbf{A}\mathbf{x}$ for some matrix $\mathbf{A}$. Anyhow, we now want to calculate the matrix of derivatives:

$$\frac{\partial \mathbf{y}}{\partial \mathbf{x}} = \begin{bmatrix} \partial y_1/\partial x_1 & \partial y_1/\partial x_2 \\ \partial y_2/\partial x_1 & \partial y_2/\partial x_2 \end{bmatrix}$$

These partial derivatives are simple, since the functions are linear.

$$\frac{\partial \mathbf{y}}{\partial \mathbf{x}} = \begin{bmatrix} \partial y_1/\partial x_1 & \partial y_1/\partial x_2 \\ \partial y_2/\partial x_1 & \partial y_2/\partial x_2 \end{bmatrix} = \begin{bmatrix} 2 & 3 \\ 4 & 5 \end{bmatrix}$$

which turns out to be the same as the original matrix **A**. So here's what you need to remember about taking derivatives of linear functions (along with the analogues in scalar calculus):

Type of linear function:	Scalar	Matrix
Always takes the form:	$y = ax$	$\mathbf{y} = \mathbf{A}\mathbf{x}$
Derivative:	$\partial y/\partial x = a$	$\partial \mathbf{y}/\partial \mathbf{x} = \mathbf{A}$

The rule is exactly the same, really.

Next, we'll do quadratic functions. A quadratic function of a single variable generally looks like $y = ax^2$. With two variables, it takes the form:

$$y = ax_1^2 + 2bx_1x_2 + cx_2^2$$

Of course, we could expand this to contain as many x variables as we like, but two are enough to give it a matrix representation:

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

where $\mathbf{x} = [x_1 \quad x_2]'$, and

$$\mathbf{A} = \begin{bmatrix} a & b \\ b & c \end{bmatrix}.$$

With a quadratic function of a single variable, we write $y = ax^2$ to denote that it is the product of x, x again, and a constant a. The same thing is true with the matrix representation of a quadratic function of several x variables: it is the product of $\mathbf{x}$, $\mathbf{x}$ again, and a constant matrix $\mathbf{A}$. However, the order of the multiplication is important with matrices, so $\mathbf{x}'\mathbf{A}\mathbf{x} \neq \mathbf{A}\mathbf{x}^2$ (you can't calculate the latter, since $\mathbf{x}$ can't be multiplied by $\mathbf{x}$ directly).

Anyhow, if the function is $y = \mathbf{x}'\mathbf{A}\mathbf{x} = ax_1^2 + 2bx_1x_2 + cx_2^2$, what is $\partial y/\partial \mathbf{x}$?

$$\frac{\partial y}{\partial \mathbf{x}} = \begin{bmatrix} \frac{\partial y}{\partial x_1} & \frac{\partial y}{\partial x_2} \end{bmatrix} = \begin{bmatrix} 2ax_1 + 2bx_2 & 2bx_1 + 2cx_2 \end{bmatrix}$$

According to scalar multiplication, the last matrix can be written:

$$\begin{bmatrix} 2ax_1 + 2bx_2 & 2bx_1 + 2cx_2 \end{bmatrix} = 2\begin{bmatrix} ax_1 + bx_2 & bx_1 + cx_2 \end{bmatrix}$$

As it turns out, the matrix on the right hand side is simply $\mathbf{x}'\mathbf{A}$. What you have is that $\partial y/\partial \mathbf{x} = 2\mathbf{x}'\mathbf{A}$. Again, let me present the rules for quadratic matrices next to the scalar analogues:

Type of quadratic function:	Scalar	Matrix
Always takes the form:	$y = ax^2$	$\mathbf{y} = \mathbf{x}'\mathbf{A}\mathbf{x}$
Derivative:	$\partial y/\partial x = 2ax$	$\partial \mathbf{y}/\partial \mathbf{x} = 2\mathbf{x}'\mathbf{A}$

And that's all there is to know about taking derivatives when matrices are involved.

The last thing on the agenda is to work out a problem, which is the basis of econometrics. We believe that there is some outcome y that is determined (in part) by some variables $x_2, x_3, \ldots, x_k$. (As a matter of convention, we start counting with variable number two; I will explain x_1 later.) We have a sample of N observations;

the letter *i* will denote a generic member of this sample. For each member of the sample, we believe that the relationship can be described:

$$y_i = \beta_1 + \beta_2 x_{i2} + \beta_3 x_{i3} + \ldots + \beta_K x_{iK} + e_i$$

e_i represents the portion of y_i that is determined by variables other than the *x*es. For example, we might think that a worker's "annual earnings" (that's the y) are determined by his education and work experience (those are the *x*es) among other things (those other factors are lumped together in *e*). This relationship is the same for everyone—the βs are all the same. We are assuming that the reason that people have different earnings is because their levels of education and experience (and other factors) differ, although the returns to education and experience are the same for everyone. The problem is that we don't know these returns, so we want to "guess" them from the data.

