Multivariate Data Analysis

Practical and theoretical aspects of analysing multivariate data with R

Nick Fieller

Sheffield

(Left blank for notes)

Contents

Multivariate Data Analysis

0. Introduction

0.0 Books

Gnanadesikan , R. (1997) ***Methods for Statistical Data Analysis of Multivariate Observations*.** (2nd Edition). Wiley.

Mardia, K., Kent, J. & Bibby, J. (1981) ***Multivariate Analysis***. Wiley.

Cox, Trevor (2005) ***An Introduction to Multivariate Analysis***. Arnold.

Everitt, Brian (2005), ***An R and S-PLUS® Companion to Multivariate Analysis*.** Springer. Support material is available at

http://biostatistics.iop.kcl.ac.uk/publications/everitt/

Manly, Bryan (2004) ***Multivariate statistical methods: a primer*,** (3rd Edition). Chapman & Hall. Data sets are available at

http://www.crcpress.com/e_products/downloads/download.asp?cat_no=C4142

Venables, W. N. & Ripley, B. D. (2002) ***Modern Applied Statistics with S***, (4th Edition), Springer. Support material is available at:

http://www.stats.ox.ac.uk/pub/MASS4

Ripley, B.D. (1996) ***Pattern Recognition and Neural Networks***. Cambridge University Press.

Hand, D.J., Mannila, H. & Smyth, P. (2001) ***Principles of Data Mining.*** MIT Press.

Johnson, R.A. & Wichern, D.W. (2002) ***Applied Multivariate Statistical Analysis***. (5th Edition). Prentice-Hall.

Everitt, B.S. & Dunn, G. (2001) ***Applied Multivariate Data Analysis.*** (2nd Edition). Arnold

Barnett, V. (ed.) (1981) ***Interpreting Multivariate Data.*** Wiley

Morrison, D.F. (1960) ***Multivariate Statistical Methods.***

Jolliffe, I.T. (2002) ***Principal Component Analysis***. Springer.

Krzanowski, W. (1990) ***Multivariate Analysis.*** Oxford

Krzanowski, W. & Marriot, F.H.C. (1994 & 5) ***Multivariate Analysis, parts 1 & 2.*** Arnold.

Hartung, J, & Elpelt, B, (1995) ***Multivariate Statistik.*** Oldenburg. (*In German).*

The first two texts are the prime sources for much of the material in these notes and specific acknowledgements to examples from them have not been provided. The two texts by Trevor Cox and Brian Everitt cover most of the material in the course and are modestly priced. The text by Bryan Manly provides an excellent introduction to many of the techniques in this course and is the source of some of the examples in this course. It is recommended for those who want a preliminary introduction on specific topics before returning to the presentation in these notes.

0.1 Objectives

The objectives of this book are to give an introduction to the practical and theoretical aspects of the problems that arise in analysing multivariate data. Multivariate data consist of measurements made on each of several variables on each observational unit. Some multivariate problems are extensions of standard univariate ones, others only arise in multidimensions. Similarly, the statistical methodology available to tackle them is in part an extension of standard univariate methods and in part methods which have no counterpart in one dimension. One of the aims of the book is to give emphasis to the practical computational implementation of the analyses using **R**. On accasions reference is made to other packages, notably S-PLUS and MINITAB, especially where there are differences of any note.

0.2 Organization of course material

The main Chapters 1–9 are largely based on material in the first two books in the list of recommended texts above (i.e. Gnanadesikan and Mardia et al), supplemented by various examples and illustrations. Some background mathematical details (properties of eigenvalues & eigenvectors, Lagrange multipliers, differentiation with respect to vectors, maximum likelihood estimation) are outlined in Appendix 0. If you want more details of matrix algebra, in particular how to use **R** to manipulate matrices, then see the notes *Basics of Matrix Algebra with **R*** (which are still under development) at

http://nickfieller.staff.shef.ac.uk/sheff-only/BasicMatrixAlgebra.html

These notes go considerably further than is required for this course.

Appendices 1–8 are provided for those wanting an introduction to some of the useful recently developed techniques that are widely used in industry and which may be of use in other courses involving project work. This material is largely based on the book by Venables & Ripley. There are some chapters (5 – 7) and a few individual sections that are marked by a star,*, which indicates that although they are part of the course they are not central to the main themes of the course. Largely these contain technical material or further topics. In some cases they provide the underlying justification of the material and and are not needed for practical applications but their existence gives confidence in the interpretation and application of the data analytic methods described.

The expository material is supplemented by simple ‘quick problems’ (*task sheets*) and more substantial *exercises.* These task sheets are designed for you to test your own understanding of the material. If you are not able to complete the tasks then you should go back to the immediately preceding sections (and re-read the relevant section (and if necessary re-read again & …). Solutions are provided at the end of the book.

0.3 A Note on S-PLUS and R

The main statistical package for this course is **R.**

R is a freely available programme which can be downloaded over the web from http://cran.r-project.org/ or any of the mirror sites linked from there. It is very similar to the copyright package S-PLUS and the command line commands of S-PLUS are [almost] interchangeable with those of **R.** Unlike S-PLUS, **R** has only a very limited menu system which covers some operational aspect but no statistical analyses. Almost all commands and functions used in one package will work in the other.

However, there are some differences between them. In particular, there are some options and parameters available in **R** functions which are not available in S-PLUS. Both S-PLUS and **R** have excellent help systems and a quick check with `help(`*`function`*`)` will pinpoint any differences that are causing difficulties. A key advantage of **R** over S-PLUS is the large number of libraries contributed by users to perform many sophisticated analyses. These are updated very frequently and extend the capabilities substantially. If you are considering using multivariate techniques for your own work then you would be well advised to use **R** in preference to S-PLUS. Command-line code for the more substantial analyses given in the notes for this course have been tested in **R.** In general, they will work in S-PLUS as well but there could be some minor difficulties which are easily resolved using the help system. The version of **R** at time of going to press is 3.0.1. but some parts of the material were prepared with slightly earlier versions. Both packages are fully upwardly compatible but there could be slight discrepancies between output produced now and those in thebook. These are most unlikely to be substantial or of practical importance.

0.4 Data sets

Some of the data sets used in this course are standard example data sets which are automatically installed with the base system of **R** or with the MASS library. Others are available in a variety of formats on the associated web page available here.

0.4.1 R data sets

Those in **R** are given first and they have extensions .Rdata; to use them it is necessary to copy them to your own hard disk. This is done by using a web browser to navigate to the web page here, clicking with the right-hand button and selecting 'save target as…' or similar which opens a dialog box for you to specify which folder to save them to. Keeping the default .Rdata extension is recommended and then if you use Windows explorer to locate the file a double click on it will open **R** with the data set loaded and it will change the working directory to the folder where the file is located. For convenience all the **R** data sets for the course are also given in a WinZip file.

NOTE: It is not in general possible to use a web browser to locate the data set on a web server and then open R by double clicking. The reason is that you only have read access rights to the web page and since **R** changes the working directory to the folder containing the data set write access is required.

