Note for 302 students

This book also provides an introduction to some techniques not covered in the course.
Past students have found it a useful reference source later in life.
Material covered in the course : Chapters 1-7,10

If you are doing STATS 302 you do not have to read chapters 8 and 9!

Contents

Chapter 1. Introduction to Matrix Algebra.

Since, no matter how hard one tries to avoid it, a familiarity with some of the jargon of matrix algebra is necessary for even a superficial understanding of multivariate statistics; an introductory section on the subject is inescapable. That was the bad news. The good news is that I will try to use a simple, intuitive, non-rigorous approach. The main aim of the section is to provide a highly selective, very superficial introduction to the subject.

As might be guessed from the context (a course on multivariate statistics), matrices are commonly used for handling multivariable systems. Simple systems (say of 2 or 3 variables) can also be described and manipulated by the more traditional graph paper. It should therefore come as no surprise that the two, matrices and graph paper, are linked and our entry into the arcane world of matrices is via a concept of multidimensional graph paper - Euclidean space.

WARNING: attempts to visualise spaces of more than 3 dimensions can damage your mental health.

Vectors.

A **vector** is a set of coordinates that define a point in a space (e.g. graph paper) relative to a set of axes (a basis). More usefully it defines a line from the **origin** (the zero point, the intersection of the axes) to that point. This line is a vector in the strict meaning of the word having magnitude and direction. Such a set of coordinates can be referred to by a variable name e.g. **x** (note: bold lower case type usually refers to a vector). So, a set of 5 measurements on the nose of an anteater would define a point in a 5-dimensional space. This concept naturally generalises to any number of dimensions. Such a space is called an *n*-dimensional hyperspace, (they do exist outside science fiction), or an *n*-space,(the in-person's jargon term).

We have talked of multi-dimensional hyperspaces, but in fact the most used space of all is one-dimensional! All real numbers can be thought of as points lying on a line running from plus infinity to minus infinity - the real line. Any number also defines a line from zero. It has length and direction (positive or negative) - it is a vector. Thus the number 5 defines a vector running from the origin in the positive direction having length 5. The number (-3) has length 3 in the negative direction. Ordinary real numbers can be thought of vectors in a one dimensional space. This will allow the operations and algebra of multidimensional spaces to be explained with reference to the arithmetic and algebra of real numbers, which everyone understands (I hope).

Properties of Vectors.

Vectors, like numbers on the real line, have both length (distance from zero - the origin) and direction. Like numbers you can add (or subtract) two vectors to get a third, just add (or subtract) corresponding elements of the two vectors.

Later on in the course we shall be concerned with the length of vectors so it may need some explanation now. Given the vector, how do we calculate its length? In one dimension it is trivial, the vector (-3) has length 3. In two dimensions it becomes easy as soon as you draw it. The length of the vector (x_1, x_2), usually written $||\mathbf{x}||$, is clearly given by Pythagoras's theorem. Thus

$$||\mathbf{x}||=(x_1^2+x_2^2)^{1/2}.$$

The length of a three dimensional vector (x_1,x_2,x_3) can be similarly calculated.

$$||\mathbf{x}||=(x_1^2+x_2^2+x_3^2)^{1/2},$$

or more generally for a space of n dimensions

$$||\mathbf{x}||=(\sum_{i=1}^{n} x_i^2)^{1/2} \quad .$$

The concept of the length of a vector now lets us measure the distance between two vectors. First we calculate the difference between two vectors. The operation of subtraction in 1 dimensional space defines the vector that joins the end points of the two vectors (think about it, better still draw it and look).

The same operation can be described for two dimensions. Subtracting one vector **a** from another **b** gives a third which defines the position vector **c** (fig.2b). This gives the direction and distance between the two vector end points.

The actual calculation is as one would expect:

$$\begin{bmatrix} a_1 \\ a_2 \end{bmatrix} - \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} = \begin{bmatrix} a_1 - b_1 \\ a_2 - b_2 \end{bmatrix} = \begin{bmatrix} c_1 \\ c_2 \end{bmatrix}$$

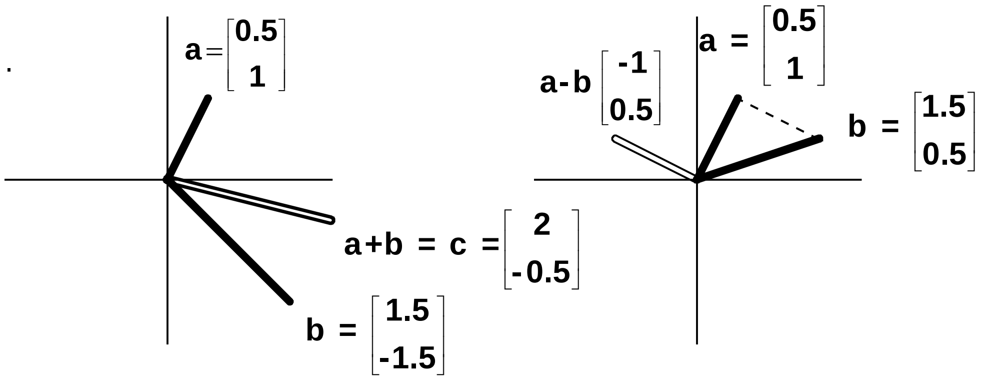
The difference vector has direction and length. Its length is the distance between the two points(vectors). Notice the vectors **a** and **b** are written as **column vectors** (i.e. the elements are aligned vertically). This is the usual default, if you are not told otherwise assume that vectors will be considered as having 1 column and many rows. As you will see this can become important later on.

The difference between two data points (e.g. animals or other sampling unit) is a most useful thing to know, and is called the Euclidean distance, after Euclid (lived around 300 B.C.) the father of classical geometry. We showed how to get the length of a vector earlier; so the distance between two vectors **a** and **b** is the length of the difference vector:

$$\|\mathbf{a}\text{-}\mathbf{b}\| = \left(\sum_{i=1}^{n}(a_i - b_i)^2\right)^{1/2}$$

So the Euclidean distance depends on Pythagoras's theorem- the ancient Greeks knew a thing or two.

We can therefore say how dissimilar two vectors (data points, animals) are by calculating the Euclidean distance between them. This will be useful later on when we talk of dissimilarity (and similarity) matrices in the context of cluster analysis and multidimensional scaling.



$$\begin{bmatrix} a_1 \\ a_2 \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} = \begin{bmatrix} a_1 + b_1 \\ a_2 + b_2 \end{bmatrix} = \begin{bmatrix} c_1 \\ c_2 \end{bmatrix} \qquad \begin{bmatrix} a_1 \\ a_2 \end{bmatrix} - \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} = \begin{bmatrix} a_1 - b_1 \\ a_2 - b_2 \end{bmatrix} = \begin{bmatrix} c_1 \\ c_2 \end{bmatrix}$$

Figure 1.1

a) addition of vectors to produce resultant vector
b) subtraction of vectors

Vectors have two different multiplication operations associated with them:

1) Like numbers you can multiply the vector by a single number. Every element of the vector is then multiplied by this same number. Because it therefore rescales every element of the vector such numbers are called **scalars**. Multiplication by a scalar has a particular geometric interpretation that will be important later, it doesn't change the direction of the vector (except to reverse it if the scalar is negative). But it does change the length - see fig 1.2.

2) You can multiply two vectors together (provided they have the same number of coordinates). The usual method produces a single number, the **inner product**. The process is simple, but since it is a special case of matrix multiplication - the vectors are treated as little matrices- we will consider it later.

Matrices.

What is a matrix? The most obvious use of the word is to refer to a table of numbers. The starting point for most multivariate analyses is the **data matrix**, whose **rows** contain the values for each sampling unit, and whose **columns** contain the values for each variable. Such a table may be referred to by a name e.g. **X** (note the bold type to show it's a matrix not a simple variable, and that it is a capital letter to show it is not a vector). Each individual number in the table, an **element** of the matrix, can be referred to using subscripts: x_{15} refers to the element in the 1st row (sampling unit) and the 5th column (variable). The convention is the row subscript first, then the column. (Please repeat 20 times: rows then columns, rows then columns...).

Sometimes the data is needed with the variables as rows and the sampling units as columns. This new matrix is the **transpose** of **X**, and is written **X**' (or sometimes $\mathbf{X}^T$ or $\mathbf{X}^t$). The value of the 1st variable on the 5th sampling unit (x_{51}) would now be in the 1st row, 5th column, i.e. x'_{15}. The transpose is simply given by reversing the

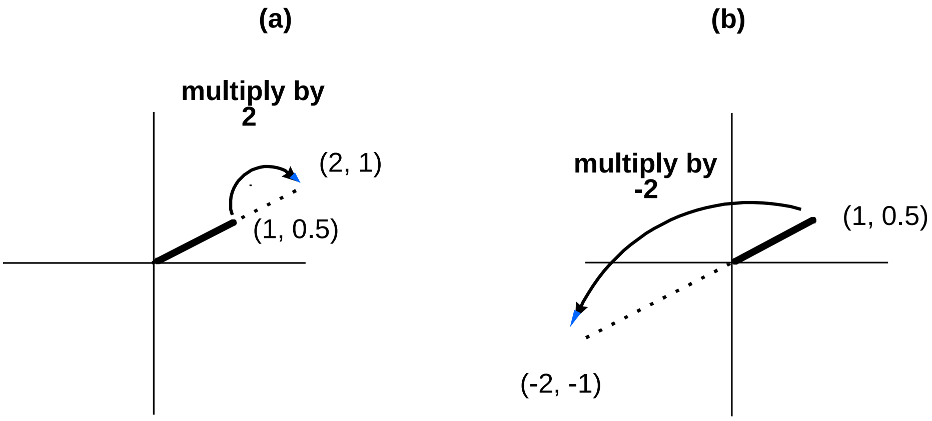


Figure 1.2
a) multiplication by a positive vector (2)
b) by a negative vector (-2)

subscripts of the elements.

The data matrix, **X**, could also be thought of as a table of vectors. Each row could be considered a vector - each sampling unit thus defining a vector in variable space. So, the (nxp) data matrix defines a cloud of n points (observation vectors) in a p dimensional space. Most multivariate statistical methods can be imagined as performing various operations on this data cloud, looking or testing for trends or structure.

In an introduction to matrix algebra perhaps the most useful way of viewing matrices is as operators on vectors. Just as ordinary algebra allows the manipulation of variables representing real numbers (i.e. vectors on the real number line), so the use of matrices allows a matrix algebra to manipulate vectors and vector spaces.

Matrix multiplication.

We can for example look at the matrix as an operator which moves points (vectors) around in a space. If you multiply a point on the real line by a number it moves it to a new position. If you multiply a vector by a matrix it moves it to a new vector. For example if you multiply

a = $\begin{bmatrix} 1 \\ -1 \end{bmatrix}$ by the matrix **M =** $\begin{bmatrix} 4 & 1 \\ 6 & 1 \end{bmatrix}$ we get the vector **b** = $\begin{bmatrix} 3 \\ 5 \end{bmatrix}$.

Multiplying **b** by the matrix **N** = $\begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix}$ gives the vector **c** = $\begin{bmatrix} -3 \\ -5 \end{bmatrix}$.

Thus **M.a = b**, **N.b = c**, just like ordinary algebra. Like numbers, this sequence of operations can be condensed into one equation **NMa = Pa = c**. Clearly matrices can be multiplied together, so **M** multiplied by **N** gives a new matrix **P**. This matrix does the same thing as multiplying a vector first by **M** and then multiplying the resulting vector by **N**; multiplying a vector by **P** just does it in one go.

IMPORTANT: Unlike multiplying numbers **NM** is not usually equal to **MN;** the order of multiplication is important. In Figure 1.3 using the matrices **A =** $\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$ which reflects vectors in the Y axis, and **B** = $\begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$ which rotates them 90° anticlockwise, we can see a reflection followed by a rotation does not give the same results as a rotation followed by a reflection.

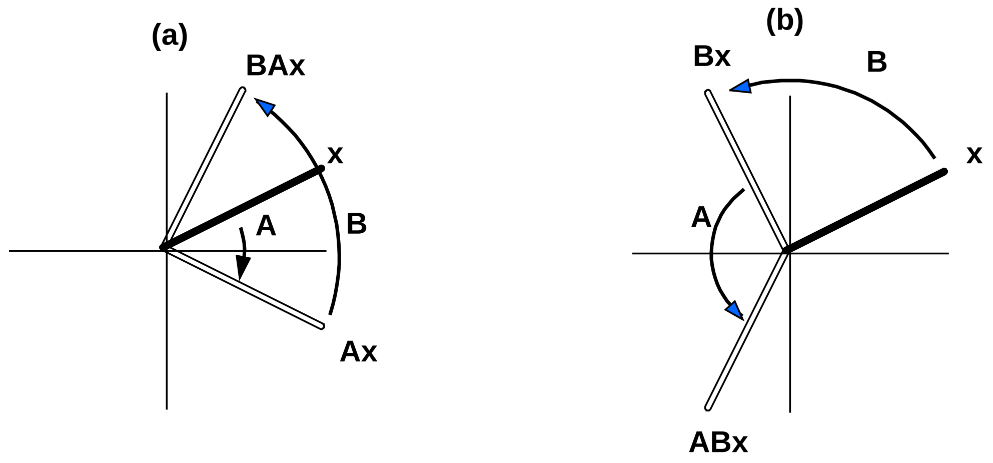


Figure 1.3 The order of multiplication can be important.

a) multiplication by a matrix A (reflection) then B (rotation)
b) multiplication by B (rotation) then A (reflection).

The results of multiplication by a square matrix are easy to comprehend; it simply changes the position of a vector to a new position in the space. However multiplication by a rectangular matrix is slightly more complex. a 2 x 1 vector lies in (defines a point in) a 2 dimensional space, multiplying it by a 2 x 2 matrix will leave it in that space. Multiplying it by a 1x 2 matrix will squeeze it into a 1 dimensional space, and by a 3 x 2 inflate it into a 3 -D space. Notice, for the multiplication to be possible the number of columns must be the same as the dimensionality of the vector, the number of rows of the matrix indicates the number of dimensions of the resulting product vector. As we shall see in section 1.2.2 **rectangular** matrices (number of rows ^{1} number of columns) can lead to problems.

On the real line there are two multiplication operations that are of particular importance: multiplication by zero, and multiplication by one. The first maps all vectors on the real line to zero - the **null vector**. The second leaves all vectors unchanged. It is called the identity operation. Matrix algebra has equivalents. Multiplication by a matrix of zeros will move any vector to the origin, the null vector. The matrix **I** = $\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$, the **identity matrix,** will leave any 2 dimensional vector unchanged. **I** = $\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$ is the identity matrix for 3 dimensional spaces, and so on. Any **square matrix** (equal number of rows and columns) with ones down the **diagonal** and zeros elsewhere is an identity matrix.

As we mentioned above the inner product of two equal length vectors can be regarded as the product of two matrices. For this to be possible the multiplication of the column vector **y** by **x** is actually **x'y** (**y'x** has an identical value). The result is a single number.

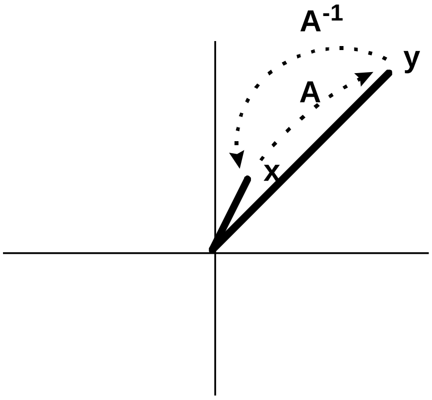


Figure 1.4. The action of an inverse ($\mathbf{A}^{-1}$) takes **y** back to **x** again

As we shall see later this can be a short hand way of representing an important statistical calculation.

Matrix division.

In comparing ordinary algebra on the real line with matrix algebra we have defined: addition of real numbers (cf addition of vectors (and matrices)), and multiplication by a number (cf multiplication by a matrix); how about division by a number? To see if there could be a matrix equivalent it is sensible to have a closer look at what division means on the real line. Division by a number *a* is equivalent to multiplication by 1/*a*, or a^{-1}. Division has now become multiplication by the reciprocal. How might we define the reciprocal? If multiplication by *a* moves a point *x* to *ax* then we can define a^{-1} as that operation that takes *ax* back to *x*, so that $a^{-1}a$ equals one. We can therefore call a^{-1} the **inverse** of *a*, since on the real line it does the opposite to multiplication by *a*.

We can now define the equivalent matrix operation to division by a number as multiplication by the inverse of a matrix. The inverse of a matrix, by analogy, is going to be that matrix, if it exists, that acts to negate the action of the matrix of which it is the inverse. So if **M** takes **a** to **b**, then $\mathbf{M}^{-1}$ takes **b** to **a** (Figure 1.4), and $\mathbf{M}^{-1}\mathbf{M} = \mathbf{I}$ the identity matrix (the matrix equivalent of the number 1).

Are there any situations when this is not going to work, an inverse does not exist? Even on the real line there is one such case - the inverse of zero does not exist. If we look at why, we may be able to predict situations where the inverse of a matrix will not exist.

The problem with multiplication by zero is that it maps all points on the real line into the same spot - zero. This makes an inverse impossible. Given that we are currently at zero which of the infinity of points should we map back to? There is no unique inverse. If we look at it in vector space terms, multiplication by zero moves any vector in the one dimensional space (the real line) into a zero dimensional space (the point zero). It then becomes apparent that any matrix that moves vectors into a space of lower dimension is not going to have an inverse. Many points in the original space will map to a single point in the resulting space, trying to jam a quart (1136ml) into a pint (568ml) pot, making an inverse operation impossible. Similarly matrices that map into higher dimensional spaces also lack inverses. Once in the higher dimensional space there is no unique way back. Rectangular matrices therefore do not have inverses. As we shall see below not all square ones have inverses, such matrices are called **singular**, because they map to a singularity - like a black hole. Well behaved square matrices that have inverses are called **non-singular**.

One advantage of the inverse is that it allows us to do algebraic manipulations just as with real numbers. We can therefore rewrite the equation

$\mathbf{Mx} = \mathbf{y}$ as $\mathbf{x} = \mathbf{M}^{-1}\mathbf{y}$, just as though this was univariate algebra.

Addition of matrices.

If **Ax** is the vector **x** after it has been moved, then it is also a vector and all the operations between vectors can be applied (e.g. vector addition and subtraction). Suppose we have two vectors **Ax** and **Bx** then the vector that is their sum will be **Ax**+**Bx** which should (and does) equal (**A**+**B**)**x** = **Cx**, where **C**=(**A**+**B**). How does one add matrices? The two matrices must be the same shape, same number of rows and columns, then we can simply add the elements of one to those in the corresponding position in the other matrix.

Subtraction of matrices is similarly defined.

Matrix properties.

The determinant.

Like vectors and numbers on the real line, matrices have unique properties. Vectors have length and direction. Numbers have size and direction. Matrices also have a concept of "size" . The size of a number X is mathematically defined by the absolute value or modulus, usually written |X|. The analogous property of a matrix is given by the **determinant**, |**X**|, which measures of the "size" of the matrix. It is a single number and as such is often used as a summary statistic for statistical matrices of various kinds. However its most common use is to find out if a matrix is non-singular (i.e. has a inverse).

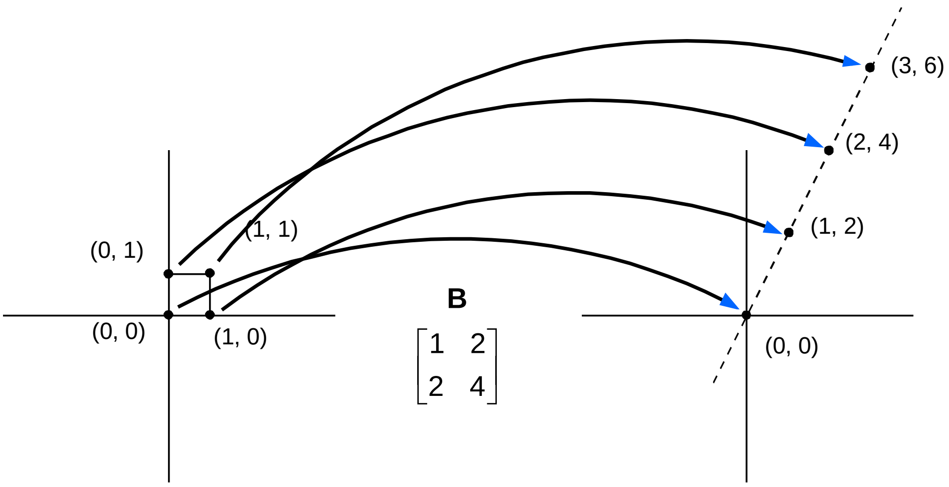


Figure 1.5. Multiplying by a singular matrix. No inverse is possible. The matrix does not have two dimensions worth of information. Its determinant is zero.

If a number has an absolute value of 0 then no inverse exists (the reciprocal of 0 is -∞). Similarly, if the determinant of a matrix is 0 then there can be no inverse. As we have seen earlier, rectangular matrices do not have inverses; it's difficult to go back the way you came if you move between spaces of different dimensionality. But why should a square matrix ever have a determinant of zero. Look at Figure 1.5. A square matrix multiplies vectors, but consistently puts them into a space of lower dimension. Clearly it cannot have an inverse - its determinant is zero. If you look carefully you will see that its rows are effectively the same, differing only by a scale factor, read on...

The determinant can also be regarded as a measure of the amount of independent information in the rows (or columns) of the matrix. If the determinant is small then one or more of the rows is nearly linearly dependent on one or more of the others, that is it shares nearly all of its information with one or more of the other rows (or columns). In this case that row is effectively redundant, adding little to the information in the matrix. There is less independent information in the matrix than the number of rows suggests. This leads to one of its statistical applications (see section 1.4.2).

Eigenvalues and eigenvectors.

Some square matrices have an underlying structure that is not usually apparent. If you multiply a vector repeatedly by a matrix, usually (not always), the resulting vectors will move closer and closer to lying on a line (figure 1.3).

For example

M **x**

$$\begin{bmatrix}1.3 & 0.4\\ 0.4 & 0.7\end{bmatrix}\begin{bmatrix}1\\ 1\end{bmatrix}=\begin{bmatrix}1.7\\ 1.1\end{bmatrix} \qquad \mathbf{M}^2\mathbf{x}=\begin{bmatrix}2.65\\ 1.45\end{bmatrix}$$

$$\mathbf{M}^3\mathbf{x}=\begin{bmatrix}4.025\\ 2.075\end{bmatrix} \qquad \mathbf{M}^4\,\mathbf{x}=\begin{bmatrix}6.062\\ 3.062\end{bmatrix} \qquad \mathbf{M}^5\mathbf{x}=\begin{bmatrix}9.106\\ 4.560\end{bmatrix}$$

Once the vectors are close to the line further multiplication by the matrix only shifts them further out, stretches the current vector. this is equivalent to multiplication by a scalar, in this example 1.5. In fact multiplication of any vector v that actually lies on the line is equivalent to multiplication by a scalar λ. So $\mathbf{Mu}=\lambda\mathbf{u}$. For points on this line, and only on this line the matrix is equivalent to a single number! This number labours under a variety of names : characteristic root, latent root, but probably the most common, and the one we shall use in this course is **eigenvalue**. All vectors that lie on the invariant line are multiples of the characteristic vector, latent vector, or **eigenvector**.

For example

$$\mathbf{M} \quad \mathbf{u} \; = 1.5 \times \mathbf{u}$$

$$\begin{bmatrix}1.3 & 0.4\\ 0.4 & 0.7\end{bmatrix}\begin{bmatrix}2\\ 1\end{bmatrix}=\begin{bmatrix}3\\ 1.5\end{bmatrix} \qquad \mathbf{M}\text{ times }\begin{bmatrix}10\\ 5\end{bmatrix}=\begin{bmatrix}15\\ 7.5\end{bmatrix}$$

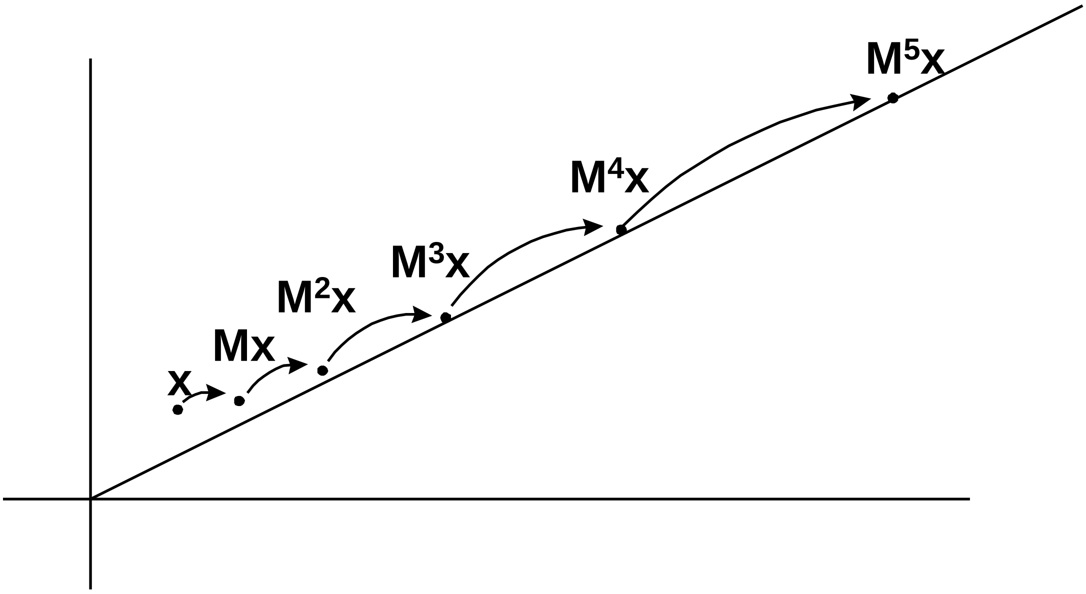


Figure 1.6. The invariant line. Repeated multiplication of the vector **x** by the matrix **M** brings the resulting vectors closer and closer to the invariant line; where multiplication by the matrix is equivalent to multiplication by a scalar - the eigenvalue. (c.f. figure 1.2a)

You could try any vector where the elements are in the proportion 2:1, in other words lie on the invariant line for matrix **M**; multiplication by **M** is equivalent to multiplying by 1.5. In fact most matrices have more than one such invariant line, matrix **M** has another defined by the vector (-1,2) or any multiple thereof. The corresponding scalar is 0.5 (try it). Thus matrix M has two eigenvectors ($\mathbf{u}_1$ and $\mathbf{u}_2$) each with its own eigenvalue (λ_1 and λ_2).

Intuitively, eigenvalues can be thought of as a measure of the stretching power or size of the matrix. In particular the largest or dominant eigenvalue is often used in this way. Since the determinant is sometimes used similarly, you would expect a relationship between the two. The product of the eigenvalues ($\prod \lambda_i$) equals the determinant |**M**|. So if any of the eigenvalues are zero then so is the determinant - no inverse - singular matrix. In fact the number of non-zero eigenvalues give the number of linearly independent rows there are in the matrix - its **rank**. Eigenvalues have many useful properties, but the only other one to which I shall refer is that the sum of the eigenvalues is equal to the sum of the diagonal elements of the matrix, which is called the **trace**. $\boldsymbol{\Sigma}\lambda_i$=$\boldsymbol{\Sigma}m_{ii}$.

We have discussed the meaning of the eigenvalues, what about the eigenvectors, the vectors that define the invariant lines? The full set can be used to define a new set of axes (a basis) that is implicit or latent in that matrix - the **basic structure** of the matrix. The directions defined by these vectors are somehow important to that matrix, are characteristic of that matrix, and in some way summarise the information in that matrix. These new axes are sometimes very useful for re-expressing the data points, especially in principal component analysis, and canonical discriminant analysis (sections 6 and 12). If the matrix **M** is symmetric then the eigenvectors are orthogonal to each other (these new axes are at right angles) - most useful.

One way in which the eigenvalues and eigenvectors express the underlying structure of a square matrix is in the **eigenvalue** (or **spectral**) **decomposition**. Any square matrix **M** can be broken down into 3 matrices. So: **M** = **UΛU'**, where **U** consists of all the eigenvectors of **M** stacked like books in a bookshelf, **U'** is its transpose (the books are now piled on top of each other) **Λ** is a diagonal matrix (i.e. only the diagonal elements can be non zero) with the eigenvalues as the diagonal elements in the same order as the eigenvectors. It is worth pointing out that these eigenvalues can under certain circumstances be imaginary numbers, but DON'T PANIC, we generally throw those away if we meet them. All this seems like just some wonderfully useless mathematical fantasy. This decomposition will however be an extremely useful way of summarising certain data based matrices. We will find that eigenvalues and eigenvectors have quite simple statistical interpretations when used properly, and this decomposition of a matrix into its basic structure underlies most of the techniques found in this course.

Simple applications and some useful matrices.

Crossproduct matrices.

Most multivariate methods, as conventionally described do not work on the data matrix **X** directly. They usually convert it to the form **X'X**, the data matrix multiplied by its transpose. Such a matrix is called a crossproduct matrix. Let us take as a simple (and extremely useful example) the **inner product** of the vector **x** with itself, i.e. **x'x**.

Let us take a data vector that has had the mean of each variable (column) subtracted from every value in that column, e.g.

$$\begin{bmatrix} (x_1 - \bar{x}) \\ (x_2 - \bar{x}) \\ (x_3 - \bar{x}) \end{bmatrix}$$

So we are now dealing with deviations from the mean (such data are called mean-centred - see section 4.1.1). If we now multiply this vector by its transpose we get a number that should be instantly familiar to anyone who has done 1st year Stats.

$$\begin{array}{ccc} \mathbf{x'} & \mathbf{x} & = \quad \mathbf{x'x} \\ \begin{bmatrix} (x_1 - \bar{x}) & (x_2 - \bar{x}) & (x_3 - \bar{x}) \end{bmatrix} & \begin{bmatrix} (x_1 - \bar{x}) \\ (x_2 - \bar{x}) \\ (x_3 - \bar{x}) \end{bmatrix} & = \sum_{i=1}^{n} (x_i - \bar{x})^2 \end{array}$$

In other words, when the data have been centred by subtracting the mean, **x'x** is equal to the corrected sum of squares; a most useful number.

We can generalise this to the bivariate case:

$$\mathbf{X} = \begin{bmatrix} (x_{11} - \bar{x}_1) & (x_{12} - \bar{x}_2) \\ (x_{21} - \bar{x}_1) & (x_{22} - \bar{x}_2) \\ (x_{31} - \bar{x}_1) & (x_{32} - \bar{x}_2) \end{bmatrix}$$

If we now multiply **X** by its transpose **X'** we will get a matrix whose elements should look familiar. Thus:

$$\begin{array}{ccc} \mathbf{X'} & \mathbf{X} & \mathbf{D} \\ \begin{bmatrix} (x_{11} - \bar{x}_1) & (x_{21} - \bar{x}_1) & (x_{31} - \bar{x}_1) \\ (x_{12} - \bar{x}_2) & (x_{12} - \bar{x}_2) & (x_{32} - \bar{x}_2) \end{bmatrix} & \begin{bmatrix} (x_{11} - \bar{x}_1) & (x_{12} - \bar{x}_2) \\ (x_{21} - \bar{x}_1) & (x_{22} - \bar{x}_2) \\ (x_{31} - \bar{x}_1) & (x_{32} - \bar{x}_2) \end{bmatrix} & = \begin{bmatrix} \sum (x_{i1} - \bar{x}_1)^2 & \sum (x_{i1} - \bar{x}_1)(x_{i2} - \bar{x}_2) \\ \sum (x_{i1} - \bar{x}_1)(x_{i2} - \bar{x}_2) & \sum (x_{i2} - \bar{x}_2)^2 \end{bmatrix} \end{array}$$

If you examine the elements of **D** you should recognise d_{11} as the corrected sum of squares for variable X_1, d_{12} as the corrected sum of crossproducts between variables X_1 and X_2 (you may have used this to calculate regressions and correlations) and d_{22} as the sum of squares of variable X_2. It should be obvious that d_{21} is the same as d_{12} - the matrix **D** is symmetric. If we think of the sums of squares d_{ii} as lying on the **diagonal** of the matrix, then the sums of crossproducts lie in the upper (d_{12}) and lower (d_{21}) **triangle**. The matrix **D** is usually known as the **sums of squares and crossproducts** or SSCP matrix. The elements of **D** immediately suggest (or should do) that if we were to divide every element by (the number of sampling units-1), i.e. multiply by 1/(*n*-1), we would convert the sums of squares to variances and the crossproducts to covariances. Thus 1/(*n*-1)**D** = **S**, the **variance-covariance**, or just **covariance**, matrix. The elements s_{11} and s_{22} (i.e. s_{ii} the diagonal elements) are the estimated variances of X_1 and X_2, while s_{12} and s_{21} estimate the covariance between them. This matrix is the starting point for many of the classical multivariate methods and we will come across it repeatedly. More generally, the *ij*th element is the covariance of the *i*th and the *j*th variable. Note that since this matrix is symmetric about the diagonal, i.e. $s_{ij}=s_{ji}$, then **S**'=**S**. For example, in Table 1.1 the variance of sepal length is 12.4, its covariance with petal width is 1.6.

The covariance matrix contains information about the spread of the data points over the variables (the variances), and also about whether the variables are correlated, how elliptical the data cloud is. (Data from a heavily correlated pair of variables typically lie along a line, forming a narrow ellipse). Thus for many types of data the covariance matrix summarises the size and shape of the data cloud.

There is another crossproduct matrix that is important statistically. If we divide the mean corrected values in data matrix **X** by the appropriate standard deviations we get a matrix of z-scores, standardised deviations from the mean (see section 4.1.2a). The resulting variance covariance matrix actually contains correlation coefficients as the off diagonal terms. This **correlation** matrix (usually called **R**) has all the diagonal elements equal to 1 (*z*-scores have unit variance), and the *ij*th element, r_{ij}, is the correlation between variables X_i and X_j. For example, in Table 1.1 the correlation between sepal width and petal width is 0.23, the reverse correlation, between petal width and sepal width is of course the same so the matrix is symmetric. To point out the obvious, the correlation between petal length and itself is of course 1. It is of course quite simple to calculate the correlation matrix directly from the covariance matrix (you might like to work out how) but thinking of the correlation matrix as the covariance matrix of *z* scores, standardised data, will help you to understand methods that use the correlation matrix as their starting point.

Programming Notes

In R/S Plus the variance covariance matrix can be got by:

```
var(data)
```

The correlation matrix by:

```
cor(data)
```

Table 1.1. Covariance and Correlation matrices for measurements (mm´ 10) on the plant *Iris setosa.*

	Covariance Matrix					Correlation Matrix	
Sepal	Sepal	Petal	Petal		Sepal	Sepal	Petal
Length	Width	Length	Width		Length	Width	Length
12.4	9.9	1.6	1.0		1.00	0.74	0.27
9.9	14.3	1.2	0.93		0.74	1.00	0.17
1.6	1.2	3.01	0.61		0.27	0.17	1.00
1.0	0.93	0.61	1.1		0.28	0.23	0.33

The generalised variance.

If two variables in a data matrix are nearly perfectly correlated, then their two columns will be almost identical and the determinant of the covariance matrix will be close to zero - there is less independent information in the data matrix than its size suggests. Some of the off diagonal elements of the covariance matrix, the covariances, will be large - there are large correlations present. If on the other hand all the variables are independent - all the covariances near zero - then the determinant will be larger. Also the larger the diagonal terms (variances) the larger the determinant. Thus the determinant of a covariance matrix is sometimes used as a single number that summarises the overall independent variance of a system of variables - the **generalised variance**. We mentioned earlier that the determinant was sometimes used as a measure of the 'size' or 'magnitude' of the matrix. Graphically, we can think about the determinant as measuring the (hyper)volume of the data cloud in space, the larger the variance the larger the volume; the more correlated the variables, the smaller the volume.

Some revision questions.

1) What is a vector?

2) What is an inner product of two vectors?

3) What is the transpose of a matrix?

4) How do you multiply two matrices together?

5) What is the inverse of a matrix?

6) Why doesn't a rectangular matrix have an inverse?

7) What is a singular matrix?

8) What is the determinant of a matrix?

9) What is the value of the determinant of a singular matrix?

10) What does the determinant of a variance –covariance matrix measure?

11) What is an eigenvalue? How many can a matrix have?

12) What is the trace of a matrix?

Chapter 2. Data exploration and checking.

Checking for errors.

When the data is first entered into the computer you must resist the temptation to dive straight into a sophisticated multivariate analysis. No matter how close the deadline, time spent checking and preparing the data now will save time and embarrassment later. For a start the data almost certainly contains copying mistakes and outliers, almost certainly violates the assumptions of the analysis, and, if you are lucky, may not need a sophisticated analysis anyway.

The first step of any analysis is to look for the copying errors that have probably crept into the data while it was being entered into the computer. The most thorough method is to have someone else read out what ought to be in the computer while you check what is actually there. Regrettably, very large data sets can strain friendships and eyesight alike; so this method is often impractical. Generally the most disastrous errors are those that generate outliers, a slipped decimal point for example. These can often be detected by preparing the univariate data displays (histograms, stem-and-leaf or box plots) that are standard in 1st year statistics courses. At the very least every variable should be displayed and outliers investigated to check they are genuine. A word of warning: if you fail to discover any mistakes - worry. I have never had a data set of any size that did not have some errors in it. Typing mistakes, observations missed out, observations repeated, they all happen somewhere. Only when you are sure the data set is as correct as possible is it worth investing time in its analysis.

The univariate displays are not only for detecting errors and outliers, they also give you your first insight into the data. Is there any evidence of structure in the data - polymodality for example? Do any patterns emerge when you compare means between variables, or between groups of observations? In environmental statistics the sampling units can often be plotted on a map. If the values of each variable are mapped do any patterns emerge? It may be that the patterns or results are so obvious that no multivariate analysis is necessary. Though before you make that decision have second thoughts; what is obvious to you, a true believer, may not be obvious to a sceptical client, referee or examiner. If you decide to go ahead with an appropriate multivariate analysis, the insights you have gained from the univariate displays will help you interpret and check the results of the multivariate analysis. If the analysis flatly contradicts what is clear in the univariate displays, some further investigation is needed.

Multivariate Exploratory data analysis.

Scatterplots.

Exploratory data analysis (**EDA** to its advocates) is more of a philosophy than an analysis. Its chief proponent was John W. Tukey (Bell Labs and Princeton University): "Exploratory data analysis is an attitude, a flexibility, and a reliance on display, NOT a bundle of techniques" (Tukey 1980:23). The approach is based on a "recognition that the picture-examining eye is the best finder we have of the wholly unanticipated". Clearly, by relying on such a (over)sensitive pattern recognition machine, EDA is aimed at the generation of hypotheses, not their testing; to the identification of the unexpected, not the confirmation of the already known. To a large extent the rest of this course is concerned with methods that make multivariate data available to that "picture-examining eye"; techniques that allow the data to be examined, patterns or relationships detected and ideas generated. They are seldom used for tests of *a priori* hypotheses.

