Chapter 2
An introduction to matrix algebra and regression.

Julius van der Werf
Brian Kinghorn
University of New England, Armidale, Australia

Matrices take a lot of the tedium out of both presenting algebra and calculating results. They are widely used in scientific animal breeding, particularly in selection indices and BLUP.

A matrix is a rectangular array with dimensions Rows x Columns.

For example:

$$A = \begin{pmatrix} 4 & 6 \\ 7 & 3 \\ 2 & 1 \end{pmatrix}$$

is a 3 (rows) x 2 (columns) matrix. Element a_{ij} has value 6. Note the convention of using a small letter for elements, and subscripts denoting row and column, in that order.

$$B = \begin{pmatrix} 4 \\ 7 \\ 9 \end{pmatrix} \quad \text{and} \quad C = \begin{pmatrix} 8 & 3 & 14 \end{pmatrix}$$

are matrices which can be referred to as vectors.

B is a column vector and C is a row vector, both of length 3.

Matrices of equal dimensions can be added and subtracted

Weaning weight, Kg	Blue Angus	Leanford	Meatmaker
High Nutrition	120	140	150
Low Nutrition	90	100	105

Yearling weight, Kg	Blue Angus	Leanford	Meatmaker
High Nutrition	260	290	320
Low Nutrition	220	250	275

These hypothetical data can be represented in Nutrition x Breed matrices. Note the special meaning that element locations have - they indicate breed and nutrition:

$$W = \begin{pmatrix} 120 & 140 & 150 \\ 90 & 100 & 105 \end{pmatrix} \quad Y = \begin{pmatrix} 260 & 290 & 320 \\ 220 & 250 & 275 \end{pmatrix} \quad G = \begin{pmatrix} 140 & 150 & 170 \\ 130 & 150 & 170 \end{pmatrix}$$

Note that the matrix of Growth (Y - W = G) is of the same dimension as the others, and is simply got by subtracting elements of W from corresponding elements of Y

Matrices can be multiplied by a scalar - a simple constant:

For example, to express G in Lbs rather that KG, multiply by 2.2:

$$G_{Lbs} = 2.2\begin{pmatrix} 140 & 150 & 170 \\ 130 & 150 & 170 \end{pmatrix} = \begin{pmatrix} 308 & 330 & 374 \\ 286 & 330 & 374 \end{pmatrix}$$

Matrix multiplication:

In this hypothetical example we have information to make two matrices:

M: a matrix of merit for breeds (X, Y and Z) by traits (Body weight and backfat).
P: a matrix of dollars per unit for traits (Body weight and Backfat) by markets (Domestic and Export).

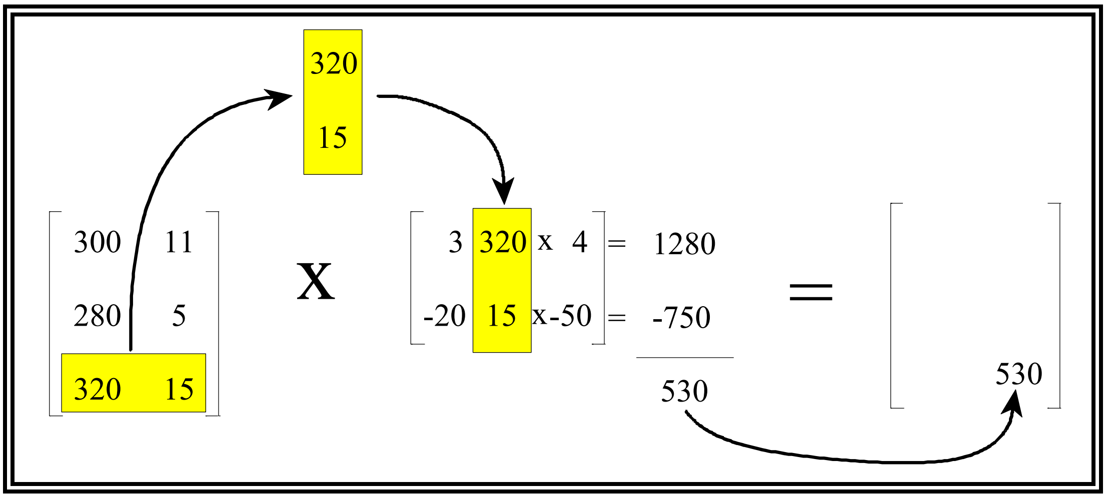
And the product of these matrices will be:
R: a matrix of dollars per head for Breeds (X, Y and Z) by markets (Domestic and Export).

M		P		R
MERIT MATRIX	**x**	**PRICE MATRIX**	**=**	**RETURNS MATRIX**

$$\begin{array}{l c} & \begin{array}{cc} \text{Wt.} & \text{Fat} \end{array} \\ \begin{array}{l} \text{Breed X} \\ \text{Breed Y} \\ \text{Breed Z} \end{array} & \begin{pmatrix} 300 & 11 \\ 280 & 5 \\ 320 & 15 \end{pmatrix} \end{array} \times \begin{array}{l c} & \begin{array}{cc} \text{Dom.} & \text{Exp.} \end{array} \\ \begin{array}{l} \text{Wt.} \\ \text{Fat} \end{array} & \begin{pmatrix} 3 & 4 \\ -20 & -50 \end{pmatrix} \end{array} = \begin{array}{l c} & \begin{array}{cc} \text{Dom.} & \text{Exp.} \end{array} \\ \begin{array}{l} \text{Breed X} \\ \text{Breed Y} \\ \text{Breed Z} \end{array} & \begin{pmatrix} 680 & 650 \\ 740 & 870 \\ 660 & 530 \end{pmatrix} \end{array}$$