0.4.2 Data sets in other formats

Most but not all of the data sets are available in other formats (Minitab, SPSS etc). It is recommended that the files be downloaded to your own hard disk before loading them into any package**.**

0.5 Brian Everitt's Data Sets and Functions

The website http://biostatistics.iop.kcl.ac.uk/publications/everitt/ provides many data sets and useful functions which are referred to in these notes. There is a link to the website on the associated webpage. These can be downloaded individually or all together in a single zipped file. If you want to use some of these functions and data sets then download the zip file from the webpage and unpack it (or download the files separately).

The file containing all of the functions is an ordinary text file *functions.txt*, the data sets are stored as commands which can be read into S-PLUS or **R** with the `source` command (see below). Additionally script files are provided to reproduce all output in each chapter and these come as versions for **R** and S-PLUS.

To load the functions into your **R** or S-PLUS session you can either include them in the `.First` function or you can open the file in a text editor (e.g. Notepad or Word), select the entire contents, copy to the clipboard (with CTRL+C) and then paste to the command window (with CTRL+V). The functions will then be available for the entire session (or longer if you save it). Alternative you can keep them in an **R** script file with extension .R.

The data files have names with a prefix indicating the chapter in which they are described, e.g. *chap2airpoll.dat*; however they are not ordinary data files but need to be read into the **R** or session with the `source` command. First, you need to know the full pathname to the file which depends on which path you gave when downloading or unzipping the file. It can be found using Windows Explorer, navigating to the directory containing the file and noting the address in the navigator window at the top. For example, it might be (as on my laptop), `C:\Documents and Settings\Nick Fieller\My Documents\My Teaching\MVA\EverittRcompanion\Data` .

To read the file into the **R** dataset `airpoll` issue the command `airpoll<-source("pathname")$value`, where pathname is the full path as above but with double backslashes followed by the name of the file: `airpoll<-source("C:\\Documents and Settings\\Nick Fieller\\My Documents\\My Teaching\\MVA\\EverittRcompanion\\Data\\chap2airpoll.dat")`, on my particular machine. Incidentally, note that **R** uses a double backslash `\\` in pathnames for external files but S-PLUS uses only a single one `\`.

In S-PLUS the `source` command has a slightly different syntax and the instruction is `airpoll<-source("pathname")`.

An alternative method is to open the data file in a text editor, copy the contents to the clipboard and paste it into the command window or script window.

0.6 R libraries required

Most of the statistical analyses described in this book use functions within the `base and stats` packages and the `MASS` package. It is recommended that each **R** session should start with

```
library(MASS)
```

The `MASS` library is installed with the base system of **R** and the `stats` package is automatically loaded. Other packages which are referred to are `lattice, ICS, ICSNO, mvtnorm, CCA.`

0.7 Subject Matter

The course is concerned with analysing and interpreting multivariate data:

i.e. measurement of *p* variables on each of *n* subjects

e.g.

(i) body temperature, renal function, blood pressure, weight of 73 hypertensive subjects (*p=4, n=73*).

(ii) petal & sepal length & width of 150 flowers (*p=4, n=150*).

(iii) amounts of 9 trace elements in clay of Ancient Greek pottery fragments (*p=9*).

(iv) amounts of each of 18 amino acids in fingernails of 92 arthritic subjects (*p=18, n=92*).

(v) presence or absence of each of 286 decorative motifs on 148 bronze age vessels found in North Yorkshire (*p=286, n=148).*

(vi) grey shade levels of each of 1024 pixels in each of 15 digitized images (*p=1024, n=15*)

(vii) Analysis of responses to questionnaires in a survey

(*p = number of questions, n = number of respondents*)

(viii) Digitization of a spectrum (*p=10000, n=100 is typical)*

(ix) Activation levels of all genes on a genome

(*p=30000* genes, *n=10* microarrays is typical)

Notes

- Measurements can be *discrete* e.g. (v) & (vi), or *continuous*, e.g. (i)-(iv) or a *mixture* of both, e.g.(vii).
- Typically the variables are ***correlated*** but individual sets of observations are ***independent***.
- There may be more observations than variables (n > p), e.g. (i)-(iv), or they may be more variables than observations (n < p), e.g. (v) & (vi) and especially (viii) [where n << p] and (ix) [where n <<< p].

- Some multivariate techniques are **only** available when n>p (i.e. more observations than variables) e.g. discriminant analysis, formal testing of parametric hypotheses etc. Other techniques can be used even if n < p (e.g. Principal Component Analysis, Cluster Analysis).

  [*technical reason*: standard estimate of covariance matrix is singular if $n \leq p$ so techniques requiring inversion of this will fail when $n \leq p$]

0.8 Subject Matter / Some Multivariate Problems

(i) Obvious generalizations of univariate problems: t-tests, analysis of variance, regression, multivariate general linear model. e.g. model data Y' by $Y'=X\Theta + \varepsilon$,

where Y' is the $n\times p$ data matrix, X is an $n\times k$ matrix of known observations of k-dimensional regressor variables, Θ is $k\times p$ matrix of unknown parameters, ε is $n\times p$ with n values of p-dimensional error variables.

(ii) Reduction of dimensionality for

(a) exploratory analysis

(b) simplification (MVA is easier if p=1 or p=2)

(c) achieve greater statistical stability

(e.g. remove variables which are highly correlated)

Methods of principal component analysis, factor analysis, non-metric scaling....

(iii) Discrimination

Find rules for discriminating between groups, e.g. 2 known variants of a disease, data X', Y' on each. What linear combination of the p variables best discriminates between them. Useful to have diagnostic rules and this may also throw light onto the conditions (e.g. in amino acids and arthritis example there are two type of arthritis :— psoriatic and rheumatoid, determining which combinations of amino acids best distinguishes between them gives information on the biochemical differences between the two conditions).

(iv) Cluster Analysis/Classification

Do data arise from a homogeneous source or do they come from a variety of sources, e.g. does a medical condition have sub-variants.

(v) Canonical Correlation Analysis

Of use when looking at relationships between sets of variables, e.g. in particular in questionnaire analysis between response to 2 groups of questions, perhaps first group of questions might investigate respondents expectations and the second group their evaluation.

After an initial review of methods for displaying multivariate data graphically, this course will begin with topics from the second category (dimensionality reduction) before considering the estimation and testing problems which rest on distributional assumptions and which are straightforward generalizations of univariate statistical techniques.

> “Much classical and formal theoretical work in Multivariate Analysis rests on assumptions of underlying multivariate normality — resulting in techniques of very limited value”.
>
> (Gnanadesikan, page 2).

0.9 Basic Notation

We deal with observations $x = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_p \end{pmatrix}$, a column p-vector.

Transpose: a dash ′ denotes transpose: $x' = (x_1, x_2, \ldots, x_p)$

The ith observation is $x_i = \begin{pmatrix} x_{i1} \\ x_{i2} \\ \vdots \\ x_{ip} \end{pmatrix} = (x_{i1}, x_{i2}, \ldots, x_{ip})'$.