The techniques in the later sections are usually sophisticated and often difficult to interpret, their advantage is that they are truly multivariate. The spirit of EDA, on the other hand, is simplicity. As I pointed out above, simple univariate methods, histograms and the like, can often be useful in gaining

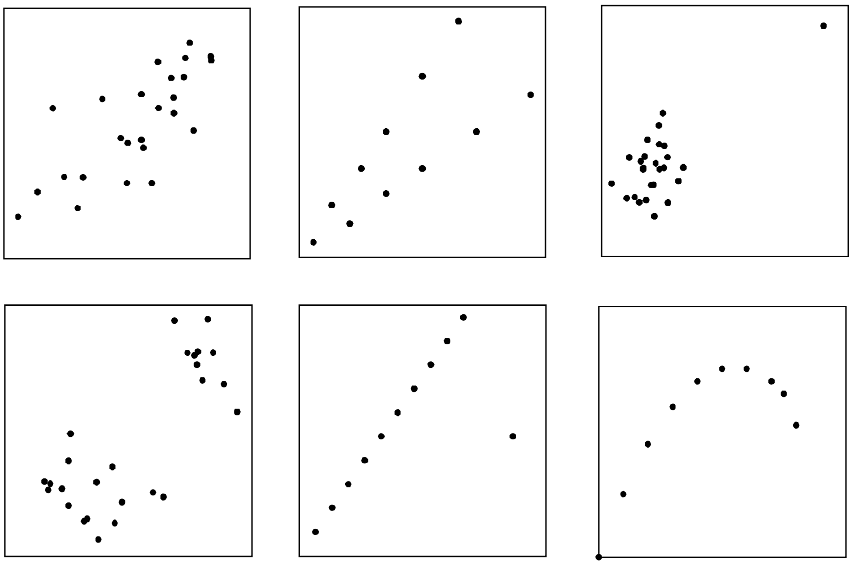


Figure 2.1. All these scatterplots have correlations of 0.75, yet all tell very different stories. Do not rely on summary statistics, always plot your data before analysis.

insight into the data. Such simple graphical methods are certainly more useful than tables of summary statistics, and are a lot easier to read.

The chief problem with multivariate data is that if there are more than three variables it is usually impossible to visualise the data cloud properly. We can produce all the summary statistics: mean vectors, covariance matrices, correlations etc. but we cannot see what the data cloud actually looks like. Figure 2.1 shows how important this can be - even with just two variables.

2. Before we apply a sophisticated multivariate analysis (with their often restrictive assumptions) we ought to try simpler, more direct, approaches to the data. If we cannot plot the data cloud in all its multidimensional glory, then we can at least plot it on each pair of variables in turn. Subtle, truly multivariate patterns will probably not emerge, but the grosser patterns and relationships should be apparent. The simplest way of organising all these plots, $p(p - 1)/2$ of them (if there are p variables) is in a **scatterplot matrix** (Figure 2.2). This is like the lower triangle of the correlation matrix (section 1.4.1) with plots instead of correlation coefficients.

EXAMPLE 2.1.

In 1936 the great geneticist and biometrician R. A. Fisher introduced a data set into the multivariate literature that now appears in nearly every text course on the subject - it is almost a tradition. The data consists of measurements of sepal length and width and petal length and width on three *Iris* species: *I. setosa*, *I. versicolor*, and *I. virginica*. He used them to demonstrate the application of Linear Discriminant Function Analysis (LDFA - not covered in this course), and since then they have become the classic data set for the purpose. LDFA tries to establish rules (functions) to classify vectors (individual flowers) into classes (the three species). For this to work, the species must be relatively separate. By doing all the bivariate plots we can see if they are.

Figure 2.2 shows the scatterplot matrix for the four variables with each of the species having a different symbol. Clearly *I. setosa* is separate from the others, while *I. virginica* and *I. versicolor*, though less obviously separable, do not appear to overlap too much. Indeed, in the full four dimensions they could be clearly distinct. A further point worth noting is that these plots clearly suggest that the species have very different shaped data clouds, i.e. that their covariance matrices are different. This could have consequences for LDFA, which, to be optimal, requires the classes (species) to have identical population covariance matrices - i.e. their data clouds must have similar shapes. I show how to check this assumption more formally in section 0.

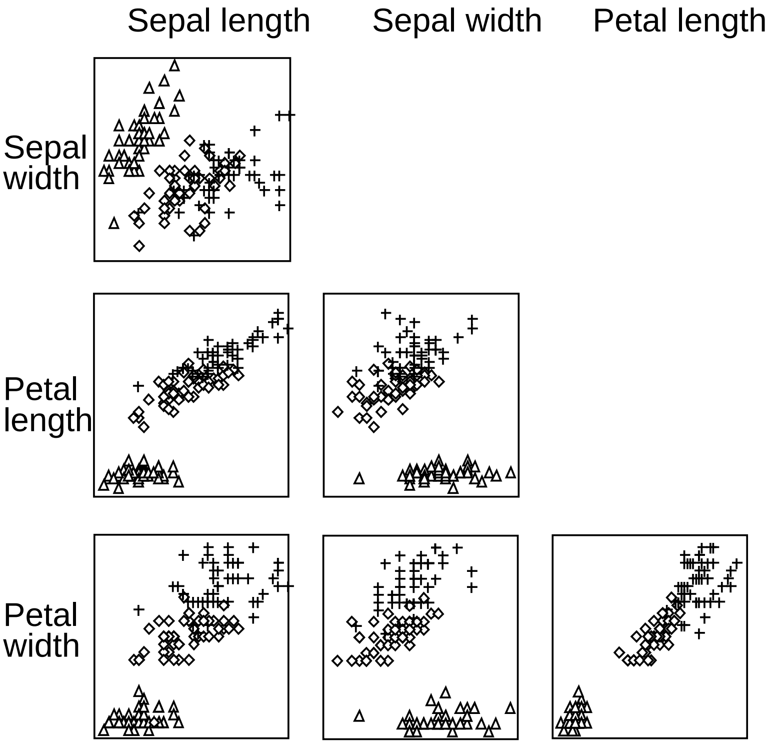


Figure 2.2. Scatterplot for Fisher's Iris data. The species seem well separated.

Programming note.

In R the basic instruction is: **pairs(***variable list***)**.

Interactive 3-D Graphics.

While the scatterplot matrix is invaluable for the preliminary examination of the data, recent advances in computer graphics provide a new, more powerful, more flexible tool for the biologist - **interactive 3-D graphics** or "**brush and spin**".

While a 3 dimensional plot is clearly better than a 2-D scatterplot, how can you decide on the orientation of the plot? Patterns that are clearly visible from one angle might be totally obscured at another. Recent advances in graphics algorithms now make it possible to plot the data in 3 dimensions and then rotate it in any direction you want, at a reasonable speed. This allows you to view the 3-variable data cloud from any angle, and to be able to try many angles in a short time. Such spin programs also usually allow you to interact with the data by allowing a subset of the points to be highlighted ("brushed" or "painted") so their position in the rotating cloud can be clearly seen (Figure 2.3). The more sophisticated programs allow data points to be selected in one display - for example in one of the bivariate scatterplots - and they will be highlighted in all the other plots, including the rotating ones. This allows patterns that emerge in one display to be checked in the others. Such interactive displays are clearly going to be a great help in identifying patterns that bivariate or univariate displays simply cannot pick up. As I mention later, these displays can also be used to check other assumptions or conclusions of multivariate analyses.

One word of warning. When looking at the data cloud the distances between the points, and the shape of the cloud are important. If the cloud is long and flat you want to see that in your rotating plot.

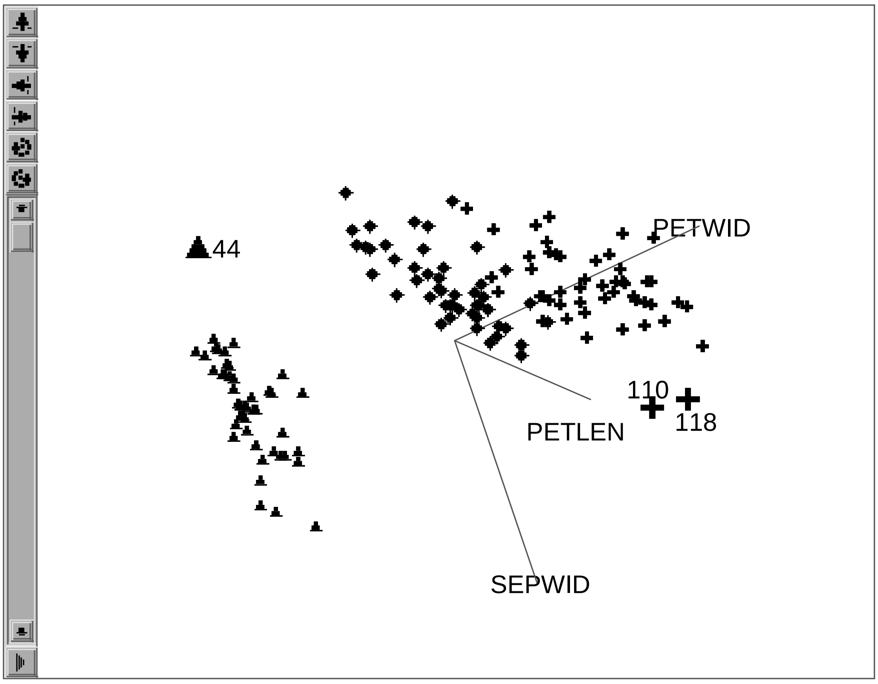


Figure 2.3. Rotated 3-D plot of Fisher's Iris data (petal width, sepal width and length). I rotated it using the tool bar at the left to find a useful picture. Note 3 observations have been selected as possible outliers. They are highlighted and their observation numbers (row in the data matrix) displayed so they can be easily identified.

Unfortunately most packages that do this type of plot try to be too helpful. If there is little variation on one axis, they expand the plot in that direction so that the range of each variable has the same size on its axis. As a result the shape of the cloud is distorted, sometimes badly. Unless the package has the facility to turn off this clumsy assistance, the only recourse is to put into the data file eight dummy data points that define a cube that will completely enclose the cloud. Because the range on each variable is now the same (the dummy points now define the range) the cloud will be undistorted, hanging in the cube which protects it from getting rescaled.

Programming Note.
At present there is no convenient 3-D rotation in R.

Checking for multivariate normality.

Many techniques for the analysis of continuous data assume a **multivariate normal distribution** (MVN). Unfortunately, though univariate normality of each of the variables is a necessary condition of MVN; it is not sufficient. So, though checking for univariate normality is a useful first step, if you fail to find non-normality it does not mean that the data are MVN.

Techniques are available for testing the null hypothesis of MVN, but generally the relatively informal graphical methods are best. They can kill two birds with one stone: not only do they check the shape of the distribution, they also highlight multivariate outliers. Though univariate and bivariate plots (see

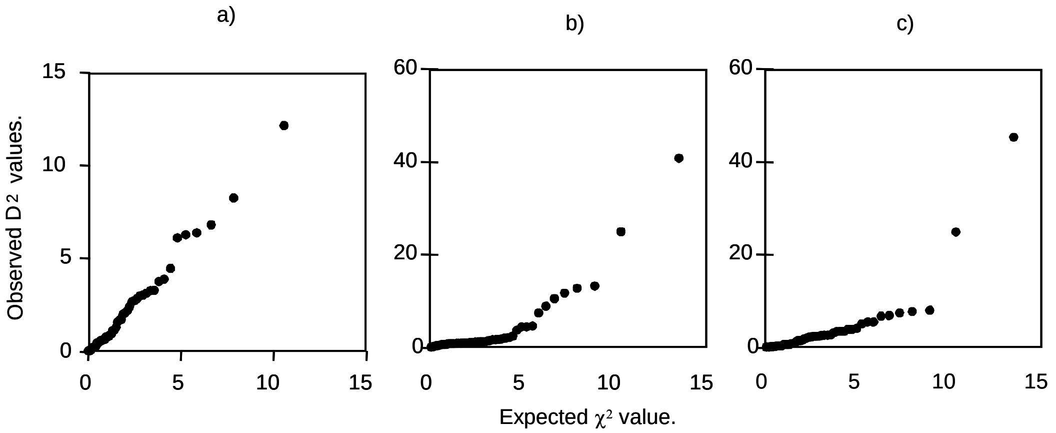


Figure 2.4 Multivariate probability plots for 3 artificial data sets
a) from a multivariate normal
b) from a multivariate lognormal (a very skewed distribution)
c) from a multivariate normal with 2 outliers
Note the difference in the scale of the vertical axes

below) are helpful in detecting outliers, they cannot be expected to find all of them. An outlier in two dimensions need not be detectable in univariate plots; similarly, bivariate plots can miss outliers in higher dimensions.

Probability plotting.

The simplest check for multivariate normality is based on a simple property of the MVN: the squared distance (suitably standardised) from a point to the centroid (the mean vector - the centre of the data cloud) is approximately χ^2 distributed. This is understandable if we look at the definition of the χ^2 statistic: χ^2 (df 1) is defined as $((x-\mu)/\sigma)^2$ i.e. the squared standardised distance from x to the mean (univariate centroid). To show the link with the multivariate case, this can be rewritten as $(x-\mu)(\sigma^2)^{-1}(x-\mu)$. The multivariate equivalent is $(\mathbf{x} - \boldsymbol{\mu})'\boldsymbol{\Sigma}^{-1}(\mathbf{x} - \boldsymbol{\mu})$ where $\boldsymbol{\Sigma}^{-1}$ is the inverse of the covariance matrix. If the data are MVN these squared distances will have a χ^2 distribution with p degrees of freedom - where p is the number of variables. This standardised distance is known as the squared Mahalanobis distance D^2. We shall meet it again in a number of places later in the course.

So, if the data are MVN the distances should be χ^2. If the distances are not χ^2 then the data are not MVN. How can the χ^2 distribution of these distances be checked? Most people taking an undergraduate course in statistics are shown how to check the normality of a univariate distribution using probability or normal quantile-quantile (qq) plots. Such a plot is possible for other distributions - in the present case by using the inverse χ^2 transformation in place of the inverse normal. In this case the D_i^2 squared distance values would be used in place of x_i.

This technique will work well if $\mathbf{S}$ is close to $\boldsymbol{\Sigma}$, i.e. if it is based on a large sample. I suggest $n > 10p$ (p is the number of variables); though for practical purposes it can work well for smaller numbers.

The plot will be approximately straight if the data conform to the multivariate normal distribution (Figure 2.4a). If they are curved or stepped, the data are not normal, e.g. Figure 2.4b where the data come from a multivariate lognormal distribution (i.e. heavily skewed). Stepping may suggest groups in the data, but as a general rule polymodality is extremely difficult to detect using these plots.

Since the distances are ordered from smallest to largest on the transformed cumulative % axis, the extreme outliers will be at the right hand end, and it should be obvious if they are further away from the centroid than expected from an MVN distribution (Figure 2.4c).

Most multivariate techniques that assume the MVN distribution are more sensitive to skewness than to kurtosis or polymodality. So, even if the probability plots detect non-normality, the sample sizes are small, and the analysis is not considered robust, don't panic. Provided the distribution is still symmetric the analysis will usually be able to go on as planned. Check the symmetry using the univariate displays; if they are all symmetric then the multivariate distribution must be symmetric.

Programming notes

In R use:

```
xx<-dataset name – quantitative variables only
mah<-mahalanobis(xx,apply(xx,2,mean),var(xx))
qqplot(qchisq(ppoints(mah), ncol(xx)), mah)
```

Testing equality of covariance matrices.

Some methods, like multivariate analysis of variance, and canonical discriminant analysis (Chapter 11), make comparisons between groups of observations. One assumption that they all share, and to which they can be very sensitive, is that the observations within each group are MVN distributed with equal covariance matrices (section 1.4.1). It is a sensible precaution to check the assumptions before employing any of these techniques.

While it might be possible to compare covariance matrices element by element, this is usually impractical. There are a number of more sophisticated techniques described in the literature, but I shall only discuss two:

i) The maximum likelihood test for homogeneity of covariance matrices.
ii) The multivariate extension of **Levene's test**.

One approach would be to condense each covariance matrix down to a single measure of 'size' and compare those. The most commonly used value is the determinant |**S**| - the generalised variance (section 1.4.2). Sometimes $\log_e$|**S**| is used instead.

Unfortunately, just looking at the values of the generalised variance conveys little to the inexperienced eye (and not much to the experienced one), so some more formal approach is usually needed.

<u>Maximum likelihood test for the homogeneity of covariance matrices.</u>

This is the most widely available test. It is discussed in all major multivariate texts and is present in all major statistical packages; yet for most purposes it is of little use. It is a multivariate generalisation of Bartlett's test that is described in many introductory statistics courses..

It is calculated as

$$M = \frac{\prod_{i=1}^{g} |\mathbf{S}_i|^{\frac{(n_i-1)}{2}}}{|\mathbf{S}|^{\frac{(n-g)}{2}}}$$

	Iris setosa				*Iris versicolor*			
Sepal length	12.42	9.921	1.635	1.033	26.64	8.518	18.28	5.577
Petal length	9.921	14.36	1.169	.9298	8.518	9.846	8.265	4.120
Sepal width	1.635	1.169	3.015	.6069	18.28	8.265	22.08	7.310
Petal width	1.033	.9298	.6069	1.110	5.577	4.120	7.310	3.910

Iris virginica			
40.43	9.376	30.32	4.909
9.376	10.40	7.137	4.762
30.32	7.137	30.45	4.882
4.909	4.762	4.882	7.543

Table 2.1. Covariance matrices for the Iris species.

Where $n = \Sigma n_i$, g = number of groups.

Thus $M = 1$ when the group covariance matrices $\mathbf{S}_1 = \mathbf{S}_2 = \mathbf{S}_i = \mathbf{S}$ the pooled group covariance matrix; when the $\mathbf{S}_i$ are not equal, $M > 1$.

A number of approximations are used to let M be compared with a tabulated statistic. The commonest is a χ^2 approximation. Despite its popularity in statistical computing packages this test is usually not very useful, for two reasons:

i) It is too powerful. It can detect differences between covariance matrices that in normal circumstances can have little or no effect on the robustness of any of the techniques in this course. It is therefore of little use in detecting troublesome data sets.

ii) It is very sensitive to non-normality in particular kurtosis a form of non-normality that usually has minimal effects on multivariate techniques.

A significant result could be due to differences between the covariance matrices, non-normality or both, and could be irrelevant anyway given the robustness of most multivariate methods.

Multivariate Levene's test.

In Levene's univariate test each data value is replaced with the absolute value of its deviation from the median (or, sometimes, the mean) of its group. If one sample is more variable than the others then the average deviation for that sample will be larger than for the others; this can be detected by a simple analysis of variance (ANOVA) on the deviations.

With multivariate data each variable is treated as above (except we nearly always use the mean rather than median). Each value is replaced by the absolute value of its deviation from the sample mean for that variable in that group. However a multivariate ANOVA (section 11.1) is used to detect any differences, rather than a lot of simple ANOVAs. The multivariate ANOVA combines (sort of) all the ANOVAs into one significance test with one p-value. A significant result does not necessarily mean that the differences are large enough to be statistically (or biologically) important. But this technique is more robust than the maximum likelihood method described above, and the results are correspondingly more reliable indications of potential problems.

EXAMPLE 2.2

Using the *Iris* data again, we want to test the null hypothesis that the population covariance matrices of the three species are equal. The sample covariance matrices are shown in Table 1. At first sight they appear different, particularly the variances. Our multivariate Levene's test rejects the null hypothesis with $p<0.0001$. A preliminary interpretation based on univariate Levene's tests suggests that the major differences are in sepal length and width; which from Table 1 are clearly less variable in the smaller species. This relationship between the variance and the mean is a common phenomenon with morphometric data. The situation can often be improved by using a logarithmic transformation (but not in this case!)

This function will do a Multivariate Levenes test.

```
MVlevene= function(y,x)
{
  res=lm(as.matrix(y)~as.factor(x))
  resid=abs(res$residuals)
  lev=lm(resid~as.factor(x))
  mvlev=manova(lev)
  summary.manova(mvlev)
}
```

Checking for linearity of relationship among variables.

Many of the multivariate techniques described in this course assume that the covariance or correlation matrix adequately describes the relationship amongst the variables, i.e. that the relationships are linear. Some, like multiple regression and canonical correlation (Sections 9 & 10) have their own techniques for detecting non-linearities. Even so, it is often useful to check for linearity at an early stage before deciding on a final analysis. The checks for multivariate normality in section 2.3 are to some extent also checks for linearity - an MVN distribution has linear relationships among the variables. But non-linearity cannot usually be distinguished from other types of non-normality. It is therefore often useful to make a simple and direct check by plotting a matrix of scatter plots (see above - section 2.2.1). By examining each plot non-linear relationships can often be spotted. Dynamic 3D graphics (section 2.2.2) can also help. If there are a lot of variables there will be lots of plots; still, time spent in front of the computer screen examining them is seldom wasted. Once the shape of the bivariate relationships has been seen you can decide on the potential usefulness of transformations; whether or not to drop some variables; which analysis is going to be the most appropriate; or even whether or not you need a multivariate analysis at all.

Some revision questions

1) What is a scatterplot matrix?
2) What is "brush and spin"?
3) How can you check for multivariate normality?
4) How can you detect outliers form a multivariate normal distribution?
5) How can you detect heterogeneity of covariance matrices?
6) How does a multivariate Levene's test work?
7) How might you check for the linearity of relationships between the variables in a multivariate data set?

Chapter 3. Data Preparation.

Missing values.

Many real data sets have missing values, i.e. some of the observations have one or more of their variables unmeasured. This leaves gaps in the data matrix. There are a number of ways of handling this situation, though there must be sufficient data remaining to do the job - information cannot be created from nothing. Problems can also arise if the missing values are non-randomly distributed (as they often are). The simplest methods described below assume that values are missing completely at random (MCAR). If all the large values for a variable were missing, the sample would be biased and this assumption violated. However, the more sophisticated methods only assume that missing values in Y_1 cannot depend on the value of Y_1 but can depend on values of the other variables e.g. Y_2. In other words, the probability of Y_1 being missing can depend on the other variables, but if there were a number of observations with the same value of Y_2 then the pattern of missing values of Y_1 within them must be completely random. This weaker assumption (missing at random - MAR) means that the missing values of Y_1 can be validly predicted from the other variables.

Table 3.1a shows a complete, highly artificial, data set. Table 3.1b shows what it might look like if values of Y_1 were missing completely at random, i.e. there is no tendency for the missing values to correlate with Y_1 or Y_2. Table 3.1c shows what it might look like if the probability of a missing value were correlated with Y_2 but is random within Y_1 for any value of Y_2 (MAR). This will lead to a biased estimated of say the mean if you averaged over the whole data set. But unbiased if you broke the data up into subsamples based on Y_2 and estimated separate means. Table 3.1d shows what might happen if the probability of a missing value is correlated with both Y_1 and Y_2 - the large values of Y_1 are missing within Y_2 = 1. This last situation is irretrievably biased and cannot be handled by any currently available technique.

Table 3.1. The different patterns of missing values.

a) Complete		b)MCAR		c) MAR		d) biased	
Y_1	Y_2	Y_1	Y_2	Y_1	Y_2	Y_1	Y_2
17	1	17	1	.	1	.	1
16	1	.	1	16	1	.	1
15	1	15	1	.	1	15	1
14	2	14	2	14	2	14	2
13	2	.	2	13	2	13	2
12	2	12	2	12	2	12	2

For MCAR data the usual strategy is to remove missing values from the data matrix.

If the data set has many variables and the missing values are confined to a few, you could consider dropping the affected variables. However, it is more usual to drop all the observations that have missing values - the complete-observation method. This is the standard solution used by the main

statistical packages. Clearly, depending on the distribution of the missing values, a combination of the two could be a sensible option. This method has several drawbacks:
If too many values are missing this pruning of the data set may shrink the data set too far for it to yield useful results.
Potentially useful information is being thrown away, some of the discarded observations may be of particular interest.
It assumes the values are missing completely at random. If the missing values are non-randomly distributed over the observations you will be left with a biased sample.

This crude method has one major advantage over the alternatives: it is simple. This and the fact that the more complicated alternatives (that we do not coever in this course) sometimes do not work well make it attractive if the data set is not too small, the values are MCAR, and the number of missing values is not too large.

Transformations and standardisations.

Many data sets require modification before analysis. At first sight this might appear a questionable activity - fudging the data, fiddling the results. In fact there are several perfectly respectable reasons for transforming the data, and if it is done properly the results should be just as valid (you hope even more valid) and more useful than if the analysis had been performed on the original values. The choice of transformation of standardisation is, as we shall see, possibly the most important decision to be made when planning a multivariate analysis.

The main reasons for using transformations are:

to put a variable on a more interpretable, appropriate, scale;
to remove size effects;
to put variables on similar scales;
to make the data conform to the assumptions of the chosen analysis, e.g. multivariate normality, homogeneity of covariance matrices, linearity.

I will follow convention by distinguishing between four different types of univariate transformation: centering, standardising, recoding, and transforming by mathematical function. I don't discuss truly multivariate transformations here because to a large extent that is what the rest of the course is about.

Centering.

This involves shifting the centroid of the data cloud, i.e. shifting the multivariate centre of the data. It is simply done by subtracting a number from each of the values. Though there can be a number of reasons for doing this, the most common is to remove what I shall call size effects.

a) Centering by column means.
Very often differences between the absolute values of variables are not of interest. The differences between the averages for each variable obscure the interesting patterns. For example in morphometrics: that the average length of the humerus (the upper bone in the foreleg) of a mammal is longer than that of the ulna - the lower leg bone of the foreleg - (or perhaps vice versa) is usually of little interest, it is how these variables (co-)vary that is interesting. The effect of the averages is removed by subtracting the mean of each variable from all its values:

i.e. $x^*_{ij} = (x_{ij} - \bar{x}_j)$. Where x_{ij} is the value from the *i*th row (observation) and *j*th column (variable), and $\bar{x}_j$ is the mean of the *j*th column.

This can therefore be thought of removing simple size differences between the variables and focusing the analysis on variability of the variables - the data has been transformed to deviations from the variable means.

Geometrically this transformation is a shift of the centroid (the vector of means) to the origin (the zero vector). All vectors are now deviations from the centroid (mean). In fact column centering is performed whenever the covariance or correlation matrix is calculated, i.e. when interest is focused on how the variables vary together, so it is easily the most commonly used transformation.

Programming notes.

In R we can use the apply() function that repeats a function separately for each row (put 1 in as the second argument) or for each column (as here, a 2 :as the second argument)

```
xx<-data
col.mean.stdised<-apply(xx, 2, function(z){z-mean(z)})
```

or more simply

```
col.mean.stdised=scale(xx,scale=F)
```

b) Centering by row means.
Sometimes you may want to remove size differences between observations (rows) before the analysis. For example, with ecological data, say a site (rows) by species (columns) matrix, you might want to get rid of differences in average abundance between the sites; so the analysis can focus on the patterns of the species abundances within the sites. Each data point is therefore converted to a deviation from the row (site) mean:

i.e. $x^*_{ij} = (x_{ij} - \bar{x}_i)$,

where $\bar{x}_i$ is the mean of the *i*th row. Each species is now represented by its deviation from the site mean, a high value no longer necessarily means an abundant species, just one that is relatively common (dominant) within its site. It is worth pointing out that in this example the data would usually be logarithmically transformed before centering.

Programming notes:

In R use the same code as before but change it to row means instead of column ones:

```
xx<-data
row.mean.stdised<-<-apply(xx, 1, function(z){z-mean(z)})
```

or more simply

```
row.mean.stdised<-xx-rowMeans(xx)
```

Standardising.

i) Standardise the columns (variables) by a measure of dispersion.
There are two main reasons to standardise variables: to put them on similar, standard, scales; or to make the scale more interpretable. There are a number of different standardisations; I will only mention the most common.

If all the variables have different units, then it is often sensible to convert them to the same, standard, scale, so that the variation within the variables is of comparable magnitude. The commonest such standardisation is to divide by a sample measure of the dispersion, most commonly the standard deviation, (though some workers recommend the range): i.e. $x^*_{ij} = x_{ij}/s_j$,. Where s_j is the sample standard deviation of the jth variable (column). This converts all the values to scores in standard deviation units. If the x_{ij} have already been centred by columns this standardisation will yield scores with zero mean and unit variance. Geometrically this has the effect of equalising the variation on each axis (the variables now all have unit variance) so the data cloud usually tends to become more spherical. Also the covariance matrix of such standardised variables is now the correlation matrix (section 1.4.1), an advantage if the main interest is in the relationships amongst the variables.

In practice this standardisation means that all variables will have roughly equal weight in the analysis.

Programming notes.

In R

```
xx<-data
col.sd.stdised<-apply(xx,2,function(z){z/sqrt(var(z))})
```

Or if you don't mind it being centred (subtracting the column mean) then this is easier:

```
col.sd.stdised<-scale(xx)
```

This is often an acceptable thing to do as the overall column means are seldom considered useful.

ii) No standardisation.
Despite the fact that most people do not explicitly standardise their data, this is a fairly rare situation. Many multivariate techniques use the correlation matrix, and many distance and similarity metrics also standardise implicitly (section 4). A principal component analysis on the covariance matrix (section 5) or any techniques performed on a simple Euclidean distance matrix would be effectively unstandardised though centered. These would be appropriate if the variables are on similar scales and you want the observations to contribute in proportion to their variability.

iii) Standardising the columns by a measure of size.
One way of putting variables on comparable scales is to convert them to proportions. For example by dividing each value by the column total: i.e. $x^*_{ij} = x_{ij}/\sum_i x_{ij}$, where $\sum_i x_{ij}$ is the total for each variable. ($x_{ij}/\ \bar{x}_j$, where $\bar{x}_j$ is the mean for the jth species is equivalent.)

Proportions are often easier to interpret than the original data, and can be an effective way of removing size differences between the variables. They also ensure that all the variables will have equal weight in the analysis - with the problems that this can entail (see above, section 3.1.2.i).

In R

```
xx<-data
x<-apply(xx,2,function(z){z/sum(z)})
```

iv) Standardising the rows by size.
More frequently the interest lies in the proportions of the variables within observations:
i.e. $x^*_{ij} = x_{ij}/\Sigma_j x_{ij}$,

where $\Sigma_j x_{ij}$ is the total for each observation. ($x_{ij}/\ \bar{x}_i$, where $\bar{x}_i$ is the mean for the ith observation, is equivalent - it differs only by the number of columns, i.e. variables). For example the chemical composition of soils, the particle structure of sediments, the species composition of sites, the proportion of time spent performing different behaviours are often made more interpretable if they are expressed as proportions (or percentages) of the row, i.e. observation, totals. It also ensures that each observation is given equal weight. However, it can lead to problems with techniques that require non-singular matrices - any technique that calculates an inverse covariance matrix for example (e.g. canonical correlation, MANOVA and related techniques). Because the proportions for each observation are forced to add up to one, converting say a two dimensional data set to proportions of the observation total collapses the data to one dimension. If the data are originally 3 dimensional, converting them to proportions of the observation (row) totals collapses the space to 2 dimensions. This means that there may be *p* columns to a data matrix, but if they are proportions that add to one within each observation, the data points will all fall in a *p*-1 dimensional space so the covariance matrix will be singular and have no inverse (section 1.2.2). Such data are often referred to as compositional.

In R:

```
xx<-data
x<-t(apply(xx,1,function(z){z/sum(z)}))
```

or more simply

```
x=xx/rowSums(xx)
```

Row proportions are routinely further standardised by √column total by any procedure based on Correspondence Analysis, a popular technique in ecology and market research. This has the effect of making Euclidean distances between the row vectors into chi-squared distances. It also does the same to the transposed data matrix (i.e. after flipping the matrix so rows become columns and columns rows). This transformation has the effect of evening out the contribution made by the variables.

Transformations by mathematical function.

Mathematical transformations are used for 4 main purposes:

a) To re-express a variable on a more appropriate scale to help interpretation.
b) To improve the normality of the frequency distribution.
c) To remove heterogeneity of variances (or covariance matrices).
d) To improve the linearity of relationships between variables.

Re-expression on a more appropriate scale.

Most scientists, myself included, were suspicious of using transformations when they first met them: changing the data looks like cheating, fudging the results. (Many scientists, myself included, remain suspicious of transformations, though for different reasons - see below). It was then usually pointed out by an instructor that many commonplace measures are on non-linear scales to make them more useful and interpretable: pH, decibels, the Richter scale for earthquakes - all on the logarithmic scale; frequency is the reciprocal of wavelength; there are many others. Many variables of interest to workers are easier to display and interpret on a scale different to the one they were measured on, e.g. the duration of an occurrence can be re-expressed as a rate (reciprocal transformation), changes in raw abundance of an organism can be re-expressed as changes in proportional abundance (log transformation). Changing the data to improve its interpretability is probably the most obvious and defensible use of transformations.

Like standardisations, transformations are also used to ensure that all the variables in a multivariate data set are on similar scales. For example, if all the variables are linear measurements except one which is an area (or volume), it might be appropriate to square root the area (or cube root the volume) to bring it on to a similar linear scale to the others. Alternatively, a log transform will ensure that they are all on the same scale, this time a unit free scale of proportional change.

<u>To improve the normality of the frequency distribution.</u>

As discussed earlier many multivariate methods make the assumption of multivariate normality (MVN). Most techniques are fairly robust to this assumption, but when they are not it is usually skewness and non-linearity that cause the trouble. It is therefore sometimes useful to transform to improve the normality of the data. There are methods to transform to MVN but they are not widely available, and I shall concentrate on univariate methods. This section is therefore concerned with the normality of the marginal, univariate, distributions; linearity of the relationships between the variables (another requirement of MVN) is discussed separately below.

There are two major approaches to find the right transformation for a given data set: using rules of thumb based on the type of data being collected, or by using the data itself to suggest the appropriate function.

a) Rules of thumb.

Most statisticians are aware of the tradition that the square root transformation is used for Poisson distributed data, logarithms for log-normal data, log(X+3/8) or log(X+1) for negative binomial counts, arcsin(square root)[1] or logit (i.e. log(p/(1-p))) for proportional (binomial) frequencies and other proportions. These transformations will often be adequate to make the distribution more symmetric. However, often the data do not fall unambiguously into one of the above classes.

One thing to remember is that even the optimum transformation will not be able to normalise some data sets. Distributions of counts with many zeros, truncated distributions, continuous distributions of positive values with the mode at zero all may be irredeemably non-normal. And though it may be worth trying to make them more normal, it may be more sensible to try to find a method of analysis that does not care what shape the distribution is.

<u>To remove heterogeneity of variance.</u>

It has always been a source of wonder and delight to me that the best transformation for normalising a distribution within a single sample is often (usually) the best for ensuring the homogeneity of the variances (homoscedasticity) of a number of samples.

Like transforming for normality, transforming for homogeneity of variance is usually done on each variable separately. This is in the hope that bringing the variances into line will also improve the covariances - a pious hope, not always achieved. Still, since the truly multivariate techniques to do the job are not readily available, I will introduce the standard univariate methods. As in transforming for normality, you can use the traditional rules of thumb.

[1] I included the arcsin($p^{1/2}$) more out of respect for the tradition than because it is useful. If the sample sizes for the proportions are unequal then it will not perform as advertised; if the proportions are between 0.25 and 0.75, no transformation is usually necessary; and when the proportions are close to 0 or 1 it doesn't normalise the data particularly well anyway (nor does any other transformation). As a general rule I prefer the logit. Not because it performs better, but at least it is interpretable as a log(odds).

To improve linearity.

Many multivariate methods assume that the relationships amongst the variables are linear. This assumption is frequently violated by biological data and the consequences are often, even usually, serious. Having detected non-linearities (perhaps using the techniques of section 2.5) how can you correct the situation? The simple answer is: you may not be able to; particularly if there are many variables. There may be no set of simple transformations that can remove the non-linearities, which could be very complex. For example, linearising a multidimensional spiral is not a simple proposition. Indeed there are situations, particularly with ordinations like principal component analysis (section 5) where displaying the non-linearities is more interesting biologically than removing them before the display. However in some data sets, and with most of the modelling techniques (multiple regression, canonical correlation and related methods) the usefulness of the results can often be improved by a sensible choice of transformations.

A major problem is that while the non-linearity may be truly multivariate (like the spiral) most available methods for choosing transformations progress by attempting to improve the bivariate relationships. This is at best inefficient and at worst pointless. Improvements on one bivariate relationship may ruin the linearity of another. Be that as it may, it is sometimes worth trying different transformations to see if any overall improvements can be made.

In some situations experience has shown that a single transformation applied to all the variables will help to linearise the multivariate relationships in a data set. For example, relationships amongst morphometric variables (e.g. length of body parts) are often improved by the $\log(x)$ transformation. Linear relationships amongst organism abundances in ecological data sets are often improved by a $\log(x+1)$ transformation. These are rules of thumb that have emerged from experience, there are no doubt others.

Problems with transformations.

The dominant problem is that transformations to improve any one of: interpretability, normality, homoscedasticity or linearity may make one or more of the others worse.

Since MVN data are by definition normal, they have no relationship between the mean and the variance (and so are more likely to be homoscedastic between samples); and have linear relationships. So transformations that achieve multivariate normality often (usually) improve homoscedasticity and linearity. Even so, if you are using the techniques described in this section to improve any of the distributional properties of your data, always check the other properties after transforming; you may have made the overall situation worse.

Perhaps more important, and less recognised, is that transforming the data to improve the distributional properties will inevitably change the interpretability of the results. When the aim of the analysis is simply to display or describe the data this may not matter very much (particularly if the data is back transformed when appropriate). However, if a model is being fitted and coefficients are being examined then the interpretation can become very complicated.

Let me take a univariate example. If we do a one way analysis of variance on log transformed data, there are no problems.

The model is (leaving out the error term):

$\log(Y_{ij}) = \mu + t_i$

where μ is the grand mean and t_i is the effect due to the ith level of the treatment. This is equivalent to:

$Y_{ij} = e^{\mu} e^{t_i}$

By testing the null hypothesis: all t_i are 0, we are testing for differences between the geometric means of the original data rather than the arithmetic. But if the geometric means are different then it is extremely unlikely that the arithmetic ones will be the same (and vice versa). So the test on transformed data is telling us about the raw data. Indeed, if the log transform improved normality (or at least symmetry) then the analysis is closer to an analysis on medians and it is often useful to consider it as such (the arithmetic mean of a skewed distribution is no longer a good description of the typical or central value, a median is).