Breeds** x Traits* **x** *Traits **x Markets = ***Breeds x Markets***

First note that the number of columns of M (= traits, 2) must equal the number of rows of P (also traits, 2) in order to be able to multiply. The following shows calculation of $r_{3,2}$



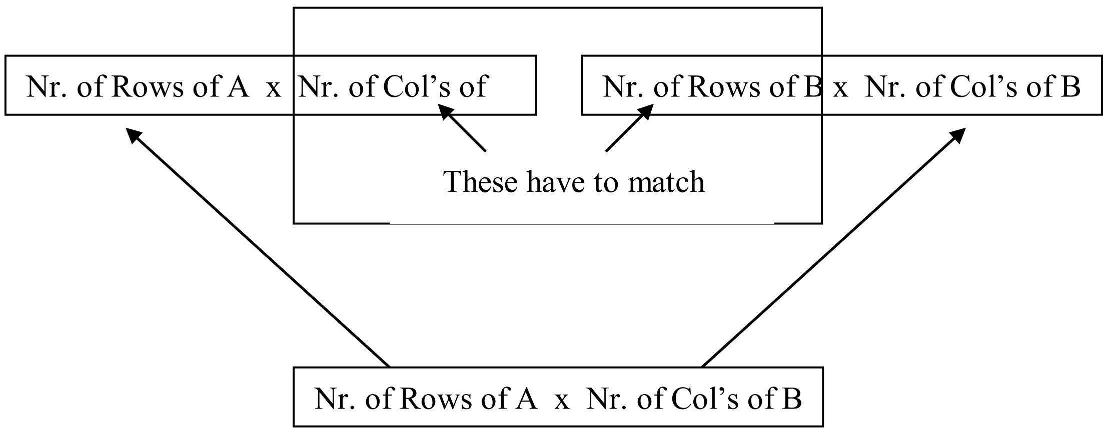
Matrix multiplication: Here is the same information in words:	
To calculate the elements of R:	The $(i,j)^{th}$ (row, column) of R is the sum of the products of the elements of the i^{th} row of M and the j^{th} column of P
For example, $r_{3,2}$:	$R_{3,2} = m_{3,1} \times p_{1,2} + m_{3,2} \times p_{2,2}$ $530 = 320 \times 4 + 15 \times -50$

Another example:

$$\begin{pmatrix} 5 & 3 \\ 4 & 2 \\ 1 & 5 \end{pmatrix} \begin{pmatrix} 7 \\ 3 \end{pmatrix} = \begin{pmatrix} 5x7 + 3x3 \\ 4x7 + 2x3 \\ 1x7 + 5x3 \end{pmatrix} = \begin{pmatrix} 44 \\ 34 \\ 22 \end{pmatrix}$$

For matrix multiplication to be legal, the first matrix must have as many columns as the second matrix has rows. This, of course, is also the requirement for multiplying a row vector by a column vector. The resulting matrix will have as many rows as the first matrix and as many columns as the second matrix. Because **A** has 2 rows and 3 columns while **B** has 3 rows and 2 columns, the matrix multiplication may legally proceed and the resulting matrix will have 2 rows and 2 columns.

Matrix A x Matrix B



Dimension of resulting matrix

Because of these requirements, matrix multiplication is usually not commutative. That is, usually **AB** ≠ **BA**. And even if **AB** is a legal operation, there is no guarantee that **BA** will also be legal. For these reasons, the terms **premultiply** and **postmultiply** are often encountered in matrix algebra while they are seldom encountered in scalar algebra.

Whenever you propose matrix multiplications, make sure they 'fit'

One special case to be aware of is when a column vector is postmultiplied by a row vector. In this case, one simply follows the rules given above for the multiplication of two matrices. Note that the first matrix has one column and the second matrix has one row, so the matrix multiplication is legal. The resulting matrix will have as many rows as the first matrix (3) and as many columns as the second matrix (2).

$$\begin{pmatrix} 3 \\ 2 \\ 5 \end{pmatrix} \begin{pmatrix} 3 & 4 \end{pmatrix} = \begin{pmatrix} 9 & 12 \\ 6 & 8 \\ 15 & 20 \end{pmatrix}$$

Similarly, multiplication of a matrix times a vector (or a vector times a matrix) will also conform to the multiplication of two matrices. For example,

$$\begin{pmatrix} 9 & 12 \\ 6 & 8 \\ 15 & 20 \end{pmatrix} \begin{pmatrix} 3 \\ 5 \\ 2 \end{pmatrix}$$

is an illegal operation because the number of columns in the first matrix (2) does not match the number of rows in the second matrix (3).

There are a couple of identities worth noting:

- Matrix multiplication is not commutative: AB $\neq$ BA.
- Matrix multiplication is associative. In other words: (AB)C = A(BC)
- Matrix multiplication is distributive. In other words: A(B + C) = AB + AC
- Scalar multiplication commutative, associative, and distributive.

There are a couple of examples that are worth looking let. Let us define the column vector **e**. By definition, the order of **e** is (N, 1). We can take the *inner product* of **e**, which is simply:

$$e'e = [e_1 \quad e_2 \cdots e_N] \begin{bmatrix} e_1 \\ e_2 \\ \vdots \\ e_N \end{bmatrix} = e_1e_1 + e_2e_2 + \cdots + e_Ne_N = \sum_{i=1}^{N} e_i^2$$

The inner product of a column vector with itself is simply equal to the sum of the square values of the vector, which is used quite often in the regression model. Geometrically, the square root of the inner product is the length of the vector. One can similarly define the outer product for column vector **e**, denoted **ee**$'$ which yields an order (N,N) matrix. There are couple of other vector products that are interesting to note. Let **i** denote an order (N, 1) vector of ones, and **x** denote an order (N, 1) vector of data. The following is an interesting quantity:

$$\frac{1}{N} i'x = \frac{1}{N}(x_1, \ldots, x_n) = \frac{1}{N}\sum_{i=1}^{n} x_i = \bar{x}$$ is the mean of all x_i

From this, it follows that: $i'x = \sum_{i=1}^{n} x_i$ is the sum of all x_i

Similarly, let **y** denote another (N, 1) vector of data. The following is also interesting:

$\mathbf{x'y} = x_1 y_1 + x_2 y_2 + \cdots + x_N y_N = \sum_{i=1}^{n} x_i y_i$ is the crossproduct of all $x_i y_i$

The identity matrix, I.