The n×p matrix $X' = \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1p} \\ x_{21} & x_{22} & \cdots & x_{2p} \\ \vdots & & \ddots & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{np} \end{pmatrix} = \begin{pmatrix} x_1' \\ x_2' \\ \vdots \\ x_n' \end{pmatrix}$

is the **data matrix.** Note that X is p×n and X′ is n×p.

Note that we do **not** specifically indicate vectors or matrices (by underlining or bold face or whatever). Additionally, whether x_i denotes the ith (vector) observation of x or the ith element of vector x is dependent on context (usually it is the ith observation since it is rare to need to refer to individual elements of vectors or matrices).

Define the sample mean vector

$\bar{x}' = (\bar{x}_1, \bar{x}_2, \ldots, \bar{x}_p) = \frac{1}{n}1'X'$, where 1 is the column vector of n 1s.

and the sample variance (or variance-covariance matrix)

$$\mathrm{var}(X') = S = \tfrac{1}{n-1}(X - \bar{X})(X - \bar{X})',$$

where $\bar{X} = (\bar{x}, \bar{x}, \ldots, \bar{x})$ is the p×n matrix with all columns equal to $\bar{x}$.

0.9.1 Notes

- S is a p×p matrix, the diagonal terms give the variances of the p variables and the off-diagonal terms give the covariances between the variables.

- S is (in general) non-singular and positive definite, provided all measured variables are ‘distinct’ (i.e. none is a linear combination of any of the others).

- S can also be written as

$$S = \tfrac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})(x_i - \bar{x})' = \tfrac{1}{n-1}\Big(\sum_{i=1}^{n} x_i x_i' - n\bar{x}\bar{x}'\Big)$$

- Also S=$S = \tfrac{1}{(n-1)}\left\{(1-\tfrac{1}{n})\sum_{i=1}^{n}(x_i - \mu)(x_i - \mu)' - \tfrac{1}{n}\sum_{i\neq j}\sum(x_i - \mu)(x_j - \mu)'\right\}$

- If w is any vector then var(X′w) = w′var(X′)w = w′Sw

- If A is any p×q matrix then var(X′A)=A′var(X′)A=A′SA

0.9.2 Proof of results quoted in §0.9.1

- $S = \frac{1}{n-1}(X-\bar{X})(X-\bar{X})' = \frac{1}{n-1}\sum_{i=1}^{n}(x_i-\bar{x})(x_i-\bar{x})' = \frac{1}{n-1}\sum_{i=1}^{n}x_ix_i' - n\overline{xx}'$

  The first step follows on noting that $X-\bar{X}$ is a matrix consisting of columns $b_i = x_i - \bar{x}$ and so B = $X-\bar{X}$ has columns (b_1,b_2,…,b_n) where each b_i is a column p-vector and $BB' = \sum_{i=1}^{n}b_ib_i'$. The second step follows directly from multiplying out the terms in the summation sign (keeping the ordering the same and taking the transpose ′ inside the brackets, i.e. noting (x–y)(w–z)′ =xw′–xz′–yw′+yz′) and noting that $\bar{x}$ is a constant and can be taken outside the summation and the sum of the individual x_i is $n\bar{x}$, and that summing $\overline{xx}'$ from 1 to n gives $n\overline{xx}'$.

- $S = \frac{1}{(n-1)}\left\{(1-\frac{1}{n})\sum_{i=1}^{n}(x_i-\mu)(x_i-\mu)' - \frac{1}{n}\sum\sum_{i\neq j}(x_i-\mu)(x_j-\mu)'\right\}.$

  It is easiest to work backwards from the target expression: We have

$$\left\{(1-\tfrac{1}{n})\sum_{i=1}^{n}(x_i-\mu)(x_i-\mu)' - \tfrac{1}{n}\sum\sum_{i\neq j}(x_i-\mu)(x_j-\mu)'\right\}$$

$$= \sum_{i=1}^{n}(x_i-\mu)(x_i-\mu)' - \tfrac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{n}(x_i-\mu)(x_j-\mu)'$$

$$= \sum_{i=1}^{n}x_ix_i{}' - 2n\bar{x}\mu' + n\mu\mu' - \tfrac{1}{n}\sum_{i=1}^{n}(x_i-\mu)\sum_{j=1}^{n}(x_j-\mu)'$$

$$= \sum_{i=1}^{n}x_ix_i{}' - 2n\bar{x}\mu' + n\mu\mu' - \tfrac{1}{n}n^2(\bar{x}-\mu)(\bar{x}-\mu)' = \sum_{i=1}^{n}x_ix_i' - n\overline{xx}' = (n-1)S$$

  This result is included here because it provides a quick proof (see §8.1.4) that the sample variance of independent observations is [always] an unbiased estimate for the population variance. It is a direct generalization of the standard proof of this result for univariate data but it is very far from ‘obvious’, especially if attempting to work from the left hand side of the identity rather than simplifying the right hand side.

- If w is any p-vector then $\mathrm{var}(X'w) = w'\mathrm{var}(X')w = w'Sw$,

  This actually follows directly from the expression for var(Y) putting $y_i=w'x_i$ etc.

$$\mathrm{var}(Y') = \tfrac{1}{n-1}\sum_{i=1}^{n}(y_i - \overline{Y})(y_i - \overline{Y})' = \tfrac{1}{n-1}\sum_{i=1}^{n}(w'x_i - w'\overline{X})(w'x_i - w'\overline{X})'$$

$$= \tfrac{1}{n-1}\sum_{i=1}^{n}w'(x_i - \overline{x})(x_i - \overline{x})'w = w'Sw\,,$$

(noting $(w'x_i - w'\overline{X})' = (x_i{}'w - \overline{X}'w) = (x_i{}' - \overline{X}')w = (x_i - \overline{X})'w$ etc.).

0.10 R Implementation

The dataset `airpoll` referred to above consists of 7 measurements relating to demographic and environmental variables on each of 60 US towns, i.e. n=60, p=7. To calculate the mean of the seven variables use the function `apply(.)`, (see `help(apply)`) . First read in the dataset as described in §0.5,

```
> apply(airpoll,2,mean)
 Rainfall Education Popden Nonwhite   NOX    SO2 Mortality
   37.367    10.973 3866.1    11.87 22.65 53.767    940.38
>
```

To calculate the variance matrix use the function `var()`

```
> var(airpoll)
           Rainfall  Education      Popden  Nonwhite
 Rainfall   99.6938   -4.13921     -128.66   36.8061
Education   -4.1392    0.71453     -290.89   -1.5744
   Popden -128.6627 -290.88847  2144699.44 -166.5103
 Nonwhite   36.8061   -1.57437     -166.51   79.5869
      NOX -225.4458    8.78881    11225.58    7.5995
      SO2  -67.6757  -12.55718    39581.66   90.0827
Mortality  316.4340  -26.83677    23813.64  357.1744

                NOX        SO2  Mortality
 Rainfall  -225.4458    -67.676    316.434
Education     8.7888    -12.557    -26.837
   Popden 11225.5771  39581.656  23813.642
 Nonwhite     7.5995     90.083    357.174
      NOX  2146.7737   1202.425   -223.454
      SO2  1202.4254   4018.351   1679.945
Mortality  -223.4540   1679.945   3870.386
>
```

Thus, for example the variance of the amount of rainfall is 99.69 inches^2, that of SO2 measurements is 4018.35, the covariance between Popden & education is –290.89 and that between NOX and Mortality is –223.45.