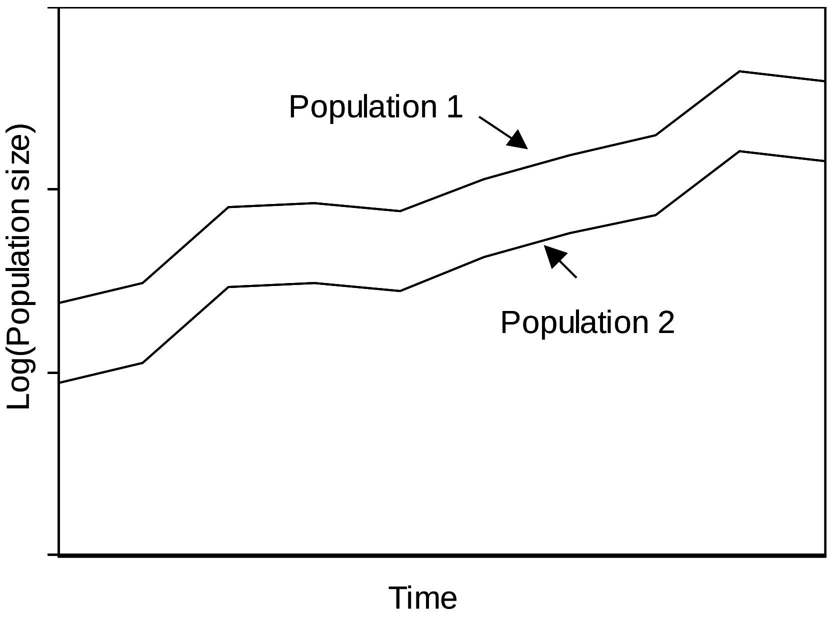
However, the situation changes if the analysis is more complicated. For example let us look at a two way analysis of variance with interaction. The model is now (again without the error term):

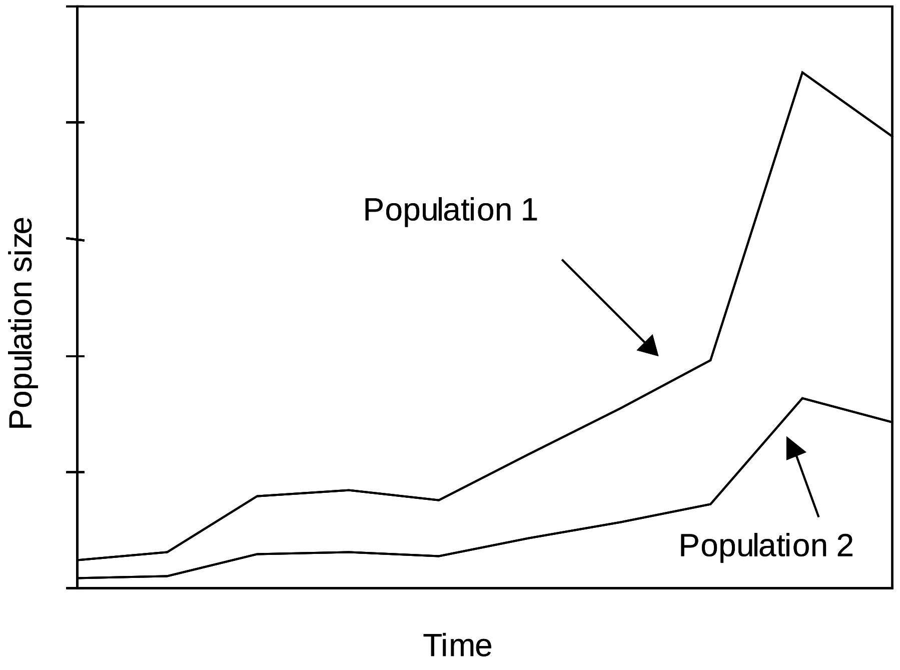
$\log(Y_{ijk}) = \mu + a_i + b_j + (ab)_{ij}$.
which implies that:
$Y_{ijk} = e^{\mu} e^{a_i} e^{b_j} e^{(ab)_{ij}}$ In other words the model is now multiplicative. If all the $(ab)_{ij}$s are 0, then the interaction term is said to be zero. However on the original scale the two factors (a and b) are still interacting in a biological sense; the effect of a_i on Y depends on the value of b_j (they are multiplicative on the original scale). Equally, if the effect of the treatment a are truly independent of those of treatment b on the original scale then analysing the data on the log scale would very likely produce a misleadingly large interaction term. If the log scale is a natural, interpretable one, then this presents no problem: the analysis is interpreted on the log scale and the results taken at face value. If however the transformation was made solely for convenience, to improve some distributional property, then the results of the analysis must be interpreted with care. The situation is even worse when the scale is less intuitive than logs, e.g. $\arcsin(Y^{1/2})$ or $Y^{-1/2}$.

Attempts to interpret the "relative importance" of variables by examining the coefficients in principal component analysis, multiple regression, canonical correlation and related techniques will only be meaningful on the transformed scales. If these are not natural and interpretable then it may be difficult or impossible to infer the role of the original, untransformed, variables.

The conclusion to be drawn from the above is that the advantages gained from transforming to improve distributional properties may be dearly bought. Interpreting the final results must be done in the context of the transformation.

MORAL. Do not transform unless you really have to or unless it enhances the relevance and interpretability of your results. It is often better to rely on the robustness of the analysis or use a non-parametric method than use arbitrary, uninterpretable, scales.





Which data set is is simpler to model?

Additional topics

Redundant variables

Most multivariate methods are adversely affected by redundant variables. A variable can be redundant for two reasons:
a) It contains no information on the structure being investigated. For example, a variable that has no information about groups in the data will get in the way in cluster analysis, multivariate ANOVA, and discriminant function analysis.
b) It contains little information that is not already carried by other variables in the data set. For example if two variables are very heavily correlated, one can substitute for the other, and there is no point in including both. Such collinear variables can have a very bad effect on some analyses and are seldom an advantage.

Unnecessary variables should ideally be dropped before the analysis. So whatever the reason for wanting to drop variables, the most informative variables must be retained.

Chapter 4. Distances and similarities.

Many of the techniques described in this course operate not on the matrix of observation vectors - the data matrix - but on a matrix of distances between the vectors. Such a matrix will contain all the information about the relative positions of the observations in multivariate space. However, deciding how 'far apart' two observations are may not be straightforward. We have already met the standard measure of the distance between two vectors in section 1 - the Pythagorean or Euclidean distance; but sometimes this measure will not convey the interesting differences between the vectors.

In fact there are an enormous number of ways of measuring the 'distance' or dissimilarity between two observation vectors. How then can one possibly decide which to use? It is difficult. The most important criterion is: does the measure agree with your intuition. When it says two observations are far apart (or close), after comparing the two data points does your intuition agree? If it does not, the results of the analysis will be uninterpretable and irrelevant. Thus the criterion for choosing a distance measure is the same as for choosing a transformation or standardisation: will it give interpretable, relevant results? Indeed, in many cases, the behaviour of a distance measure is largely determined by the standardisation it imposes on the data.

The choice of distance measure is not a trivial one; for many techniques (e.g. clustering and multidimensional scaling (MDS) - both covered later in the course) the results of the analysis may depend more on the distance measure used than on the particular method of clustering or MDS used. So if there is no obvious *a priori* choice of measure on grounds of relevance, it would be sensible to try out the analysis with more than one measure to show that the results do not depend too strictly on the choice.

While relevance to the current problem must be the dominant consideration in choosing a measure there are other factors that can be important. For example, some distance measures may behave counter intuitively with some techniques. The problem is that we live in a universe where, locally at least, distances obey certain rules, and our interpretation of distances is based on the tacit assumption that these hold. However some of the measures commonly used do not obey the rules, which can lead to problems of interpretation and also representation, how do you plot a set of points whose distances apart do not obey the rules of Euclidean distance?

The rules are:

1) If vector A equals vector B then they are zero distance apart.
2) If vector A does not equal vector B then there is a positive distance between them (you are not allowed a negative distance).
3) The distance from A to B equals that from B to A.(Distances should be symmetric).
4) The route from A directly to B must be shorter than or equal to the route that goes via any other point. (In other words the shortest distance between two points is a straight line). This is called the **triangle inequality** as it can be rephrased to state that any side of a triangle can never be longer than the sum of the other two sides

If a distance measure obeys all these rules then it is called a **metric**, if it only obeys the first 3 then it is a **semimetric** and if it violates any more it is **nonmetric**. Our Euclidean distance, the natural measure in our part of the universe, is metric. However, as most travellers have experienced, travelling time is not. The direct route between two places is often longer than one that goes via somewhere else: the "we're-in-a-hurry-so-we'll-just-try-this-short-cut" corollary of Murphy's Law.

The ideal distance measure (one that we can interpret as we do geometric distance) is however not merely metric, it should be a **Euclidean metric**. This does not mean that only the Euclidean distance fits the bill; it means that a distance matrix made up of such a distance measure can be treated as Euclidean distances and can be used to plot the relative positions of the vectors in a Euclidean space. Their geometric distances apart will exactly reflect the values of our distance measure. To avoid confusion some workers would like to call Euclidean distance (the geometric distance), Pythagorean distance, and reserve the adjective Euclidean for Euclidean metric - any distance measure that can be exactly reproduced in a Euclidean space. However, the name Euclidean distance is so well entrenched that it seems futile to try to change it now, we will simply have to live with the potential for confusion.

The advantage of Euclidean metrics is that given a matrix of distances it is possible to exactly reconstruct the relative position of the points - they may lie in a high dimensional space but there will be no distortion. This may not be possible with non-Euclidean metrics and is especially unlikely with semi- and nonmetrics: if the distance from A to B is larger than that from A to B via C, how can you plot the relative positions of the three points? It can't be done without distortion, so semimetrics and nonmetrics can be a problem with ordination techniques that try to preserve all the information in the distance matrix while reconstructing the relative positions of the observations for plotting (e.g. see section 6.1.5: principal coordinates analysis). Non-metrics are even worse. However for most purposes if the measure conveys the important differences between observations you should not worry too much whether it is a Euclidean metric or not. If you insist on worrying you can always use a nonmetric technique (e.g. nonmetric multidimensional scaling, single linkage clustering - both covered later in the course).

In the following sections I will outline a few of of the commonest measures. It would take an entire book to examine the whole range of measures that have been suggested at different times; but the vast majority of workers use a fairly restricted subset.

In fact some of the measures I discuss are more often calculated as similarities rather than distances (most analyses that accept distances also take similarities). But it is so simple to convert similarities to distances that it is easier to be consistent and refer to the distance form throughout. The commonest conversion is $d_{ij} = (1 - s_{ij})$, the one complement of the similarity, though sometimes you have to use $d_{ij} = (1 - s_{ij})^{1/2}$.

Continuous variables.

i) Euclidean distance.

$$d_{ij} = (\Sigma_k(x_{ik}-x_{jk})^2)^{1/2}$$

This is the same geometric, Pythagorean distance between two vectors that we met in Section 1 (section 1.1.1). Confusingly some statisticians apply the name to the squared distance as well, but this can lead to confusion (and different results) so I will reserve the term for the distance.

As one would expect this is a Euclidean metric, though the squared distance is not. Though it is very popular (it is the default distance in most computer packages), it has a number of characteristics that can make it dangerous to the unwary:

a) It is scale dependent, any change in the units of one of the variables could completely change the pattern of distances. For example: suppose three individuals are being compared on two variables: length (in cm) and weight (in gms).

	Length	Weight
A	10	50
B	15	100
C	20	75

Their distance matrix is therefore:

A	B	C
0	50.2	26.9
50.2	0	25.5
26.9	25.5	0

A is closer to C (26.9) than to B (50.2). If we now express the length variable in millimetres instead of centimetres, the distance matrix becomes:

0	70.7	103.1
70.7	0	55.9
103.1	55.9	0

Now A is closer to B than to C. This is particularly likely to occur if the variables are in different units. The usual solution is to transform or standardise the data first so as to make them dimensionless or their units comparable (section 2).

b) Because it squares the differences during the calculations it is very sensitive to outliers, or to variables where the size of the differences depends on their average value, e.g. weight of animals: large animals tend to have larger differences from each other than small ones. Such problems can often be avoided by using an appropriate transformation (section 3.3).

c) For some data, e.g. species abundances, Euclidean distance will seldom be appropriate. The problem lies in the way it handles double zeros. If two sites are missing the same species, they will be regarded as similar as if they had the species present in the same numbers at each site. Imagine two ends of a gradient, perhaps a transect down the shore in an intertidal community. Sites at the top and bottom of the shore will be missing the species from the mid-tide zone, should they be regarded as more similar as a result? Ecologically, of course not. The community above the high tide mark is ecologically more similar to the mid tide zone than to the subtidal; using Euclidean distance could obscure that fact. For this reason, few ecologists knowingly use the Euclidean distance (without some appropriate transformation) - though anyone who uses principal component analysis (section 6) does so by default.
If the mutual absence of a variable or attribute conveys no information on the difference between the observations then the Euclidean distance is not appropriate. There are a number of standardisations that removes the effect of double zeros on the Euclidean distance. But we will not discuss them in this course.

Programming notes:

In R.

```
x<-data
d<-dist(x,method="euclidean")
```

ii) Manhattan distance.
a.k.a. the city-block or taxi-cab distance, or the L_1-norm.

$$d_{ij} = \sum_k \left| x_{ik} - x_{jk} \right|$$

Or if the value is averaged over the *p* variables:

$$d_{ij} = 1/p \ \sum_k \left| x_{ik} - x_{jk} \right|$$

it sometimes called the mean character difference or the unrelativised Czekanowski coefficient.

It is metric but not Euclidean (though if the variables are standardised by the range - Gower's quantitative distance - it is). It is one of the most widely used measures in all branches of biology, social sciences or market research. It is quite frequently used in ecology, because it has the attractive feature that it will change the same amount if two sites differ by 2 individuals in 1 species as if they differ by 1 individual in 2 species. Thus individuals in the different species are all treated as equivalent. Since common species tend to vary over sites more than do rare ones this measure is sensitive to abundant species; though since the differences are not being squared it is less sensitive than the Euclidean distance. Like the Euclidean distance it is scale sensitive so it should only be used on variables with comparable units. Also, like the Euclidean distance, the Manhattan distance incorporates double zeros, so it should not be used where joint absences are uninformative.

Programming notes:

In R.

```
x<-data
city.matrix<-dist(x,method="manhattan")
```

Binary Variables.

Most measures comparing binary observation vectors are similarities, but all those I present here can be converted to distances by (1-s_{ij}).

A comparison of two observations A and B on a binary variable leads to one of four outcomes: 1 in both, 0 in both, 1 in A but 0 in B, and vice-versa. Accumulating these comparisons over all the variables leads to a two way table. To simplify the formulae I shall follow convention and use the following terminology

		Observation A	
		1	0
Observation B	1	*a*	*b*
	0	*c*	*d*

The number of variables compared is therefore a+b+c+d.

i) Simple matching coefficient.

a.k.a. Sokal and Michener's association coefficient.

$s_{ij} = (a + d) / (a + b + c + d)$,
i.e. the probability of the observations being in the same state on a randomly chosen variable. It is particularly appropriate when the variables being compared are two-state qualitative variables, like male/female, or black/white, rather than presence/absence where the meaning of double absences can become a problem (see section 4.1.1a). It is identical to the average Euclidean (or manhattan) distance between the binary vectors.

Programming notes:

In R. Simple matching distance cannot be got directly, we have to turn the data into binary form and use Manhattan distance. Work out why for yourself.

```
x<-as.matrix(data)
x[x>0]<-1
simple.matrix<-dist(x,method="manhattan")/ncol(x)
```

ii) Jaccard's coefficient.

$s_{ij} = a / (a + b + c)$,

i.e. the probability of a randomly chosen variable being present in both observations, ignoring double absences. The distance (1-s_{ij}) is equivalent to Gower's distance (ignoring double zeros - section 4.1.1 c) applied to the binary data. It is clearly appropriate when all the variables are presence/absence and mutual absences are uninformative. The one complement, (1-s_{ij}), is metric but not Euclidean. Because it is insensitive to double zeros it has been widely used in ecology and animal behaviour. However this problem - uninformative double-zeros - is actually very common and so perhaps this measure should be more commonly used than it is.

Programming notes:

In R.

```
x<-as.matrix(data)
jacc.matrix<-dist(x,method="binary")
```

Some revision questions

1) What are the rules defining a metric?

2) Which rules don't apply to semi-metrics?

3) Which rules don't apply to non-metrics?

4) What are the problems with Euclidean distance?

5) What is the city-block distance?

6) What is the difference between Jaccard's distance and the simple matching coeffient?

Chapter 5. Principal Components Analysis

Principal Component Analysis (PCA) is a technique for the analysis of an unstructured sample of multivariate data. Its aim is to display the relative positions of the observations in the data cloud in fewer dimensions (while losing as little information as possible) and to help give insight into the way the observations vary. It is not a hypothesis testing technique (like t-test or Analysis of Variance); it is an exploratory, hypothesis generating tool that describes patterns of variation, and suggests relationships that should be investigated further.

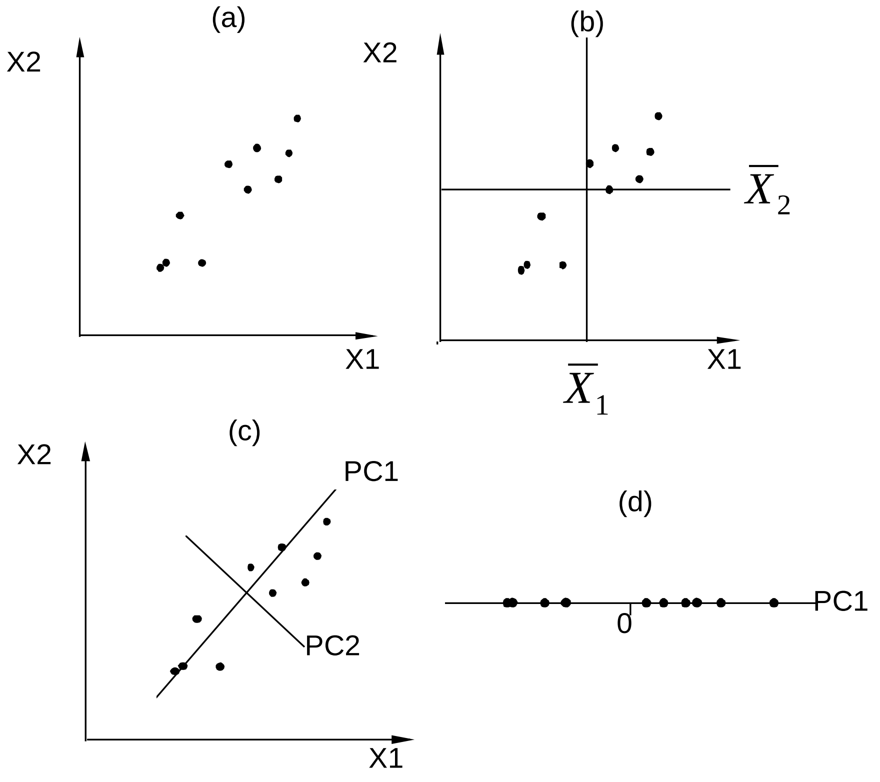
PCA is a member of a family of techniques for dimension reduction (ordination). I have chosen to give PCA a chapter on its own because, while relatively easy to understand, it provides a good introduction to many other more complex methods. Anyway, its wide use is sufficient justification on its own. Other common ordination techniques are described in Chapter 6.

The word ordination was applied to dimension reduction techniques by botanical ecologists whose aim was to identify gradients in species composition in the field. For this reason they wanted to reduce the quadrat ´ species (observations ´ variables) data matrix to a single ordering (hence ordination) of the quadrats which they hoped would reflect the underlying ecological gradient.

An Intuitive Explanation.

The aim of PCA is to reduce the dimensionality of the data, and to help us to say something about the patterns of variation. How can this be done? Consider figure 5.1. Here we have an imaginary data set of 10 observations on 2 variables that conform, roughly, to a multivariate normal distribution. How can we re-express these data in fewer dimensions (i.e. one), while at the same

Figure .5.1 The geometry of PCA. a) the data cloud. b) the new axes are shifted to the centroid. c) the axes are rotated to lie along the major axes of the data. d) The data are projected onto the first component axis, giving a reduced space (1 dimensional) plot.

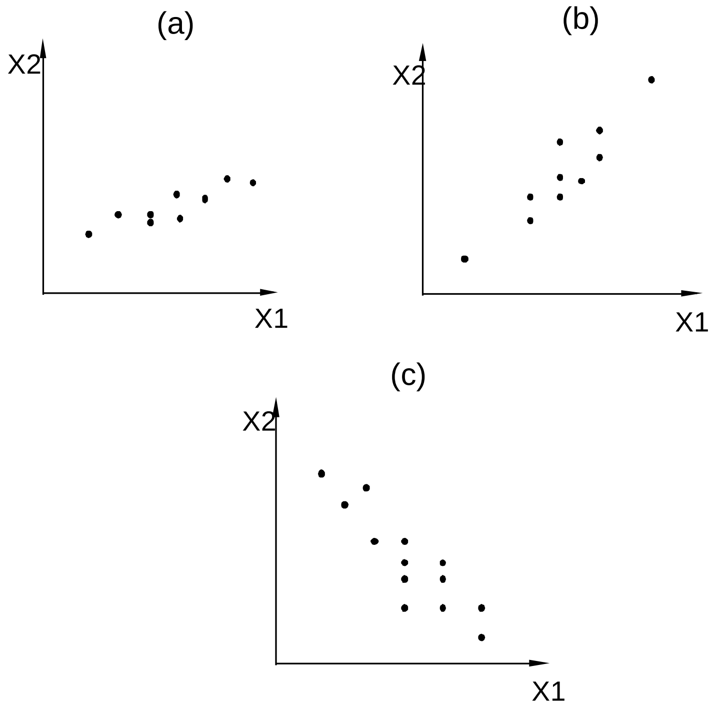


time keeping their relationships to each other as undistorted as possible? It is actually very easy, even by hand. First we shift the origin of our present axes to the centroid ($\bar{X}_1$, $\bar{X}_2$)of the data cloud (fig.5.1b) - centering the data. Now all the observations are in terms of deviations from the mean, and have means of zero. Now we trace these axes on a transparent sheet, and rotate the tracing till one axis lies along the main axis of the ellipse (fig 5.1c). Then we mark off the position of every observation onto this new axis. If we simply look at this new axis on its own (fig 5.1d) we see that the relative positions of the observations on it are quite close to those of the original data. In other words we have effectively reduced the data from 2D to 1D while leaving their relative positions largely unchanged.

How does this help us interpret the patterns of variation present in the data? Which variable contributes most to the trend? Figure 5.2a shows a situation where variable X_1 is responsible for the major variability in the data. In figure 5.2b X_1 and X_2 contribute roughly equally. Both variables also contribute equally in figure 5.2c, though here they are negatively related. So by looking at where the new axis lies relative to the old ones we can say something about how our variables vary together. We can describe the major trend in the data.

All this is rather trivial in two dimensions, but the same process can be done in spaces of higher dimension. Though you won't use transparent sheets to rotate the axes, the idea remains the same - it is just rather harder to visualise. To help, let us extend the idea to three dimensions. Imagine a data cloud that looks like a loaf of French bread: long, thin, and slightly flattened. We now relocate and rotate our axes till one lies along the main line of the data, as though we pushed a skewer down the length of the loaf. This identifies the dominant trend in the data. We then use that line as our axle, and rotate the two remaining axes till one lies along the next longest line of the data, i.e. the width of the loaf. The line running down the length of the data we call the first **component** and it identifies and describes the major trend in variation of the data, in the same way that the component

Figure 5.2. Major trends in the data: a) X1 and X2 positively related, X1 contributing most. b) X1 and X2 positively related, both contributing equally, c) negative relationship, equal contributions.

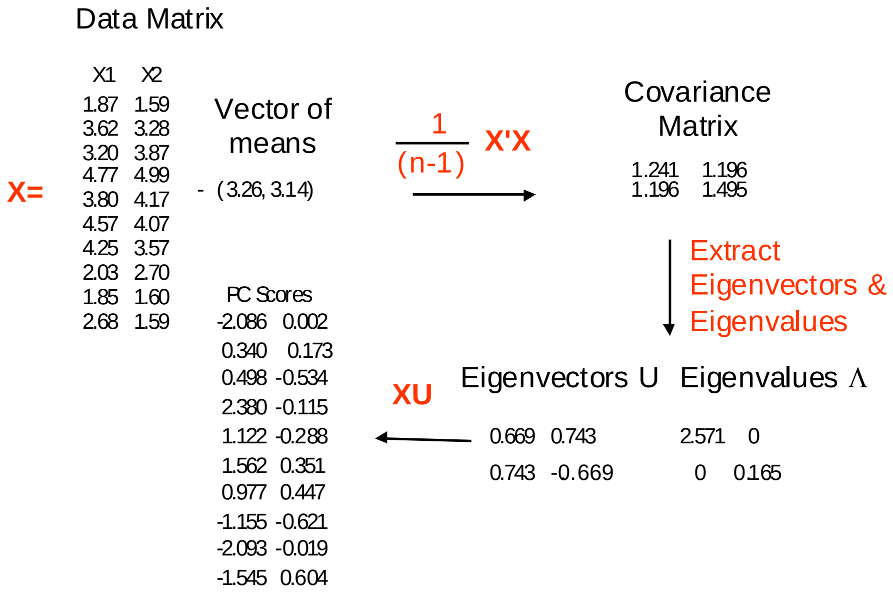


did in Figure 5.1. The line that runs across the width of the data describes a second trend in the data. This is the second component. The third axis, running at right angles to the first two, is called, unsurprisingly, the third component. If we wished to draw the data in two dimensions, while maintaining their relative positions, we could plot them on the first two of the new axes and get quite a reliable picture. In fact, provided the loaf was long and thin enough we would probably get quite an adequate representation if we only used the first component. Of course if the loaf was round like a ball, i.e. no correlation between the variables and equal variances, then there would be no way of getting a good picture in fewer dimensions.

Of course, actually calculating the positions of each observation on these new axes is not a trivial exercise. We need to first identify the position of the new axes and then get the score of each observation on each axis. If you worked through the matrix algebra in chapter 1, I can use the intuitive approach used there to show how it's done. The shape of the data cloud is summarised by the variance covariance matrix (section 1.4.1). What we want is a set of vectors that identify the major axes of this data cloud. These vectors are clearly *latent* in that matrix and *characterise* the structure of the cloud. If you look back at section 1.3.2 I am clearly hinting fairly broadly that the vectors we want are the latent, characteristic or eigenvectors. These vectors identify the new axes and allow us to calculate the positions of each of the observations.

The position of an observation **x** on the first, most important new axis is simply given as $\sum_i u_{i1}X_i$, a linear combination of the X variables. Multiply the value of each variable by the corresponding element (u_{i1}) of the first eigenvector and add them all up. The resulting number is the **principal component score** of that observation for component 1. By using the eigenvectors for the second axis, third etc., we can get the position of the observation on all the new axes (of course we will only be interested in the more important axes).

If the eigenvectors identify the directions of the major axes of the data cloud what do the eigenvalues do? This is easily understood if we remember that the sum of the eigenvalues $\Sigma\lambda_i$ = trace of the matrix, i.e. Σm_{ii}. What is the trace of a variance covariance matrix? It is the sum of all the variances, the total variance. So the sum of the eigenvalues adds up to the sum of the variances. The eigenvalues are actually giving us the variance of the scores of the observations on each of the new axes. The first component lies along the axis of the data cloud that will have the largest amount of associated variance. The variance of the scores of each observation on the line in fig. 5.1d is clearly going to be large; and is measured by the eigenvalue associated with that eigenvector. The total variance is being partitioned into bits associated with the new axes. The largest bit with the first new axis, the first principal component; the next largest bit with the next most important axis, and so on.



Data Matrix

$$X= \begin{array}{cc} X1 & X2 \\ 1.87 & 1.59 \\ 3.62 & 3.28 \\ 3.20 & 3.87 \\ 4.77 & 4.99 \\ 3.80 & 4.17 \\ 4.57 & 4.07 \\ 4.25 & 3.57 \\ 2.03 & 2.70 \\ 1.85 & 1.60 \\ 2.68 & 1.59 \end{array}$$

Vector of means

- (3.26, 3.14)

$$\frac{1}{(n-1)} X'X \longrightarrow$$

Covariance Matrix

$$\begin{array}{cc} 1.241 & 1.196 \\ 1.196 & 1.495 \end{array}$$

Extract Eigenvectors & Eigenvalues

Eigenvectors U

$$\begin{array}{cc} 0.669 & 0.743 \\ 0.743 & -0.669 \end{array}$$

Eigenvalues Λ

$$\begin{array}{cc} 2.571 & 0 \\ 0 & 0.165 \end{array}$$

XU

PC Scores

$$\begin{array}{cc} -2.086 & 0.002 \\ 0.340 & 0.173 \\ 0.498 & -0.534 \\ 2.380 & -0.115 \\ 1.122 & -0.288 \\ 1.562 & 0.351 \\ 0.977 & 0.447 \\ -1.155 & -0.621 \\ -2.093 & -0.019 \\ -1.545 & 0.604 \end{array}$$

The basic steps of a principal components analysis, performed on the data from figure 1.

The matrix of PC Scores is calculated by XU

$$\mathbf{X} = \begin{array}{cc} -1.394 & -1.553 \\ 0.356 & 0.137 \\ -0.064 & 0.727 \\ 1.506 & 1.847 \\ 0.536 & 1.027 \\ 1.306 & 0.927 \\ 0.986 & 0.427 \\ -1.234 & -0.443 \\ -1.414 & -1.543 \\ -0.584 & -1.553 \end{array} \qquad \mathbf{U} = \begin{array}{cc} 0.669 & 0.743 \\ 0.743 & -0.669 \end{array}$$

PC1 score for obs 1

(0.669*-1.394) + (0.743*-1.553)

= -2.086

PC2 score for obs 1

(0.743*-1.394) + (-0.669*-1.553)

= 0.002

*j*th Observation's score on *i*th component
$= a_{i1}X_{j1} + a_{i2}X_{j2} + \ldots a_{ip}X_{jp}$
where *a* is element of eigenvector

Standardisations and transformations.

A major problem with PCA is that the components are not scale invariant. That means if we change the units in which our variables are expressed, we change the components; and not in any simple way either. So, every scaling or adjustment of the variables in preparation for the analysis could (and usually does) produce a separate component structure. As I showed in section 4 sensible pre-treatment of the data by standardisation or transformation can often increase the interpretability and biological relevance of the results. It is therefore important to choose a standardisation or transformation carefully. In particular PCA will give different results depending on whether we analyse the covariance matrix, where the data have merely been centred (corrected for the column, variable, mean), or the correlation matrix, where the data have been standardised to *z*-scores (centred and converted into standard deviation units). This is particularly important, as many computer programs to do PCA automatically analyse the correlation matrix . If you do not want that standardisation; you may have to explicitly ask for the covariance matrix. As you would expect, the results from the two analyses will usually be very different.

How many components should you keep?

Since the interpretation of a full PCA can take a lot of effort, it is often worth using the techniques of section 3 to check for trends and correlations in the data before investing time in doing a PCA.

If you decide to do a PCA one of the first decisions you are faced with is how many components to use. PCA attempts to reduce dimensionality and/or identify trends; so you will want to look at fewer dimensions than there are variables (p). But PCA extracts as many components as there are variables (unless p, the number of variables, is more than n, the number of sampling units; in which case you should probably use another technique, e.g. principal coordinates, see section 6.1). However, the components are selected in sequence as the best descriptors of the data (subject to the constraints mentioned earlier). The last few components usually won't account for much of the variance (information), so, if you ignore them you should lose little. But how many should you drop? What amount of variance is small enough to ignore? We need a cut-off point.

An example may illustrate the problem. Suppose we have 4 variables but only 2 trends (linear and orthogonal - wishful thinking but it's only an example). Once the first 2 components have identified the trends, the remaining variation is amorphous, random - in geometric terms, spherical. There are, therefore, no major axes to be identified, no trends left, but PCA still identifies 2 more components. Their direction will be arbitrary, determined by sampling variation, and the variances that they explain will be small and about equal. They can tell us nothing about the major trends of variation in the data, and we would usually want to forget about them. These 'residual' components can sometimes be useful in a regression context but most of the time we will want to ignore them.

Unfortunately the problem of how many components to use doesn't have a standard, rigorous solution. We will meet similar situations in other analyses. There is a general absence of appropriate significance tests. This is probably a good thing; too slavish an adherence to significance tests can blind an investigator to what the data is actually saying. However it makes interpreting the results of an analysis inevitably appear somewhat subjective. To a large extent, therefore, we resort to rules of thumb and various approximate devices, and accept that most multivariate techniques suggest rather than test hypotheses. It is still more unfortunate that there is no general agreement in which rules of thumb to use.

Probably the most obvious method of selecting which components to use or interpret is to look at the first component - the one with the largest eigenvalue. It describes some amount (λ_1) of the total variation ($\sum\lambda_\mathrm{i} = \sum s_i^2$). Do we consider this amount a sufficient percentage of the total? If not, we consider the second component. In combination with the first it will explain some larger percentage

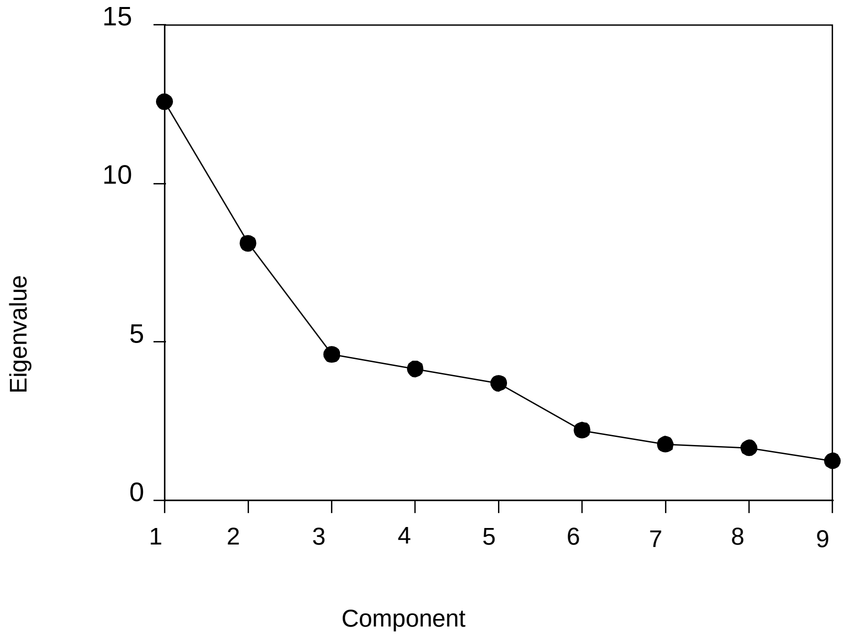


Figure 5.3. Scree diagram from the log(X+1) transformed microzooplankton data

- is this sufficient yet? And so on till we have explained a sufficient percentage of the variation. The obvious problem is - what is sufficient? In the literature this ranges from 75% to 90%. Morrison(1976) suggests 75% and not more than 4 or 5 components. Timm(1975) suggests that experimenters should be satisfied if no more than 5 or 6 components explain 70 to 80%. Pimentel(1979) suggests 80-90%, but keep any others that make sense. While Mardia et al(1979) suggest 90%. As you can guess, these will usually give very different sized sets of components, particularly if the number of variables (*p*) is large.

The other suggested methods also give a wide variety of results. Some suggest using only those components that have eigenvalues greater than the average ($\sum_i \frac{\lambda_i}{p}$). If you analysed the correlation matrix this average will always be 1, (confirm this yourselves - hint: $\sum\lambda_i$ = trace of the correlation matrix). This cut off point (**Kaiser's criterion**) tends to retain too few components when *p*<=20; but in most cases that is no bad thing and it is a widely used technique.

.

Another potentially useful way of identifying a good cutoff is using a "**scree graph**". If we plot λ_i against *i* (figure 5.3) we get a declining curve, which may allow us to spot where "large" eigenvalues end and "small" ones begin. This technique can apparently lead to including too many components for most purposes. So a compromise between this and Kaiser's criterion is sometimes used by analysts.

In a study that compared these three methods on data with a large number of variables (*p*>= 36): Kaiser's criterion, the scree graph and significance tests, Kaiser's criterion retained too many components but the scree graph performed well (provided the user was experienced).

Finally there is the pragmatic attitude: use as many components as give an interesting result, interpret as many as give a reasonable story. PCA is generally a hypothesis generating technique, so there is no harm in exploring the data space as much as you like to suggest ideas; and speculate all you wish on the meaning of the components.

AUTHOR'S CHOICE: For data exploration a scree graph is often useful. If it includes too many components it doesn't matter. Look at them anyway.

EXAMPLE 5.1

Table 5.5.1 Eigenvalues and cumulative proportion of variance summarised for the PCA on zooplankton (log(X+1))

PRIN1	12.5608	0.31435
PRIN2	8.1066	0.51722
PRIN3	4.5990	0.63231
PRIN4	4.1533	0.73626
PRIN5	3.6865	0.82851
PRIN6	2.2047	0.88369
PRIN7	1.7687	0.92795
PRIN8	1.6431	0.96907
PRIN9	1.2359	1.00000

I performed a PCA on the covariance matrix of the log(X+1) transformed microplankton data. The eigenvalues and the cumulative percentage variance explained are shown in table 5.1 and the scree graph is in figure 5.3. Let us apply some of the rules of thumb I outlined above. If we use the cumulative percentage variance explained then we find 70% would give us 4 components, 80% gives us 5 and 90% 7. The scree graph suggests 2 or 5, Kaiser's criterion 3. This disagreement is quite standard, and should not worry you. I personally would use 2, I put more reliance on the scree graph than the other methods and always tend towards fewer rather than more. I might still look at the other components but would be careful to keep a very sceptical attitude to them. I would still be fairly sceptical about the two I retain - they will be suggesting patterns not stating truths.

Programming notes.

In R when using multivariate methods you may first want to enter library(MASS). It may not be necessary, it depends on the installation.

```
x<-dataset name
prin<-princomp(x)
```

If you want to do it on the correlation matrix do princomp(x,cor=T)
The eigenvector coefficients are in prin$loadings, the scores in prin$scores, and the square roots of the eigenvalues in prin$sdev.