The number 1 is quite special, in that if you multiple any number by 1 that number retains its identity - it is not changed.

The same property holds for the identity matrix , which is a square matrix. There is not just one identity matrix, but one for each size, populated with zeros, except for the 'leading diagonal' (top left to bottom right) which contains one's.

You can check that the following are in fact true:

$$\text{IA} = \text{A} \qquad \underset{2\text{x}2}{\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}} \text{ x } \underset{2\text{x}3}{\begin{pmatrix} 4 & 6 & 7 \\ 3 & 2 & 1 \end{pmatrix}} = \underset{2\text{x}3}{\begin{pmatrix} 4 & 6 & 7 \\ 3 & 2 & 1 \end{pmatrix}}$$

$$\text{AI} = \text{A} \qquad \underset{2\text{x}3}{\begin{pmatrix} 4 & 6 & 7 \\ 3 & 2 & 1 \end{pmatrix}} \text{ x } \underset{3\text{x}3}{\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}} = \underset{2\text{x}3}{\begin{pmatrix} 4 & 6 & 7 \\ 3 & 2 & 1 \end{pmatrix}}$$

Note, that a scalar multiplied by an identity matrix becomes a diagonal matrix with the scalars on the diagonal.

Diagonal matrix

A diagonal matrix has only non-zero elements on its diagonal,

For example $\begin{pmatrix} 2.45 & 0 & 0 \\ 0 & 1.71 & 0 \\ 0 & 0 & 1.69 \end{pmatrix}$

Transpose - pivot the matrix about the top left element

$$b = \begin{pmatrix} x \\ y \end{pmatrix} \text{ gives } b' = \begin{pmatrix} x & y \end{pmatrix} \text{ "b transpose"}$$

The transpose of a matrix is denoted by a prime (′): **A'** or a superscript t or T ($\mathbf{A}^t$ or $\mathbf{A}^T$).

$$\begin{pmatrix} 4 & 6 \\ 7 & 3 \\ 2 & 1 \end{pmatrix}' = \begin{pmatrix} 4 & 7 & 2 \\ 6 & 3 & 1 \end{pmatrix} \qquad \text{Note that } a'_{i,j} = a_{j,i}$$

The transpose of a product takes an interesting form: $(AB)' = B'A'$

Symmetrical matrix

A matrix is symmetrical if A = A'.
A symmetrical matrix has to be also a squared matrix (equal numbers of rows and columns)

Matrix inversion

Scalar: $X^{-1} = {}^{1}/_{x}$
Only square matrices can be inverted. We do this in order to achieve matrix division – we just multiply by the reciprocal, or inverse! Just as 20/5 = 4, we have $20 \times 5^{-1} = 4$
The inverse of a matrix is denoted by the superscript "-1"

Inverse of a 2 x 2 Matrix: $$\begin{pmatrix} a & b \\ c & d \end{pmatrix}^{-1} = \frac{1}{ad - bc} \begin{pmatrix} d & -b \\ -c & a \end{pmatrix}$$

For matrices larger that 2 x 2, inversion is quite tedious, and best left to a computer!

Exercise: Just as $X \cdot {}^{1}/_{x} = 1$ show that for matrices, $X X^{-1} = I$, the identity matrix.

In scalar algebra, the inverse of a number is that number which, when multiplied by the original number, gives a product of 1. Hence, the inverse of x is simple $1/x$. or, in slightly different notation, x^{-1} In matrix algebra, the inverse of a matrix is that matrix which, when multiplied by the original matrix, gives an identity matrix. Hence,
$\mathbf{AA}^{-1} = \mathbf{A}^{-1}\mathbf{A} = \mathbf{I}$
A matrix must be square to have an inverse, but not all square matrices have an inverse. In some cases, the inverse does not exist, that is, when the determinant equals zero (see below).
For covariance and correlation matrices, an inverse will always exist, provided that there are more subjects than there are variables and that every variable has a variance greater than 0.

Determinant of a Matrix:

The determinant of a matrix is a scalar and is denoted as |**A**| or det(**A**). The determinant has very important mathematical properties, but it is very difficult to provide a substantive definition. For covariance and correlation matrices, the determinant is a number that is sometimes used to express the “generalized variance” of the matrix. That is, covariance matrices with small determinants denote variables that are redundant or highly correlated. Matrices with large determinants denote variables that are independent of one another. The determinant has several very important properties for some multivariate stats (e.g., change in R^2 in multiple regression can be expressed as a ratio of determinants.) The determinant is tedious to obtain, using similar calculations as those used for calculating an inverse.

Determinant of a 2 x 2 Matrix: $$\begin{pmatrix} a & b \\ c & d \end{pmatrix} = \frac{1}{ad - bc}$$

In linear models we use often ‘incidence matrices’ and these can have redundancies if different effects together explain the same thing. In that case, coefficient matrices that are formed can not be inverted because the determinant is equal to zero. We call such redundant matrices ‘singular’

Matrix singularity and matrix rank

Matrices are often used to defined and solve a set of equations (see next section). A set of equations can only be solved if the number of equations is more or equal to the number of parameters to solve for. An important condition is that these equations need to be non-redundant or independent. For example, if we have 2x + 3y = 9 and 4x + 6y = 18, we still can’t solve for x and y.