Note that both the mean and the variance can be assigned to variables, a vector in the case of the mean and a matrix in the case of variance.

```
> airmean<- apply(airpoll,2,mean)
> airmean
 Rainfall Education Popden Nonwhite   NOX    SO2 Mortality
   37.367    10.973 3866.1    11.87 22.65 53.767    940.38
> airvar<- var(airpoll)
> airvar
           Rainfall  Education      Popden  Nonwhite
 Rainfall   99.6938   -4.13921     -128.66   36.8061
Education   -4.1392    0.71453     -290.89   -1.5744
   Popden -128.6627 -290.88847  2144699.44 -166.5103
 Nonwhite   36.8061   -1.57437     -166.51   79.5869
      NOX -225.4458    8.78881    11225.58    7.5995
      SO2  -67.6757  -12.55718    39581.66   90.0827
Mortality  316.4340  -26.83677    23813.64  357.1744

                  NOX        SO2  Mortality
 Rainfall   -225.4458    -67.676    316.434
Education      8.7888    -12.557    -26.837
   Popden  11225.5771  39581.656  23813.642
 Nonwhite      7.5995     90.083    357.174
      NOX   2146.7737   1202.425   -223.454
      SO2   1202.4254   4018.351   1679.945
Mortality   -223.4540   1679.945   3870.386
>
```

Further example: The version of the Anderson’s Iris Data available from the course webpage is a dataframe with five columns named `irisnf` (the system data set `iris` is a different version of the same data). The data give the lengths and widths of petals and sepals of iris flowers of 3 varieties. The fifth column is a factor indicating the varieties and the first four are named `Sepal.l, Sepal.w, Petal.l, Petal.w.`

Since the dataframe contains one variable which is a factor the mean and variance commands above cannot be applied top the whole dataframe. Instead individual variables must be selected and named in the commands.

Assuming that the dataset has been downloaded and read into **R**, e.g. by double clicking on it in Windows Explorer or running directly from the webpage (this latter option is not recommended), the summary statistics of the lengths of the sepals and petals can be obtained by:

```
> attach(irisnf)
> var(cbind(Sepal.l, Petal.l))
        Sepal.l Petal.l
Sepal.l 0.68569  1.2743
Petal.l 1.27432  3.1163
> apply(cbind(Sepal.l,Petal.l),2,mean)
 Sepal.l Petal.l
  5.8433   3.758
```

Note the use of `attach()` and `cbind()`,(try `help(attach)` and `help(cbind)` to find out more).

Notes

- The commands `apply()` and `var()`are also available in S-PLUS. The command `mean()` operates slightly differently in **R** and S-PLUS: in S-PLUS it produces a scalar which is the overall mean of all values (which is rarely useful); in **R** it produces a vector if the argument is a dataframe but an overall mean of all values if the argument is a matrix (as produced by `cbind()`) or an array.
- Summary statistics of individual variables can be obtained in S-PLUS by using the menus (`Statistics>Data Summaries>`) but this is not recommended since firstly doing so treats each variable separately (i.e. lots of univariate observations on the same cases) instead of regarding the data as a multivariate data set and each observation as a vector measurement on each case. Secondly, the menus do not allow the mean p-vector and the variance p×p matrix to be stored as variables for later calculations.

0.10.1 Illustration in R that var(X′A)=A′var(X′)A

The data frame `scor` found in `openclosed.Rdata` available from the course webpage gives the examination marks of 88 students who took five examinations under either open or closed book examinations in mathematics at the University of Hull in c.1970 and are given in Mardia, Kent & Bibby (1981). The five examinations were in Mechanics, Vectors, Algebra, Analysis and Statistics. The first two of these were taken under closed book conditions and the last three under open book conditions. The five variables in the data set are labelled `mec, vec, alg, ana` and `sta.`

Below is a transcript from an **R** session with comments which illustrates the result var(X′A)=A′var(X′)A as well as basic matrix manipulation in **R**.

The session is started by locating the file `openclosed.Rdata` on your hard disk in Windows Explorer (after downloading it to a suitable directory) and double clicking on it. This starts **R**, changes the working directory to that where the file `openclosed.Rdata` is located and loads the data. We will take A as the 5×2 matrix

$$A = \begin{pmatrix} 1 & \tfrac{1}{2} \\ 1 & \tfrac{1}{2} \\ 1 & -\tfrac{1}{3} \\ 1 & -\tfrac{1}{3} \\ 1 & -\tfrac{1}{3} \end{pmatrix}$$

```
> ls()  # list all objects in the work space
[1] "scor"
>
> scor[1:5,] # print the first few lines of the data set
  mec vec alg ana sta
1  77  82  67  67  81
2  63  78  80  70  81
3  75  73  71  66  81
4  55  72  63  70  68
5  63  63  65  70  63
>
>
> dim(scor) #  find out how many rows and columns scor has
[1] 88  5
>
> # want X' to be the data matrix scor so
> # define X to be the matrix of the transpose of scor
>
> X<-as.matrix(t(scor))
>
> dim(X)
[1]  5 88
>
> var(t(X)) # find the variance matrix of X'=scor
    mec   vec   alg   ana sta
mec 306 127.2 101.6 106.3 117
vec 127 172.8  85.2  94.7  99
alg 102  85.2 112.9 112.1 122
ana 106  94.7 112.1 220.4 156
sta 117  99.0 121.9 155.5 298
>
> A<-matrix(c(1,1/2,1,1/2,1,-1/3,1,-1/3,1,-1/3),5,2,byrow=T)
# enter the matrix A
>
> A  # check it is correct
     [,1]   [,2]
[1,]    1  0.500
[2,]    1  0.500
[3,]    1 -0.333
[4,]    1 -0.333
[5,]    1 -0.333
>
```

```
> Y<-t(A)%*%X # let Y=A'X so that Y'=X'A
> t(Y)[1:5,] # and print first few lines of t(Y)
  [,1]  [,2]
1  374  7.83
2  372 -6.50
3  366  1.33
4  328 -3.50
5  324 -3.00
>
> # note that Y’ gives the total score of each candidate
# in column 1 and the difference in mean scores
# on closed and open book exams in column 2
#
> scor[1:5,] # print the first few lines of the data set
  mec vec alg ana sta
1  77  82  67  67  81
2  63  78  80  70  81
3  75  73  71  66  81
4  55  72  63  70  68
5  63  63  65  70  63
>
>
> var(t(X)%*%A)  # calculate var(X'A)
        [,1]   [,2]
[1,] 3351.31  -2.81
[2,]   -2.81 138.57
>
> t(A)%*%var(t(X))%*%A  # calculate Avar(X'A)A'
        [,1]   [,2]
[1,] 3351.31  -2.81
[2,]   -2.81 138.57
>
```