To get a screeplot screeplot(prin,type="lines")

To get the eigenvectors printed out neatly:

```
print (prin$loadings, cutoff=0, digits=3)
```

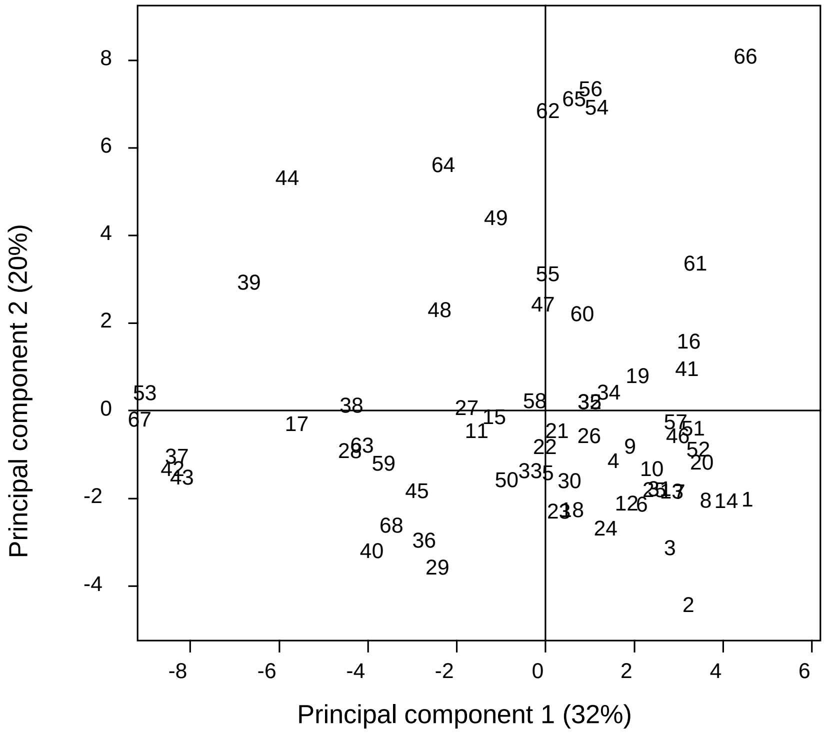


Figure 5.4. Plot of the zooplankton samples in the space defined by the first two principal components. The numbers are the sample number. Pretty uninformative isn't it?

Displaying the observations in fewer dimensions.

One of the main uses of PCA is to reduce the dimensionality of the data so that the relative positions of the observations can be examined and patterns and trends identified. If you have decided that the first *k* components are likely to preserve the important information, then use the proportion of variance explained $\frac{\sum_{j=1}^{k}\lambda_j}{\sum_{i=1}^{p}\lambda_i}$ as a measure of goodness of fit. If you are content that the first 2 components are the only interesting ones then a simple scatter plot of the observations' positions on the components (the component scores) is all that is necessary; the relative positions of the observations in the plot will show their relative positions in the full space.

If the dimensionality of the data is greater than 2 then it may be necessary to do more than one scatterplot, say by plotting component 1 against component 2, 1 against 3, and 2 against 3 and so on. A very powerful alternative approach is to use the Brush-and-spin techniques described in section 2.2.2. These allow you to look at 3 dimensional plots of the data from all directions and explore the trends that (you hope) will emerge.

Programming notes

In R:

```
eqscplot(prin$scores[,1:2])
```

The bubble plots we want the area to be proportional to the variable. So we must plot bubbles whose radius is the square root of the value.

In R we can get a bubble plot using just:

```
var.scaled<-apply(dataset,2,function(z){z<-5*(z-min(z))/diff(range(z))+.5})
eqscplot(prin$scores[,1:2],cex=var.scaled[,3])
```

`var.scaled` contains the variables scaled to produce sensible sized bubbles. The `cex=` option requests that variable (in this case the 3rd column) to be plotted as bubbles.

EXAMPLE 5.2
Figure 5.4 shows the reduced space plot for the first two principal components from the log(X+1) transformed plankton data. At first sight there are no great biological insights to be gained from the plot, the only apparent trend is that the bulk of the observations have high values of component 1 and low values of component 2. It must also be remembered that this plot only summarises 52% of the variation, so the picture we get may not be very reliable; this is typical of real ecological data (and many other kinds) - we seldom do much better. Real insight will usually be gained (if at all) when we investigate the relationship between the plot and the data.

Interpreting apparent trends.

Looking for trends in the observations

The presence and nature of trends in the data can often be simply displayed by adding information to the basic reduced space scatter plot. Which method you use will depend in part on how crowded the plot is; how many plots you are prepared to do; and how much extra information you have. In any case the ease of producing these plots on computers means that even a large number of plots can be scanned in a relatively short time. So, if you can, try all the following methods and see which bring out the trend(s) most obviously.

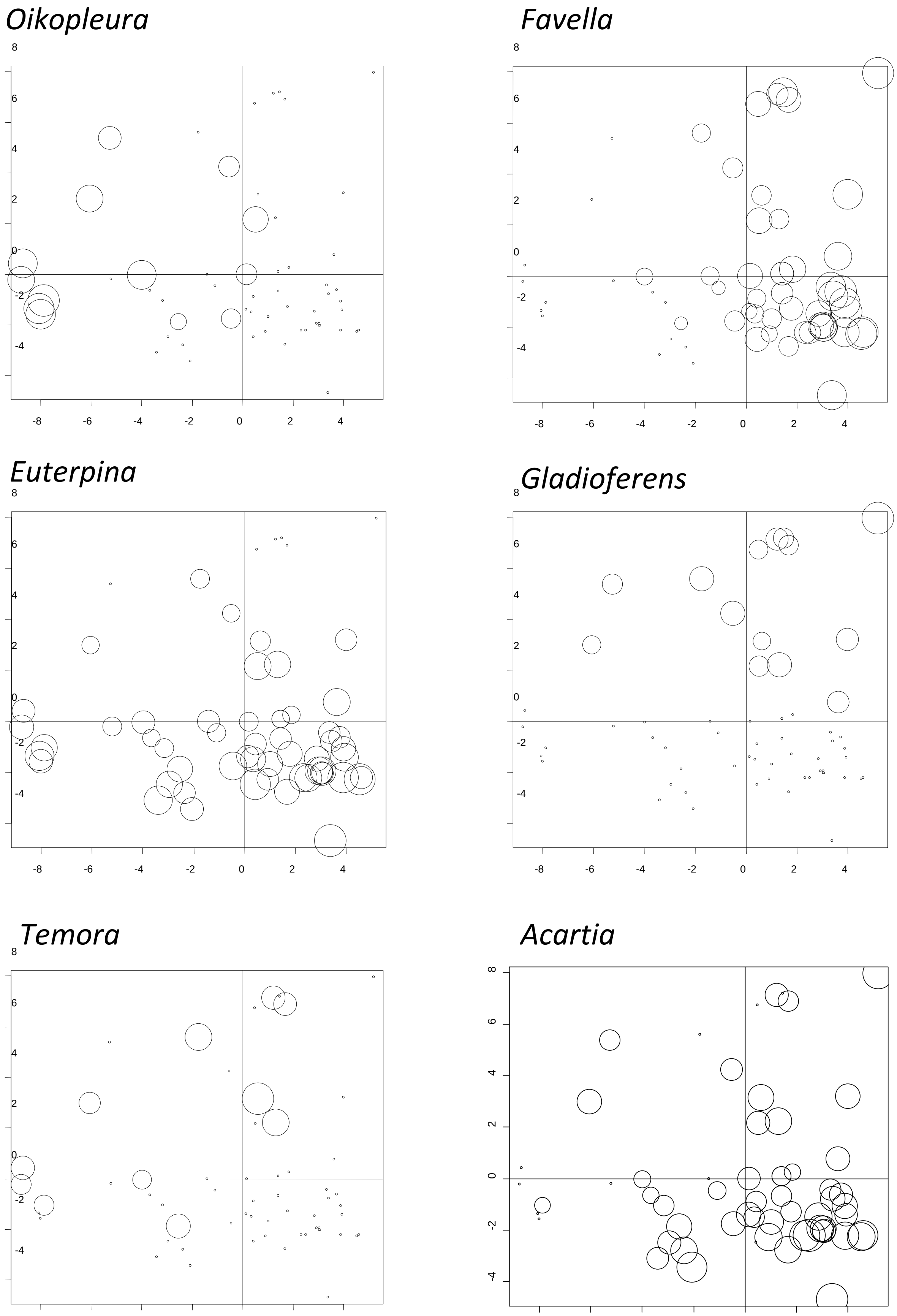
i) Label the observations.
Sometimes the observations have names or identities that are meaningful. Plotting these labels instead of the plotting symbol in your reduced space scatter plot can make trends more obvious. Identifying which observations are close to which other ones is an obvious way to spot pattern. Unfortunately a crowded plot can be made almost indecipherable if labels are added. This is can be avoided by using Brush-and-spin programs where individual observations or groups of observations can be highlighted interactively and their labels shown. The labels can then be hidden again to avoid overcrowding the plot.

ii) Include the original variables.
Bubble plots: A simple way of identifying trends in the original variables is to make the plotting symbol of the reduced space scatter plot reflect the values of one of the variables. For example make the radius of the plotted circle proportional to the value of the variable – so they look like bubbles. If observations in one part of the plot have large values (large bubbles) and in another have small, a trend is obvious. Of course this means that there will be one plot for each variable, which could be a lot of plots. But the speed of modern computers and the effectiveness of the human eye at detecting pattern makes this a relatively simple job even for large data sets.

EXAMPLE 5.3

If we plot the plankton species as bubbles in the reduced space plot some interesting trends emerge. The best six are presented in Figure 5.5. Clearly there are strong trends in the observations which are clearly visible from the plots. I tentatively identify 3 basic trends: One running from left to right - *Favella* and *Oikopleura*, one from top to bottom - *Gladioferens* and *Euterpina*, and finally one running from top left to bottom right - Harpacticoids and *Temora*. It will require more information to determine what these trends mean, but I would suggest that these diagrams make a good start. They are certainly more useful than the plot using just observation number (Figure 5.4)

Oikopleura


Oikopleura
Favella
Euterpina
Gladioferens
Temora
Acartia
8
6
4
2
0
-2
-4
-8
-6
-4
-2
0
2
4

iii) External variables.
In many situations there is more information available than went into the PCA. As a general rule your analysis should be chosen to take advantage of all the information you have available, but sometimes for one reason or another some is held back. You may be able to identify or interpret trends in the observations by superimposing this information on the plots.

Programming Notes.

To name the sample units in a plot (i.e. plot a label on each point) in R

use:

```
eqscplot(prin$scores[,1:2],type="n")
text(prin$scores[,1:2],label=as.vector(names))
```

In R if you want to put observation numbers in then use:

```
numbers<-as.character(1:nrow(prin$scores))
eqscplot(prin$scores[,1:2],type="n")
text(prin$scores[,1:2],label=as.vector(numbers))
```

EXAMPLE 5.5
Figure 5.7 shows the plot when we identify the site from which each sample came. Clearly there is a split, albeit not a clean one, between the observations from Whau Creek (W) and those from Mangere (M). The Mangere observations seem to be largely concentrated in the lower right corner of the plot. These differences seem to coincide with the trend in the numbers of *Gladioferens* and *Euterpina*.

Interpreting the axes as trends.

Though PCA is primarily concerned with displaying trends among the observations, your understanding of these trends can sometimes be enhanced by attempting to interpret these new axes, the components, as new variables each with a biological interpretation. After all they do lie along the axes of major variation that are most likely to be due to real trends in the data. Interpreting the components in this way, sometimes known as **reification**, - turning the component into a thing (*res* in Latin) - is a virtually automatic part of any PCA, and can often be useful; though the resulting 'entities' should not be taken too seriously. To be useful, these new, synthetic variables should have a plausible biological identity; at the very least finding out which variables vary most along a particular axis can help interpret trends apparent in the plots.

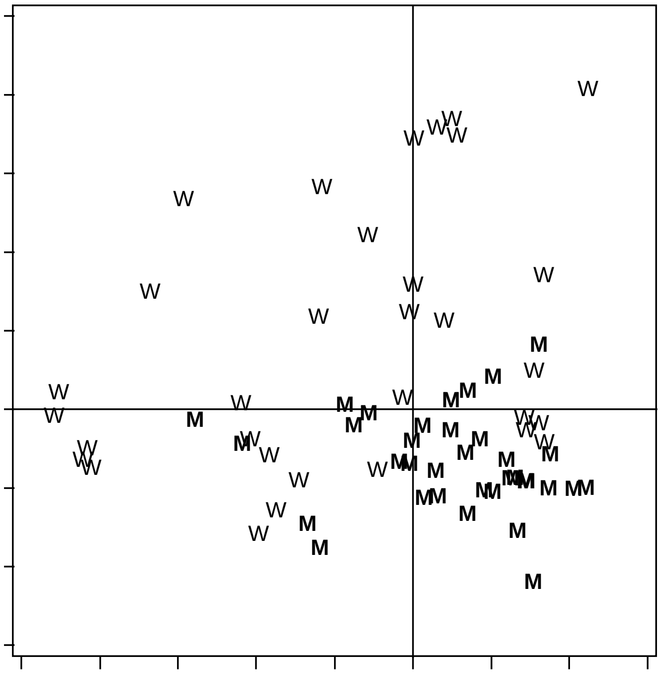


Figure 5.5. Using external variables to aid the interpretation of trends. The main cluster of observations in the bottom right are largely from Mangere, Whau Creek observations are far more variable. The major trends are probably related, at least partly, to organic and nutrient enrichment from the Mangere sewage oxidation ponds.

There are three main ways of trying to interpret the new axes.

i) Interpreting the coefficients.
Having decided which components to use, it is common to try to interpret each of them (the process of reification). You hope they will correspond to some easily identified phenomenon If these new, synthetic, variables have simple and useful interpretations they can be used on their own in further analyses and data displays. This is probably best done by attempting to identify the variables that are most influential in determining the position of the component. We do this by looking at the coefficients (u_{ij}) of the eigenvectors determining the components. These coefficients are usually **normalised to one**; i.e. presented scaled to unit length (that is the sum of their squared values is one). They give the relative separate contribution of each variable to the component score.

When looking at the u_{ij}s how do we select those variables that have 'significant' influence on the position of the component? There are no formal tests; just more rules of thumb. If we are lucky, one or more variables will have 'large' (in absolute value) coefficients on a component that are clearly distinct from the 'smaller' values for the remaining variables; in which case it's easy. However, often there will be a graded series of values (no group of variables clearly dominating); so either you come up with an interpretation that includes all the variables, or you impose an arbitrary cutoff value. There are few guidelines on where this cutoff should go. It's usually a matter of experience; but there are a couple of potentially useful values that can be used.

The first is the coefficient the variable would have had if all the variables had contributed equally to this components. Called the **equilibrium contribution** it usually takes the value of $1/\sqrt{p}$, where p is the number of variables. It has little theoretical justification, it just happens to provide a cut off value in roughly the right place. It is basically identifying "above average" coefficients. This method will not work when the component is a “growth” component – i.e. where all the eigenvector coefficients are of much the same size. In this case all those below the average will be dropped even though they are large. An alternative method that works better in that situation is a rough cut off point: say 0.7 times the largest coefficient in an eigenvector. Though totally *ad hoc* this value (I call it Mardia's criterion) seems to work quite well.

EXAMPLE 5.6.
Table 5.2 gives the normalised eigenvector coefficients (the u_{ij}s) for the first three components of the plankton data. (Please remember that there are in fact 9 components in all).

Let us try to interpret the first component. The equilibrium contribution will be $1/\sqrt{p} = 1/\sqrt{9} = .333$. Mardia's criterion will give us 0.7 x 0.774 = 0.54. The equilibrium contribution suggests that the major trend in the data is determined by the abundances of *Favella*, *Oikopleura*, and *Acartia*. Mardia's criterion identifies *Favella* alone as the important species for separating the observations. In fact, as we have seen in Figure 5.5 the *Favella-Oikopleura* trend is very clear; but it is not obvious that *Acartia* is part of this - it seems more closely related to the Harpacticoid trend. The interpretation of a principal component can often be better understood if we remember that the score an observation has on the *j*th PC = $\Sigma_i u_{ij} x_i$ (section 5.3 ii). So, if we consider only the largest coefficients we get:

PC score = 0.77xlog(*Favella* density+1) - 0.45xlog(*Oikopleura* density+1) + 0.37xlog(*Acartia* density +1).

All the other coefficients are considered to be effectively zero. Clearly a large value for *Favella* will give a large value for the PC. A large value for *Oikopleura* will tend to give a large negative PC score. We can also reverse the logic: a large PC score implies large amounts of *Favella* (and/or *Acartia*) but a small value for *Oikopleura*. Similarly large negative PC scores are only possible if there are many *Oikopleura* and few *Favella*. We might therefore conclude that component one is identifying a trend between sites containing large numbers of *Favella* and small numbers of *Oikopleura* (giving positive values on the PC axis) and sites with few *Favella* but lots (relatively) of *Oikopleura* (sites with negative values on the PC axis). Such a component is called a bipolar component because it identifies a contrast: where there are *Favella* you do not find *Oikopleura* and vice-versa. A bipolar component is easily identified by looking at the eigenvector coefficients. If some of the important coefficients are positive and some negative then a contrast is involved - think about it.

If we try to interpret component 2 we identify a bipolar component contrasting the densities of *Gladioferens* (0.68) and *Euterpina* (-0.59) - both criteria agree on these two. Sites with large

Table 5.5.2. The eigenvectors associated with the first 3 principal components of the zooplankton data.

	PC1	PC2	PC3
ACARTIA	-0.374	-0.203	0.636
EUTERP	-0.01	-0.601	0.078
GLADIO	-0.068	0.682	0.38
HARPACT	-0.065	-0.236	0.506
OITHONA	-0.129	0.075	-0.178
PARACAL	0.042	-0.04	0.02
TEMORA	0.163	0.217	0.304
FAVELLA	-0.774	0.136	-0.194
OIKOPL	0.455	0.067	0.163

positive scores on the 2nd PC axis will tend to have lots of *Gladioferens* and few *Euterpina*. Those with large negative values will tend to have many *Euterpina* and few *Gladioferens*.

Clearly, interpreting the components from their eigenvector coefficients gives us no more than was apparent from the bubble plots of figure 5.5. Indeed, arguably it gives less. This will usually be true. However, when there are many variables, scanning the coefficients may be an efficient way of suggesting which variables to plot.

NOTE: Some computer packages when they perform PCA do not label the component (eigenvector) coefficients as such in the output. They call them factor or score coefficients, a relic of the bad old days when PCA was seen as a form of Factor Analysis (see later in this chapter). princomp() calls them loadings.

ii) Interpreting the component-variable correlations.

There is another, less direct, approach to the problem of interpreting the components. We can characterise the component by identifying the variables that vary closely with it. In essence we are not identifying the variables that determine the value of the component, we are identifying variables that are well described by the component. We can then try to use them to infer an identity for the component. For this interpretation we simply get the correlation coefficient (c_{ij}) between variable x_i and the scores on the jth component (the component correlation). I refer to them as c_{ij} to avoid confusion with the elements r_{ik} of the correlation matrix. The set of component correlation coefficients is sometimes called the factor structure, a singularly uninformative designation derived from psychology. They can be calculated from the eigenvector coefficients by $u_{ij}\sqrt{\lambda_j}/s_i$. When we are using coefficients from a PCA on the correlation matrix, the variables have already been standardised into standard deviation units; so this standardisation leaves the $u_{ij}\sqrt{\lambda_j}$ unchanged. In this case the interpretation will be the same as for contribution to the component, as described above. However, when we have left the variables unstandardised, (we did the PCA on the covariance matrix **S**), the c_{ij}s and the u_{ij}s will often (usually) produce different interpretations. Those variables that correlate most closely with the component may have small variances, and thus exert little influence on the position of the component, and therefore on the trend that it describes. If the analysis was performed on the covariance matrix, a variable with large variance is regarded in some sense as important, otherwise the correlation matrix would have been used. Thus, those authors who say that the c_{ij}s (correlations between X_i and *j*th component) are a measure of the contribution of X_i to the *j*th component are correct only when they refer to analyses of the correlation matrix, where interpretation of c_{ij} and u_{ij} produce the same results. When analysing the covariance matrix, the contributions of the variables to the component values, the u_{ij}s, are generally to be preferred to the correlations (c_{ij}s).

This is not to say that the c_{ij}s cannot be useful. They identify which variables are well described by the components. Indeed, c_{ij}^2 is the proportion of the variance of a variable explained by a component. It may be interesting to know if 99% of a variable's variance is explained by a component; particularly if it was a component you were considering dropping. However, the importance of a variable to the component is not measured by c_{ij} or c_{ij}^2, (except almost coincidentally when you've analysed the correlation matrix). The importance of a component to a variable is usually not the same as the importance of a variable to the component. It may be an overall unimportant variable, i.e. it may have a small variance.

The difference between the u_{ij} and the c_{ij} can most simply be understood by the useful fact that if you regress the values the observations have on the PC (the PC scores) against their values on the original variables the u_{ij}s emerge as the partial regression coefficients (section 8.****). That is, the influence of the *i*th variable on the PC score keeping all the other variables constant. The c_{ij}s are of course the simple correlations between PC score and the variable.

Programming note.

In R if the PCA object is called prin then the coefficients (u_{ij}) are in prin$loadings. Print them using print(prin$loadings,digits=3,cutoff=0) or something similar. The PC-variable correlations (c_{ij}) are easily got by cor(*original variables,* prin$scores)

EXAMPLE 5.7

The component-variable correlations (c_{ij}s) for components 1 and 2 of the microplankton data are presented in table 5.3. Their interpretation is substantially similar to the coefficients (u_{ij}s - Table 5.2) - *Favella, Oikopleura* and *Acartia* on the first component and Gladioferens and *Euterpina* on the second - though this would not always be the case.

Table 5.3. The component correlations associated with the first 3 principal components of the log(X+1) transformed zooplankton data.

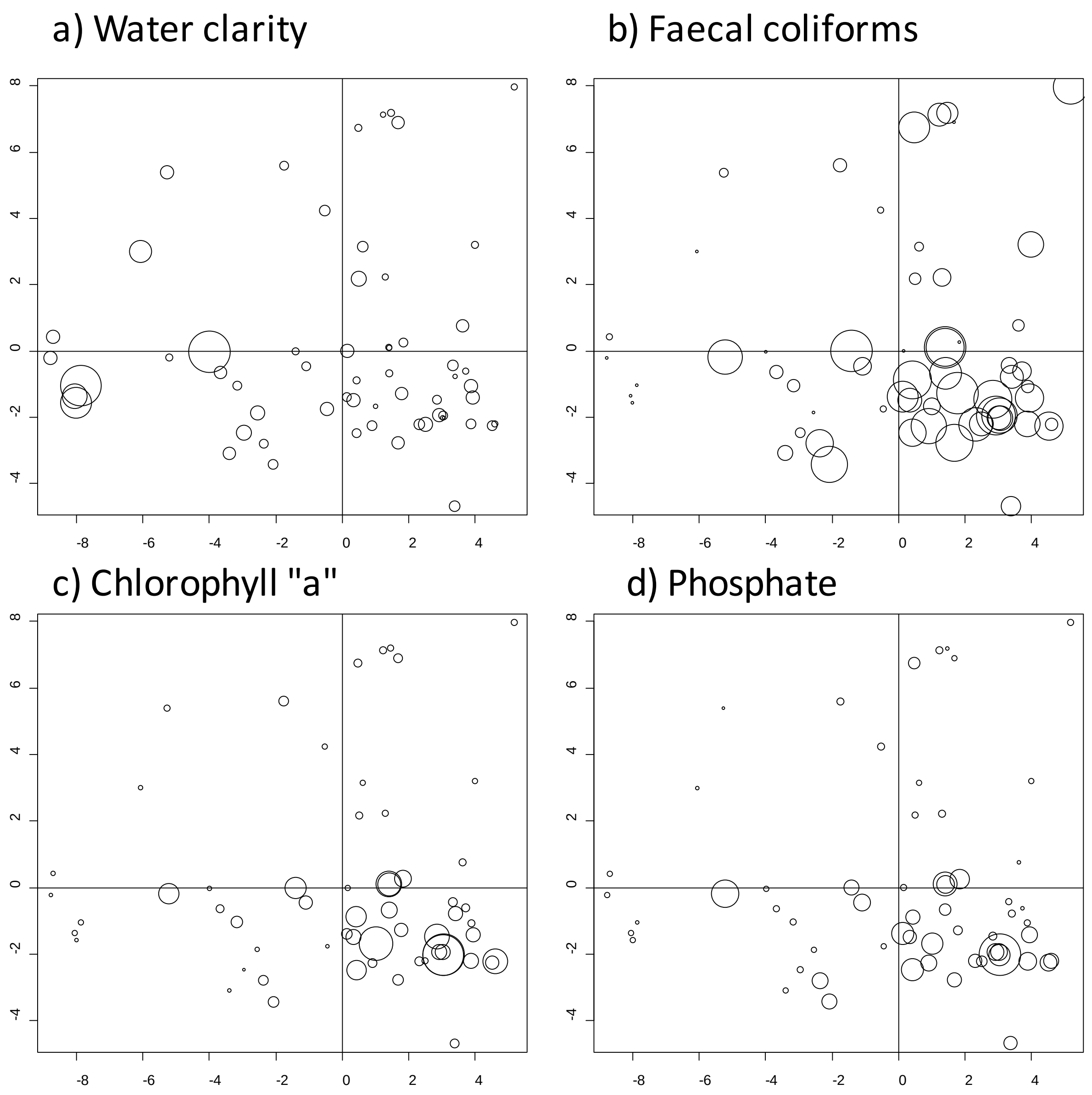
	PC1	PC2	PC3
ACARTIA	-0.585	-0.258	0.576
EUTERP	-0.017	-0.836	0.077
GLADIO	-0.107	0.877	0.347
HARPACT	-0.119	-0.353	0.538
OITHONA	-0.336	0.158	-0.267
PARACAL	0.106	-0.082	0.029
TEMORA	0.357	0.388	0.386
FAVELLA	-0.913	0.131	-0.132
OIKOPL	0.717	0.086	0.148

iii) Using external information.
There are other methods that may help to assign a biological interpretation to a particular component. Some authors ignore the direct interpretation of the component through the u_{ij}s, and, treating the component scores as observations on a new variable of an unknown identity, try to infer its nature from its behaviour and its relationship with other variables. In particular other variables that were measured in the same study but not included in the data matrix. We looked at how these variables can help us interpret trends in the reduced space plots; they can also be used to interpret the individual components. The relationship with these external variables may give some insight into the nature of the new variable, the component. The bubble and labelled plots used earlier would help here.

Alternatively you could use descriptive statistics (e.g. correlation coefficients) to describe the relationships with the external variables. If any of these external variables were qualitative rather than quantitative, then analysis of variance could be used instead of correlation, to see if the component was related to the differences between the levels of the external variable.

If you have a number of external variables then you should not take the *p* values from the tests too seriously, particularly if you selected the variables on the basis of some preliminary data exploration. Such tests have no formal validity, they are just checking that the pattern observed earlier was probably not your imagination. Such a test cannot have formal validity because the hypothesis being tested was suggested by the data it is now being testing on. The circular logic invalidates the test. Of course had you intended to perform the test before you started fishing around in the data then the test would be quite valid. Still, even if the *p* value is not strictly correct, you can still use the *F* value or the correlation coefficient as a description of the strength of the relationship.

It is important to realise that there are methods for directly describing the relationship between one set of variables and another (the multivariate linear model e.g. Canonical correlation, redundancy analysis - Section 9). However there are advantages to at least visually relating the results of a simple ordination like PCA to these external variables first. The ordination is (hopefully) describing the patterns the physical variables are to explain; a preliminary examination of possible patterns may give insight into the results of the more sophisticated method (particularly since these make assumptions about the form of the relationship that may not be justified).



a) Water clarity

b) Faecal coliforms

c) Chlorophyll "a"

d) Phosphate

Figure 5.6. Bubble plots of external variables displayed in the reduced space from a PCA on the Zooplankton data (log(X+1) transformed). Note the trends that clearly relate the physical variables to the patterns in the animal data. By identifying groups of observations (areas of the space) with particular physical properties we can go back to the bubble plots (**Error! Reference source not found.**) and see which species are found there.

EXAMPLE 5.8

We saw in Figure 5.5 that there were differences between the two harbours on the first component (apparently attributable to the Mangere sewage oxidation ponds). The bottom right corner of the reduced space plot contained nearly all the Mangere samples. The Whau samples were spread over the arc from bottom left to upper right. The bubble plots of some other external variables are shown in Figure 5.6. These plots clearly suggest that components one and two both relate to the physical characteristics associated with pond effluent (sewage bacteria, nutrient enrichment etc.). It is clear from the plots that if the individual components are to be given separate meaning they should be interpreted simultaneously. Clearly, given the concentration of the Mangere observations in the lower right quadrant, component 1 is a contrast between a subset of Whau stations and the Mangere sites, while component 2 is contrasts a different subset of Whau stations with the Mangere sites.

The Whau stations low on component 1 seem to be more offshore with relatively clean water. Those Whau stations high on component 2 are more inshore and though cleaner than Mangere are dirtier than the other Whau stations. Therefore we might tentatively identify component 1 as relating to a comparison between offshore and inshore water while component 2 could be mainly describing the difference between the dirty inshore water of Mangere with the (somewhat) cleaner less nutrient enriched inshore water in Whau creek. This is entirely consistent with the plankton associated with these trends (see **Error! Reference source not found.**). *Oikopleura* is on offshore species coming in on the high tide or at more offshore stations. *Gladioferens* on the other hand tends to be an inhabitant of mangrove swamps and inshore brackish water which, though contaminated with sewage, have not had the excessive fertilising of the oxidation ponds.

Problems (with solutions where possible.)

Exclusion of vital space.

PCA, as you will appreciate by now, though conceptually fairly simple (it is, really) has a number of problems associated with it. It is not a magic wand to be waved at the data set, which will then open up to reveal its secrets; there are many potential pitfalls. One of the most important is associated with throwing away useful information. When you use a reduced space for representing the data, you have no guarantee that you have not thrown away interesting information. Just because the variance associated with a component is small relative to other trends, doesn't mean it might not be important.

Unreliability of 'major' components.

I talked above of how 'minor' components could still hold important information. The other side of this coin is that some 'significant' components may be unreliable or describe irrelevant variation. For example, if measurement error is a relatively large contribution to total variance, it may well be partially described by a component with a large eigenvalue. This would usually not be interpretable, and a reduced space based on such an axis would probably not be useful.

Most workers try to use the sample defined components as estimates of population trends. Unfortunately, it would appear that both the eigenvalues and the eigenvectors are subject to considerable sampling variation. There are formulae for the calculation of the appropriate standard errors, but these are only accurate at large sample sizes, when you usually don't have to worry anyway.

To make matters worse, PCA can throw up unreliable components even with large sample sizes. These components may explain large amounts of variation but do not identify a trend at all, but are pointing in an arbitrary direction. It is easiest to explain this by an example. If we were analysing a 3-D data cloud shaped like a salami, after identifying the major axis we are left with the insoluble problem of where to put the 2nd and 3rd axes. Unless someone has sat on it, the space at right angles to the major axis (i.e. its cross section) is circular. So there is no obvious direction for the 2nd and 3rd axes. PCA will still place two new axes, but they will be in arbitrary directions, influenced by sampling error. The eigenvalues will be nearly equal, because both axes will explain equal amounts of variation. This problem is called **sphericity**, because when a data cloud is spherical there are no meaningful components to be found. What's the major axis of a soccer ball? The sign to look for is two or more components having the same sized eigenvalues. In which case their directions must be arbitrary, and they should not be interpreted. However, there is no reason why they should not be used together for a reduced space, they may still show up interesting patterns among the observations.

Assumptions of PCA.

Since PCA is normally an exploratory, hypothesis generating technique it doesn't really make any assumptions (providing you are not using any of the tests). However, PCA is more useful and the results easier to interpret (and biologically more relevant) if certain assumptions can be made: in particular, random or at least representative sampling, and linearity of the relationships between variables.

PCA assumes that the covariances (or correlations) adequately describe the relationships between variables. This is only true when the relationships are linear, or at least monotonic. In real data the relationship between two variables may be markedly non-linear; or the covariances may fail to describe the situation for other reasons, so the resulting PCA may not give good results. .

Outliers.

Another feature of the data that may distort the components is the presence of outliers. There are two major ways in which outliers can affect the results: by distorting the covariances between the variables; and by adding spurious dimensions or obscuring the cutoff point for choosing components. We can detect points with excessive influence on the covariance matrix by searching the plots of sampling units on the first few components for extreme values. We can detect the other kind of outlier by considering the minor components. These last few components (the discarded ones) can be thought of as being residual variation left after a model (the higher components) has been fitted to the data. Looking for outliers among these residuals can identify observations that are adding unimportant dimensions to the data. One of the less important components may be in place solely due to the presence of a single outlying data point. These outliers, by inflating the variance of these residual components, may obscure the discontinuity between the 'large' components and the 'small' ones. This will make it more difficult to decide which components to discard. These outliers may be picked up by looking at the scatter plots of the sampling units scores on these 'minor' components.

If you are using PCA to generate hypotheses, there is no reason not to drop outliers and reanalyse. It could also be interesting to look more closely at those observations; their deviations may be a clue to something important.

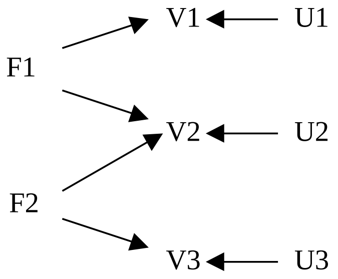
Factor Analysis

The versatility of PCA actually creates problems. There is a family of techniques closely related to, and partly based on PCA-like mathematics, called Factor Analysis (FA). The similarities between Factor Analysis and PCA are so great that the boundaries have become blurred. This blurring is not helped by the varying interpretations that workers in different disciplines put on the words Factor Analysis. For my purposes I shall maintain that PCA is primarily concerned with the variation between the observations, summarising, describing and possibly explaining it. The new variables we extract from the data are defined by the pattern of observations. Even when we are looking at the relationships between the original variables we do so to identify which combinations of variables are related to the major variation among the observations. Factor Analysis on the other hand is concerned with the pattern of covariation among the variables, the observations are only relevant in so far as they define the pattern of correlations among the variables.

There are two main streams of thought on what factor analysis is trying to do, so when reading anything about factor analysis it is important to identify which is being discussed. Most statisticians and a lot of social scientists see factor analysis as a way of identifying underlying latent, causal, variables that are responsible for the observed values of the the measured variables. In other words factor analysis is attempting to fit a particular type of causal model to the data. Other practitioners see it more as a way of identifying groups of covarying variables which may allow the efficient summary of information. For the purposes of the book, I shall be pedantic and restrict factor analysis to the fitting of latent variable models - because in this form I don't like it and it makes an easier target.

Factor Analysis has been widely used in psychology and the other social sciences, but it has tended to receive a bad press from statisticians outside the social sciences. Unfortunately I tend to agree with them, though not necessarily for the same reasons, so I cannot recommend its use, particularly by beginners. For this reason I am not giving it a chapter to itself. However since it does appear in the literature, and to show the similarities with, and more importantly, the differences from PCA; I will give a brief outline of the core methodology.

Factor Analysis like PCA is designed to identify underlying patterns in the data (the variables you have measured). However it has a more ambitious goal, and is consequently based on much more restrictive assumptions. It assumes that the pattern of covariation observed in the set of variables you have measured are due to a set of underlying factors- some unique to each variable, some shared between variables. The goal of factor analysis is to identify and interpret all the factors that influence the variables. This can probably be best illustrated by a path diagram that shows a possible set of causal links between the variables in your data set (V1, V2, V3) and the factors that you cannot directly measure but which you think affect the observed values for the variables:



F1 and F2 are the common factors, they each effect more than one variable. U1, U2 and U3 are the unique factors. The fundamental idea is that the pattern of variation and covariation that we observe in the covariance or correlation matrix for the variables can be explained in terms of the underlying causal factors. The common factors are described in terms of their contribution to each variable (**factor score coefficients**) or by their correlation with each variable (**factor correlation**). The components in PCA are described by the contribution of the variable to the component (the component coefficient, equivalent to the factor score), or by the correlation of the component with each variable,(the component correlations). The importance of the unique factors can be inferred from the size of their contribution to the variables. However the real problem lies in identifying the way in which the common factors combine to produce the pattern of covariance observed among the variables. The final set of factor score coefficients that describe this, is called the **factor pattern**. The corresponding set of factor correlations is called the **factor structure**.

There are two major forms of Factor Analysis: exploratory Factor Analysis when little or nothing is known about the factor structure - this is the commonest use; and confirmatory Factor Analysis where the number of factors is known *a priori* and a certain amount is known about the structure of a hypothesised factor pattern. This last form, though apparently the most robust use of Factor Analysis, is, by its nature, rare outside the social sciences. It is generally performed using Maximum Likelihood Factor Analysis or Latent Variable Modelling which allows the appropriate hypothesis tests to be performed. However due to its sophistication I shall not describe it further. I shall instead concentrate on the exploratory form.

There are three parts to most exploratory Factor Analyses: i) to identify what proportion of the variation in each variable is shared with other variables and what is unique; ii) to identify how many common factors are involved and iii) to try to pin down and interpret these factors.

5.7.1 Removing unique variance.

The analysis usually starts with the correlation matrix of the variables. Clearly the off-diagonal elements (the correlations) will reflect the activity of the common factors, and will be unaffected by the unique ones. The diagonal elements (the variances, standardised to one) however, can in theory be split into two parts, one unique to that variable and another caused by the action of one or more common factors. So, when we analyse the matrix to detect the common factors, it would be sensible to remove the variation due to the unique factors before we start. This would leave us with the **communalities**, estimates of the variance due to the common factors, as the elements along the diagonal of the **reduced correlation matrix** that results. There are a number of methods for estimating the communalities, some simple but approximate, others requiring successive approximations (iterative methods), and yet others that provide exact results derived, somewhat laboriously from theory. I will not go into detail.

5.7.2 Extraction of preliminary factors

Now that we have a correlation structure that reflects the pattern of the common factors, we will now want to know how many factors there could be. The most commonly used Factor Analysis technique, Principal Factor Analysis, performs a PCA on the reduced correlation matrix. If the factor model holds, there should be as many major components as underlying factors, the remaining components are assumed to be trivial, representing random error. The number of 'significant' eigenvalues should therefore identify the number of factors. Of course this step, guessing how many eigenvalues are 'significant' is fraught with problems (section 5.3). Most computer packages use Kaiser's criterion. If we accepted the eigenvectors of the reduced correlation matrix as describing the factors, and some workers do, our interpretation would in fact probably not differ markedly from the solution derived from an orthodox PCA on the correlation matrix. However these are more usually regarded as preliminary values that require more manipulation before an interpretation is attempted. PCA is being used to identify the subspace in which the factors must lie. It may not identify the factors themselves.