To solve a system of equations, we need to premultiply: Ax = y would give $A^{-1}Ax = A^{-1}y$ would give $x = A^{-1}y$.

Hence, basically we use a same idea as division in scalar algebra: 4x=8 → x=8/4.

Now, we can only solve such a system of equations if the inverse of *coefficient matrix* A exists. For this, the determinant has to be non-zero.

For what sorts of matrices is this a problem? It can be shown that matrices that have rows or columns that are *linearly dependent* on other rows or columns have determinants that are equal to zero. For these matrices, the determinant is undefined. We are given an order (*k*, *k*) matrix **A**, and denote this by using column vectors:

$\mathbf{A} = [\mathbf{a_1}\ \mathbf{a_1} \cdots \mathbf{a_k}]$

Each of the vectors $\mathbf{a_i}$ is of order (*k*, 1). A column $\mathbf{a_1}$ of **A** is said to be *linearly independent* of the others if there exists no set of scalars a_j such that:

$$a_i = \sum_{j \neq i}^{n} \alpha_j a_j$$

Thus, given the rest of the columns, if we cannot find a weighted sum to get the column we are interested in, we say the matrix is linearly independent.

We can define the term *rank*. The *rank* of a matrix is defined as the number of linearly independent columns (or rows) of a matrix. If all of the columns are independent, we say that the matrix is of *full rank*. We denote the rank of a matrix as *r*(**A**). By definition, *r*(**A**) is an integer that can take values from 1 to *k*. This is something that can be computed by software packages.

Some important things to remember:

- For an inverse to exist, **A** must be square. This is a necessary, but not sufficient, condition
- If the inverse of a matrix does not exist, we say that it is *singular*.

- The following statements are equivalent: full rank <> nonsingular <> invertable.
- If the determinant of **A** equals zero, then **A** is said to be singular, or not invertable.
- More generally, if $|A| \neq 0$ then A singular.
- If the determinant of **A** is non-zero, then **A** is said to be nonsingular, or invertable. In other words, the inverse exists. More generally, If $|\mathbf{A}| = \mathbf{0}$ then **A** nonsingular.
- If a matrix **A** is not of full rank, it is not invertable; i.e., it is singular.

- $\mathbf{AA}^{-1} = \mathbf{I}$ $\quad$ $\mathbf{A}^{-1}\mathbf{A} = \mathbf{I}$ $\quad$ $\mathbf{A}^{-1}$ is unique.
- $(\mathbf{A}^{-1})^{-1} = \mathbf{A}$. In words, the inverse of an inverse is the original matrix.
- Just as with transposition, it can be shown that $(\mathbf{AB})^{-1} = \mathbf{B}^{-1}\mathbf{A}^{-1}$
- One can also show that the inverse of the transpose is the transpose of the inverse.
- Symbolically, $(\mathbf{A}')^{-1} = (\mathbf{A}^{-1})'$

Generalized Inverse

For matrices not of full rank, an inverse does not exist. If matrix A has order M x N, the maximum possible order is n. The generalized inverse (g-inverse) G of matrix A is such that

$$AGA = A.$$

The generalized inverse is not unique. G is also not necessarily symmetric.
Whereas a normal inverse is usually written as A^{-1}, a generalized inverse has the notation A^{-}.

Trace of a Matrix:

The trace of a matrix is sometimes, although not always, denoted as tr(**A**). The trace is used only for square matrices and equals the sum of the diagonal elements of the matrix. For example,

$$tr\begin{pmatrix} 4 & 6 & 2 \\ 3 & 7 & 5 \\ 2 & 3 & 9 \end{pmatrix} = 4 + 7 + 9 = 20$$

Orthogonal Matrices:

Only square matrices may be orthogonal matrices, although not all square matrices are orthogonal matrices. An orthogonal matrix satisfied the equation

$$\mathbf{AA}' = \mathbf{I}$$

Thus, the inverse of an orthogonal matrix is simply the transpose of that matrix. Orthogonal matrices are very important in factor analysis. Matrices of eigenvectors (discussed below) are orthogonal matrices.

Eigenvalues and Eigenvectors

The eigenvalues and eigenvectors of a matrix play an important part in multivariate analysis. This discussion applies to correlation matrices and covariance matrices that (1) have more subjects than variables, (2) have variances > 0.0, and (3) are calculated from data having no missing values, and (4) no variable is a perfect linear combination of the other variables. Any such covariance matrix **C** can be mathematically decomposed into a product:

$$\mathbf{C} = \mathbf{ADA}'$$

where **A** is a square matrix of **eigenvectors** and **D** is a diagonal matrix with the **eigenvalues** on the diagonal. If there are n variables, both **A** and **D** will be n by n matrices. Eigenvalues are also called **characteristic roots** or **latent roots**. Eigenvectors are sometimes refereed to as **characteristic vectors** or **latent vectors**. Each eigenvalue has its associated eigenvector. That is, the first eigenvalue in **D** (d_{11}) is associated with the first column vector in **A**, the second diagonal element in **D** (i.e., the second eigenvalue or d_{22}) is associated with the second column in **A**, and so on. Actually, the order of the eigenvalues is arbitrary from a mathematical viewpoint. However, if the diagonals of **D** become switched around, then the corresponding columns in **A** must also be switched appropriately.

Example: $C = \begin{pmatrix} 100 & 90 & 10 \\ 90 & 100 & 10 \\ 10 & 10 & 100 \end{pmatrix}$,

then eigenvalues in D $\begin{pmatrix} 10.0 & 0 & 0 \\ 0 & 192.17 & 0 \\ 0 & 10 & 97.83 \end{pmatrix}$ and eigenvectors in A $\begin{pmatrix} -.71 & .70 & .11 \\ .71 & .70 & .11 \\ 0 & .15 & -.99 \end{pmatrix}$

Imagine C to be a matrix with variances and covariances between 3 variables. We see that C contains 3 variables each with equal variance and the first two highly correlated. The first eigenvalues has little variance and is mainly made up of the difference between the first two variables. The second eigenvalues has a lot of variance and mainly reflects the sum of the first two variables. The third eigenvalues is pretty much made up of the just the third variable, with only small contribution from the first two.