Tasks 1

(see §0.0–§1.5 & A0.1)

1) Read the Study Guide for this course if you have not already done so.
2) Verify the final result referred to in Chapter 0, §0.9.1 Notes that if A is any p×q matrix then var(X′A)=A′var(X′)A=A′SA.
3) Access the Iris Dataset which is stored as an **R** data set `irisnf.Rdata`
   - i) Find the 4-vector which is the mean of the four dimensions `Sepal.l, Sepal.w, Petal.l, Petal.w` and the 4×4 matrix which is their variance (see 'Further example' in §0.10).
   - ii) Plot sepal length against sepal width using:
     - a) the default choices
     - b) using different symbols for each variety (with `pch=` and `col=`)
   - iii) Construct a matrix plot of all four dimensions, using first the default choices and then enhancing the display as above.
   - iv) Try the commands

```
var(irisnf)
diag(var(irisnf))
```

4) Try these simple exercises both 'by hand' and using **R**:

Let $a=\begin{pmatrix}1\\2\\3\end{pmatrix}$, $A=\begin{pmatrix}1&2&3\\4&5&6\end{pmatrix}$, $B=\begin{pmatrix}1&2\\3&4\\5&6\end{pmatrix}$,

Find AB, B′A′, BA, a′A, a′Aa

5) Read through the sections on eigenvalues and eigenvectors, differentiation w.r.t. vectors and use of Lagrange Multipliers in Appendix 0: Background Results in course booklet. This material will be used in Chapter 2. At least some of the material will be familiar to almost all of you but probably the vector differentiation will be novel: the only cases that are required in this course are those listed in that section. The important point to appreciate in the eigenanalysis section is that if we are trying to determine some vector x and we can shew that this vector must satisfy an equation of the form $Sx = \lambda x$ (S a matrix and λ a scalar) then we have essentially solved the problem and x is determined as one of the eigenvectors of S. This is equivalent to establishing that an unknown scalar x satisfies the equation $ax^2+bx+c=0$ means that x must be one of the roots of the quadratic. In **R** we can find roots of polynomials by the function `polyroot()` and similarly we can solve an eigenvalue problem with the function `eigen()`. Try `help(polyroot)` and `help(eigen)`.

6) Read the Study Guide for this course [again] if you have not already done so [or have done so only once].

1 Graphical Displays

1.1 1 Dimension

- Small # points – 1-dimensional scatter plots, dot plots.

- Large # points – stem & leaf plots, histograms, box plots.
  + other special techniques dependent on particular form of data, e.g. circular data need circular histograms.

Examples:

(i) Karl Pearson's data:

1230	1318	1380	1420	**1630**	1378
1348	1380	1470	1445	1360	1410
1540	1260	1364	1410	1545	

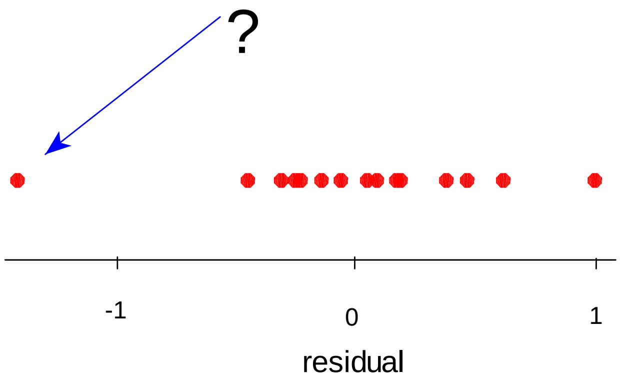
capacities of male Moriori skulls

(ii) Lt. Herndon's Venus Data

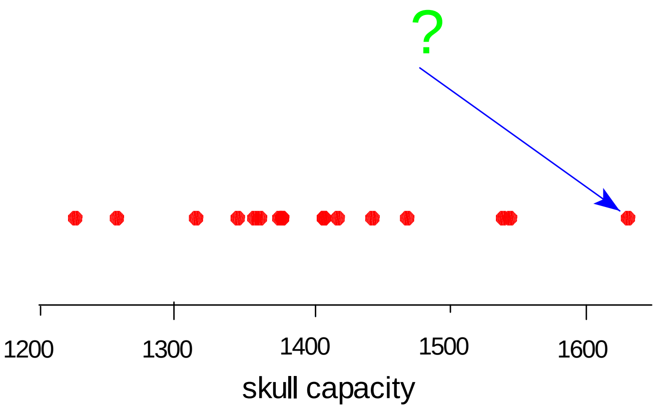
–.30	+0.48	+0.63	–0.22	+0.18
–0.44	–0.24	–0.13	-0.15	+0.39
+1.01	+0.06	**–1.40**	+0.20	+0.10

Semi-diameters of Venus (deviations from mean)

Both the above sets of data have suspiciously large values: A simple plot allows assessment of how much these deviate from the rest of the data, taking into account the internal scatter of the values.