5.7.3 Rotation and interpretation of the factors.

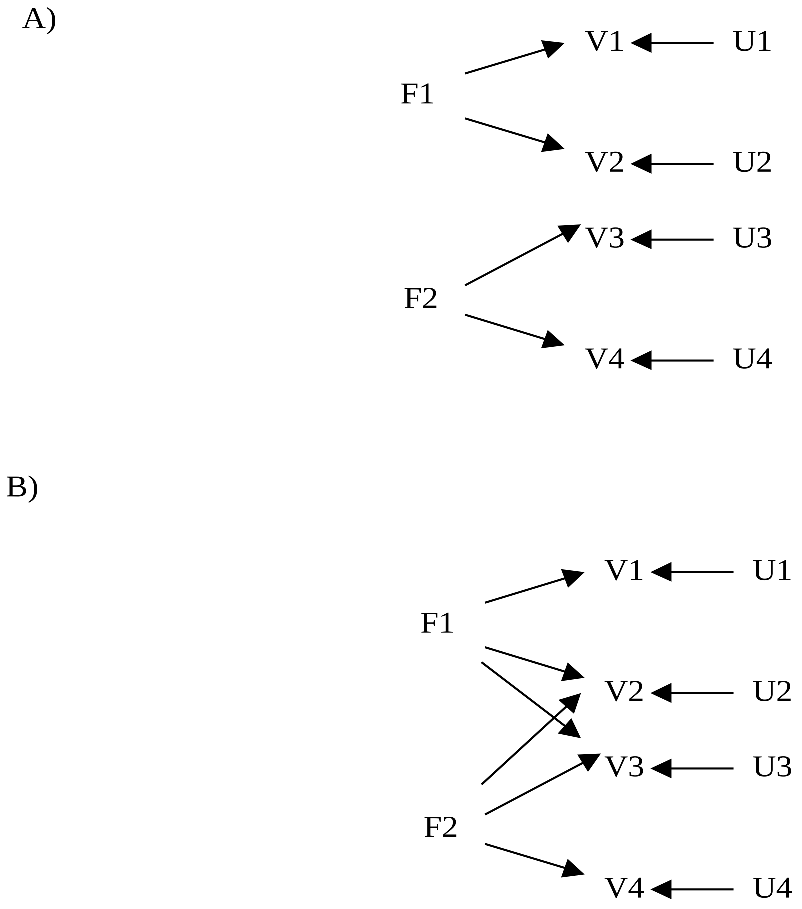


Figure 5.7 A) Assumed simple structure, B) is not allowed.

Let us assume that we have guessed the correct number of factors. The corresponding preliminary factors, like the principal components that they so resemble, define a reduced space. This is the space spanned by the factors. Let us now envisage these factors as axes that define the space. There are an infinite number of ways of orienting them in the space. How should we chose just one orientation? Of course, we have one quite sensible set, the principal components of the reduced correlation matrix, the preliminary factors. These describe the maximum amount of shared variance. But most practitioners of Factor Analysis do not regard these as adequate. They are led to a choice of orientation by a desire for what they call 'simple structure'. Some argue that the number of factors that contribute to an individual variable is likely to be small. They believe that the structure a) in figure 5.8 is more likely than b).

The problem is that the preliminary factors, the component axes of the reduced correlation space, are liable to have a complex interpretation, like the second path diagram above. So, after identifying the factor space, it is usual to rotate the axes to a new orientation that makes their factor structure and therefore their interpretation simpler. Unfortunately there are a number of definitions of "simple" as applied to factor structure- so there are a number of rotation techniques. The commonest of these is probably the **varimax rotation**. This attempts to line up a factor axis so that a variable will have either a high correlation with it or none at all. This will avoid the coefficients of intermediate size that can make PCA derived components so tricky to interpret (section 5.5.2.i). It will also usually make the pattern of the derived factors more like that of the "simple" path diagram above. It achieves all this by rotating each axis to maximise the variance of the factor loadings (hence "varimax"), while keeping the axes orthogonal. There are other rotation methods. Some are designed around alternative views on simplicity, others around *a priori* conceptions of how a factor structure should look. This raises the spectre, mentioned by many critics of Factor

Analysis, that an expert can get virtually any result he wants by appropriate choice of a rotation technique. Indeed, one of my teachers used to call principal factor analysis, rather unkindly, "PCA with fudge factors".

Some of the rotations, like varimax, keep the axes orthogonal. They therefore assume that the factors are independent of each other. Other techniques allow for the possibility that the underlying factors may themselves be correlated. The factors produced by these **oblique rotations** appear to be somewhat fragile and are often difficult to interpret.

The common factors, whether oblique or orthogonal, are interpreted in much the same way as PCA derived components - by examining the strength of the relationship between the variables and each factor in turn. Most workers use the correlation between the variable and the factor.

5.7.4 Problems with Factor Analysis.

The major problems with Factor Analysis can be divided into two groups: those associated with the assumptions and those resulting from its performance.

As explained above, Factor Analysis assumes that the variance of a variable can be partitioned into a unique component and one due to one or more factors that also affect other variables, as in the path diagram above. This assumption explicitly excludes the possibility of one of the variables causing some of the variation in one of the others.

This might considerably limit Factor Analysis's usefulness in some disciplines. In ecology, for example; where it is reasonable to suppose that there are at least some direct interactions between species, e.g.predation, mutualism or competition. The assumption of 'simple structure' is also of concern. It is true many scientists have recourse to the principle of parsimony. They look for the simple, elegant solution. But their relationship with this principle tends to be ambivalent. Lagrange said "seek simplicity, but distrust it". My personal prejudice is that the form of simplicity sought by the common rotations in Factor Analysis has little relevance to many, if not most situations

In performance exploratory Factor Analysis appears to be rather fragile. Seber(1984) outlines in some detail a test, performed by Francis(1973), of the effectiveness of Factor Analysis. Francis investigated the ability of Principal Factor Analysis, with and without rotations (both orthogonal and oblique), to identify the factor structure from covariance matrices derived from known underlying factor models. Thus, he knew the right answer, and he wanted to see how well the technique did in finding it. He concluded that if the correct number of factors were known, then Principal Factor Analysis may provide a reasonable factor structure, that may or may not improve with rotation. The big problem is, of course that there is no satisfactory method for estimating the number of factors and fictitious factors are all too easily generated. Seber further concludes that unless the underlying factor structure really is simple in an appropriate way, orthogonal rotations may be worse than useless if the wrong number of factors is chosen. He also suggests that oblique rotations will seldom be useful. J. Scott Armstrong in a satirical paper in the American Statistician (1967) also demonstrated the weaknesses of exploratory factor analysis.

The practical limitations of exploratory factor analysis are therefore largely in the appropriateness of simple structure and the choice of number of factors. If the number of factors is known *a priori*, and the approximate structure of the causal relationships is known to be "simple" - no direct influence between observed variables, no loops in the path diagram, then factor analysis may work quite well. In other words, confirmatory factor analysis is fine, but exploratory analysis is at best weak, at worst plain misleading. Furthermore, some workers have concluded that there is no practical difference between principal factor analysis where the communalities were set to 1 (essentially a PCA with factor rotation), image factor analysis (yet another method) and maximum likelihood factor analysis - provided the number of factors was known in advance and they used the same rotation methods. The results can be expected to diverge if the number of factors is wrongly estimated.

If Factor analysis is so fragile and unreliable how is it that so many workers (particularly in the social sciences) swear by it? It may be impolite to suggest it, but one reason that factor analysts claim it to be a useful technique may be because, even if the factors generated are spurious, they can nearly always be given plausible interpretations. The interpretations can often then be rationalised as supporting some particular viewpoint. The fact that many factor analysts are working in complex subject areas (like the social sciences) where it is very difficult to confirm the factor interpretations, may explain why these workers stay satisfied with their results. That it is depressingly easy to fall into this trap of finding satisfying interpretations of spurious factors is shown by an embarrassing episode in my own past.

I was once involved in performing a PCA for a graduate student - in those days computing was so difficult students often did not do it themselves. I got the print out and together we pored over it as I gave an inspired, eminently plausible and authoritative interpretation of the component coefficients (the factor score coefficients). The student went away happy. Imagine how I felt when I discovered the next day that during the reading of the data into the computer there had been a mix up in the input formats and I had been effectively interpreting a PCA on random data! I had had no difficulty in providing meaningful interpretations to the components, yet they were complete rubbish.

There were two main reasons why I was able to fall into the trap. The first was that there were no standard errors available to show me that the coefficients were not reliable, the second was that the student was exploring the data with an open mind. He had no idea of what to expect, and the range of possible, biologically plausible relationships among the variables was enormous. When little is known about a system there is nothing to contradict any interpretation of its structure. The more complex the system being studied and the less that is known about it, the easier it is to be tricked by spurious, non-existent factors (perhaps why factor analysis has proved so popular in the social sciences).

Factor analysis as a summary of correlation structure.

I mentioned above that there is a more exploratory, less formal approach to factor analysis. Some people use it simply to identify groups of covarying variables, to suggest patterns among the variables. They do not claim to be identifying underlying, hidden factors or to be modelling causal systems. Their factors are simply new variables that best summarise the pattern of correlations among the variables after a certain amount of variation has been dropped as irrelevant. While in many, particularly exploratory situations this may be more reasonable than attempting to fit causal models, it still suffers from some of the problems outlined above. In particular the choice of how much information to discard may still be crucial. As an *ad hoc* check on the robustness of the choice, do not rely on any single method for choosing the number of factors, try a range of values. If after rotation they all give substantially the same interpretation to a common subset of factors then one might have faith that you were identifying a stable and real pattern. If however the patterns detected were closely dependent on the number of factors extracted then I would abandon factor analysis altogether and rely on a simple PCA performed on the correlation matrix.

Rotations and Principal Components

I feel it is a regrettable fact that the use of axis rotations has crept from Factor Analysis to PCA. With the increased availability of computer packages that allow a rotation by the addition of a single command, people have begun doing them as a matter of course. Some say they rotate to increase the interpretability of the components. They are in fact not interpreting the principal components but the subspace spanned by the subset of components they have retained. Whatever the rotated axes may be they are no longer principal components. These people are performing a principal factor analysis where all the variance is assumed shared - i.e. that the communalities=total variance. Their results are therefore subject to the same uncertainties as those from a more orthodox Factor Analysis, and will probably not be much different.

In conclusion, PCA and Factor Analysis are for two distinct uses. PCA is exploratory and is primarily concerned about patterns in the observations; apart from the assumption of linearity, it makes no *a priori* assertions about the patterns of covariation. It can be used to suggest descriptions of trends in the data, but it should never be concluded that all the important trends can be identified (sections 5.3 and 5.). It can only be suggested that if a trend can be clearly identified then it is probably important; there are exceptions here - the "horseshoe effect" (section 6.8.4).
Factor Analysis on the other hand is primarily concerned with modelling the patterns among the variables. It assumes that the observed pattern of covariation is the result of a particular arrangement of shared and unique factors and further, if rotations are used, that the common factors conform to some particular 'simple' structure. If the number of factors is known *a priori* then confirmatory factor analysis can be performed which when the model structure is appropriate can be very powerful. However when the number of factors is not known the factor analysis will be exploratory and subject to so many problems that it is doubtful if the results should carry much credence. (You should be fairly sceptical about the results of an unrotated PCA how much more so for an exploratory factor analysis.) Even if exploratory factor analysis is being used simply to identify groups of covarying variables (possibly by using a principal components analysis then rotating components within a subspace), the assumption is still being made that the appropriate number of factors can be correctly identified, and as we have seen this is unlikely to be true.

AUTHOR'S CHOICE.

Reserve factor analysis for the fitting of models when the number of factors is known *a priori* and when the underlying structure is known to conform to one of the forms of "simplicity".

Some revision questions

1) What is PCA used for?
2) What do the eigenvalues measure?
3) How might you decide how many components to keep?
4) How might you identify which variables are associated with a particular trend in the data?
5) What is Exploratory Factor Analysis used for?
6) What is a communality?
7) How might we detect sphericity?
8) What is simple structure?

Chapter 6. Multidimensional Scaling.

Multidimensional scaling (MDS) is an extended family of techniques that try to reproduce the relative positions of a set of points in a reduced space given, not the points themselves, but only a matrix with interpoint distances (**dissimilarities**) - see section 3 . This sounds easier than it is. These distances might be measured with error, or even be non-Euclidean. With PCA there would be at least one Euclidean configuration of points that would exactly reproduce the dissimilarity matrix - the original data matrix. Indeed there would be an infinity, any rotation or reflection of the original data would leave the interpoint distances unchanged. This configuration might lie in a high dimension hyperspace but at least it would be exact. The problem of approximating it in fewer dimensions is not too difficult. However, if the matrix of dissimilarities is not Euclidean, then there is no guarantee that there is any exact configuration, let alone an adequate approximation. It is this kind of problem the scaling techniques are designed to solve.

The origins of many of these techniques are in psychology (psychometrics to be precise) so the extensive literature is littered with terms like stimulus, subject, attribute and preference, which makes reading it rather stressful.

Principal Coordinates.

Principal coordinates (PCO) is closely related to PC. It finds a configuration of points that optimally reproduces the distance matrix in a Euclidean space of few dimensions. Though it works best on a Euclidean dissimilarity matrix, it will nearly always produce useful results from non-Euclidean ones.

Intuitive explanation

Assume for the moment that the dissimilarity matrix is Euclidean. We can therefore try to imagine a cloud of points hanging in a space of unknown dimensionality. These points have no coordinates since as yet there are no axes (only distances were given). The problem is to impose a set of axes on the space and then locate the points on them. By a cunning transformation of the dissimilarity matrix, it can be made equivalent to a matrix **YY**', a cross-product matrix of a set of coordinates **Y**. An eigenanalysis of this cross-product matrix (like PCA and CA) will give a diagonal matrix of eigenvalues $\mathbf{\Lambda}$ and a matrix of eigenvectors **V**, so that $\mathbf{YY'=V\Lambda V'}$ (see chapter 1). The coordinates are given by $\mathbf{Y=V\Lambda}^{1/2}$, the elements of the scaled eigenvectors (the $\mathbf{\Lambda}^{1/2}$ rescales the eigenvectors to have a sum of squared coefficients equal to the eigenvalue). Like PCA the eigenvectors identify orthogonal axes that run down the major axes of the cloud of data points; though they will not be the old axes rotated, there are no old axes. As in PCA the size of the eigenvalues will give the variation of the data points along the associated axis (eigenvector). It is therefore comparatively easy to select a reduced space to display the relationships among the data points. If the original distance matrix was Pythagorean then the results will be identical to a PCA on thedata matrix from which the distances were calculated.

If the dissimilarity matrix is non-Euclidean, then there may be problems with the interpretation of the eigenvalues and in extreme cases the plots themselves, this is discussed in section 5.0.

Relationships with PCA.

The final stage of PCO, an eigenanalysis of a cross products matrix **YY'**, seems very reminiscent of PCA - an eigenanalysis of a crossproducts matrix **Y'Y**. If the original dissimilarity matrix actually contains Euclidean (Pythagorean) distances, then PCA and PCO are doing exactly the same thing, reproducing a cloud of points in fewer dimensions on orthogonal axes while optimally preserving their relative positions. It is reassuring that in such a case, a PCA on the covariance matrix will is identical to PCO on the dissimilarity matrix. The major conceptual difference between the two techniques is that PCA uses the eigenvectors to project the original data points (vectors) into the reduced space. In PCO the scaled eigenvectors are the vectors of observation scores in reduced space; but their positions in the reduced space and the positions of the corresponding axes will be identical.

By this stage you should be grasping the extraordinarily incestuous relationships amongst multivariate methods. Every so often it is demonstrated, for example, that all the methods of ordination are special cases of each other. This serves to remind us that the main difference between them is the transformation and/or standardisation they impose not the fact that they have different names.

Programming notes:

In R.

If d is your distance matrix then:

```
pco<-cmdscale(d, number ,eig=T)
```

Where *number* is the number of eigenvalues you want to extract (usually (the number of observations minus one) - so you can look for negative eigenvalues). Your scores on the principal axes will be in pco$points, your eigenvalues in pco$eig.

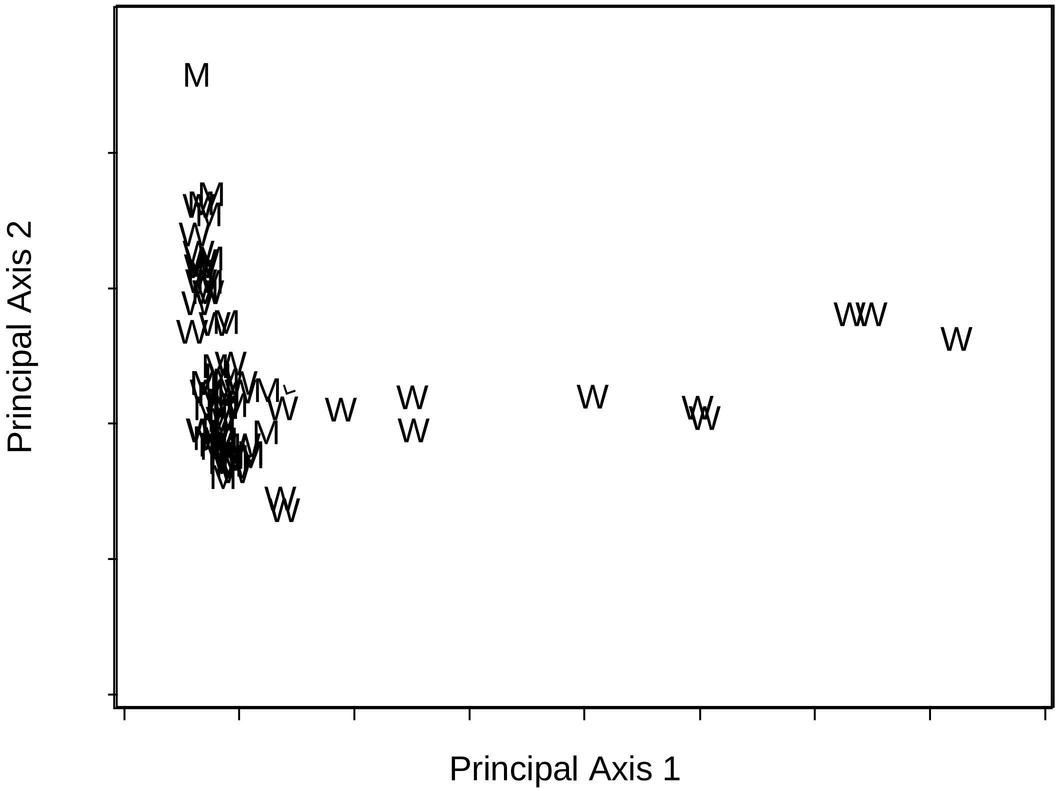
A screeplot can be done using the plot() function

```
plot(pco$eig, type=”b”)
```

EXAMPLE 6.1

Using the zooplankton data I have performed a PCO on a chi-squared distance matrix. This is of course identical to doing a PCA on chi-square transformed data. The results of an analysis are

Figure 6.1 Principal coordinates reduced space plot using chi-square distance. Because chi-square distance is a special case of the Euclidean distance this is identical to a PCA on the appropriately standardised data.



primarily determined by the transformation of the data, not the technique.

Interpretation

i) How many axes?

Since PCO is an eigenanalysis of a cross-product matrix, like PCA, the eigenvalues give the amount of variation described by the associated eigenvectors. A scree graph is therefore the most direct way of selecting a reduced space. The criteria are the same as for a PCA (see section 6.3). However there is a complication: if the dissimilarities are non-Euclidean then there can be negative eigenvalues. These indicate that there is no exact configuration possible in Euclidean space.

What you do in this situation depends on whether the negative eigenvalues are referring to potentially relevant information. These eigenvalues are summarising those components of the distances that prevent the observations being represented in a Euclidean space. If these bits can be assumed to be uninformative, then you could argue the negative eigenvalues are irrelevant and can be discarded. So the measure of goodness of fit is then the proportion of the positive eigenvalues represented by the retained ones.

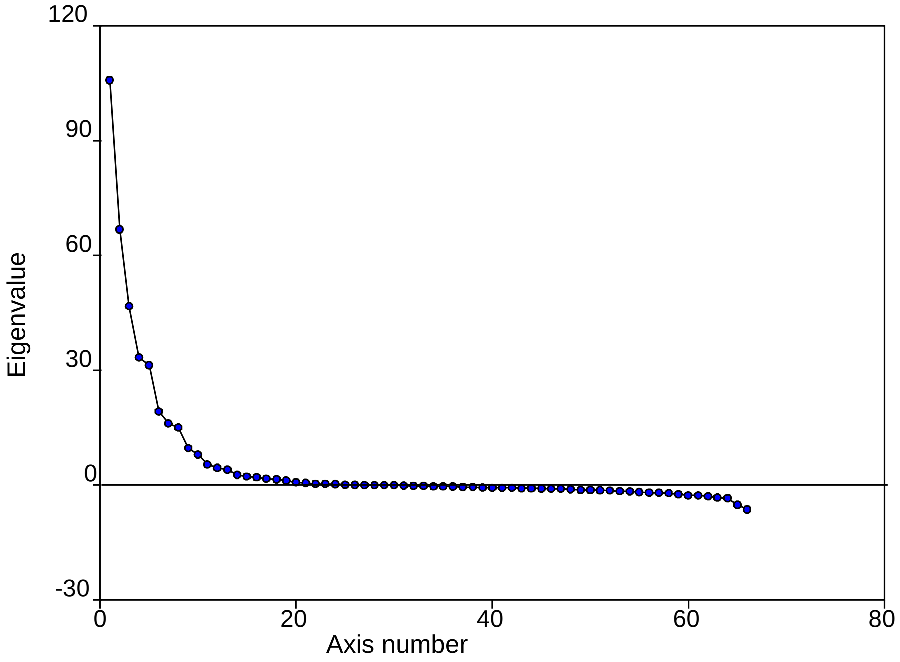
Mardia suggested two measures of the goodness of fit of a *k* dimensional reduced space:

$$a_{1k}= \Sigma_i \lambda_i / \Sigma|\lambda|,$$

$$a_{2k}= \Sigma_i \lambda_i^2/\Sigma\lambda^2.$$

I prefer a_{1k}. When there are no negative eigenvalues it is the same as the measures used in PCA. Also, if you are used to the usual measure, then a_{2k} appears to exaggerate the goodness of fit; it might mislead.

Figure 6.2 Scree diagram for the PCO on the Gower's unstandardised distance of the log(X+1) transformed zooplankton data. Note the negative eigenvalues. 40 of the 66 non-zero eigenvalues are negative, though they only represent 13.1% of the absolute variation. The first 2 axes summarise 39.6% of the absolute variation.



ii) Interpreting the principal axes.

If you still have the original data matrix from which you calculated the distances then you can use the axis variable correlations to help interpret which variables are associated with which axis. Bubble plots can also be used.

iii) Problems.

a) Adequacy of the reduced space.

Just as with PCA this is typically summarised by the eigenvalues.

b) Stability of space.

As in any other method based on eigenanalysis, if two or more of the eigenvalues are equal then there will be problems of instability (sphericity).

EXAMPLE 6.2

Using the zooplankton data set again I have performed a PCO on a site × site dissimilarity matrix using Gower's distance, ignoring double zeros, but without the range standardisation (not a distance measure you have met in this course, don't worry about it) and using a log(X+1) transformation. The distance was chosen to remove the double zero problem, and the transformation was chosen to reduce the effect of dominant species. The scree plot of the eigenvalues is shown in Figure 6.2 . 40 out of the 66 non-zero eigenvalues are negative, but they only contain 13.1% of the total variation (when we disregard the sign), so the positive eigenvalues contain most of the information. (I discuss negative eigenvalues below). The main problem is that the first two eigenvalues only explain 39.6% of the (absolute) variation. The first 3 explain 50.3% and the first 4, 57.9. There is no obvious cutoff - though the features found by our earlier analyses are apparent on the first two axes. The plot looks particularly like the PCA on the log(X+1) transformed data. As usual, what we see is largely determined by the transformation. That we are using a city-block metric rather than the Euclidean, and are excluding double zeros (and 51 % of the data set are zeros) has little effect, it is the log(X+1) transform that is important.

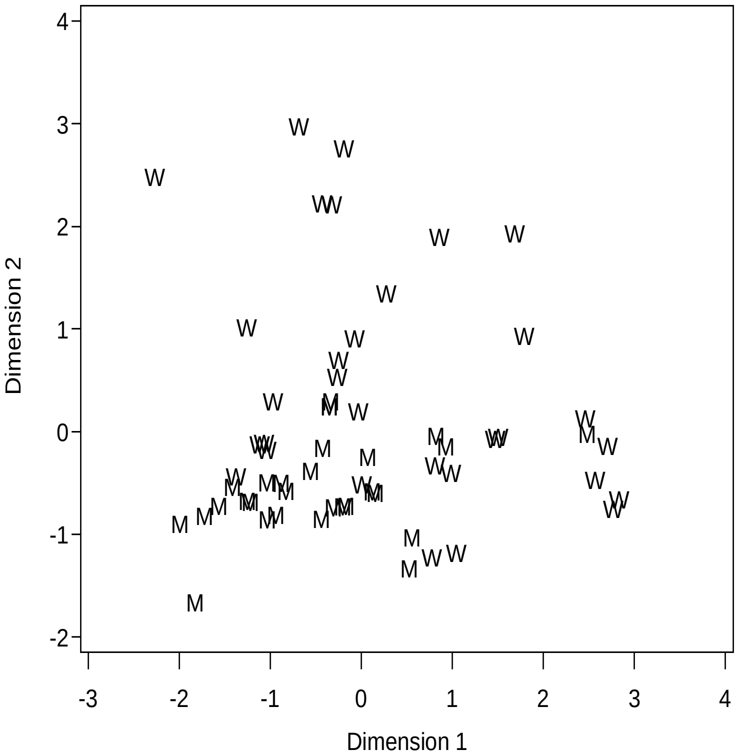
Metric Scaling.

Metric scaling tries to produce a set of coordinates (a configuration of points) in a reduced number of dimensions whose matrix of interpoint Euclidean distances approximates the original dissimilarity matrix as closely as possible. The eigenanalysis technique Principal Coordinates (PCO) does this directly. PCO *is* a metric scaling technique (it is sometimes called **classical** or **Torgerson scaling**). However, the term metric scaling is more commonly applied when computer intensive iterative algorithms are used to do the job rather than eigenanalysis. The results will seldom be very different from doing a PCO on the same dissimilarity matrix.

Intuitive explanation

The simplest approach to metric scaling is by repeated approximation, and is always done on a computer. The computer can be imagined as guessing an initial configuration in a high dimension space (less than *p*-1). It then calculates the distance matrix for these initial points. The elements of this matrix are then regressed against the elements of the given dissimilarity matrix. If by some extraordinary stroke of luck (or fudging) the fit is extremely good then this configuration will do. However, it is far, far, more likely that the fit will not be adequate. The computer then shifts each point in the configuration slightly so that its interpoint distances will fit the given dissimilarity matrix better, calculates the distances, measures the fit and calculates the improvement. If there has been little or no improvement, or the fit is adequate, then a solution has been found and the process stops. If not, then it goes through more iterations till it finds the best fitting set of coordinates (or gives up in disgust). This process is repeated for spaces of fewer dimensions, and the goodness of

Figure 6.3. Plot from a PCO on Gower's unstandardised distance using log(X+1) transformed data ignoring double zeros. Note the similarity with the PCA plot on the same transformation of the data



fit plotted against the number of dimensions (like a scree graph - section 5.3). The appropriate number of dimensions for the reduced space is chosen, and the corresponding configuration plotted and (hopefully) interpreted. Notice it does not calculate one high dimensional solution and extract all lower dimensional solutions from it as PCA or PCO does; the configuration for each reduced space is calculated anew each time. In fact, if the dissimilarities are exactly Euclidean, then a *p*-1 dimension solution, if subjected to a PCA, will give similar lower dimensional configurations to separate scalings for each solution. In other words a metric scaling on a Euclidean distance matrix will give the same results as a PCA.

<u>The goodness-of-fit statistic.</u>

To get the fit of the calculated distances to the observed, the estimated distances are regressed, linear regression through the origin, against the original dissimilarities (δ_{ij}) in the distance matrix. The fitted values ($\hat{d}_{ij}$ - the disparities, forgive the jargon) are compared with the current distances between the points in the reduced space, the d_{ij}s, to assess the fit. The usual goodness of fit statistic is analogous to r^2, the coefficient of determination in ordinary regression:

$$\Sigma(d_{ij}- \hat{d}_{ij})^2/ \Sigma d_{ij}^2$$

This is called STRESS, STRESS formula 1, or STRESS1. This is the statistic that is being minimised by the iterative procedure.

Because each configuration for a dimensionality is calculated separately, it is possible to request a solution for a particular dimensionality without calculating any others. This is nearly always a bad idea; not only is it possible that another (possibly lower dimensioned one) might be better, but as we shall see below local minima can be detected by comparing the stress values from different solutions - see section 0.ii.

Provided STRESS1 is used as the measure of goodness of fit the results of a metric scaling will be nearly always equivalent to a PCO on the same dissimilarity matrix. The reason I described it is to introduce a more widely used technique.

Non-metric scaling.

Under certain circumstances trying to preserve the actual dissimilarities might be too restrictive or even pointless. For example if there is large error in the dissimilarity estimates, if the dissimilarities or the data they were based on were ranks (ordinal), then the magnitude of the distances are too crude to be worth preserving. A method that preserved only the rank order of the dissimilarities would be more appropriate.

The algorithm to do this is virtually the same as the one given above for metric scaling.. The sole difference is that the linear regression that fitted the estimated distances for the solution to the dissimilarities is now replaced with an order preserving regression - Kruskal's least squares monotonic transformation (Kruskal 1964), sometimes known as optimal scaling.

<u>Monotonic Regression.</u>

The fitted line in this form of regression is not smooth, there is only one constraint on its shape - it must only move in one direction, always upwards or always downwards; it must be monotonic. In the algorithm described here the direction will be upwards. The process is simple if tedious. The optimal scaling procedure starts at the lower left corner of the plot and moves towards the right.(Figure 6.4) Each point is examined in turn, if it is higher than the previous one it is left undisturbed; if lower, the mean of it and the previous value is calculated and both points replaced with this value. The next point is then compared with this value, if it is lower then the mean of the three points is calculated and all three replaced. Then the next point until one is found that is higher than the current value. There is now a block of points all with the same value, the algorithm now returns to the point prior to the block to see if the fitted line still moves upwards, if not, then the points are amalgamated again. The result of this tedious process is a line that progresses upwards in a series of steps like a staircase. The tread (the horizontal bit) of each step (the predicted value - $\hat{d}_{ij}$ the disparity) is defined by the mean of all the points that occur in that stretch of the X axis (δ_{ij} - elements of the original dissimilarity matrix). It can be shown (Kruskal 1964) that this line minimises the sum of the squared deviations of the current configuration distances from the estimated distances predicted from the line, STRESS1; it is truly a least squares method. STRESS1 measures the extent that the rank order of the estimated distances differs from the rank order of the dissimilarities.

Programming notes:

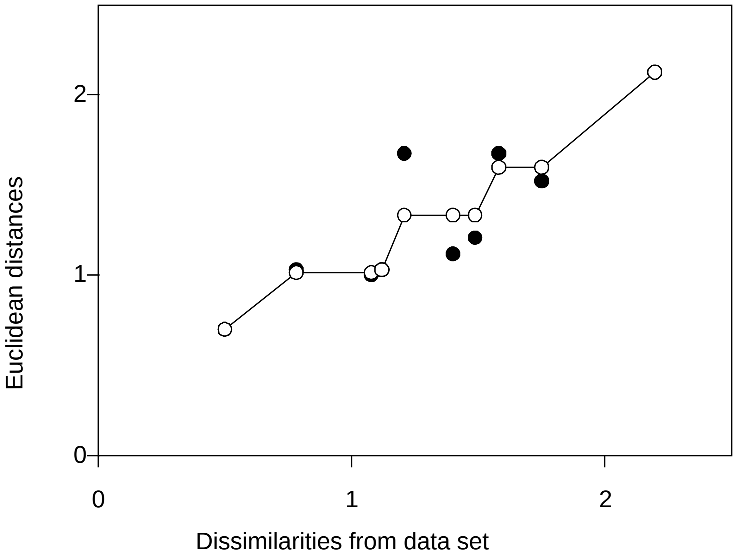
In R

if *dist* is the name of your distance matrix.

then:

```
dimen<-3
sol<-isoMDS(dist, y=cmdscale(dist, dimen),k=dimen)
```

Figure 6.4. Monotone regression. Empty markers represent the fitted values. Starting at the origin the first four observed values increase monotonically so the fitted and observed values coincide. However, the 6th and 7th are lower than the 5th so the fitted values are the average of the three. Similarly, the 8th and 9th get smoothed also. The resulting line increases monotonically.



will get you a 3 dimensional solution. The points will be in sol$points, and the final STRESS1 value in sol$stress (expressed as a percentage rather than a proportion). Plot the points with the same functions as PCA.

To do a STRESS plot:

```
STRESS=NULL      #PREPARE AN EMPTY VECTOR
mds2=isoMDS(distance matrix, k=1)
STRESS=append(STRESS,mds2$stress)      #ADDS THE STRESS VALUE ON THE END OF THE VECTOR
mds3=isoMDS(distance matrix,k=2)
STRESS=append(STRESS,mds3$stress)
```

keep going up to the desired maximum number of dimensions

When you have finished

```
plot(1:length(STRESS),STRESS,type="b")      #PRODUCES STRESS PLOT
```

<u>Interpretation</u>

a) How many dimensions?
There are three main techniques for identifying the appropriate number of dimensions for the reduced space.

i) If the program has been allowed to calculate solutions for high dimensionality down to one, then a plot of STRESS against number of dimensions is extremely useful. This STRESS plot is analogous to a scree graph of other ordination techniques but is interpreted slightly differently. It is examined for an "elbow" where the STRESS reduces rapidly and the addition of further dimensions improves the STRESS only slightly. The number of dimensions at the elbow identifies the solution that is used for the reduced space plots.

Of course if the ideal space is unidimensional (for example in many morphometric data sets) then there will be no elbow. Kruskal and Wish in their excellent little book, hereafter referred to as K & W, suggest that, in this situation, if stress for the unidimensional solution is less than .15 then it will probably give the most useful plot - provided there are more than ten sampling units so that the dissimilarity matrix is larger than 10 X 10.

K & W also offer advice on when to accept an elbow as useful. If the STRESS value at the elbow is greater than 0.1 then it should not be used, indeed an elbow at high STRESS may be suggesting a local minimum. Generally, a genuine elbow at high STRESS(near 0.1) ought to have the left hand section very steep, the right hand (higher dimensional) bit need not be very shallow. For a genuine elbow at low STRESS (e.g. 0.02) the left hand section need not be very steep but the right hand bit must be very shallow, nearly horizontal. Unfortunately if the elbow appears at two dimensions, where most workers would like it, there could be a problem. A large STRESS for unidimensional solutions does not mean much. Most existing algorithms have difficulty in fitting data onto a line (see Heiser 1987 for references), so an elbow at two dimensions could be an artefact and should be regarded with caution - a pity.

It is also unfortunate that many, possibly most, real data do not have clear elbows; in which case K & W offer the following rules of thumb: if possible do not use a solution with a STRESS greater than 0.1 or less than 0.05 (unless an extra dimension reduces the STRESS considerably. Seber (1984) presents a table (from Kruskal 1964) that gives guide-lines to the meaning of different values of STRESS.

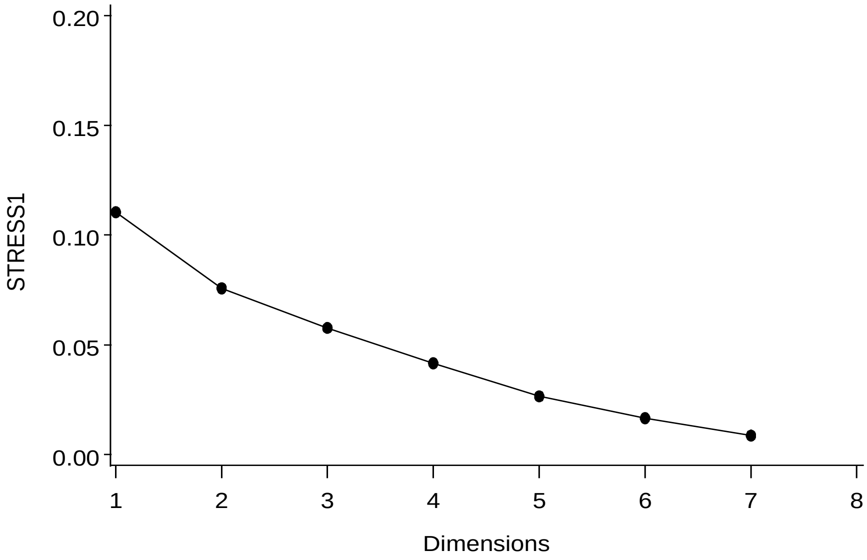


Figure 6.5. Stress diagram for non-metric MDS on chi-squared distance matrix. There is no obvious elbow (not unusual). Dimensionality was chosen by checking that there was no effective difference between the 2 - D solution and the first 2 dimensions of the 3-D solution.

0.2	Poor
0.1	Fair
0.05	Good
0.025	Excellent
0	Perfect and therefore suspicious

A STRESS value close to zero (say < 0.01) is a possible indicator of a degenerate solution - see below, section 0.iii.

EXAMPLE 6.3

Figure 6.6 shows a reduced space plot of the two dimensional solution from a non-metric multidimensional scaling minimising STRESS1. The distance measure used was the chi-squared distance to compare the solution with the PCO (classical metric scaling). The main difference is that the horseshoe has been removed. This remarkable removal of the horseshoe is by no means inevitable. It is commoner in my experience for any horseshoe in a metric technique to also appear (though sometimes a bit reduced) in the non-metric solution.

Why did I chose a 2 dimensional solution? The STRESS plot is in Figure 6.5. There is no obvious elbow so the choice of dimensionality is somewhat arbitrary. I chose 2 because that gave a reasonable STRESS level (0.075); but checked that the first 2 dimensions of the 3-D solution (STRESS 0.053) gave essentially the same picture. This analysis illustrated a typical problem. Even though it was performed with a modern algorithm (PROC MDS, in SAS version 6.12), the best 2 - D solution was not found using a PCO starting configuration (the default). After 20 random starts, 3 better solutions (over 10% improvement in STRESS) had been found. Clearly there are problems with local minima that only repeated random starts can overcome Even now I cannot be sure there is not a better solution but I am fairly sure that there would be little substantial change to the reduced space plot.

b) Interpreting dimensions and trends.
The direction of the axes on which the MDS configuration is defined are usually in arbitrary directions. Even if the final solution of a given dimensionality has been rotated by PCA, the component axes need have no meaning. As a general rule therefore there is little point in trying to interpret the axes.

If external variables are available, trends in the plots can sometimes be interpreted if it is possible to associate the plotted points (the sampling units) with their values on the external variables. Bubble plots can be useful.

Problems

Besides the problems associated with all ordination techniques: outliers, useful information associated with rejected dimensions, the horseshoe effect etc; Multidimensional scaling has three that are peculiar to it.

i) Incomplete convergence.
The iterative process may not have found the minimum before it gives up. It is inconceivable that any program would not notify you of the fact, but most, perhaps all, present the configuration at the last iteration as a final solution - which it isn't. It is therefore important to check the output to find out why the program terminated. The only acceptable reasons are that the change in STRESS is too small, so a minimum has been reached; or that the value of STRESS is too small. Any other message spells trouble. In fact as we shall see below (section 6.3.3.iii), if the STRESS is too small, then the solution must be checked for degeneracy.