The relationship holds for each observation in our sample. In other words,

$$\begin{aligned} y_1 &= \beta_1 + \beta_2 x_{12} + \beta_3 x_{13} + \ldots + \beta_K x_{1K} + e_1 \\ y_2 &= \beta_1 + \beta_2 x_{22} + \beta_3 x_{23} + \ldots + \beta_K x_{2K} + e_2 \\ &\vdots \\ y_N &= \beta_1 + \beta_2 x_{N2} + \beta_3 x_{N3} + \ldots + \beta_K x_{NK} + e_N \end{aligned}$$

We know how to represent this system of equations in compact matrix notation. We can construct a vector $\mathbf{y} = [y_1 \quad y_2 \quad \cdots \quad y_N]'_{N\times 1}$ containing all of the y values for our sample. We can construct another vector $\boldsymbol{\beta} = [\beta_1 \quad \beta_2 \quad \cdots \quad \beta_K]'_{K\times 1}$ containing all of the β coefficients. There is a little trick in constructing the matrix $\mathbf{X}$: it will contain all the variables $x_2, x_3, \ldots, x_k$ for all the people; in addition, we will have $x_1 = 1$ for everyone.

$$\mathbf{X} = \begin{bmatrix} 1 & x_{12} & x_{13} & \cdots & x_{1K} \\ 1 & x_{22} & x_{23} & & x_{2K} \\ \vdots & & & \ddots & \vdots \\ 1 & x_{N2} & x_{N3} & \cdots & x_{NK} \end{bmatrix}_{N\times K}$$

Finally, we can construct a vector $\mathbf{e} = [e_1 \quad e_2 \quad \cdots \quad e_N]'_{N\times 1}$ containing all of the "unobservable" determinants of the outcome y. The system of equations can be represented as:

$$\mathbf{y}_{N\times 1} = \mathbf{X}_{N\times K}\boldsymbol{\beta}_{K\times 1} + \mathbf{e}_{N\times 1}$$

I've added subscripts to the matrices to ensure that all the operations (multiplication and addition) conform. An $N \times K$ matrix can be multiplied by a $K \times 1$ vector; the product $\mathbf{X}\boldsymbol{\beta}$ is then an $N \times 1$. This can be added to the $N \times 1$ vector $\mathbf{e}$; the sum $\mathbf{y}$ should also be $N \times 1$. Everything checks out.

This equation describes the *true* relationship between x and y. However, we don't know this true relationship. We will make some guess as to the value of the coefficients. Let us denote an (arbitrary) guess as $\hat{\boldsymbol{\beta}}$. Now we will write the *econometric* model as:

$$\mathbf{y}_{N\times 1} = \mathbf{X}_{N\times K}\hat{\boldsymbol{\beta}}_{K\times 1} + \hat{\mathbf{e}}_{N\times 1}$$

The last term, $\hat{\mathbf{e}}$, is called the **residual** vector. It is the portion of the outcome that we the econometricians cannot explain. $\mathbf{e}$ is the portion of the outcome that in truth cannot be explained, but we don't know what this is (it is unobservable, after all!). We do know $\hat{\mathbf{e}}$, however. Once we pick $\hat{\boldsymbol{\beta}}$, $\hat{\mathbf{e}}$ is defined as:

$$\hat{\mathbf{e}}_{N\times 1} = \mathbf{y}_{N\times 1} - \mathbf{X}_{N\times K}\hat{\boldsymbol{\beta}}_{K\times 1}$$

The true relationship and the econometric model are analogues. $\boldsymbol{\beta}$ is the true vector of coefficients; $\hat{\boldsymbol{\beta}}$ is our guess of the vector of coefficients. $\mathbf{e}$ is the true effect of other factors not in the model; $\hat{\mathbf{e}}$ is our guess of the effect of other factors. Therefore, if we have guessed $\hat{\boldsymbol{\beta}}$ exactly right, then we have guessed $\hat{\mathbf{e}}$ exactly right; if we get $\hat{\boldsymbol{\beta}}$ close to the true value, then our $\hat{\mathbf{e}}$ are close to the true values.