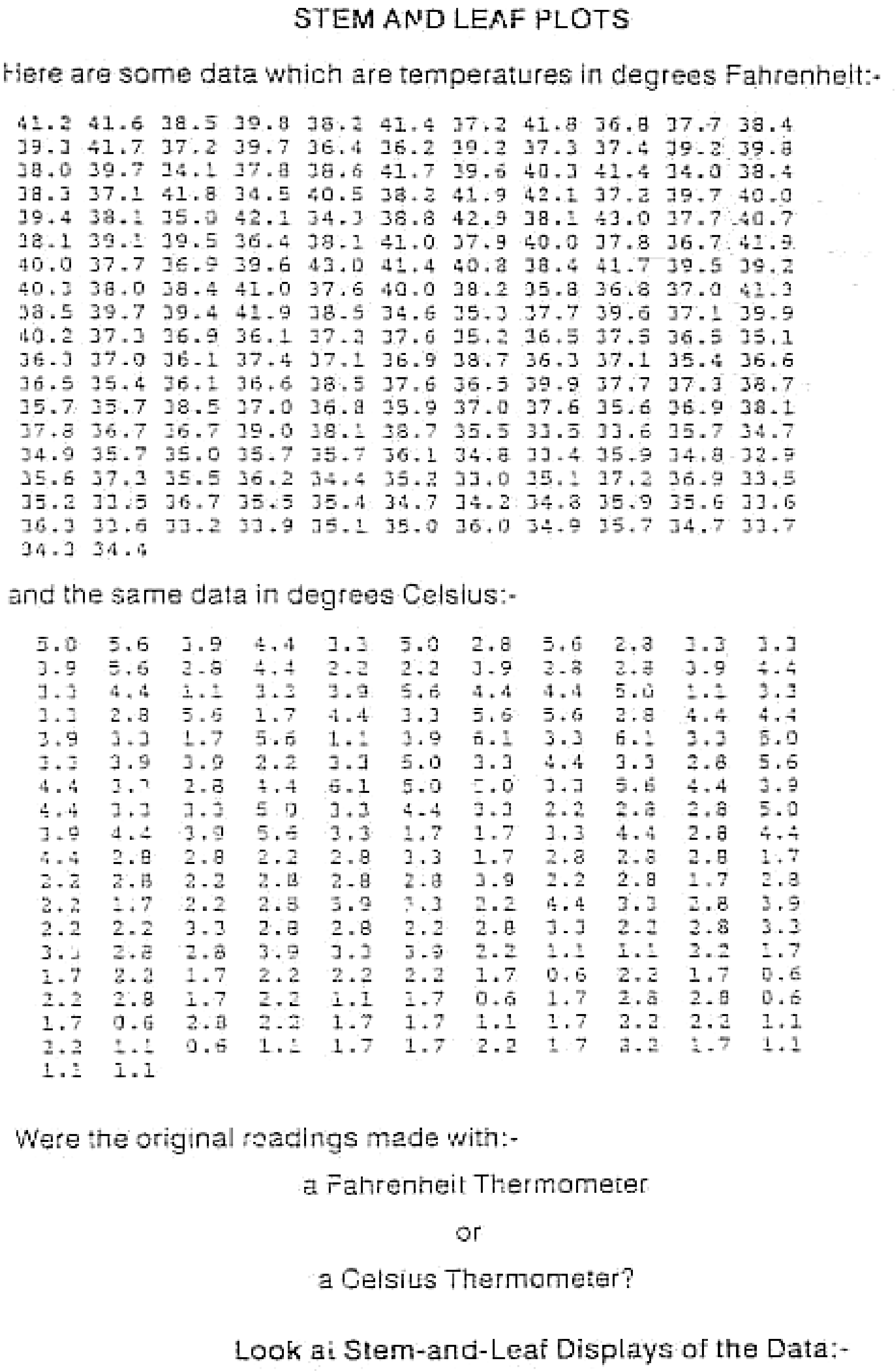


one dimensional scatterplot



one dimensional scatterplot

Examples of stem& leaf plots: temperature data



STEM AND LEAF PLOTS

Here are some data which are temperatures in degrees Fahrenheit:-

41.2	41.6	38.5	39.8	38.2	41.4	37.2	41.8	36.8	37.7	38.4
39.3	41.7	37.2	39.7	36.4	36.2	19.2	37.3	37.4	39.2	39.8
38.0	39.7	34.1	37.8	38.6	41.7	39.6	40.3	41.4	34.0	38.4
38.3	37.1	41.8	34.5	40.5	38.2	41.9	42.1	37.2	39.7	40.0
39.4	38.1	35.0	42.1	34.3	38.8	42.9	38.1	43.0	37.7	40.7
38.1	39.1	39.5	36.4	38.1	41.0	37.9	40.0	37.8	36.7	41.9
40.0	37.7	36.9	39.6	43.0	41.4	40.8	38.4	41.7	39.5	39.2
40.3	38.0	38.4	41.0	37.6	40.0	38.2	35.8	36.8	37.0	41.3
38.5	39.7	39.4	41.9	38.5	34.6	35.3	37.7	39.6	37.1	39.9
40.2	37.3	36.9	36.1	37.3	37.6	35.2	36.5	37.5	36.5	35.1
36.3	37.0	36.1	37.4	37.1	36.9	38.7	36.3	37.1	35.4	36.6
36.5	35.4	36.1	36.6	38.5	37.6	36.5	39.9	37.7	37.3	38.7
35.7	35.7	38.5	37.0	36.8	35.9	37.0	37.6	35.6	36.9	38.1
37.8	36.7	36.7	39.0	38.1	38.7	35.5	33.5	33.6	35.7	34.7
34.9	35.7	35.0	35.7	35.7	36.1	34.8	33.4	35.9	34.8	32.9
35.6	37.3	35.5	36.2	34.4	35.2	33.0	35.1	37.2	36.9	33.5
35.2	33.5	36.7	35.5	35.4	34.7	34.2	34.8	35.9	35.6	33.6
36.3	33.6	33.2	33.9	35.1	35.0	36.0	34.9	35.7	34.7	33.7
34.3	34.4									

and the same data in degrees Celsius:-

5.0	5.6	3.9	4.4	3.3	5.0	2.8	5.6	2.8	3.3	3.3
3.9	5.6	2.8	4.4	2.2	2.2	3.9	2.8	2.8	3.9	4.4
3.3	4.4	1.1	3.3	3.9	5.6	4.4	4.4	5.0	1.1	3.3
3.3	2.8	5.6	1.7	4.4	3.3	5.6	5.6	2.8	4.4	4.4
3.9	3.3	1.7	5.6	1.1	3.9	6.1	3.3	6.1	3.3	5.0
3.3	3.9	3.9	2.2	3.3	5.0	3.3	4.4	3.3	2.8	5.6
4.4	3.3	2.8	4.4	6.1	5.0	5.0	3.3	5.6	4.4	3.9
4.4	3.3	3.3	5.0	3.3	4.4	3.3	2.2	2.8	2.8	5.0
3.9	4.4	3.9	5.6	3.3	1.7	1.7	3.3	4.4	2.8	4.4
4.4	2.8	2.8	2.2	2.8	3.3	1.7	2.8	2.8	2.8	1.7
2.2	2.8	2.2	2.8	2.8	2.8	3.9	2.2	2.8	1.7	2.8
2.2	1.7	2.2	2.8	3.9	3.3	2.2	4.4	3.3	2.8	3.9
2.2	2.2	3.3	2.8	2.8	2.2	2.8	3.3	2.2	2.8	3.3
3.3	2.8	2.8	3.9	3.3	3.9	2.2	1.1	1.1	2.2	1.7
1.7	2.2	1.7	2.2	2.2	2.2	1.7	0.6	2.2	1.7	0.6
2.2	2.8	1.7	2.2	1.1	1.7	0.6	1.7	2.8	2.8	0.6
1.7	0.6	2.8	2.2	1.7	1.7	1.1	1.7	2.2	2.2	1.1
2.2	1.1	0.6	1.1	1.7	1.7	2.2	1.7	2.2	1.7	1.1
1.1	1.1									