If the program has terminated before a minimum STRESS solution has been found, then either the program can be run again requesting a larger number of iterations, or, if this is not possible then the final configuration of the run can be fed back in as the initial configuration for another run, so the program starts where it left off.

ii) Local minimum.

Since the program is an iterative one, it is sometimes possible, especially with non-Euclidean dissimilarity matrices and non-metric scaling, for the process to find more than one minimum STRESS configuration depending which starting configuration is used.

To use an analogy: with well behaved data, finding the solution with minimum STRESS is like rolling down the sides of a volcanic crater, you may bounce around a bit, but sooner or later you will end up at the bottom. However if the data are not well behaved - the structure is not clear - there may be secondary craters in the side of the main one. So where on the edge you start your trip will determine whether you end up in the bottom or in a secondary crater (a local minimum). Of course if the volcano is active it doesn't much matter either way, but in MDS it does; a local minimum can be misleading.

Even with well behaved data there will be local minima, but the configurations will usually be so similar: reflections, rotations, close data points reversed in position, etc, that it will not matter.

Local minima that are significantly different to the global arise when the structure of the data is not clear, in a way analogous to the equal eigenvalue problem in PCA. Large numbers of missing values can accentuate the problem. Local minima are particularly likely on unidimensional solutions.

In the event there is a local minimum configuration that is very different from the global, then either both will have high STRESS, in which case both are useless; or the local one will have a much higher STRESS than the global so it should stand out in the STRESS diagram as anomalous relative to the solutions of higher and lower dimensionality. The STRESS diagram must always be concave upwards, (e.g. Figure 6.5) any kink upwards in that shape will nearly always mark a local minimum.

If a local minimum is suspected, or no STRESS diagram is being produced, then the program should be run with more than one starting configuration. Some programs can generate random starting configurations, otherwise the user may have to provide them, which is very tedious. Note: do not give one where all the points lie on one axis, it can lead to problems. Make sure the starting configuration spreads the points throughout the space. If all the starting configurations lead to the same final solution then it is very unlikely that this is a local minimum. If some locate a different solution with a lower STRESS then this should be used.

iii) Degeneracy.
A degenerate solution is a configuration where for no obvious reason the program decides assert its artistic independence and present what is usually an attractive, regular but meaningless plot with all the points coalescing into a few, often equally spaced, clusters. This solution is often associated with a very small STRESS value (e.g. less than 0.01).

Degenerate solutions are largely a feature of non-metric methods. There appear to be three main causes of degenerate solutions.

a) There are a large number of equal values in the dissimilarity matrix, perhaps a lot of zeros, or a matrix where the dissimilarities take only a few values, for example where it is based on a few binary variables.

b) The data cloud is really a few (4 or less) clusters, where the intercluster distance is much larger than the intra cluster distances.

c) There are a lot of missing values in the dissimilarity matrix.

These situations are basically the same, and represent a situation where there is insufficient information for the program to come up with an meaningful configuration, so it produces a pretty one instead.

There are two main ways to recognise a degenerate solution: a STRESS value of less than 0.01, even zero, and the scatter plot of estimated distance against dissimilarities has a characteristic shape. The plot will usually consist of a few "steps", where a number of different dissimilarities (δ_{ij}) have the same distance (d_{ij}). There will nearly always be a cluster of zero distances, sometimes associated with medium to large dissimilarities.

A degenerate solution cannot be interpreted, so the simplest thing to do is re-analyse using a metric method.

iv) Outliers.

Metric scaling is sensitive to outliers; though Gower (1987) suggests it is more robust than PCO, and therefore PCA. If outliers are found there are two main solutions.

a) Drop them and repeat the analysis.

b) Use a non-metric method. These are more robust as they do not try to preserve distance, only the rank order of distances.

v) Adequacy of the reduced space.
The measure of goodness of fit, in this case STRESS, provides a crude measure.

vi) Stability of the reduced space.
The major determinant of the stability of a solution under sampling error is the sample size relative to the number of dimensions. Kruskal and Wish (1978) suggest that if the number of dimensions is three or less then the sample size should always be greater than four times the number of dimensions - plus one. If the sample size is less than twice the number of dimensions - plus one - then the solution should not be used.

Of course using dissimilarity matrices with little information in them can also lead to instability in the space, even if the solution is not degenerate.

Which to use: metric or non-metric?

Both metric and non-metric methods have their strengths. Non-metric methods can handle ordinal data or other lower quality dissimilarities, and are robust to outliers. On the other hand they are more prone to local minima and degenerate solutions. As Gower (1987) points out: when the number of sampling units is large, preserving the rank order is usually essentially the same as preserving distances; in which case it doesn't much matter which is used. A metric scaling will always have a higher stress than the corresponding global non-metric solution, but will often be more accurate. Sometimes, with non-Euclidean distances, the relationship between the fitted Euclidean distances and the dissimilarities is non-linear. In which case a linear metric scaling may not be adequate and a non-metric method will usually be appropriate. Such a situation can be recognised from the shape of the scatter diagram of the fitted distances against the dissimilarities.

Chapter 7. Cluster Analysis

Cluster analysis, or classification as it is known in the botanical literature, has the apparently simple aim of finding clusters in a data cloud of sampling units in the absence of any *a priori* information about which point belongs in which cluster. This apparently unambitious aim is unfortunately fraught with problems.

The major difficulty is that no one seems to agree on precisely what a cluster is. For a very good reason, the human eye is unexcelled as a pattern recognition device, but we recognise clusters of points in a variety of different ways. For example, it is extremely difficult to think of a single definition that would adequately describe all the clusters in fig 7.1, even though they are quite obvious (I hope). Some workers have stressed the importance of cohesiveness (like fig 7.1a); others contiguity of points (7.1b); yet others have concentrated on distances such that all or most of the distances within a cluster are less than those to any point outside the cluster; and finally others have tried to make the definition so vague that it can include most of the possibilities without the necessity of actually defining anything. Everitt's definition (Everitt 1980) seems to come as close to being useful as any:

"Clusters may be described as continuous regions of (a) space containing a relatively high density of points, separated from other such regions by regions containing a relatively low density of points."

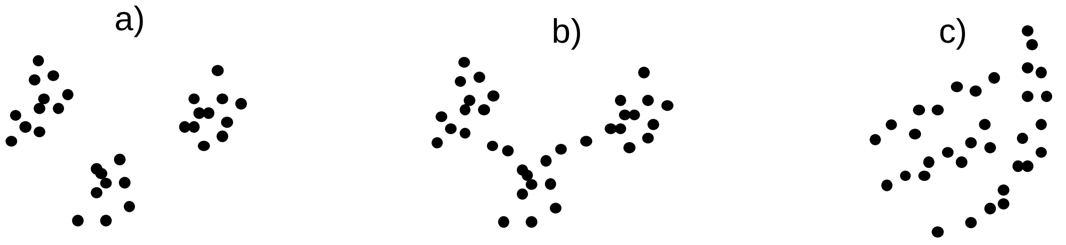
Unfortunately it does not provide a rationale for a single comprehensive technique that can handle all the data structures shown and satisfy all the requirements of workers. Indeed it is extremely unlikely that any such method could ever be found, for there lies another problem, workers want the technique(s) for a number of different purposes:

i) to find groups for classification;

ii) to reduce the number of sampling units in an analysis by using a single representative from each cluster of similar individuals;

iii) for data exploration and hypothesis generation;
iv) for fitting distribution models and estimating their parameters;
v) dissecting a continuous data cloud into relatively homogeneous zones;
and many more.
The only thing the large number of existing techniques have in common is that unlike canonical

Figure 7.1. The human eye can detect clusters in all three of these diagrams. Statistical methods find it more difficult.



discriminant analysis (section 10) and discriminant function analysis (not covered in this course) there is no prior information about which sampling unit is in which group. Like the ordination methods of the earlier chapters, cluster analysis techniques operate on an unpartitioned data matrix to find, or impose, structure in the data cloud.

One consequence of this variation in definition and use is that cluster analysis as such does not exist. The title refers to an enormous and extraordinarily diverse family of techniques. For someone to say that they used cluster analysis is about as informative as their saying they studied an insect. To cover all the techniques would take a whole (large) book. So for this course I shall content myself with covering some of the common ones and ones I think are potentially most useful.

Given the diversity of techniques it is very important to choose the technique with a clear idea of what it is required to do. Like selecting a similarity or distance metric (Section 4), the choice must be made with care after consideration of the nature of the data, your objectives, and the available alternatives. However the most important thing to remember when using a clustering technique is:- you must not believe the result. The pattern you get is at most a plausible way of viewing the data. By using an appropriate method and by employing validation techniques the plausibility can be enhanced, but no cluster analysis can be relied on to produce truth. With real data, different methods will nearly always produce different results. If the structure in the data is fairly obvious then these answers may not differ much, but if there is any ambiguity in the data then the methods may well give contradictory results.

Partitioning methods.

Though the hierarchical methods have been historically more important, the partitioning methods are becoming increasingly popular, and it is easy to see why. The hierarchical methods are restricted to an often inappropriate nested structure so that at each level (number of clusters) the solution is constrained by the previous one. In the partitioning or segmentation methods the solution at any level is independent of the others and can therefore be globally optimal - if you're lucky.

For a single run of a partitioning method, the desired number of clusters (*k*) is usually fixed - some techniques do allow some small adjustment in this number during the process. Of course since the correct number of clusters is usually not known, the program is normally run with different values of *k* and the optimum number of clusters chosen (covered later).

There are two major phases to a partitioning method:

i) an initial allocation (usually rather arbitrary) into *k* preliminary clusters;
ii) reallocation of each point either to the closest centroid, or so as to optimise some property of the clusters. This is repeated until there is no further improvement, then the program stops.

The initial allocation is usually started by choosing *k* sampling units to use as "seeds" to "crystallise" the clusters. There are a number of ways to choose these seeds; it depends on the program. As we shall see it is a tremendous advantage if you can put in your own set. These seeds are used as the initial centres of the clusters, points are allocated to the nearest cluster centre, and in most programs the cluster centroid is adjusted as they are added.

The methods we consider here (there are others) the *k*-means methods, run through the sampling units reallocating them to the cluster with the closest centroid; they pass and repass through the data till no further reallocation of points is possible. Some programs then try swapping pairs of points between clusters, to further improve the solution, and to protect against local optima.

K-means partitioning methods

The *k*-means methods are generally the fastest clustering methods, but they are inclined to be trapped by local optima and tend to produce equal volume spherical solutions. They are also very sensitive to starting strategy. Some workers suggest that random starting values should not be used. Seber reports a study as having located the global optimum only 3 times from 24 random starts! However their performance in the few Monte Carlo simulation studies that have incorporated them has been good relative to alternative methods, particularly when the solution from a hierarchical method was used as the starting configuration. In fact, it has tended to be better than the best hierarchical methods considered (Ward's and average linkage).

If the data set is particularly large, a sub-sample of the points could be clustered and the estimated centroids of the resulting clusters used as seeds for the analysis of the full data set. Some programs allow you to vary how the distance to the centroid is measured. Some programs normally use the squared distance which means that it is minimising the trace(**W**) where **W** is the within cluster variance-covariance matrix pooled over all the clusters, i.e. the total within sample variance. This is an appealingly statistical thing to optimise.

EXAMPLE 7.1.
To display the use of a *k*-means partitioning technique, I will use the microzooplankton data again. However, because I want to use the same data set again for the other clustering techniques which produce dendrograms, I will cut the data set down a bit first. All the examples in this chapter will only consider the 35 samples that were taken near to high tide. Since the PCA on log(X+1) transformed abundances seemed to give the most informative ordination I have used log(X+1) transformed data in this partitioning. A different transformation or standardisation would give a different result.

In reality I would normally not use a *k*-means techniques on a data set this small. Because of problems with local minima, the technique is best used with large data sets. We have already used a variety of ordination techniques on the complete data set and there were no obvious clusters so a *k* means method may well have problems converging to a global optimum. From the ordination plots we might suspect that a partitioning technique would really only dissect the data cloud (in marketing terms: produce a segmentation). After trying partitions with 7, 6, 5, 4, 3, and 2 clusters (allowing a couple of outliers) there was no reason to change that opinion. Using Callinski and Harabasz's Index (see later), a measure of cluster separation, none of them was clearly superior to the others (values of 13.04, 13.03, 13.12, 13.04, 11.23, 11.32 respectively). Given that the value seems to drop slightly between 4 and 3 clusters, and because I felt that 4 was a nice number of clusters - not too many, not too few - I chose to present four clusters. (Since we are dissecting the space rather than recovering true clusters, we can afford to please ourselves a bit). I used seeds from a Ward's hierarchical clustering program (see later) as starting points for the partitions.

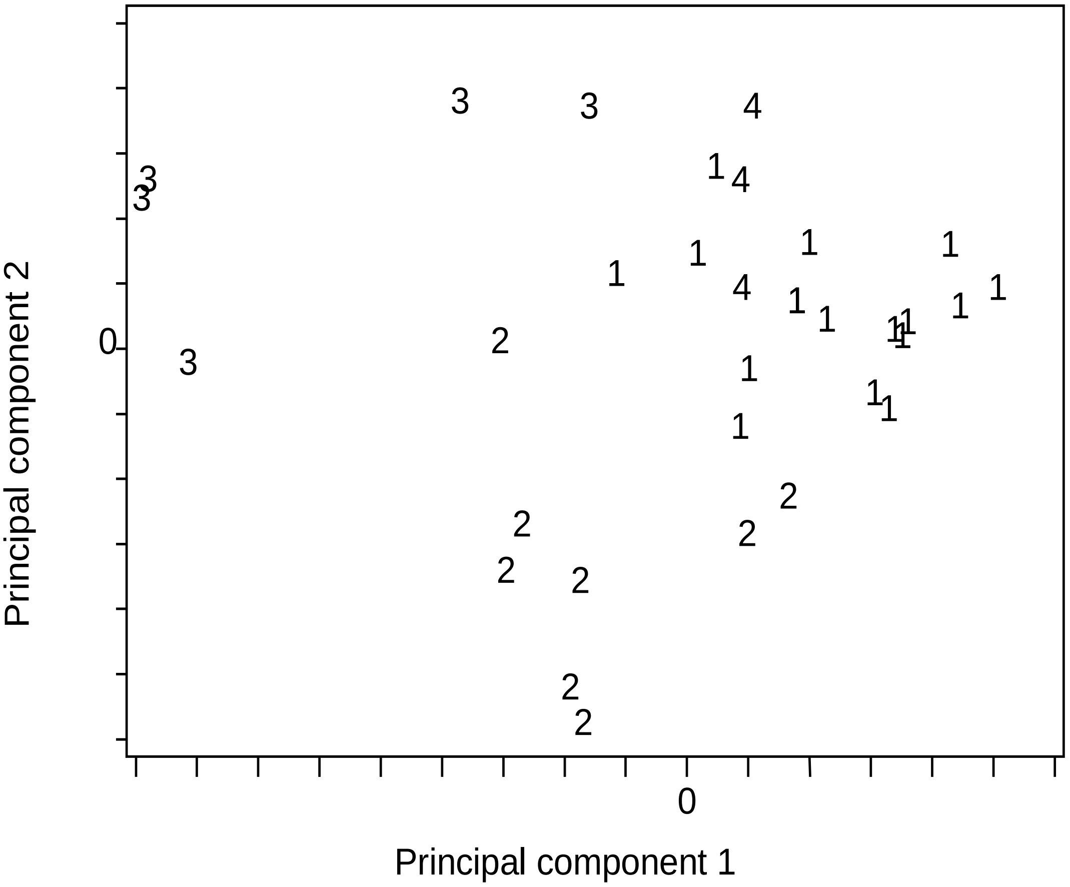


Figure 7.2. Results of k means clustering superimposed on a PCA reduced space plot of the microzooplankton data (to keep the graphs simple I am only using high tide data). For the observations, a 4 mean partition was used, the cluster number is plotted.

The results are displayed on a PCA reduced space plot (section 5) on the log(X+1) transformed data in Figure 7.2 Clearly the partitioning has identified quite consistent groups. This analysis therefore conforms to the central axiom for clustering: "I do not believe it unless I can see it." This of course has a complement: "But I do not believe it just because I can see it" - the human eye and brain are hard-wired to see pattern even when none exists. The plot allows the major differences between the clusters to be seen.

As we shall see in Example 7.2 the results of this analysis are very similar to those from the Ward's method applied to the same data. In essence it identifies the split between the Whau Creek (groups 3, and 4) and the Mangere samples (group 1), with group 2 containing 4 Mangere samples taken all on the same day, and 4 Whau samples taken on 2 different days.

It is worth pointing out that my suspicions about the possibility of local minima with this data set were well founded. Out of 20 random starts for the 5 group partition, not one found a solution as good as that found when the seed came from a preliminary Ward's clustering. Clearly when the number of observations is so small there are a large number of local minima.

Programming notes.

In R. First load my function PseudoF.

```
data<-dataset
k=number of clusters
cl<-kmeans(data,centers=k,nstart=number of random starts)
pf<-PseudoF(data,cl$cluster)
pf
```

The nstart= option tells kmeans how many random start you want it to do. It then picks the best of these. The value pf is the pseudoF statistic for this cluster solution.

When you have chosen your final solution then you may want to plot your cluster solution in a reduced space plot

Do your PCA and then

```
eqscplot(pcs$scores[,1:2],type="n")
text(pcs$scores[,1:2],labels=as.vector(cl$cluster))
```

Hierarchical methods.

These methods assume that the groupings in the data cloud have a hierarchical structure. The smaller groups form larger groups which form larger groups and so on - a nested classification. If this assumption is untrue then the techniques can be expected to distort the true structure of the data.

Most of the commonly used techniques are members of this group. They are widely available, all the major packages have a selection, and they are relatively easy to use, though often less so to interpret.

Hierarchical organisation is often difficult to justify for real data sets. Though there may be more than one level of grouping there may be no reason to assume that they are nested. For example, it has been shown that the clusterings defined by the optimum sum of squares at various levels of *k* may not be nested for all data sets; so a hierarchical method may be unsuitable for any given data set.

There are two approaches to hierarchical clustering, **agglomerative** and **divisive**. Agglomerative methods start from the individual sampling units forming them into groups and fusing the groups till there is only one that includes all the points. If we can describe this as working from the bottom up, then the divisive techniques work from the top down. The groups are formed by splitting the data set successively until there are as many groups as points.

Hierarchical agglomerative clustering.

All of the commonly used hierarchical methods are agglomerative. Most of them operate in the same way: first all sampling units that are zero distance apart are fused into clusters. The threshold for fusion is then raised from zero until two clusters (they may be individual points) are found that are close enough to fuse. The threshold is raised, fusing the clusters as their distance apart is reached until all the clusters have been fused into one big one. Thus the close clusters are fused first, then those further apart, till all have been fused. This process allows the history of the fusions, the hierarchy, to be displayed as a dendrogram. This is an advantage of the agglomerative methods, if the data have a nested structure these techniques lead to a useful way of displaying it. Other advantages are the ready availability of programs and their ability to handle quite large data sets - at reasonable expense. Unlike the optimisation or *k*-means methods, most of the agglomerative techniques can use a broad range of similarity or distance measures. This of course means that considerable care must be taken to choose the appropriate one; different measures often lead to different results.

Inevitably, given the variety of definitions of a cluster, there are a large number of different hierarchical agglomerative techniques. They mainly differ in the details of the fusion rule. For most of them the rule is simply stated: two clusters should be fused if the distance between them has been reached by the threshold. The problem is to estimate that distance. It can be done in a variety of ways and will usually affect the results. As we shall see, different types of clusters need different ways of estimating intercluster distance.

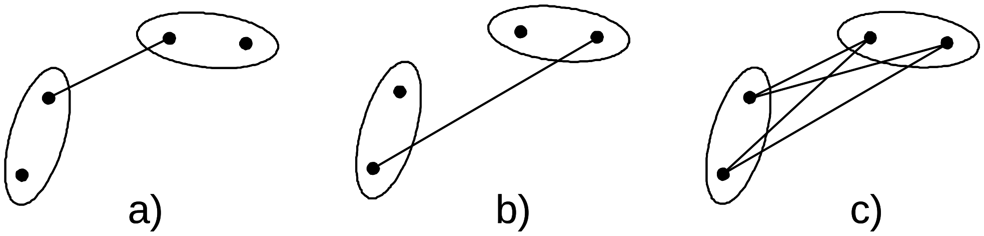
We shall consider the four most commonly used methods.

i) Single linkage (nearest neighbour) clustering.
The distance between two clusters is the distance between their nearest points (Figure 7.3a).The simplicity of this method makes it easy to program and extremely efficient. It was one of the most popular techniques in the early days of clustering; but since then, despite support from the theoreticians, it has been used less frequently. In general it has not performed well. It identifies clusters on the basis of isolation, how far apart they are at their closest points. This means that if there are any intermediate points then single linkage will fuse the groups without leaving any trace of their separate identities. This is called "chaining", which leads to characteristic and rather uninformative dendrograms. It is the chief weakness of the method. Its strength is that if the clusters are well separated in the data, then single linkage can handle groups of different shapes and sizes, even long thin straggly ones (e.g. Figure 7.1c) that other methods often cannot recover. It has other advantages, it will give the same clustering after any monotonic transformation of the distance measure - that means that it is fairly robust to the choice of measure. It is insensitive to tied distances - some methods suffer from indeterminacy if there are too many ties; a bit like degenerate solutions in non-metric MDS, (section 6.3.3.iii) and though the results are seldom as pretty, they can be just as meaningless.

As a cluster analysis single linkage is usually not very useful (unless the data is of the right type). Many investigations have found it performs badly with even slightly messy data.

ii) Complete linkage (farthest neighbour) clustering.
In many respects complete linkage clustering is the opposite of single linkage. Instead of measuring the distance between two clusters as that between their two nearest members; it uses that between the two farthest members (Figure 7.3b). In consequence the resulting clusters are compact, spherical and well defined. Unlike single linkage it can be sensitive to tied distances. There are similarities,

Figure 7.3. Three common measures of the distance between clusters.
a) nearest neighbour - used in Single Linkage.
b) furthest neighbour - used in Complete Linkage
c) the average of all the distances - Average Linkage (UPGMA)



the clustering it gives is also invariant under monotonic transformation of the distances; it is robust to a certain amount of measurement error and choice of distance. Unfortunately it is sensitive to even a single change in the rank order of the distances in the dissimilarity matrix (Seber 1984), and does not cope well with outliers. However, in Monte Carlo simulations, it nearly always performed better than single linkage; though usually not quite as well as Ward's or group average.

iii) Group average linkage (UPGMA)

This is probably the most popular hierarchical clustering method - for a very good reason - it usually works well. It could be thought of as an attempt to avoid the extremes of the single and complete linkage methods. The distance between two clusters is the average of the distances between the members of the two groups (Figure 7.3c). If the distances are Euclidean this is the distance between the centroids plus the within group scatter. As a result this method tends to produce compact spherical clusters.

Like its main rival Ward's method, average linkage has generally performed well in Monte Carlo simulations, and its continued popularity is because it consistently, though not inevitably, gives adequate results. However, Ward's generally performed better, particularly when there was some overlap between the groups. When intermediate points and outliers were removed ("trimming" or "incomplete coverage"), group average's performance was considerably improved. It performed poorly with mixtures of multivariate normal distributions probably because of the overlap between clusters..

iv) Ward's method (incremental sums of squares, minimum variance, agglomerative sums of squares).

Ward's method is the hierarchical version of the k-means partitioning method. At each fusion it attempts to minimise the increase in total sum of squared distances within the clusters. This is equivalent to minimising the sum of squared within cluster deviations from the centroids - i.e. trace(**W**). Since at any one stage it can only fuse those clusters already in existence - it is not allowed to reallocate points - it can only be stepwise optimal. It cannot find the true minimum configuration at each level, so it would not be expected to recover natural clusters as well as the non-hierarchical methods that also minimise trace(**W**). A bad start to the agglomeration process can place the algorithm on a path from which it can never reach the global optimum for a given number of clusters. Despite this, Ward's method has performed well in simulations; one of the two best hierarchical methods overall. Its chief flaw is a tendency to form clusters of equal size, regardless of the true number. So when the number of points in the clusters are different, group average and complete link may give better results. Like the complete linkage and group average methods it is also biased towards forming spherical clusters; though perhaps not as strongly as they are. It may also be rather sensitive to outliers. However it appears to perform well when there is a lot of overlap, when many of the other techniques have difficulties. It has been found in simulations that Ward's performed best of the hierarchical methods at recovering natural clusters, but that the *k*-means and optimising methods were better.

EXAMPLE 7.2.
To show the differences between the major agglomerative methods I analysed the high-tide micro-zooplankton data. I used Ward's, average , complete , and single linkage, on six different distance measures: Euclidean with log(X+1) transformed data, chi squared distance, chi-squared distance with log(X+1) transformed data, Bray-Curtis distance, Bray Curtis with √√ transformed data, and Jaccard's distance. These impose a range of standardisations on the data that try to reduce the influence of some of the very large numbers lurking in the data set. Plankton data typically consists of some very large numbers in a sea of zeros. In Figure 7.4 I have presented the results for the Euclidean log(X+1) data.

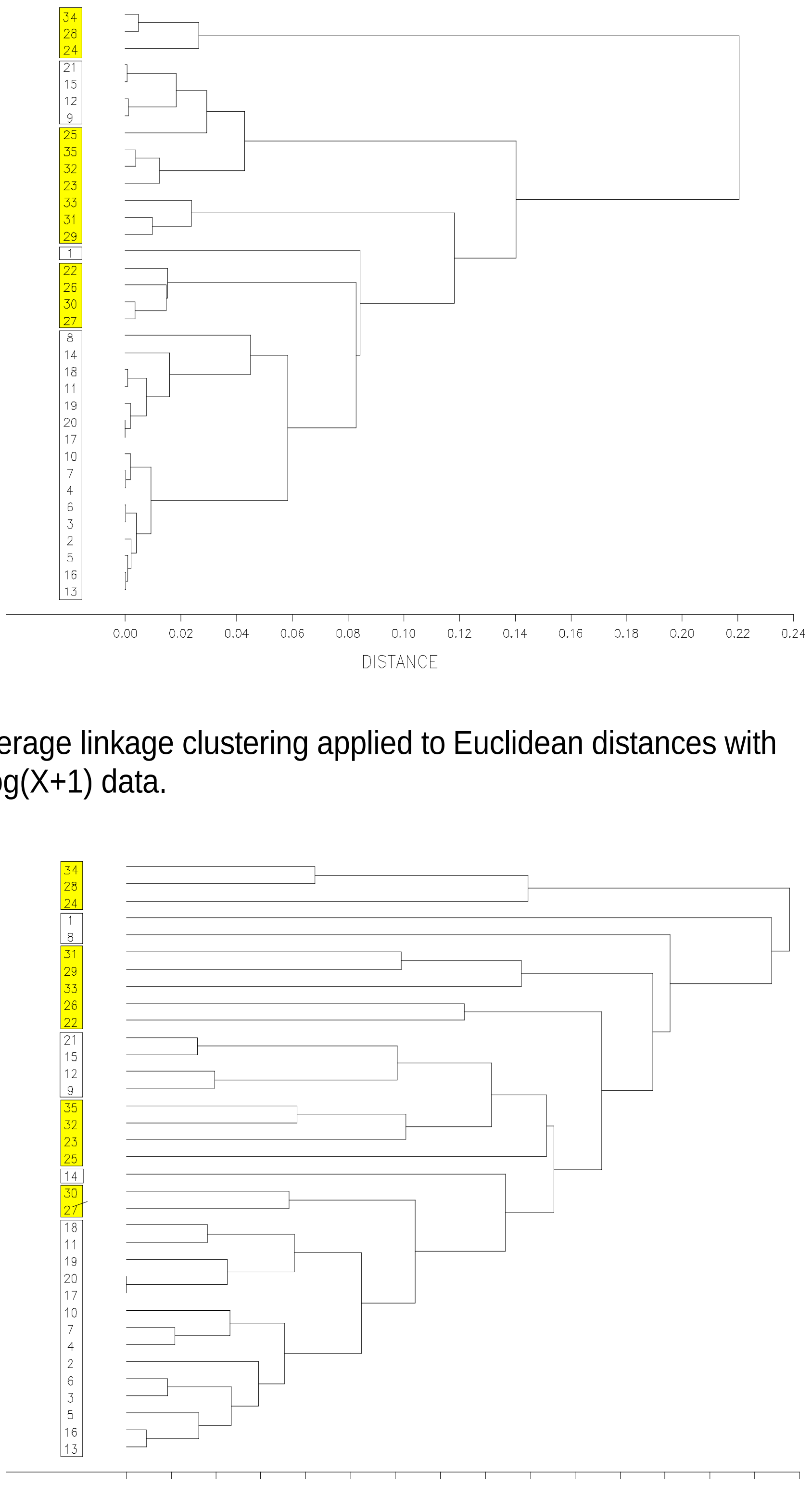
One thing that is apparent from a glance at Figure 7.4, the dendrograms look different, even though the same data had been given each of the clustering methods. It is also clear that there is a large core group of Mangere samples that form a fairly tight cluster, there is also another group of four Mangere samples that reappear as a group (12, 9, 21, 15) in 3 of the four dendrograms. Though at first sight there does not appear to be much more structure, close examination shows that other, Whau Creek, samples consistently reappear as groups or together inside groups ((24, 28, 34), (29, 31, 33), (23, 32, 35), (22, 26), (30, 27)). Much of this cluster structure is due to samples that were taken on the same day. So, even though ordinations do not suggest much cluster structure, the dendrograms do seem to be recovering some informative groupings.

One typical, pleasing, feature of Ward's method that emerges from this diagram is that the dendrogram tends to be clearer to read than other methods. Like Complete Linkage, it tends to slightly exaggerate the differences between clusters, so making them more apparent.

To conserve space, I will not present all the dendrograms that were produced for all the other transformations and standardisations,, but Figure 7.5 shows the dendrogram relating them to each other. It was produced using Ward's clustering algorithm, and indicates how similar the dendrograms produced by the various methods were. The main groups are not defined by the clustering method used, but by the standardisation or transformation. This reinforces the message of the course, that the choice of standardisation is often more important than the choice of method.

It is worth noting that while the dendrograms in Figure 7.4 were fairly successful at identifying plausible groups, most of the others were not. In particular, Bray-Curtis and chi-squared distance on untransformed data were too affected by extreme values to be useful. This is shown by their outlier position in Figure 7.5.

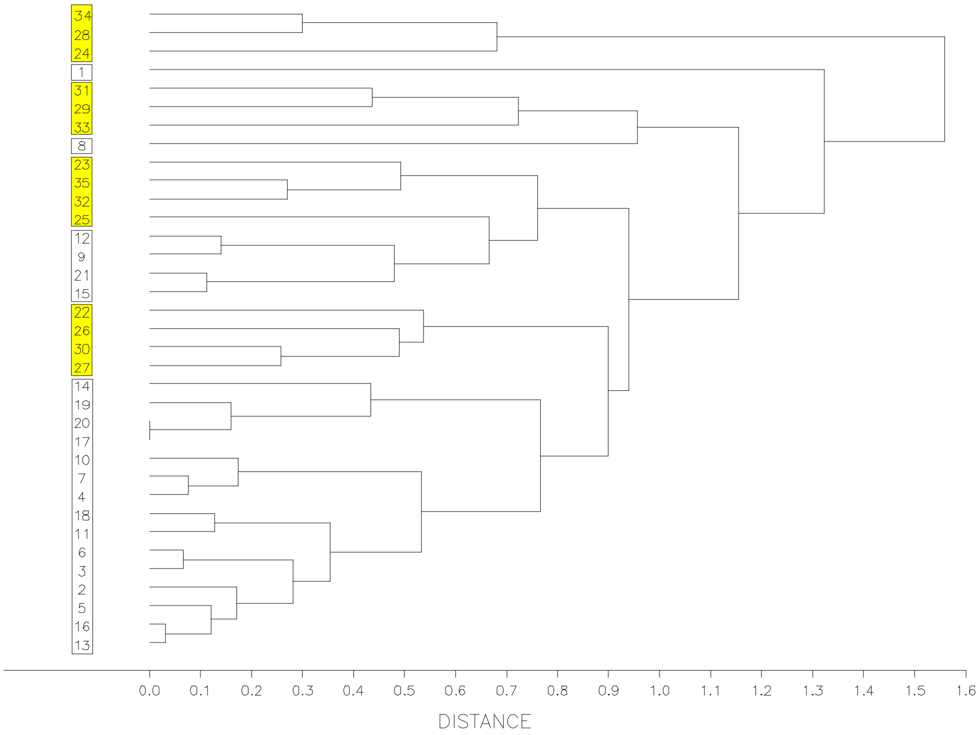
Figure 7.4. Dendrograms for hierarchical cluster analysis on zooplankton data. Whau creek observations are in the stippled boxes, Mangere in the clear. The observations have been sorted within the dendrogram so neighbouring observations are more similar to each other.



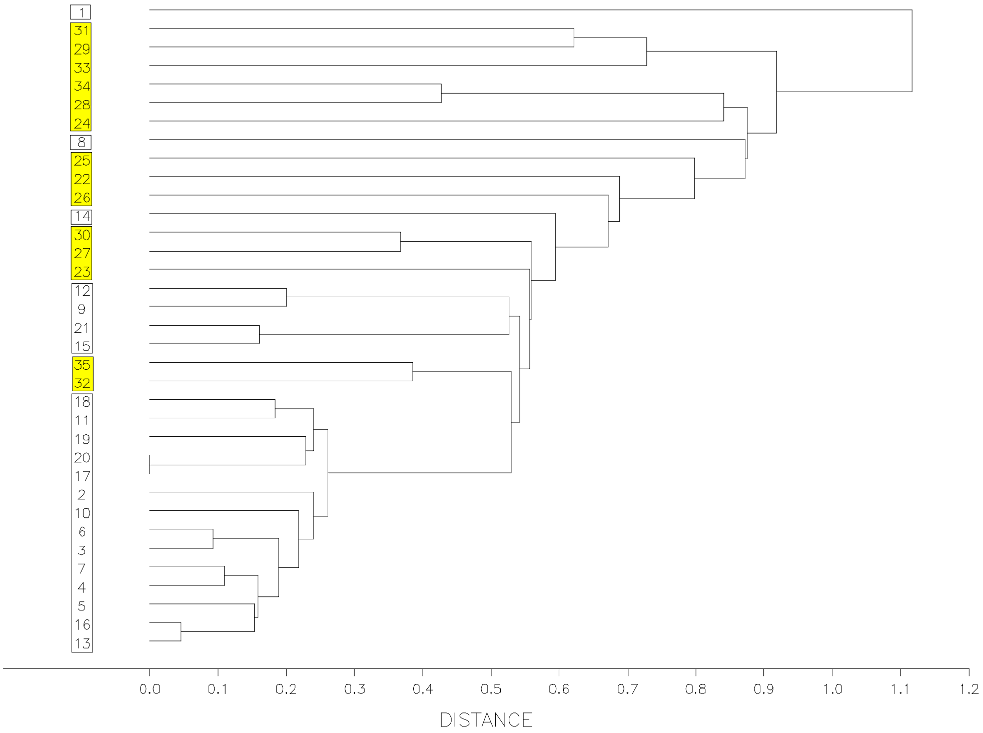
(a) Ward's method applied to Euclidean distances with log(X+1) transformed data.

(b) Average linkage clustering applied to Euclidean distances with log(X+1) data.

c) Complete linkage clustering aplied
Euclidean distances with log(X+1) data



d) Single linkage clustering applied to
Euclidean distances with log(X+1) data



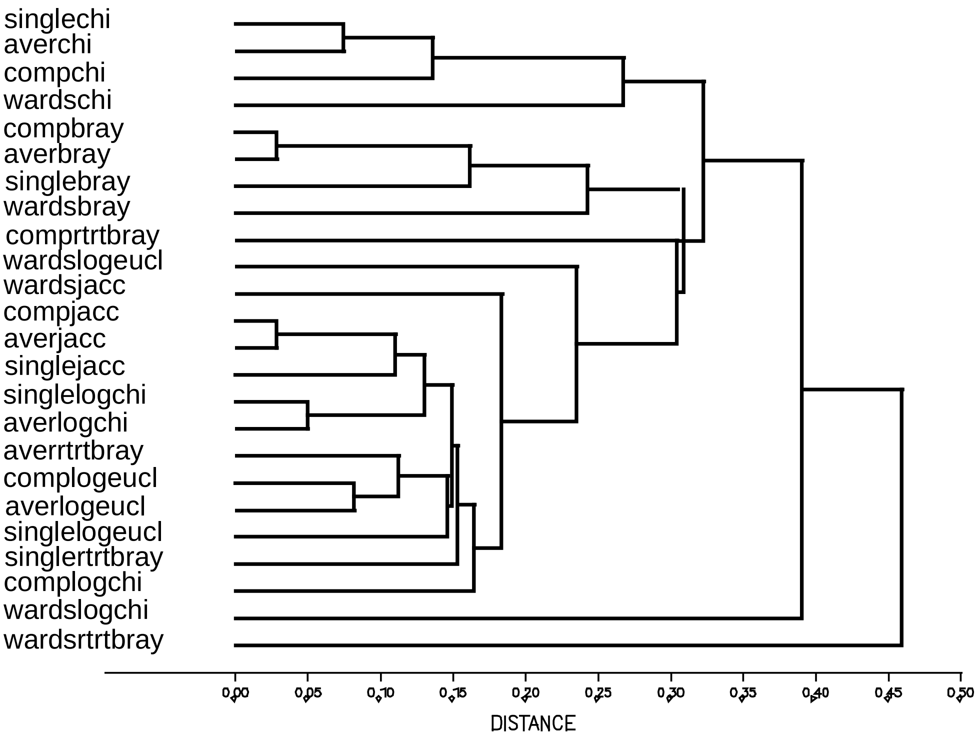


Figure 7.5.Dendrogram from a clustering of clusterings (ordered for similarity - so neighbouring points are similar). A Ward's method cluster analysis was performed on a matrix of distances between the dendrograms from 24 different clusterings of the high tide data. Four very different clustering algorithms were used (Ward's, average linkage, single linkage, and complete linkage) on six distance matrices (Euclidean on log(X+1), chi-squared, chi-squared on log(X+1), Bray-Curtis, Bray-Curtis on root-root transformed abundances, and Jaccard's. Notice how the dendrograms cluster not on the basis of the clustering algorithm but on the standardisation and transformation used in the distance measure

Programming notes

In R we can use "ward" , "average", "single", and "complete". The hang=-1 makes the dendrogram branches all have the same length

```
cl<-hclust(d,method="ward")
plot(cl,hang=-1)
```

AUTHOR'S CHOICE.