What we would like to do is to come up with a guess $\hat{\boldsymbol{\beta}}$ that minimizes the unexplained portion; that is, we want to pick $\hat{\boldsymbol{\beta}}$ to minimize the "size" of the vector $\hat{\mathbf{e}}$. The size (or norm) of the vector $\hat{\mathbf{e}}$ is defined as:

$$\|\hat{\mathbf{e}}\| = \sqrt{\hat{e}_1^2 + \hat{e}_2^2 + \ldots + \hat{e}_N^2} = \sqrt{\hat{\mathbf{e}}'\hat{\mathbf{e}}}$$

We can forget about the square root, though, since it doesn't change the solution to the minimization problem. The bottom line is that we want to pick $\hat{\boldsymbol{\beta}}$ to

$$\min\left(\hat{e}_1^2 + \hat{e}_2^2 + \ldots + \hat{e}_N^2\right) = \min \Sigma_{i=1}^N \hat{e}_i^2 = \min \hat{\mathbf{e}}'\hat{\mathbf{e}}$$

The rest is algebra. First, we'll simplify the expression $\hat{\mathbf{e}}'\hat{\mathbf{e}}$; then we'll minimize it; finally, we'll solve that for the optimal $\hat{\boldsymbol{\beta}}$, which we'll call $\hat{\boldsymbol{\beta}}_{OLS}$: the **ordinary least squares** estimator of $\boldsymbol{\beta}$.

We want to minimize the following expression:

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

Let's transpose the first matrix:

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

Now let's multiply through:

$$\min \hat{\mathbf{e}}'\hat{\mathbf{e}} = \min(\mathbf{y}'\mathbf{y} - \mathbf{y}'\mathbf{X}\hat{\boldsymbol{\beta}} - \hat{\boldsymbol{\beta}}'\mathbf{X}'\mathbf{y} + \hat{\boldsymbol{\beta}}'\mathbf{X}'\mathbf{X}\hat{\boldsymbol{\beta}})$$

To minimize the problem, we take the derivative with respect to $\hat{\boldsymbol{\beta}}$, and set that equal to zero (or really, a vector of zeros). The optimal $\hat{\boldsymbol{\beta}}_{OLS}$ must solve:

$$-\mathbf{y}'\mathbf{X} - (\mathbf{X}'\mathbf{y})' + 2\hat{\boldsymbol{\beta}}'_{OLS}\mathbf{X}'\mathbf{X} = \mathbf{0}$$

Now let's simplify and solve for $\hat{\boldsymbol{\beta}}_{OLS}$. Transposing the second term, we have:

$$-\mathbf{y}'\mathbf{X} - \mathbf{y}'\mathbf{X} + 2\hat{\boldsymbol{\beta}}'_{OLS}\mathbf{X}'\mathbf{X} = \mathbf{0}$$

We can group the first two terms together, and then move them to the right-hand side of the equation:

$$2\hat{\boldsymbol{\beta}}'_{OLS}\mathbf{X}'\mathbf{X} = 2\mathbf{y}'\mathbf{X}$$

Then we can cancel out the 2s. We want to solve for $\hat{\boldsymbol{\beta}}$, but this expression contains $\hat{\boldsymbol{\beta}}'$ instead. We can easily solve this problem by transposing everything on both sides of the equation:

$$(\hat{\boldsymbol{\beta}}'_{OLS}\mathbf{X}'\mathbf{X})' = (\mathbf{y}'\mathbf{X})'$$
$$\mathbf{X}'\mathbf{X}\hat{\boldsymbol{\beta}}_{OLS} = \mathbf{X}'\mathbf{y}$$

Finally, we need to isolate $\hat{\boldsymbol{\beta}}$. If we were working with scalars, we would divide both sides by the stuff in front of $\hat{\boldsymbol{\beta}}$; since these are matrices, we multiply by the inverse instead (provided that $\mathbf{X}'\mathbf{X}$ is in fact invertible, which we will assume for now).

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

Finally, $(\mathbf{X}'\mathbf{X})^{-1}(\mathbf{X}'\mathbf{X}) = \mathbf{I}$, the identity matrix; $\mathbf{I}\hat{\boldsymbol{\beta}}_{OLS} = \hat{\boldsymbol{\beta}}_{OLS}$. We have our formula for the estimator:

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

That's it. Remember how in the univariate regression model, $\hat{\beta}_{2OLS} = \hat{Cov}(x,y)/\hat{Var}(x)$? Here we have essentially the same thing, except that it's greatly generalized!

However, it isn't that simple in reality. $\mathbf{X}'\mathbf{X}$ is actually a very complex matrix, which contains all of the variances and covariances between the *x* variables. Its inverse is not simply the reciprocal of these terms; it is something much more complex.

Nonetheless, in an abstract sense, it *is* exactly the same.