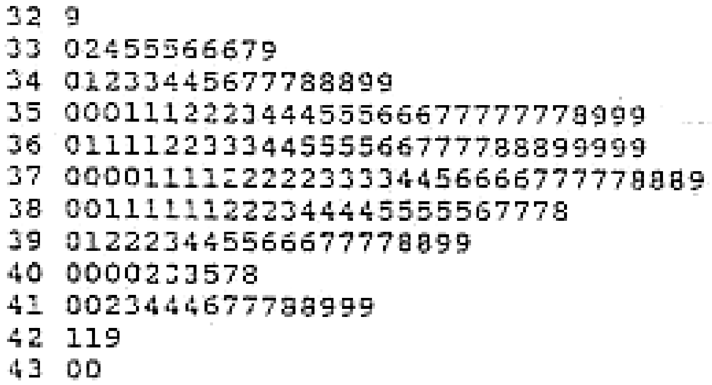
Were the original readings made with:-

a Fahrenheit Thermometer

or

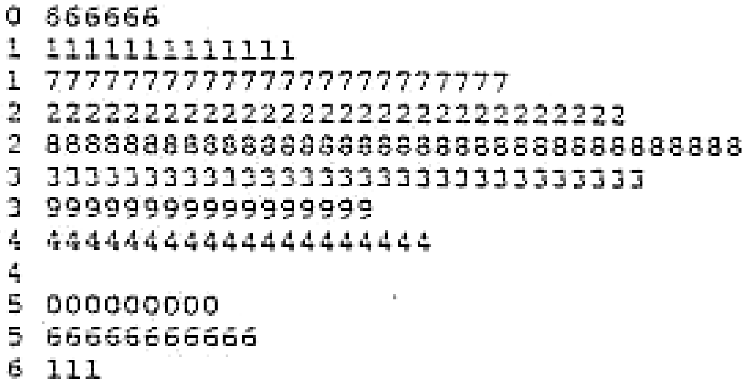
a Celsius Thermometer?

Look at Stem-and-Leaf Displays of the Data:-



```
32 9
33 02455566679
34 01233445677788899
35 000111222344455566677777778999
36 0111122333445555667777788899999
37 0000111122222333344566666777778889
38 001111111222344445555567778
39 012223445566677778899
40 0000233578
41 0023444677788999
42 119
43 00
```

Stem & leaf plot of Fahrenheit values

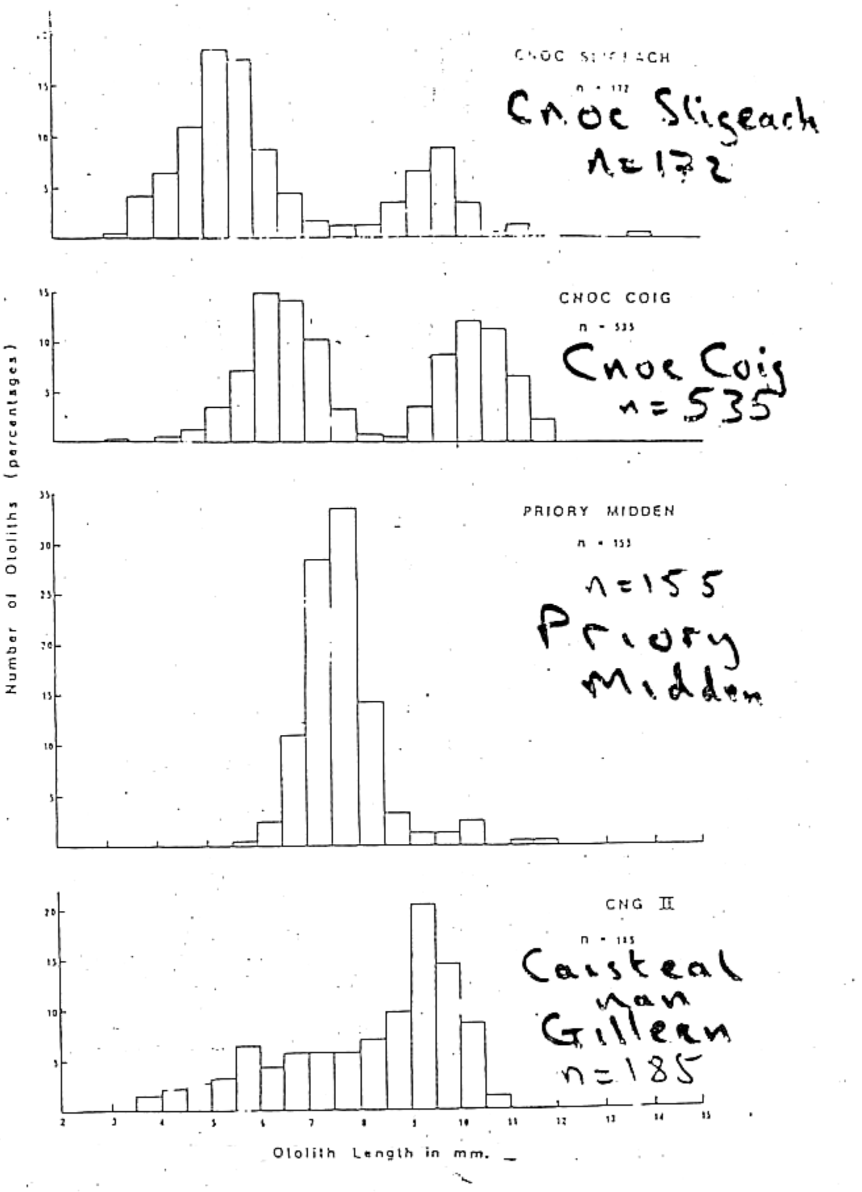


```
0 666666
1 1111111111111
1 7777777777777777777777777
2 22222222222222222222222222222
2 8888888888888888888888888888888888
3 333333333333333333333333333333
3 9999999999999999
4 44444444444444444444
4
5 000000000
5 66666666666
6 111
```

Stem & leaf plot of Celsius values.

Plots suggest that original measurements were in Fahrenheit and then converted into Celsius (and rounded to one decimal).

Examples of histograms: Data are lengths of otoliths (fossilised fishbones) found at four archaeological sites in Oronsay (Inner Hebrides), together with samples taken from contemporary fish caught on known dates. The fishbones grow larger with age and the pictures suggest that the four archaeological sites had bones from fish caught on different dates.