When a hierarchical method is needed, Ward's and group average linkage methods (UPGMA) are sensible choices. If the clusters are well separated it will not much matter which method you choose. Remember your choice of transformation or standardisation will usually be more important than your choice of algorithm. If the data consists mainly of continuous or ordered variables, and the observations do not necessarily have a nested structure then one of the non-hierarchical methods for minimising the within cluster variation would usually be more appropriate - e.g *k*-means. If you are dissecting a space rather than trying to recover real clustering then a *k*-means technique is probably best.

Interpretation.

Display.

i) Displaying a partition
a) Ordination.

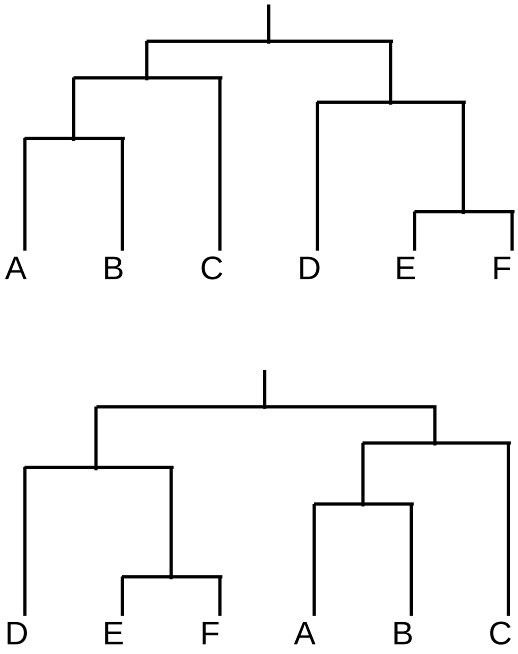
If the main aim of the study was to dissect a more or less homogeneous data cloud then you would usually want to show that the partition is reasonable and at the same time display the relative positions of the sampling units. Perform an ordination on the sampling units and then show their cluster membership in the reduced space plot (e.g. Figure 7.2).

Even if the aim was the recovery of natural clusters, this approach could be used to try to confirm the adequacy of the cluster solution. If the points cluster in the reduced space then you have your confirmation and a good display; if they do not, it means either that there are no clusters in the data or the clusters are not spherical and such splits as exist are on the minor component axes.

b) Canonical Discriminant Analysis.

If the main aim of the study was to recover natural clusters then your main concern would usually be their positions relative to each other. If the recovered clusters have approximately equal variance covariance matrices (and most clustering methods tend to force this anyway), then the relative positions of the cluster centroids can be shown in a reduced space using Canonical Discriminant Analysis (chapter 11).

Figure 7.6. Ordering dendrograms. a) and b) are equivalent. The ordering of a dendrogram is arbitrary. The human eye however cannot help but try to interpret proximity as similarity, so algorithms that impose a sensible ordering are useful.



ii) Displaying a hierarchy.

The dendrogram.

This is easily the commonest way of presenting the results of a hierarchical clustering. It provides a two dimensional record of the clustering history. Clusters (or points) are joined at the distance at which they were fused (their fusion threshold). Large vertical distances between successive fusions implies that a cluster is a long way from other clusters, and therefore discrete. Isolated clusters or solitary points join the major clusters only at the higher levels.

It is not always appreciated that the order of the sampling units along the "crown" of the tree does not necessarily reflect similarity. This means that a dendrogram, as usually presented, can seriously mislead. Two adjacent points on the dendrogram can in fact be very far apart in the full space; while points that are far apart on the dendrogram may in fact be similar to each other. A dendrogram is like a hanging mobile sculpture, the horizontal lines joining clusters are the bars, the vertical lines are the cotton. Any part of the tree can be twisted to reverse the order of its sampling units, thus Figure 7.6 a and b are equivalent, having been twisted at levels I and II.

Number of clusters.

When the purpose of the analysis is the recovery of natural clusters the number of clusters is not usually known in advance. With a hierarchical method there are as many possible clustering levels (partitions) as there are sampling units, and it is normal with partitioning methods to try a number of runs with different *k*. The problem is to choose the best partition. One study lists no fewer than 30 ways of making that decision - and they deliberately left out several more. They compared the performances of the 30 methods using the inevitable Monte Carlo simulations. Their test data was exceptionally well behaved, non-overlapping compact clusters, all roughly the same shape and volume - and unrealistic. They varied the number of units per cluster, number of clusters and dimensionality of the data. Their argument was that if a technique did not perform well on these data, it was not going to be much use with real data. Though it is worth considering that the reverse is not necessarily true.

As you will see, both the methods I present from their investigation will work best with compact spherical data which may not be present in real data. This returns the responsibility for choosing the number of clusters to the worker. There is probably no foolproof way of choosing the optimum number. Since cluster analysis, like most multivariate methods, is mainly a hypothesis generating tool, the choice that suggests the most interesting hypotheses should be used - look at more than one partition and choose the most interesting.

i) Callinski and Harabasz's Index , a.k.a. Pseudo-F statistic (SAS Institute 1987).

$$(\text{Trace}(\mathbf{B})/(k-1))/(\text{trace}(\mathbf{W})/(n-k)).$$

Trace(**B**) is the total between-cluster sum of squares (summed over all variables), so the expression is a sort of total between-cluster mean square over a within-cluster mean square - i.e. a pseudo-F statistic. It is calculated for each partition (level of the hierarchy) and the *k* with the largest value of the index is chosen. If it decreases monotonically as cluster number increases then the data possess a hierarchical structure. If it increases monotonically as cluster number increase then there is probably no cluster structure in the data. Because it ignores the off diagonal terms in the two matrices, it can be expected to work best with spherical clusters with equal variance-covariance matrices - i.e. this kind of data. In fact, in the simulations it recovered the true number of clusters 390 times out of a possible 412.

ii) Duda and Hart's index - only suitable for hierarchical solutions.

$$(W_K+W_L)/W_M.$$

Where W_K and W_L are the sums of squares (over all variables) within the two clusters K and L that are about to be fused, and W_M is the value in the resulting cluster, M. At each fusion a test is performed of the null hypothesis that there is only one cluster, not two. When this is rejected, the current level is chosen so that clusters K and L are kept separate.

In simulations it performed better than the Callinski and Harabasz index when the true number of clusters was greater than two. Most of the methods tested performed worse when there were only two true clusters in the data. Callinski and Harabasz's statistic was an exception; it was very consistent over all cluster numbers. Like Callinski and Harabasz's index, the Duda and Hart index ignores covariation among the variables, so it will also be at its best with spherical clusters, though it might be expected to be more robust to moderate variation among the within cluster variance-covariance matrices.

It is important to realise that this statistic is measuring something very different to Callinski and Harabasz's index. That attempts to assess the variability between all the extant clusters at each level; it is examining the global cluster structure. The Duda and Hart index is just looking at the two clusters currently being fused; it is examining local cluster structure. While global structure would be, ideally, the most useful, local structure is easier to detect - and sometimes easier to interpret. In my experience, with real data, Duda and Hart's index is much more sensitive and informative than Callinski and Harabasz's.

A slight variant of this statistic is in SAS, modified to become a pseudo-t^2 analogous to Hotelling's T^2 statistic (the multivariate equivalent of the *t*-test) with a scalar variance-covariance matrix. This is distributed as F with p and $p(n_K+n_L-2)$ degrees of freedom. This could imply that we could use critical values from tables to "test" whether to fuse or not. In practise it is safer not to rely on any particular cut-off but to look for peaks in the measure to identify levels of the dendrogram which suggest clustering. Certainly it should not be used for formal significance tests except with randomisation techniques.

Programming notes

R. As yet I have no function for the Duda and Hart Pseudo-t^2. However if you load my function `pseudofh` it will produce a Pseudo-F for a range of cluster numbers. So after doing your `hclust` the function is used:

psfs<-pseudofh(*datasetname*,cl,2,8)

This asks for a pseudoF for 2 through 8 clusters.

EXAMPLE 7.3
In Table 7.1 I present 3 of the cluster statistics calculated for the Ward's clustering on the log(X+1) transformed high-tide zooplankton data.

Callinski and Harabasz's index fails to find any clear clustering at all, it increases monotonically with number of clusters, and does not even show any sharp changes that might indicate solutions to examine.

Duda and Hart's statistic (in the form of the pseudo-t^2) suggests that 13 is an interesting number, the two clusters that were fused to form 12 clusters are very different. However, examining the dendrogram for this analysis (Figure 7.4a), the two clusters concerned are small and relatively uninteresting, merely {21, 15} and {12, 9}. We can afford to disregard the 13 cluster solution. There is clearly another peak around 6-8 clusters. The 5 cluster solution is formed by fusing most of the Mangere samples with a block of Whau Creek ones ({22, 26, 27, 30}), clearly not a good idea. One of these solutions would be reasonable, but which to choose would depend on how much detail you

Table 7.1. Clustering statistics for the Ward's clustering of the log(X+1) transformed high-tide zooplankton data: Pseudo-t^2 (Duda and Hart's index), Pseudo-F (Callinski and Harabasz's index). Note that a large Pseudo-t^2 refers to the previous number of clusters; it is measuring the similarity of the last two clusters to be fused. If they are different, you shouldn't have fused them. So you must backtrack one fusion to get the right number of clusters. The pseudo-F finds no clustering at all; the pseudo t^2 suggests 13 (rather too many for convenience), 8-6, or 2.

Number of clusters	Pseudo-t^2	Pseudo- F
16	3.3	18.5
15	4.3	16.7
14	1.7	15.5
13	6.2	14.8
12	21.8	14.2
11	2.4	13.4
10	5.7	12.8
9	4.4	12.4
8	3.9	11.5
7	8.6	11.1
6	9.1	10.6
5	8.3	9.7
4	6.1	9.5
3	6.8	9.0
2	6.9	9.3
1	9.3	

were hoping for from your clustering. By looking at the dendrogram again we see that the peak at 2 clusters is simply due to the outlier group {34, 28, 34}.

After considering the statistics we should now examine the dendrogram (Figure 7.4a), and the contoured ordination to see which solution seems the most informative. The pseudo-t ^{2} statistics suggest the 6 cluster solution and I agree, as this keeps the Whau Creek and Mangere samples largely separate and identifies structure among the Whau Creek samples.

AUTHORS CHOICE.

Ultimately the final choice of the number of clusters is nearly always subjective; does any particular partition suggest biologically interesting groupings? Remember that there might be more than one interesting level of partitioning, even in a non-nested data set. However the methods given here can help. For k-means partitioning methods that minimise trace(**W**) I suggest you try Callinski and Harabasz's index. It assumes multivariate normality and spherical clusters; so do these partitioning methods. It ought to be good with *k*-means methods that use absolute distance For hierarchical methods with spherical data, if there are more than two clusters, use Duda and Hart's index, but examine the dendrogram for large changes in distance between successive fusions. If the clusters are extremely non-spherical and you have used single linkage or one of its family, Duda and Hart's method might be inappropriate.

How valid is the partition?

Having produced a partition of the data into *k* groups either directly, or by sectioning a dendrogram at the appropriate level, it is important that you should demonstrate the validity of the clusters. Otherwise you cannot expect people to accept your solutions as worth looking at. Even if the data have structure, and the clustering technique has produced a partition, you cannot be sure the resulting clusters are any different from a random partition; the limitations of your method may prevent it recapturing the real structure in the data.

Replicating methods.
The simplest, if not the most direct, method is to use two different clustering algorithms, if possible with two different dissimilarity measures for each. If all four solutions are essentially the same, then the chances are the partition is a valid description of the data. It might be considered a bit underhand if the methods used were too similar. For example it would not be acceptable to use a *k*-means method and one that directly minimised a trace(**W**) criterion; they both optimise the same criterion and might be expected to give much the same results. Ward's and the group average methods, or a *k*-means instead of Ward's, might be sensible combinations. Of course in the extremely unlikely event that you got the same result from a single linkage and a complete linkage, even the most sceptical referee would have to admit the existence of the clusters in the data - the more different the analyses, the more plausible the solution. Of course, the transformations or standardisations may make agreement between the methods impossible - the clustering structure of the data cloud can be changed dramatically by transformation or standardisation (see Figure 7.5, the dendrogram of dendrograms).

Interpreting clusters.

There are three major features of a clustering solution that usually demand interpretation: which sampling units are in which groups; which groups are similar to which other groups; and which variables are most important in separating out the groups.

The interpretation of group membership depends on having extra information about the sampling units. For example in the zooplankton clusterings we can interpret the groups on the basis of what site the sampling units were taken from. We could also look at the physical variables that might be correlated with group membership. This can be legitimately done using simple univariate significance tests and descriptive statistics. These tests are not circular as the variables being tested did not contribute to the clustering.

The relationship between groups can usually be seen directly from the data displays discussed in section 7.3; though looking at the group means (centroids) for the variables can tell you a lot.

When looking at the cluster means it is important to realise that for most clustering methods they are biased estimates of the population means of the underlying distributions. The reason is not hard to see. Imagine two overlapping normal distributions. If the points are allocated to clusters solely on the basis of which mean is closer, then extreme points of one distribution will be allocated into the other cluster and vice versa. This will shift the means of the resulting clusters away from each other; they will be further apart than the true means.

One of the more important questions that can arise after a cluster analysis is: which variables are responsible for the difference between the clusters. Some of the variables may only show random differences over the clusters; the clustering may be only apparent on a subset of the variables - perhaps even only one. A simple approach is to perform a separate 1-way ANOVA for each variable, and by ranking the resulting F-values identify which variables are most effective at separating the groups. It is important to remember that these F values cannot be used for significance tests in the normal way. The standard ANOVA significance tests are totally invalid and can tell you nothing about the clusters. The data have been split to maximise some measure of difference between the clusters, or some measure of similarity within them; of course the null hypothesis of no difference between clusters can be rejected - the test is totally circular. If I split a group of people into "greater than 5ft 10 ins" and "less than 5ft 10 ins", it would be rather pointless to then test to see if their mean heights are the same.

If there are many variables, a quick way of identifying the potentially important ones is to use a Canonical Discriminant Analysis and interpret the structure coefficients (we do this later in the course: section 11.3.2). Since a reduced plot from a CDA is a good way of aiding the interpretation of the relationships among the clusters (7.3.1.i.b), this further step is a natural one.

In example 7.1 I used a PCA reduced space plot (Figure 7.2) to indicate the cluster membership.

Programming notes

Plotting labels in a reduced space plot has already been covered in the PCA section. However we have to create variables with the cluster membership information

In R,

```
clusmem<-cutree(cl,k=number of clusters)
```

EXAMPLE 7.5.

If we return to the *k*-means partition of example 7.1, we can now attempt to interpret the groups. The most obvious feature that has already been commented on is that group membership is related to the sites from which the samples were taken. Since this pattern was expected even before we looked at the data (i.e. *a priori*), a simple contingency table test is appropriate. Because of the small numbers a χ^2 test is not possible but a Fisher's Exact Test rejects the null hypothesis that cluster membership is independent of site (p<.0001).

Figure 7.2, the PC plot suggests that group 1 is associated with larger amounts of *Favella*, group 2 with Harpacticoids, group 3 with *Oikopleura*, and it is unclear what is related to group 4. We can expand on that interpretation by presenting the means of the log(X+1) transformed abundances for each group in Table 7.2.

As suggested by the PC plot, group 1 is characterised by high densities of *Favella*, the organic sewage loving ciliate. Group 1 contains nearly all the Mangere observations, close to the sewage oxidation ponds. Group 2 has the highest levels of Harpacticoids, though there does not seem to be anything else particularly characteristic. Group 3 is quite clearly the offshore water in the Whau Creek as it contains characteristically high numbers of *Oikopleura*, an offshore species. Group 4 has large amounts of *Gladioferens*, *Temora*, and *Favella*, an anomalous mix since *Temora* tends to like cleaner, more offshore water, *Favella* likes high nutrient, organically polluted water, and Gladioferens likes relatively unpolluted estuarine waters. Since the 3 members of this group were observations on the same day taken well up the Creek at high tide, I am going to suggest that they represent a mix of offshore *Temora*-bearing water, with bodies of highly polluted estuarine water carrying *Favella* and less polluted estuarine water carrying *Gladioferens*.

If we rank the *F* values from one-way ANOVA between the clusters on each of the variables we find that *Gladioferens, Oikopleura, Temora,* and *Favella* have the largest values. *Oikopleura* and *Temora* marking the cleaner, offshore waters, *Favella*, organically polluted inshore waters, and *Gladioferens* relatively unpolluted estuarine water.

Table 7.2 Means of Log(X+1) transformed high tide zooplankton data for each cluster from example 9.1 (*k*-means method).

Cluster Species	1	2	3	4	F-value from ANOVA
Acartia	5.44	5.91	1.72	6.06	7.9
Euterpina	6.38	6.16	6.16	4.03	2.28
Gladioferens	0.00	0.00	0.93	5.20	37.12
Harpacticoids	0.00	2.57	0.85	1.38	6.04
Oithona	8.26	7.72	6.49	7.91	6.85
Paracalanus	0.00	0.93	1.54	0.00	2.93
Temora	0.00	0.54	2.19	4.91	20.21
Favella	6.53	1.58	1.88	5.47	18.45
Oikopleura	0.51	0.41	6.12	0.00	31.57

Chapter 10. Canonical Discriminant Analysis and MANOVA

Multivariate Analysis of Variance - MANOVA.

Just as ANOVA is (relatively) simply multiple regression on classification variables recoded as binary dummy variables, so MANOVA (Multivariate Analysis of Variance) is a multivariate regression of the response variables onto dummy variables. We will discuss the significance tests later and will restrict ourselves here to the ordinations on the fitted values: Canonical Discriminant Analysis (CDA) based on canonical correlation and the much less often used redundancy analysis on dummy variables.

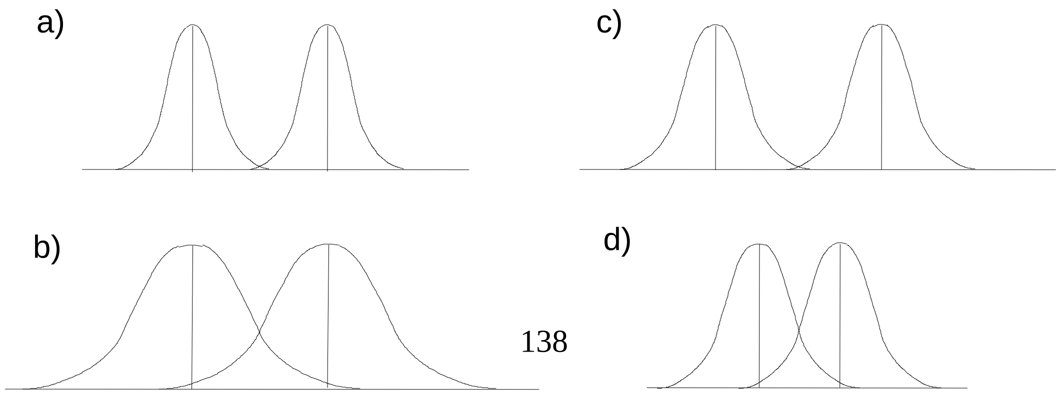
Canonical Discriminant Analysis.

Few multivariate techniques labour under so many names as Canonical Discriminant Analysis (CDA). It is called Discriminant Analysis, Multiple Discriminant Analysis, Canonical Coordinates or Canonical Variates Analysis. The common elements of the names betray its close relationships with Canonical Correlation on the one hand and Discriminant Function Analysis (a different technique not covered in this course) on the other.

In fact though it is usually explained as a separate technique it is simply canonical correlation relating the response variables to a matrix of dummy variables (the design matrix) that specifies the ANOVA design. It specifies the parameters to be estimated in that particular design: e.g. simple treatment effects, block effects, interaction terms etc. CDA is most often used with simple one way designs (more complex designs are discussed later). In this case the fitted values from the model are the sample (treatment) means. The PCA on them therefore displays the difference between the means. It is however important to realise that the space in which the display is done is standardised (this is canonical correlation). This has important consequences on interpretation as we shall see later.

In ANOVA regression is used to separate the total variation into hypothesis (treatment) and error sums of squares. In MANOVA the decomposition is of the total sums of squares and crossproduct matrix (**S**) into hypothesis (**H**) and error (**E**) matrices. The hypothesis SSCP matrix is simply the SSCP matrix of the fitted values and since those fitted values are group or treatment means then it summarises the between centroid variation. We could therefore just do a PCA on the matrix **H,** a simple redundancy analysis with dummy variables (I will describe its advantages later), but this could be misleading. This is clearest in a univariate example. Take fig 1 a & b. These two variables

Figure 10.1. The means of populations a) and b) are equal distances apart, but clearly those in a) are more distinct. When we rescale their axes so that they have the same (unit) within population variance (c and d) the true distinctness of the populations is apparent .



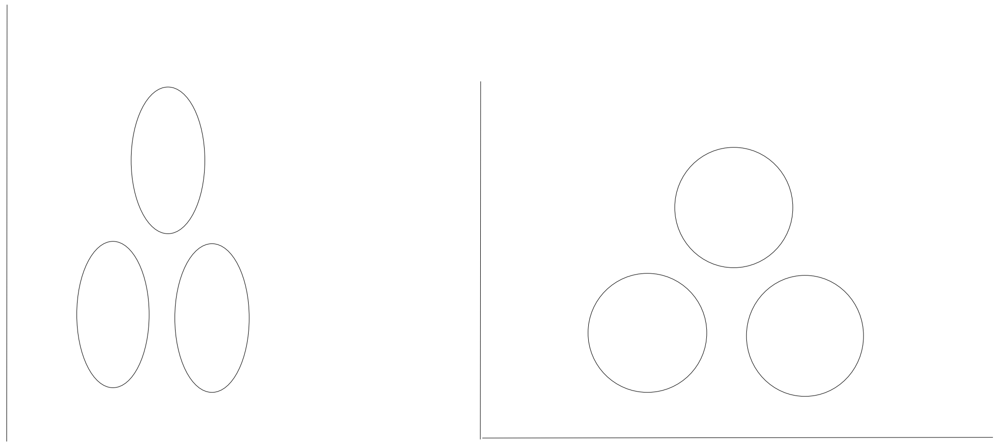


Figure 10.2 a) The distance between the centroids suggest that the centroids are not equidistant. Adjusting the space to standardise by within sample error shows that they are.

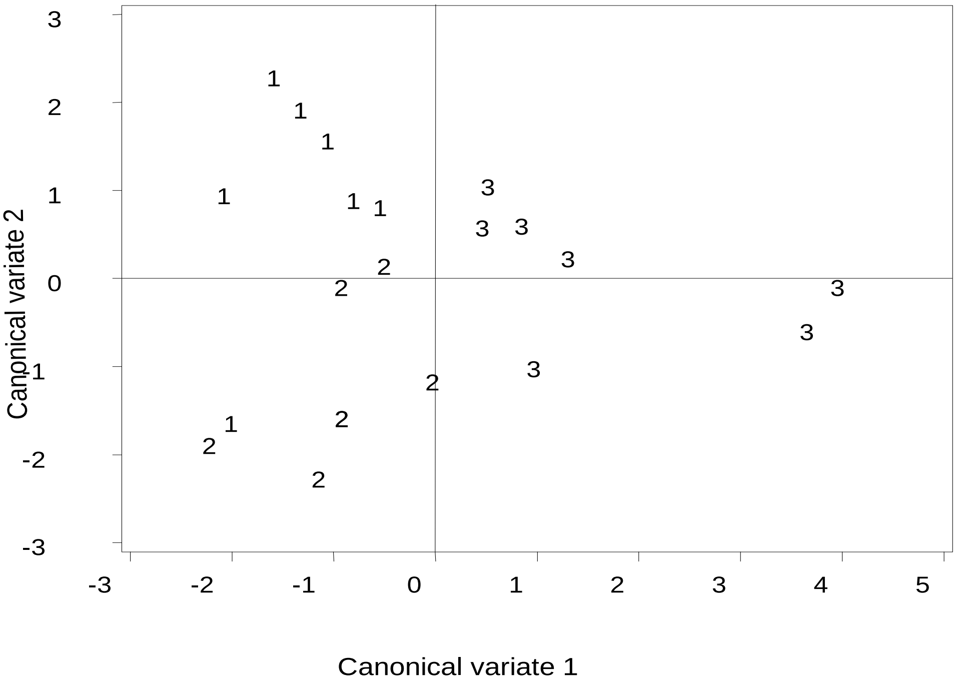
have the same distance between their means, but obviously the means on variable A are better separated than variable B. The simple distance does not reflect this, it should be considered relative to the within sample variation. The simplest solution is to rescale the variables by their within group variation, e.g. divide each value by the within sample standard deviation. The within group SDs will have the value one on the rescaled axes, and the between group distances now reflect the real differences between the means (fig 1 c & d).

This process generalises to the multivariate situation. The between centroid distance in fig 2 a does not reflect the real difference between the centroids. Rescaling the **H** matrix by $\mathbf{E}^{-1}$ (the inverse of the within group variance-covariance matrix) change the axes so that the within group variances are unity and the data clouds for the groups are now spherical. The true differences between the centroids are now apparent. Basically this is a PCA on the centroids in this rescaled space (though in fact it is weighted by the sample size for each centroid). The rescaled old axes are rotated in the new space so that one lies along the major axis between the centroids, then the second major axis, orthogonal to the first in that space, is found, and so on. The number of these axes, canonical variates, that can be found is limited by the number of variables *p* or the number of groups (*g*) minus one, whichever is the smaller. Clearly you can't discover more axes than you started with, hence *p* is a limit; but any *g* points can be perfectly represented in *g*-1 dimensions (try it), so *g*-1 is also a limit.

The result of the analysis is a reduced space display of the group centroids that reflect how separate (distinct) they are, and also a set of new axes (canonical variates) that summarise the between group variation. The contribution of the variables to the new axes can be calculated, and those that are responsible for the differences between the groups identified.

Interpretation of CDA results.

Figure 10.3. The observations on 3 dates plotted on the two canonical variables. Notice the excellent separation



Even though CDA is really a canonical correlation analysis, to some extent the terminology and methods of interpretation have diverged. Canonical correlation coefficients are often still reported but eigenvalues are usually given greater prominence. Canonical variate scores are still calculated but only for the response variables (CVs for the dummy variables are probably not very interesting), and, as we shall see, the canonical and structure coefficients for the response variables are still used - though they are some times renamed as eigenvector coefficients and canonical factor structure respectively.

i) Goodness of fit.

Clearly a significant MANOVA test is a reasonable condition for a useful CDA. As in canonical correlation the eigenvalues λ_i describe the separating power of the associated eigenvectors that define the canonical variates; they give the between group sums of squares of the observation's scores on that canonical variate axis.

An standard, intuitive, approach is to employ a scree diagram to identify the number of axes required to summarise the between group variation. Certainly, the adequacy of the reduced space plot is best described by the sum of the eigenvalues associated with the reduced space plot as a proportion of the sum of all the eigenvalues, e.g. $(\lambda_1 + \lambda_2) / \Sigma\lambda_i$.

ii) Canonical coefficients.

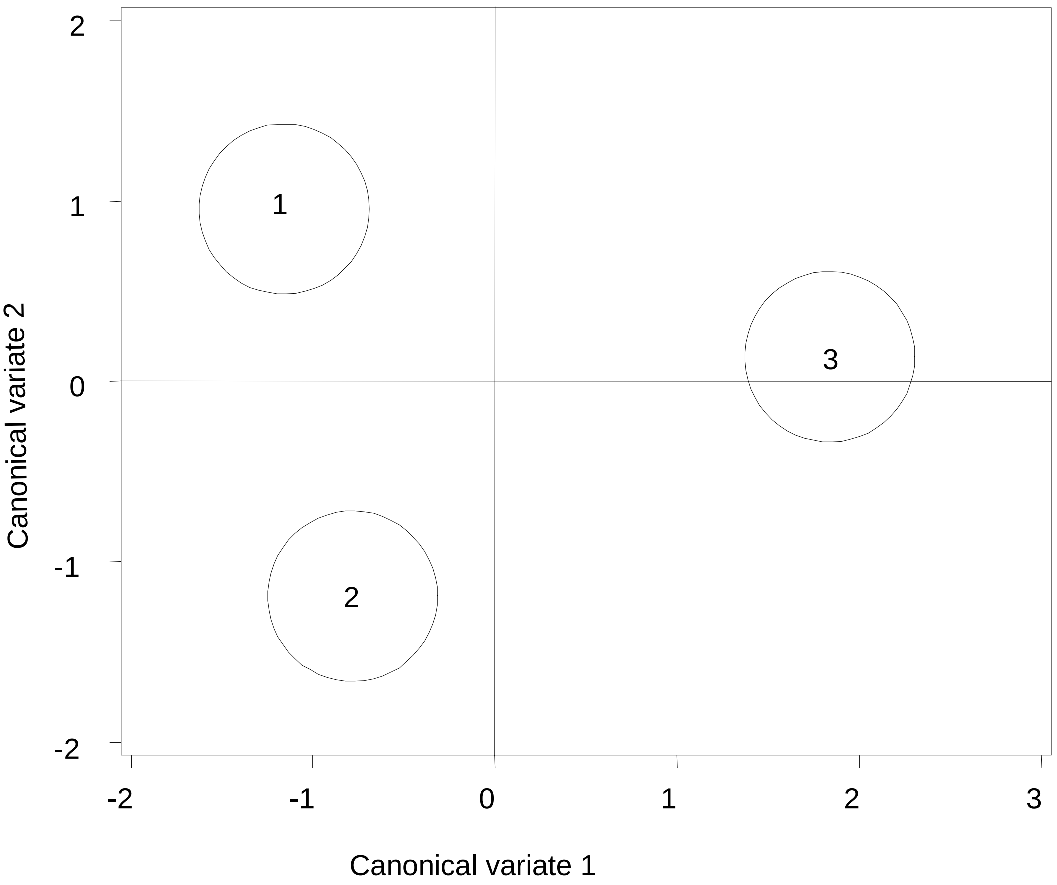
As in most metric ordinations the new axes given by the analysis provide potentially useful information about linear trends that underlie the variation in the data set. In the case of CDA, the canonical variate axes identify the major axes of the variation between the groups. It is therefore normal to attempt to interpret the relationship between these new axes and the original variables. However it is important to realise that the space being described by the ordination has been rescaled by the error so that distance between centroids reflects how distinct the samples (figure 2b), are not simply how different (figure 2a).

Since the CDA is usually being used to analyse the results of an experiment or to describe the difference between groups, the treatment or group identifiers (the dummy variables mentioned earlier) are being used as the predictors, and the *Y* variables as responses. Consider the first canonical variate, the interpretation should tell us which variables would be most distinct for two centroids separated along this line. In this case you should use the structure coefficients (r_{ij}) the correlations between the *i*th Y variable and the *j*th canonical variate.

iii) Display.
The CDA reduced plot can be used to display three different types of information:
a) Centroids and confidence intervals.
The chief aim of CDA is to display the relationships among the centroids. For this reason, the plot of the centroids on the first two canonical variates is incomplete without confidence ellipses around the centroids (intervals are around univariate means, ellipsoids are around multidimensional centroids). No plot of univariate means would get past a referee or examiner without confidence intervals or standard errors. The same rigour should be demanded if CDA is to be used for anything other than data exploration.

The standard confidence ellipses are based on the same assumptions as the MANOVA tests and CDA: random sampling, multivariate normality and homogeneous variance covariance matrices. They make use of the fact that if these assumptions are met the canonical variate scores have unit within group variance, and the data clouds for each group are spherical (the scores are uncorrelated within the groups). Thus simple confidence intervals can be drawn in the 2-D reduced plot as circles of radius $(\chi^2_{2,0.05}/n_i)^{1/2}$, i.e. $2.45/\sqrt{n_i}$.

Figure 10.4 The centroids for each date with approximate confidence regions



b) Overlap. This can be simply shown by plotting the canonical variate scores for the individual sampling units. Give each group a different plotting symbol so group membership is easily recognised. While the overall level of overlap will usually be well described (provided the reduced space plot is itself adequate), the overlap between particular pairs of groups may be badly overestimated; the difference between any two centroids may lie on one or more of the minor axes. This will be most likely when there are more than two important axes, in which case you could do multiple plots (CV1 vs CV3, CV2 vs CV3 etc) to see if such separation exists.

Programming Notes.

To calculate the cvscores:

```
ld<-lda(y variables,  as.factor(class variable))
 cvscores<-predict.lda(ld, Y variables)$x
```

To plot the cvscores:

```
eqscplot(cvscores[,1:2])
```

If you don’t want the scores, just the centroids and/or confidence circles: replace the last command with: eqscplot(cvscores[,1:2], type=”n”)

The centroids (with labels)

```
cvmeans<-predict.lda(ld, ld$means)$x
points(cvmeans[,1:2], pch=19)
text(cvmeans[,1:2], labels=rownames(cvmeans), pos=3)
```

To add confidence circles to the plot:

```
 n=tapply(Y variables, as.factor(class variable), length)
r=2.45/sqrt(n)
symbols(cvmeans[,1], cvmeans[,2],circles=r, inches=FALSE, add=TRUE)
```

EXAMPLE 10.1.

It would be logical to display the differences between the dates to find out which dates are most similar to each other? I therefore performed a CDA treating sites as replicates (though since no significance tests are involved it doesn't much matter whether they are independent or not). Since there are only three dates their centroids can be perfectly represented in two dimensions. The CV scores for the 21 observations are displayed in Figure 2. The centroids with their approximate confidence intervals (circles of radius $(\chi^2_{2,0.05}/n_i)^{1/2}$) are in Figure 3. The first canonical variate (CV) explains 69% of the scaled between centroid variation, the second therefore explains 31%.

The canonical coefficients (raw because the variables are all in the same units) and the structure coefficients are in Table 2. Since we are primarily interested in which species vary between dates we will look at the structure coefficients: the simple correlations between the CVs and the species. The structure coefficients suggest that on the first axis the differences are associated with changes in *Favella*, *Acartia*, and the Harpacticoid group. So we can infer that March (Date 3) had more of the two last species than January or February, while it had less Favella. Similarly, the differences between January and February (CV 2) seem to be associated with a decrease in *Euterpina* and *Acartia*.

Table 1. Coefficients for the first two canonical variates of the difference between dates for the log(X+1) transformed plankton data.

	Canonical variate 1		Canonical variate 2	
	Coeff.	Structure	Coeff.	Structure
Acartia	0.446	0.473	0.160	0.673
Euterpina	-0.032	-0.02	0.619	0.855
Harpacticoids	0.015	0.44	0.282	0.063
Oithona	0.126	0.119	0.133	-0.149
Favella	-0.589	-0.883	0.190	0.307

Robustness of CDA.

CDA is a descriptive technique, none of the tests mentioned later are essential to its validity. It is therefore generally less sensitive to heterogeneity of variance and non-normality than they are. Moderate heterogeneity does not seem to be a major problem, especially if the groups are well separated. Large differences between the variance covariance matrices will distort the inter-centroid distances but a useful picture will usually be produced. Non-normality on its own is usually not such a problem, but skewness can strongly amplify the distortion due to heterogeneity of variance covariance matrices.

The simplest way to check for heterogeneity of variance covariance matrices is to plot the observations as well as the centroids in the reduced space. You can then examine the similarity of the within group distributions. More formal methods were given earlier in the course.

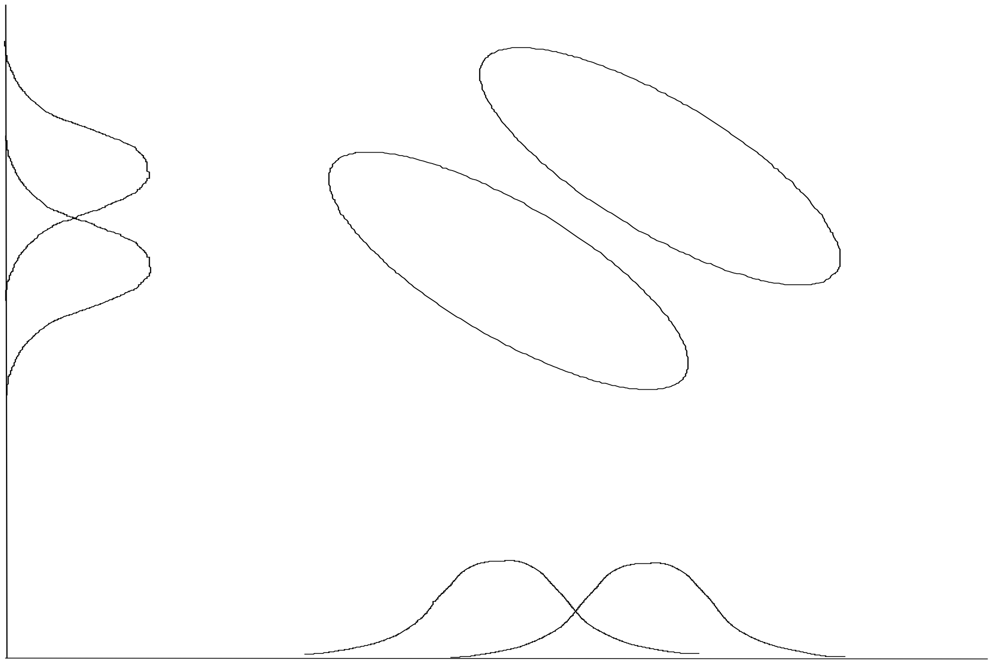


Figure 10.5. Not separate on the marginal axes. But clearly separate in multivariate space.

Significance tests: MANOVA

Most statistics students ought to be familiar with the design of experiments and their analysis by ANOVA (Analysis of Variance). This is a univariate technique; no matter how complex the design, it is only investigating the response of one variable. In many situations you may be interested in measuring the sampling units' response using a suite of variables. Sometimes it will be sufficient to use a separate ANOVA for each variable, especially if they are all measuring something distinct so that they are all relatively uncorrelated within each sample. There are problems with multiple ANOVAs, because, at a 5% significance level, each ANOVA has a 1 in 20 chance of rejecting the null hypothesis even if it is true. So, if *p* significance tests are performed (one for each variable) when there are really no differences, there is a probability of $(1\text{-}.95^{p})$ of getting at least one spuriously significant result. If there are 10 variables the probability is 0.4. Most people happily disregard this problem preferring to sacrifice rigour to power; in most cases it will not matter too much. An experiment where all the null hypotheses are true must be unusual - most biologists do not do an experiment unless there is likely to be an effect. Despite this, one reason to use a multivariate analysis of variance (MANOVA) even in an uncorrelated situation would be to protect against this inflation of the type I error rate.

If however the variables are not all distinct, but are correlated to a greater or lesser extent, MANOVA will usually be more appropriate. In this situation there are two main reasons for employing MANOVA and follow-up techniques instead of multiple ANOVAS.

i) By considering the correlations between the variables you may get increased power in your test, despite protecting yourself from spurious rejection of the null hypothesis that would result from multiple ANOVAs.

ii) You may be able to detect relationships among the variables and the experimental groups that make the differences between the treatment (or group) means more interpretable.

Informal Explanation.