Note that if the first two variables would have had a covariance close to 100 (correlation 1) the first eigenvalues would be close to 0. If the covariance would be higher (correlation >1), this eigenvalues would even become negative. Eigenvalue decomposition has an important application in checking consistency of covariance matrices; if some eigenvalues are negative, the matrix is not consistent, practically meaning that the correlation structure is not possible (e.g. correlations >1, or just inconsistent (when >2 variables, e.g. when variables a and b are highly correlated and a and c are lowly or negatively correlated, it would not be possible for b to be highly correlated to c).

Some important points about eigenvectors and eigenvalues are:

1) The eigenvectors are scaled so that **A** is an orthogonal matrix. Thus, $\mathbf{A}' = \mathbf{A}^{-1}$, and $\mathbf{AA}' = \mathbf{I}$. Each eigenvector is orthogonal to all the other eigenvectors.

2) The eigenvalues will all be greater than 0.0, providing that the four conditions outlined above for **C** are true.
3) For a covariance matrix, the sum of the diagonal elements of the covariance matrix equals the sum of the eigenvalues, or in math terms, tr(**C**) = tr(**D**). For a correlation matrix, all the eigenvalues sum to n, the number of variables.
4) The determinant of **C** equals the product of the eigenvalues of **C**.
5) Calculating eigenvalues is tedious, not very different from calculating inverses, use computerprograms.
6) VERY IMPORTANT : The decomposition of a matrix into its eigenvalues and eigenvectors is a mathematical/geometric decomposition. The decomposition rearranges the variables into linear combinations of them that become new and independent variables. This rearrangement may but is not guaranteed to uncover an important biological construct or even to have a biologically meaningful interpretation.
7) ALSO VERY IMPORTANT : Eigenvalue decomposition is used in Principal Component analysis. An eigenvalue tells us the proportion of total variability in a matrix associated with its corresponding eigenvector. Consequently, the eigenvector that corresponds to the highest eigenvalue tells us the dimension (axis) that generates the maximum amount of individual variability in the variables. The next eigenvector is a dimension orthogonal to the first that accounts for the second largest amount of variability, and so on.

Solving Systems of Equations Using Matrices

Matrices are particularly useful when solving systems of equations, which we use when we solve for the least squares estimators. Here is an example, with three equations and three unknowns:

$$x + 2y + z = 3$$
$$3x - y - 3z = -1$$
$$2x + 3y + z = 4$$

How would one go about solving this? There are various techniques, including substitution, and multiplying equations by constants and adding them to get single variables to cancel. There is an easier way, however, and that is to use a matrix. Note that this system of equations can be represented as follows:

$$\begin{pmatrix} 1 & 2 & 1 \\ 3 & -1 & -3 \\ 2 & 3 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} 3 \\ -1 \\ 4 \end{pmatrix} \qquad \rightarrow \qquad \mathbf{Ax} = \mathbf{b}$$

We can solve the problem **Ax** = **b** by pre-multiplying both sides by $\mathbf{A}^{-1}$ and simplifying. This yields the following:

$\mathbf{Ax} = \mathbf{b} \rightarrow \ \mathbf{A}^{-1}\mathbf{Ax} = \mathbf{A}^{-1}\mathbf{b} \qquad \rightarrow \mathbf{x} = \mathbf{A}^{-1}\mathbf{b}$

We can therefore solve a system of equations by computing the inverse of **A**, and multiplying it by **b**. Here **A** inverse is $\begin{pmatrix} 8 & 1 & -5 \\ -9 & -1 & 6 \\ 11 & 1 & -7 \end{pmatrix}$

And $\mathbf{x} = \mathbf{A}^{-1}\mathbf{b} = \begin{pmatrix} 8 & 1 & -5 \\ -9 & -1 & 6 \\ 11 & 1 & -7 \end{pmatrix}\begin{pmatrix} 3 \\ -1 \\ 4 \end{pmatrix} = \begin{pmatrix} 3 \\ -2 \\ 3 \end{pmatrix}$

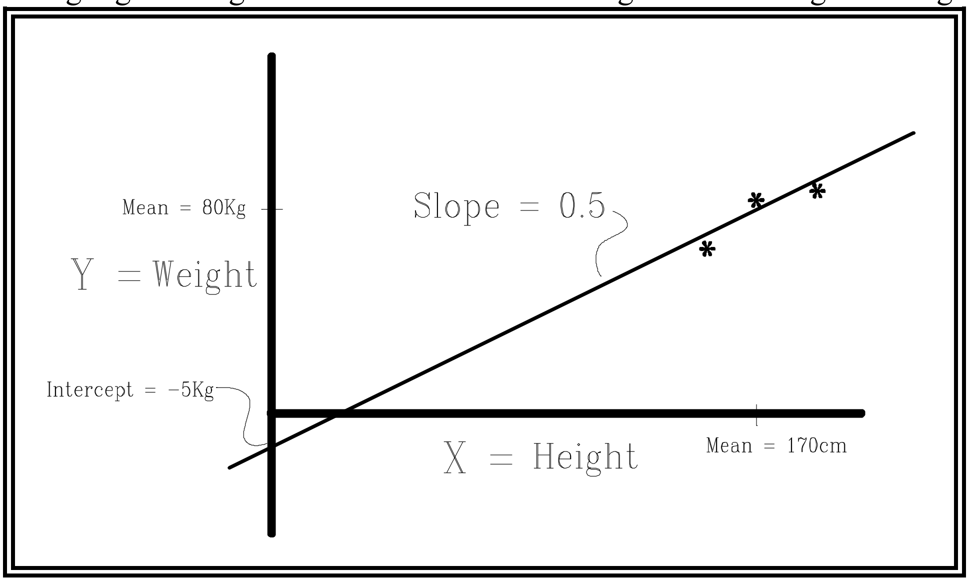
Computationally, this is a much easier way to solve systems of equations – we just need to compute an inverse, and perform a single matrix multiplication.
This approach only works, however, if the matrix **A** is nonsingular. If it is not invertable, then this will not work. In fact, if a row or a column of the matrix **A** is a linear combination of the others, there are *no* solutions to the system of equations, or *many* solutions to the system of equations. In either case, the system is said to be under-determined. We can compute the determinant of a matrix to see if it in fact is underdetermined.
Note also that for many equations, there are more efficient ways to solve such equations, using sparse matrix techniques (many coefficients in such matrices are often zero) and iteration. In act, it can be numerically quite risky to invert a very big matrix as the accumulation of very many rounding errors can become quite substantial.