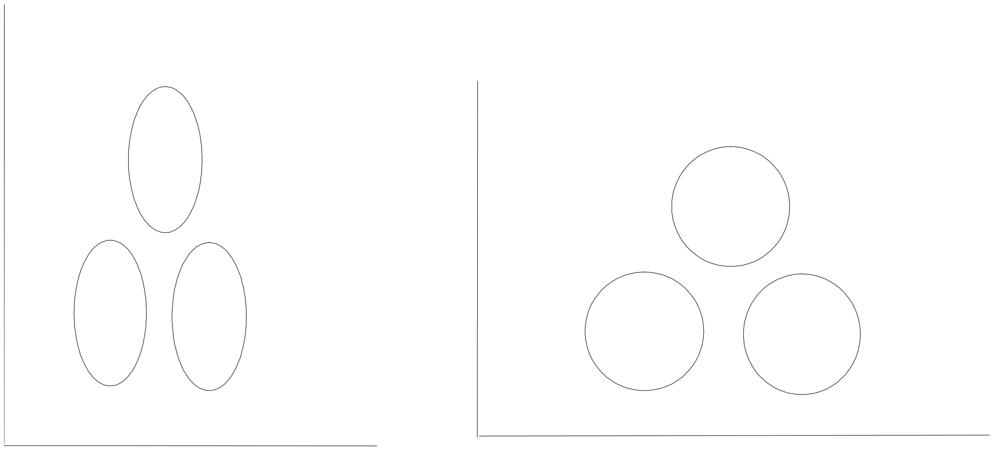


Figure 10.6 a) Between centroid distance does not reflect distinctness. b) In Mahalanobis space it does. The groups are all equally distinct.

Consider fig 5 above . If we look at the variables separately, there is considerable overlap between the two populations. However by looking at the variables in multivariate space, we see the populations are quite distinct. Similarly, if we took samples from these two populations on those variables, we might very well fail to get significant results from separate ANOVAs for each variable; but a MANOVA, which treats the populations as truly two dimensional, would nearly always detect the difference.

ANOVA starts off by partitioning the total sums of squares into an error (within groups) and treatment (between groups) sums of squares. Analogously, MANOVA partitions the total sums of squares and cross products matrix to give an error (within group) matrix **E** and a treatment (between group) matrix **H** (for hypothesis).

Now consider the points in fig6. This space is first rescaled (stretched) so that the data clouds for each group are spherical (fig 6b). This is done by rescaling the data by the inverse of the error matrix, i.e. E^{-1}. This converts all the distances in the space to Mahalanobis' D. I shall discussed this rescaling in more detail earlier - canonical discriminant analysis. CDA then rotates the axes in this new space till they are oriented along the major axes of the variation between the centroids - just like a PCA on the centroids. The data points are now given by scores on these new axes (the CVs). These new variates are used for the tests. As in PCA, these are uncorrelated, and the within group variance is one.
Now the test for the null hypothesis of equal population centroid vectors can be performed. Formally, the null hypothesis can be stated H_0: $\mu_1 = \mu_2 = ... = \mu_g$, where g is the number of groups. Unfortunately though there is one null hypothesis, there are four test statistics. These reflect different ways of relating the between, within and total sums of squares on these new axes. Because the new axes are uncorrelated these sums of squares can be validly combined in ways that would not be possible if we used the original (correlated) variables. This is how scalar multivariate test statistics can be built up that incorporate the variation in the whole set of variables.
There are four main test statistics, all relating the matrix **H** to either the total variation (**H+E)** or the error variation **E**:

i) Wilks Lamda (Λ) is the values of ss_w / ss_T on each of the new axes, multiplied together. If the groups are well separated then ss_W / ss_T will be small for each new axis, and their product smaller still. It is $|\mathbf{E}|/|\mathbf{H} + \mathbf{E}|$.

ii) Pillai's Trace (*V*) is the sum of the values for ss_B / ss_T for all the new axes. If the groups are well separated this will be large. It is trace($\mathbf{H}(\mathbf{H} + \mathbf{E})^{-1}$).

iii) Hotelling-Lawley Trace (*U*) is the sum of the values for ss_B / ss_W for all the new axes. Because the within group variances have been rescaled to 1, *U* is proportional to the sum of the ANOVA *F* values on each of the new axes. It is trace($\mathbf{HE}^{-1}$)

iv) Roy's greatest characteristic root (gcr). This is simply ss_B / ss_W on the first, major, axis. It is therefore proportional to the ANOVA *F* statistic on this axis. This axis is chosen because it maximally separates the centroids, and so the *F* value is the largest possible. It is the first eigenvalue of $\mathbf{HE}^{-1.}$

Though all these methods are testing the same null hypothesis, they do not always give the same results. I shall discuss the appropriate choice of statistic below.

Choice of statistic.

Many, perhaps most, MANOVA programs produce all four statistics; sometimes these will contradict each other. Which to believe? First we can generally disregard the p value for Roy's greatest root, it is usually based on an approximation that in practice nearly always leads to a p value that is much too small. As for choosing between the other 3, it is not an easy decision and has caused a great deal of argument One way we can choose is based on the robustness of the test statistics.

Robustness of the tests.

When the experimental or sampling design is balanced (group sizes are equal) all the tests I have described are fairly robust to non-normality and unequal variance-covariance matrices within groups. As the difference between the variance -covariance matrices increases the gcr test tends to inflate the type I error (is more likely to spuriously reject the null hypothesis). Pillai's trace is the most robust. The robustness of the tests is improved by larger differences between the centroids, by fewer response variables and by having fewer groups rather than more. This does not mean that you should use small sample sizes - larger samples give greater power - you can rely on Pillai's trace's superior robustness to protect you.

None of the test statistics given above are robust when the sample sizes are unequal. Just as in ANOVA, violations of the assumptions combined with unequal sample sizes can result in spuriously significant results; at other times it can lead to failing to detect differences that are really there.
The assumptions of normality and equal variance covariance matrices can be tested using Levene's test.

Programming Notes:

```
man= manova(lm(as.matrix(Y variables) ~ as.factor(Class variable)))
summary(man, test="Pillai")
```

You can get other test statistics using the options "Wilks", "Hotelling-Lawley", "Roy" instead of "Pillai".

EXAMPLE 10.2

Though the plankton data were not collected with strict hypothesis tests in mind, we can still use tests like MANOVA as exploratory tools to help us check for patterns in the data that we can expect *a priori*. For example some workers might test to see if there was a consistent difference between the harbours. They might, I wouldn't; the difference is so obvious even in the raw data that there is no point in testing for it. A less trivial example might be to look for differences between sampling dates or between stations within a site. Within the Mangere Inlet, samples were taken at 7 different stations on 3 different occasions (January, February and March 1979). Since some of the sites are on mudflats some samples were only possible at high tide, so I have excluded the low tide data. Though there are no replicates on each occasion we could treat the problem as a two way unreplicated design. This lumps any interaction term into the error, thus making it potentially insensitive, but since I am looking for general averaged trends over these particular stations, I am content to ignore the interaction, and consider it to be the error term.

I chose to use the Mangere Inlet rather than Whau Creek data because there are fewer species in Mangere. This is actually very important. It is impossible to do a MANOVA at Whau with all 9 species because there are not enough error degrees of freedom. The reduced diversity at Mangere (5 species) makes it possible.

The results of the analysis on the log(X+1) transformed data are in Table 12.1a. It is worth noting that the test for Roy's gcr given by R strongly suggested a difference between stations (*p*-value = 0.0023), but using the more precise values from tables of gcr failed to detect any station differences. A result that is much more in keeping with the other statistics.

Table 2a.
Test for differences between stations

		p-value
Wilks' Lambda	0.052	0.2708
Pillai's Trace	1.863	0.2802
Hotelling-Lawley Trace	5.563	0.3168
Roy's Greatest Root	3.485	>.05

Table 2b
Test for differences between months

		p-value
Wilks' Lambda	0.063	0.0029
Pillai's Trace	1.463	0.0017
Hotelling-Lawley Trace	0.486	0.0053
Roy's Greatest Root	4.728	<.05

There does seem to be some consistent differences between months, but the stations do not seem to be detectably different on average. That is not to say that the stations might not be different on any given date, nor that if we had sampled another phase of the tide we might not have got more obvious differences. However it does suggest that it may more productive to carry on to examine differences between months rather than stations.

Interpretation of MANOVA results.

As in ANOVA, a significant test statistic is the start not the end of the analysis. In ANOVA the next step is the investigation of the differences among the means by simple plots, followed either by comparisons planned in advance (planned contrasts) or by using simultaneous test procedures like Fisher's LSD, Duncan's multiple range test, Tukeys HSD (my favourite) or the Student-Neuman-Keuls test. In MANOVA you not only have to investigate the differences between the centroids, but also identify which variables are responsible for them.

Univariate Anovas.

The simplest way to identify which variables have responded to the experimental regime is to perform separate Anovas for each of the response variables. This will allow you to identify which

variables are most different between the groups. The *p*-values should not be taken too literally because you have done a number of tests and so are subject to Type I error rate inflation.
One simple approach that I have found useful is to ignore the significance tests altogether. Which variables have responded most can be identified by ranking them on their ANOVA *F* statistics; remember the *F* value is the between variation scaled by the within. Which means are different from which others is perhaps best investigated using Canonical Discriminant Analysis described earlier.

EXAMPLE 10.3
Having demonstrated that differences exist, we should now try to find out which variables are responsible. As mentioned above, one of the simplest approaches is to examine the univariate ANOVAs. The tests for differences between months are shown in table 10.3.

Table 10.3 Results of univariate ANOVAs

Species	*F* statistic	*p*-value
Acartia	5.74	0.0178
Euterpina	6.25	0.0138
Harpacticoids	1.26	0.318
Oithona	0.17	0.847
Favella	17.6	0.0003

R tips for STAT302

1) How to read in a .csv file

Comma delimited files (.csv) are simple text file with every data value separated from the other by a comma. They can be read in and created by virtually every statistics package/language. CSV files can be created in Excel (Save as and choose CSV Comma delimited (.csv) from the Save as Type pull down menu).

I will provide assignment data as .csv files.
To read them into R.

dataset name=read.csv(file.choose(),header=T)

Then check it went in OK with

fix(*dataset name*)

This opens a spreadsheet window so you can look at the data set and check what you got. You must close the window before you can execute any more R commands.

2) Some functions are from libraries that need to be loaded first. e.g. eqscplot() is often used but you must enter library(MASS) first, because eqscplot is from that library

3) How to log transform a variable or a bunch of variables.

I typically refer to a bunch of variables by the column numbers in the data set.

Thus

a) To transform one variable in column 3 of the data set (log(X+1) transformation)

var.log=log(*dataset*[,3]+1)

Note: a matrix or dataset has a row index and a column index. Leaving an index blank says use all its possible values. So, in the example above, the row index is missing which says use all the rows in the matrix.

b) To transform a block of columns (3 to 10):

vars.log=log(*dataset*[,3:10]+1)

c) To transform a selection of columns (3,5 and 9):

vars.sel.log=log(*dataset*[,c(3,5,9)]+1)

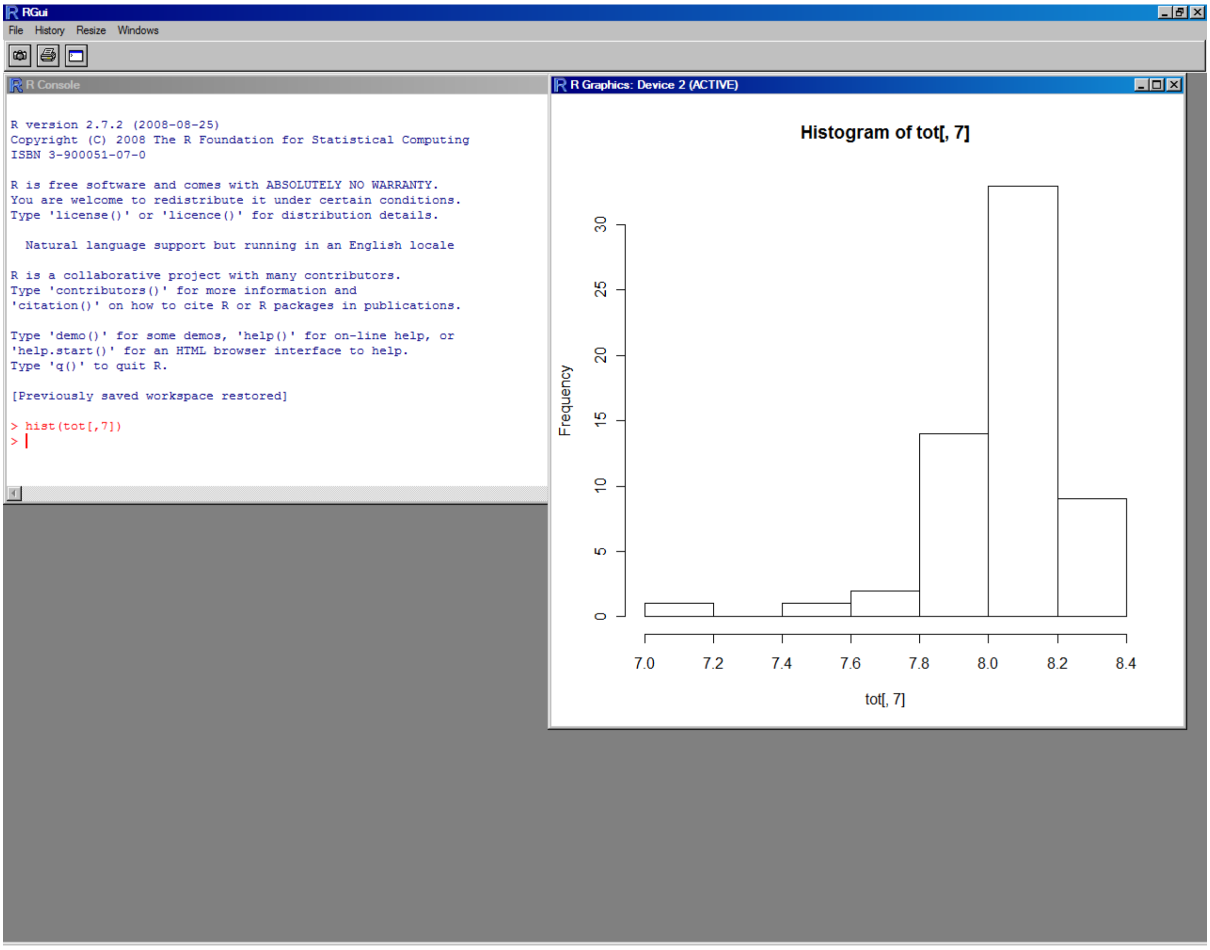
function c() creates a list.

Note: To find which variables are in which columns use names(*dataset*).

d) Sometimes it is easier to do the transformation in Excel and then save it as a .csv file and read it in to R again.

4) To harvest a plot and paste it into a MSWord document (or Powerpoint if you want to edit it).

Have the plot window active (the bar is blue as in the screen below) then use Ctrl-W to copy it in a suitable format to the clipboard.

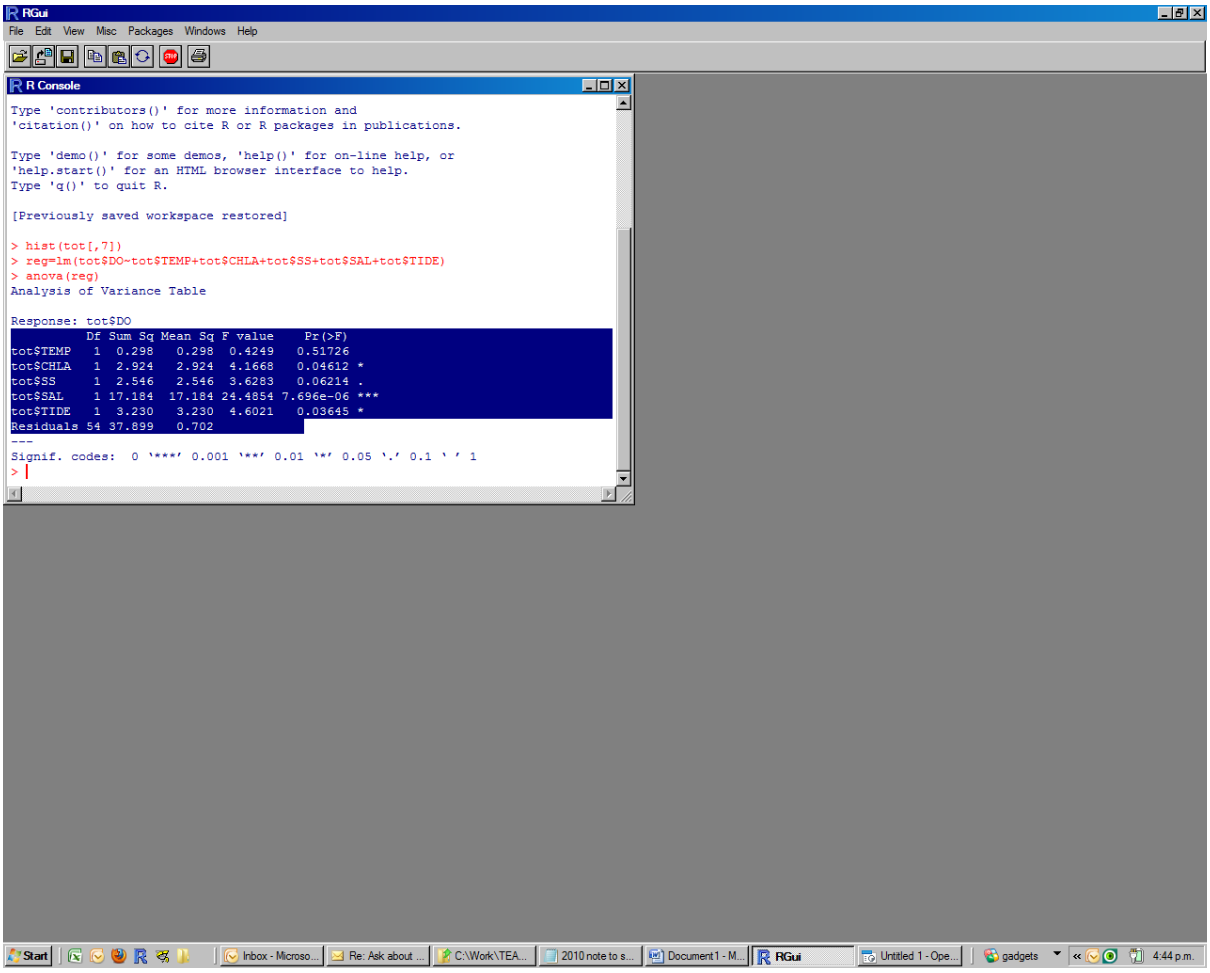


```
R version 2.7.2 (2008-08-25)
Copyright (C) 2008 The R Foundation for Statistical Computing
ISBN 3-900051-07-0

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

  Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

[Previously saved workspace restored]

> hist(tot[,7])
> 
```

Then paste into MSWord or another program.

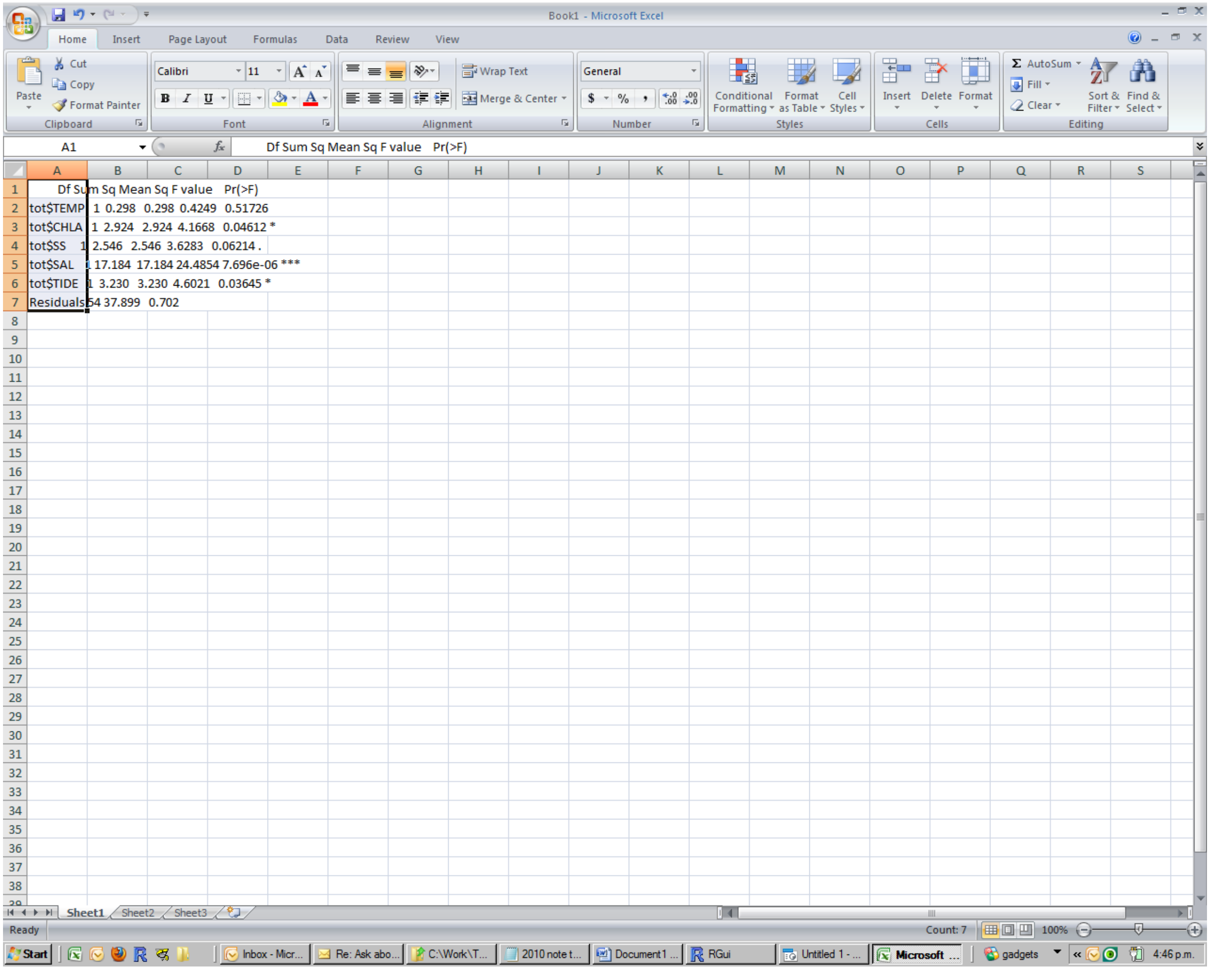
5) To copy a table from R and paste it neatly into MSWord go by way of Excel.

First select and copy the R table you want – an ANOVA table in the example below



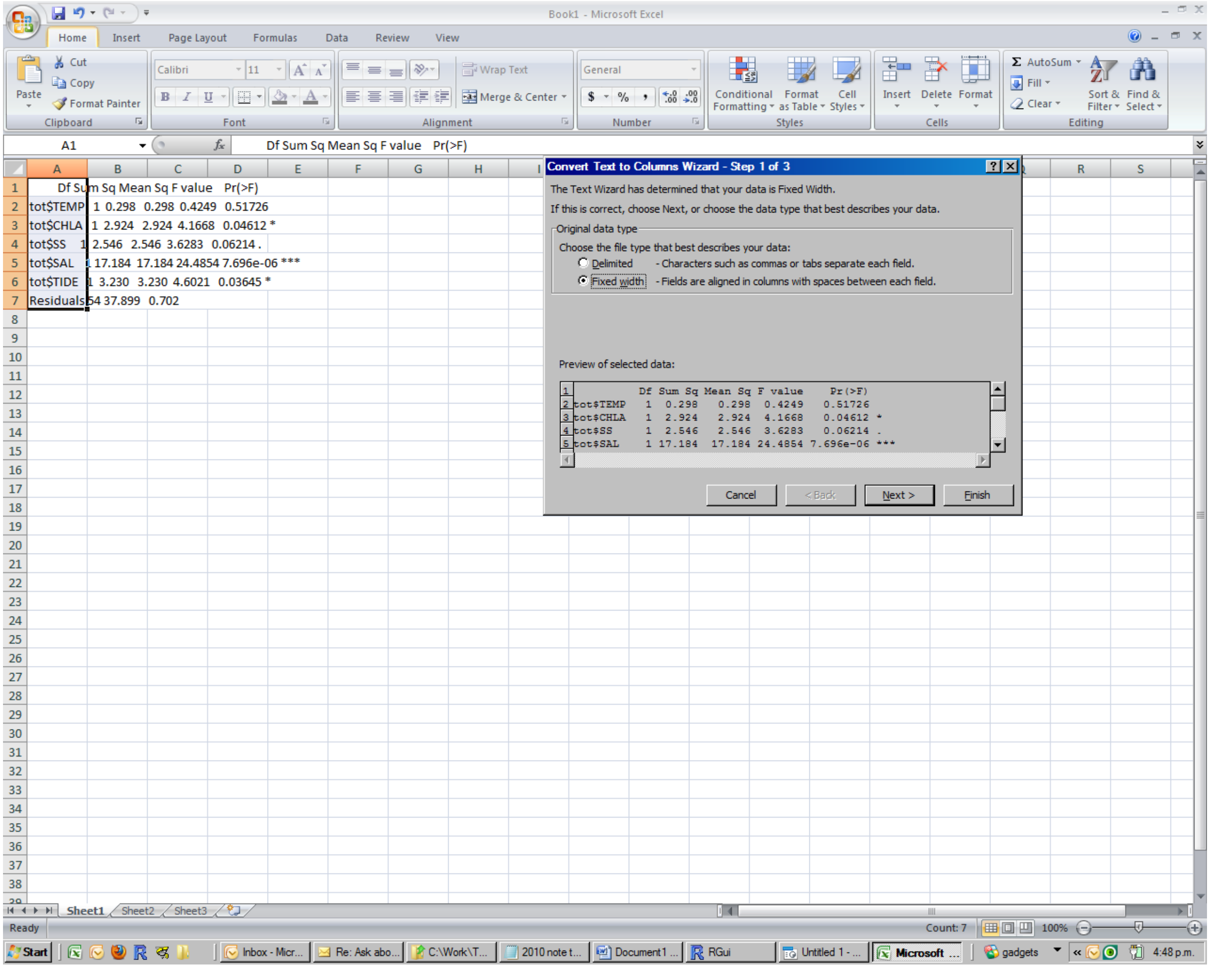
```
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

[Previously saved workspace restored]

> hist(tot[,7])
> reg=lm(tot$DO~tot$TEMP+tot$CHLA+tot$SS+tot$SAL+tot$TIDE)
> anova(reg)
Analysis of Variance Table

Response: tot$DO
          Df Sum Sq Mean Sq F value    Pr(>F)
tot$TEMP   1  0.298   0.298  0.4249   0.51726
tot$CHLA   1  2.924   2.924  4.1668   0.04612 *
tot$SS     1  2.546   2.546  3.6283   0.06214 .
tot$SAL    1 17.184  17.184 24.4854 7.696e-06 ***
tot$TIDE   1  3.230   3.230  4.6021   0.03645 *
Residuals 54 37.899   0.702
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
> 
```

Now paste it into Excel.



```
        Df Sum Sq Mean Sq F value    Pr(>F)
tot$TEMP  1  0.298   0.298  0.4249   0.51726
tot$CHLA  1  2.924   2.924  4.1668   0.04612 *
tot$SS    1  2.546   2.546  3.6283   0.06214 .
tot$SAL   1 17.184  17.184 24.4854 7.696e-06 ***
tot$TIDE  1  3.230   3.230  4.6021   0.03645 *
Residuals 54 37.899   0.702
```

Now: Alt-d, then Alt-e



A1 — Df Sum Sq Mean Sq F value Pr(>F)

```
          Df Sum Sq Mean Sq F value   Pr(>F)
tot$TEMP   1  0.298   0.298  0.4249  0.51726
tot$CHLA   1  2.924   2.924  4.1668  0.04612 *
tot$SS     1  2.546   2.546  3.6283  0.06214 .
tot$SAL    1 17.184  17.184 24.4854 7.696e-06 ***
tot$TIDE   1  3.230   3.230  4.6021  0.03645 *
Residuals 54 37.899   0.702
```

Convert Text to Columns Wizard - Step 1 of 3

The Text Wizard has determined that your data is Fixed Width.

If this is correct, choose Next, or choose the data type that best describes your data.

Original data type

Choose the file type that best describes your data:

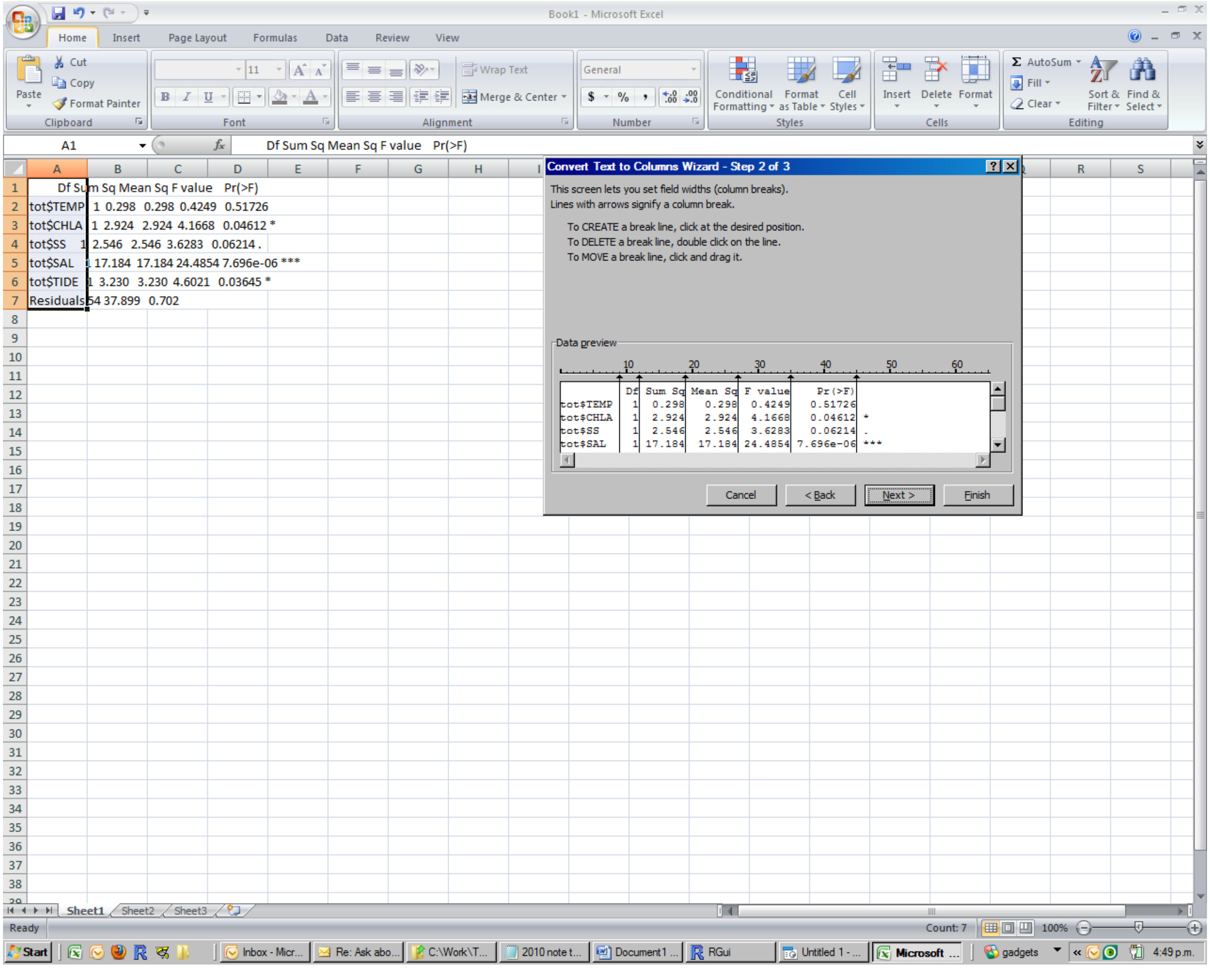
○ Delimited - Characters such as commas or tabs separate each field.

● Fixed width - Fields are aligned in columns with spaces between each field.

Preview of selected data:

```
1           Df Sum Sq Mean Sq F value    Pr(>F)
2 tot$TEMP   1  0.298   0.298  0.4249   0.51726
3 tot$CHLA   1  2.924   2.924  4.1668   0.04612 *
4 tot$SS     1  2.546   2.546  3.6283   0.06214 .
5 tot$SAL    1 17.184  17.184 24.4854 7.696e-06 ***
```

Then press Next in the Box



```
        Df Sum Sq Mean Sq F value   Pr(>F)
tot$TEMP  1  0.298  0.298  0.4249  0.51726
tot$CHLA  1  2.924  2.924  4.1668  0.04612 *
tot$SS    1  2.546  2.546  3.6283  0.06214 .
tot$SAL   1 17.184 17.184 24.4854 7.696e-06 ***
tot$TIDE  1  3.230  3.230  4.6021  0.03645 *
Residuals 54 37.899  0.702
```

Convert Text to Columns Wizard - Step 2 of 3

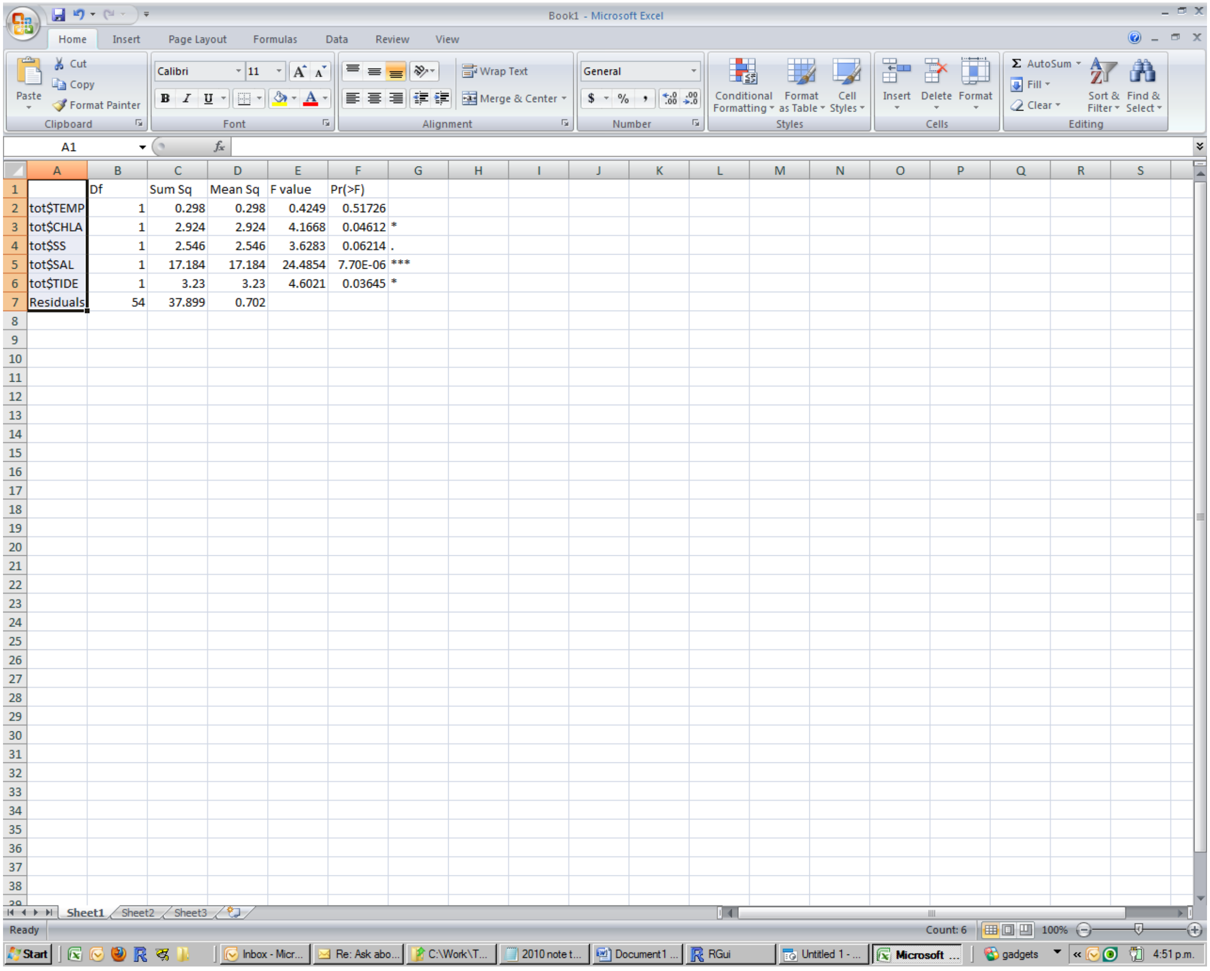
This screen lets you set field widths (column breaks).
Lines with arrows signify a column break.

To CREATE a break line, click at the desired position.
To DELETE a break line, double click on the line.
To MOVE a break line, click and drag it.

Data preview

```
         |Df|Sum Sq|Mean Sq|F value|   Pr(>F)|
tot$TEMP | 1| 0.298|  0.298| 0.4249|  0.51726|
tot$CHLA | 1| 2.924|  2.924| 4.1668|  0.04612| *
tot$SS   | 1| 2.546|  2.546| 3.6283|  0.06214| .
tot$SAL  | 1|17.184| 17.184|24.4854|7.696e-06| ***
```

Check that the splits into separate columns are in the right place. Double click on an arrow to remove a split click in a gap to put one in
Now click Finish



	A	B	C	D	E	F	G
1		Df	Sum Sq	Mean Sq	F value	Pr(>F)	
2	tot$TEMP	1	0.298	0.298	0.4249	0.51726	
3	tot$CHLA	1	2.924	2.924	4.1668	0.04612	*
4	tot$SS	1	2.546	2.546	3.6283	0.06214	.
5	tot$SAL	1	17.184	17.184	24.4854	7.70E-06	***
6	tot$TIDE	1	3.23	3.23	4.6021	0.03645	*
7	Residuals	54	37.899	0.702			

Now copy and Paste Special > Formatted text (RTF) the table into MSWord. Pretty isn't it?

	Df	Sum Sq	Mean Sq	F value	Pr(>F)	
tot$TEMP	1	0.298	0.298	0.4249	0.51726	
tot$CHLA	1	2.924	2.924	4.1668	0.04612	*
tot$SS	1	2.546	2.546	3.6283	0.06214	.
tot$SAL	1	17.184	17.184	24.4854	7.70E-06	***
tot$TIDE	1	3.23	3.23	4.6021	0.03645	*
Residuals	54	37.899	0.702			

Useful resources on the net:

http://maths.anu.edu.au/~johnm/courseweb/r-courseprep.html

http://www.youtube.com/off2themovies2

http://www.ms.unimelb.edu.au/~andrewpr/r-users/icebreakeR.pdf