Example use of Matrices: Regression.

Consider that we have a tiny data set on height and weight of individuals:

Trait	**Data**			**Means**
Weight (Y)	74	82	84	80
Height (X)	160	170	180	170

To predict weight given height we need to calculate the regression of weight on height:

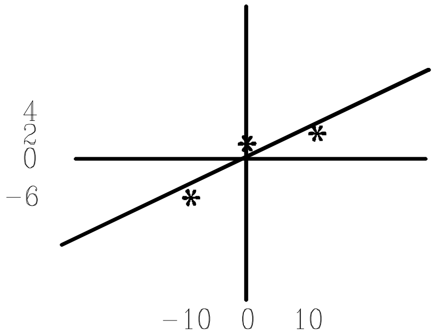


$$\hat{b}_{(Y \text{ on } X)} = \frac{Cov(X,\ Y)}{V_X} = \frac{\dfrac{\sum_i [\,(X_i - X)\,(Y_i - Y)\,]}{n-1}}{\dfrac{\sum_i (X_i - X)^2}{n-1}} = \frac{\sum_i [\,(X_i - X)\,(Y_i - Y)\,]}{\sum_i (X_i - X)^2} = \frac{\sum_i x_i\, y_i}{\sum_i x_i^2}$$

Where x_i and y_i are height and weight expressed as deviations from their respective means.

Notice that if we make vectors $X = \begin{pmatrix} -10 \\ 0 \\ 10 \end{pmatrix}$ and $Y = \begin{pmatrix} -6 \\ 2 \\ 4 \end{pmatrix}$ containing deviations from means, then notice that ...

$$X'X = \begin{pmatrix} -10 & 0 & 10 \end{pmatrix} \begin{pmatrix} -10 \\ 0 \\ 10 \end{pmatrix} = (200) = \sum_i x_i^2$$



$$X'Y = \begin{pmatrix} -10 & 0 & 10 \end{pmatrix} \begin{pmatrix} -6 \\ 2 \\ 4 \end{pmatrix} = (100) = \sum_i x_i \, y_i$$

So, just as $\hat{b} = \dfrac{\sum_i x_i \, y_i}{\sum_i x_i^2}$ we have: $\hat{b} = (X'X)^{-1} X'Y = 0.5$ in this case.

The model we have used here is: $y_i = b \quad x_i + e_i$

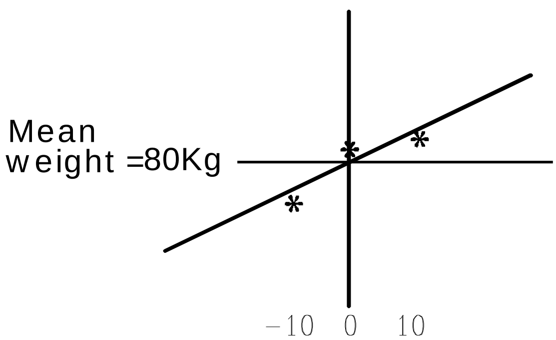
The 'e' is for error. For example @ i=1:	$(-6) = 0.5\,(-10) + (-1)$
We must have an 'e' to be able to use '='.	
We drop the 'e' to get predictions of weight from height:	$\hat{y}_i = \hat{b} \quad x_i$ $(-5) = 0.5\ (-10)$

We can write this model in matrix notation: $\mathbf{Y} = \mathbf{X} \quad \mathbf{b} + \mathbf{e}$

$$\begin{pmatrix} -6 \\ 2 \\ 4 \end{pmatrix} = \begin{pmatrix} -10 \\ 0 \\ 10 \end{pmatrix} \begin{pmatrix} .5 \end{pmatrix} + \begin{pmatrix} -1 \\ 2 \\ -1 \end{pmatrix}$$

A more common model is:

$$Y_i = b_1 \times 1 + b_2 \times (X_i - \bar{X}) + e_i$$



Note that the scalars Y and X are capital - not expressed as deviations from means. We now have 2 b's to be estimated:

Vector b now contains:	**Matrix X now contains:**
b_1 - the effect of the mean of Y (weight) The weight at height $(X_i - \bar{X}) = 0$	1 (100%) is the degree of expression of the mean's effect in *each* observation.
b_2 - the effect on Y a unit deviation in X (i.e. the regression slope of 0.5)	$(X_i - \bar{X})$ is the amount of expression of the effect of height on the i^{th} weight observation.

Now we have $\mathbf{X} = \begin{pmatrix} 1 & -10 \\ 1 & 0 \\ 1 & +10 \end{pmatrix}$ and $\mathbf{Y} = \begin{pmatrix} 74 \\ 82 \\ 84 \end{pmatrix}$

The Model can be written in matrices: Y = X B + e

$$\begin{pmatrix} 74 \\ 82 \\ 84 \end{pmatrix} = \begin{pmatrix} 1 & -10 \\ 1 & 0 \\ 1 & +10 \end{pmatrix}\begin{pmatrix} 80 \\ .5 \end{pmatrix} + \begin{pmatrix} -1 \\ 2 \\ -1 \end{pmatrix}$$

And, as before, $\hat{b} = (X'X)^{-1} X'Y$

$$\hat{b} = (\quad X' \quad X \quad)^{-1} \quad X' \quad Y$$

$$\begin{pmatrix} \hat{b}_1 \\ \hat{b}_2 \end{pmatrix} = \left(\begin{pmatrix} 1 & 1 & 1 \\ -10 & 0 & +10 \end{pmatrix}\begin{pmatrix} 1 & -10 \\ 1 & 0 \\ 1 & +10 \end{pmatrix}\right)^{-1}\begin{pmatrix} 1 & 1 & 1 \\ -10 & 0 & +10 \end{pmatrix}\begin{pmatrix} 74 \\ 82 \\ 84 \end{pmatrix}$$

The result is $\begin{pmatrix} \hat{b}_1 \\ \hat{b}_2 \end{pmatrix} = \begin{pmatrix} 80 \\ 0.5 \end{pmatrix}$ as you would expect, mean weight 80Kg, regression slope 0.5 Kg/cm. You can check this by hand calculation, or e.g. use matrices in Excel.

A model which uses raw data is: $Y_i = b_1 \times 1 + b_2 \times X_i + e_i$

Vector b now contains:	**Matrix X now contains:**
b_1 - is now the intercept - the predicted value of Y (weight) at X (height) = zero	1 (100%) is the degree of expression of the intercept effect in *each* observation.
b_2 - the effect on Y a unit deviation in X (i.e. the regression slope of 0.5)	X_i is the amount of expression of the effect of height in the i^{th} observation.

Now we have $X = \begin{pmatrix} 1 & 160 \\ 1 & 170 \\ 1 & 180 \end{pmatrix}$, $Y = \begin{pmatrix} 74 \\ 82 \\ 84 \end{pmatrix}$, and ...

The Model can be written in matrices: Y = X B + e

$$\begin{pmatrix} 74 \\ 82 \\ 84 \end{pmatrix} = \begin{pmatrix} 1 & 160 \\ 1 & 170 \\ 1 & 180 \end{pmatrix} \begin{pmatrix} -5 \\ .5 \end{pmatrix} + \begin{pmatrix} -1 \\ 2 \\ -1 \end{pmatrix}$$

Note the intercept is at -5Kg as in the original graph.

And we can predict weights (**Y**) as: $\hat{Y} = X B$

$$\begin{pmatrix} 75 \\ 80 \\ 85 \end{pmatrix} = \begin{pmatrix} 1 & 160 \\ 1 & 170 \\ 1 & 180 \end{pmatrix} \begin{pmatrix} -5 \\ .5 \end{pmatrix}$$

$$\hat{b} = (X' \quad X)^{-1} \quad X' \quad Y$$

$$\begin{pmatrix} \hat{b}_1 \\ \hat{b}_2 \end{pmatrix} = \left(\begin{pmatrix} 1 & 1 & 1 \\ 160 & 170 & 180 \end{pmatrix} \begin{pmatrix} 1 & 160 \\ 1 & 170 \\ 1 & 180 \end{pmatrix} \right)^{-1} \begin{pmatrix} 1 & 1 & 1 \\ 160 & 170 & 180 \end{pmatrix} \begin{pmatrix} 74 \\ 82 \\ 84 \end{pmatrix}$$

gives the result: $\begin{pmatrix} \hat{b}_1 \\ \hat{b}_2 \end{pmatrix} = \begin{pmatrix} -5 \\ 0.5 \end{pmatrix}$

Again this is as expected: -5Kg is the intercept - where the regression line cuts the vertical axis (this is the expected weight for x=0, i.e. for a height of zero – does this make sense-?) , and 0.5 Kg/cm is the regression slope.

$\hat{b}$ = (X'X)$^{-1}$ X'Y is a very powerful formula. It forms the basis of multiple regression and Analysis of Variance.
X'X has the number of observations, X'Y has the totals

With some modification, it forms the basis of BLUP.

Reference Books

Searle, S.R. 1982. Matrix Algebra Useful for Statistics.Wiley & Sons.
(this books gives a llot of formal prrofs and mathematical detail)
Mrode, R.A. 1996. Linear Models for the Prediction of Animal Breeding Values. CAB Int. Oxon, UK.(The appendix is simple, similar to these notes)

Exercises

Matrix Review

Some Simple Matrix Problems

$$A = \begin{pmatrix} 3 & 1 \\ 2 & -1 \end{pmatrix} \qquad B = \begin{pmatrix} 1 & 3 & 0 \\ 2 & 4 & 2 \end{pmatrix} \qquad C = \begin{pmatrix} 1 & 2 \\ 2 & 6 \end{pmatrix}$$

$$D = \begin{pmatrix} 2 & 0 \\ 0 & 3 \end{pmatrix} \qquad E = \begin{pmatrix} 1 \\ 4 \end{pmatrix} \qquad I = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{pmatrix} \qquad I = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

Which matrices are square, which are symmetrical, which are diagonal, which are identity? Which matrices are of not of full rank?

Compute the following. One or two of these cannot in fact be computed. You should attempt at least those marked with an asterisk. You may use Excel to check you work, but attempt a few 'by hand' to get a feel for matrix calculations

(*) A + C	A - C	(*) B + C	6B	
(*) AC	CA	(*) AB	BA	AE
(*) AI	IA	IE		
AD	DA	DE		
A^2	D^2	I^2		
A'	(*) B'	E'	C'	

(*) B'A' and compare with (AB)'

(*) B'B	BB'	E'E	EE'	
(*) A^{-1}	(*) AA^{-1}	$A^{-1}A$	$(AC)^{-1}$	$C^{-1}A^{-1}$

Matrix calculations using Excel

You can do some basic matrix calculations with MS Excel.

First put in the values of your matrices

To multiply two matrices:
- select an area of the size of the resulting matrix
- type: =**MMULT(**
- select the area of the first matrix
- type a comma (,)
- select area of the second matrix
- type a close bracket)
- press: Ctrl_Shift_Enter

To add or subtract a matrix (vector):
- select an area of the size of the resulting matrix
- type: = (
- select the area of the first matrix
- type a **+** or **-**
- select area of the second matrix
- type a close bracket)
- press: Ctrl_Shift_Enter

To invert a matrix:
- select an area of the size of the resulting matrix
- type: =**MINVERSE(**
- select the area of the first matrix
- type a close bracket)
- press: Ctrl_Shift_Enter

To transpose a matrix (vector):
- select an area of the size of the resulting matrix
- type: =**TRANSPOSE(**
- select the area of the first matrix
- type a close bracket)
- press: Ctrl_Shift_Enter

A more specialized matrix calculation program is MATLAB. It contains many more matrix functions and mathematical function than excel. MATLAB allows you to make and run programs, draw graphs, and run simulation). A MATLAB student version is very well suitable for animal breeding problems and quite easy to use.

Matrix Commands in R

(from: Gareth James, Daniela Witten , Trevor Hastie and Robert Tibshirani
An Introduction to Statistical Learning with Applications in R, Springer)

R uses *functions* to perform operations. To run a function called funcname, we type funcname(input1, input2), where the inputs (or *arguments*) input1 and input2 tell R how to run the function. A function can have any number of inputs. For example, to create a vector of numbers, we use the function c() (for *concatenate*). Any numbers inside the parentheses are joined together.
The following command instructs R to join together the numbers 1, 3, 2, and 5, and to save them as a *vector* named x. When we type x, it gives us back the vector.

```
x = c(1,6,2)
> x
[1] 1 6 2
y = c(1,4,3)

length (x)
[1] 3
length (y)
[1] 3
x+y
[1] 2 10 5

x=matrix (data=c(1,2,3,4) , nrow=2, ncol =2)
> x
[,1] [,2]
[1,] 1 3
[2,] 2 4
```

Note that we could just as well omit typing data=, nrow=, and ncol= in the matrix() command above: that is, we could just type

```
x=matrix (c(1,2,3,4) ,2,2)
```

As this example illustrates, by default R creates matrices by successively filling in columns. Alternatively, the byrow=TRUE option can be used to populate the matrix in order of the rows.

```
matrix (c(1,2,3,4) ,2,2,byrow =TRUE)
[,1] [,2]
[1,] 1 2
[2,] 3 4
```

Taking subsets of matrices. Suppose we have

```
A=matrix (1:16 ,4 ,4)
> A
     [,1] [,2] [,3] [,4]
[1,] 1 5 9 13
[2,] 2 6 10 14
[3,] 3 7 11 15
[4,] 4 8 12 16
```

Then, typing

```
A[2,3]
[1] 10

A[c(1,3) ,c(2,4) ]
   [,1] [,2]
```

```
[1,] 5 13
[2,] 7 15

A[1:3 ,2:4]
  [,1] [,2] [,3]
[1,] 5 9 13
[2,] 6 10 14
[3,] 7 11 15
```

Adding matrices: Use A+B
Transpose: use At = t(A)

```
> X
    [,1]
[1,]    1
[2,]    1
[3,]    1
[4,]    1

> t(X)
    [,1] [,2] [,3] [,4]
[1,]    1    1    1    1
```

Multiplying matrices use A %*% B

```
> t(X)%*% X
    [,1]
[1,]    4
```

Note that A * B multiplies each element individual of 2 matrices of equal dimension. This is NOT a matrix multiplication

```
Inverse of A: AINV = solve(A)

#setting up a small example for a mixed model

#First set up the variables
data      = array(c(10,20,30,40))                  #we have 4 records here
nrecords  = dim(data)
y         = matrix(data,ncol=1)                    # put those in vector y
X         = matrix(1,nrow=nrecords, ncol=1)        # make X as a vector which just ones
Z         =diag(1,nrecords)                        #make Z as an identity matrix
alpha = 3                                          # just a scalar
```

```
# example of a relationship matrix
A        =matrix(c(1,0,.5,.5, 0,1,.5,.5, .5,.5,1,.5, .5,.5,.5,1),nrow=4)
A
     [,1] [,2] [,3] [,4]
[1,]  1.0  0.0  0.5  0.5
[2,]  0.0  1.0  0.5  0.5
[3,]  0.5  0.5  1.0  0.5
[4,]  0.5  0.5  0.5  1.0

AI=solve(A)
> AI
     [,1] [,2] [,3] [,4]
[1,]    2    1   -1   -1
[2,]    1    2   -1   -1
[3,]   -1   -1    2    0
[4,]   -1   -1    0    2
```

Setting up a coefficient matrix for mixed model, putting submatrices together

```
XX= t(X) %*% X
XZ= t(X) %*% Z

ZX= t(Z) %*% X
ZZ= t(Z) %*% Z
C22=ZZ + alpha*AI

Ctop = cbind(XX, XZ)
Cbot = cbind(ZX,C22)
MMElhs = rbind(MMEtop,MMEbot)

Xy= X %*%y
Zy= Z %*%y

MMErhs = rbind(Xy,Zy)

# Solve the equations
Cinv   = solve(MMElhs)
MMEsoln=Cinv %*% MMErhs
```

Or simply:

```
MMEsoln= solve(MMElhs,MMErhs)
```