Archaeological samples

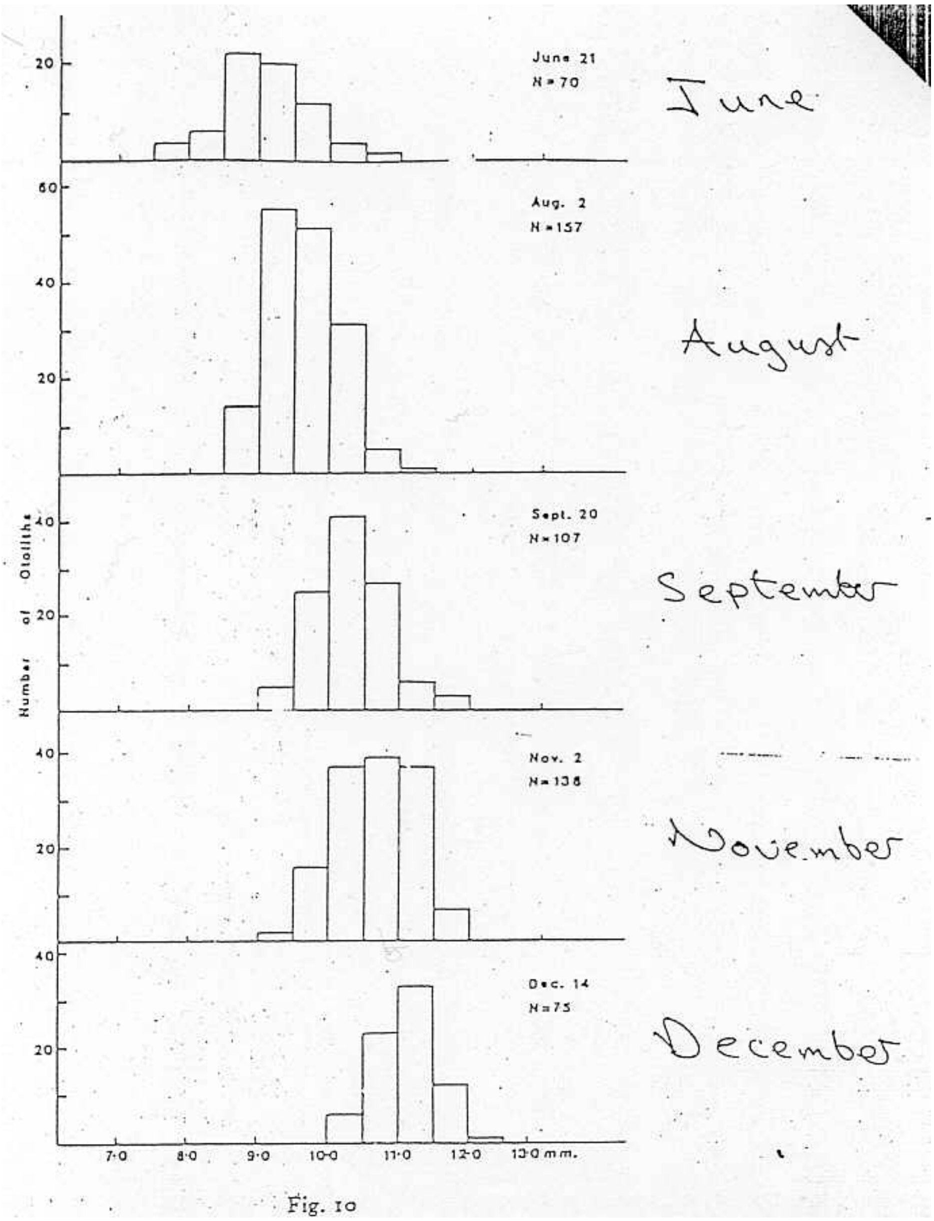
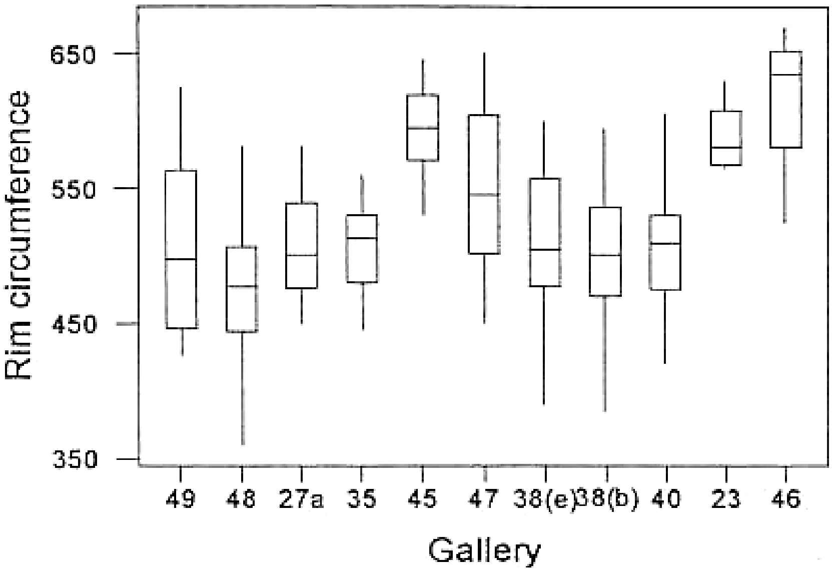


Fig. 10

Contemporary samples

Comparing a small number of histograms is possible. The histograms reveal the bimodality of the first two archaeological samples and the consistent increase in size of the contemporary ones. With a larger number of samples (e.g. too large for a separate histogram of each to fit on a page) boxplots are preferable.

Example of Boxplots: Data are rim-circumferences of mummy-pots (containing mummified birds) found at various different galleries in the Sacred Animal Necropolis in Saqqara, Egypt. The boxplots illustrate marked variation in sizes of pots between galleries.



Note that boxplots may conceal features such as bimodality — do some galleries contain a mixture of big and little pots? Those with exceptionally large boxes (e.g. galleries 47 & 49) indicate large variances which might be a reflection of bimodality.

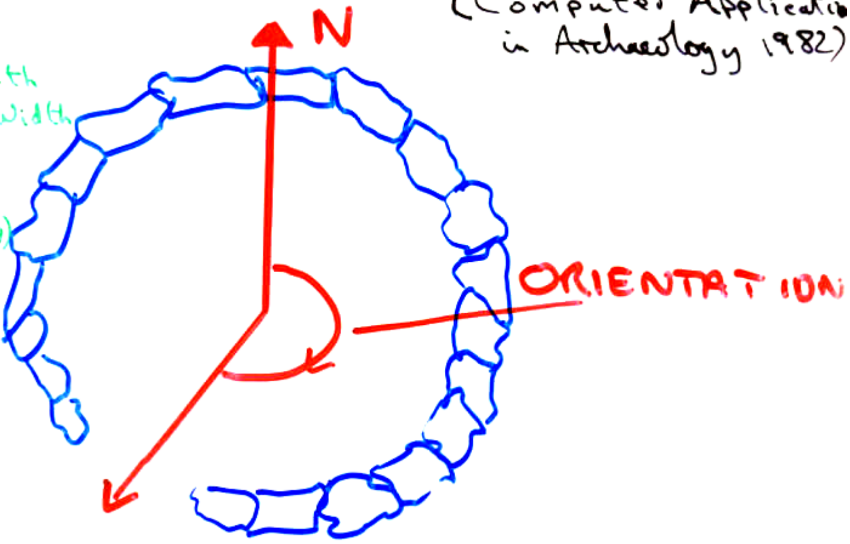
Example of circular data: data are orientations (i.e. compass directions) of the doorways of Bronze Age houses in the upper Plym Valley, Dartmoor. There are two groups of houses and the question of interest was whether the two groups were equally consistently orientated. Ordinary histograms can be misleading but *circular dotplots & histograms* capture the features of the data.

3) Orientations of Bronze Age Houses

3

(Computer Applicatio in Archaeology 1982)

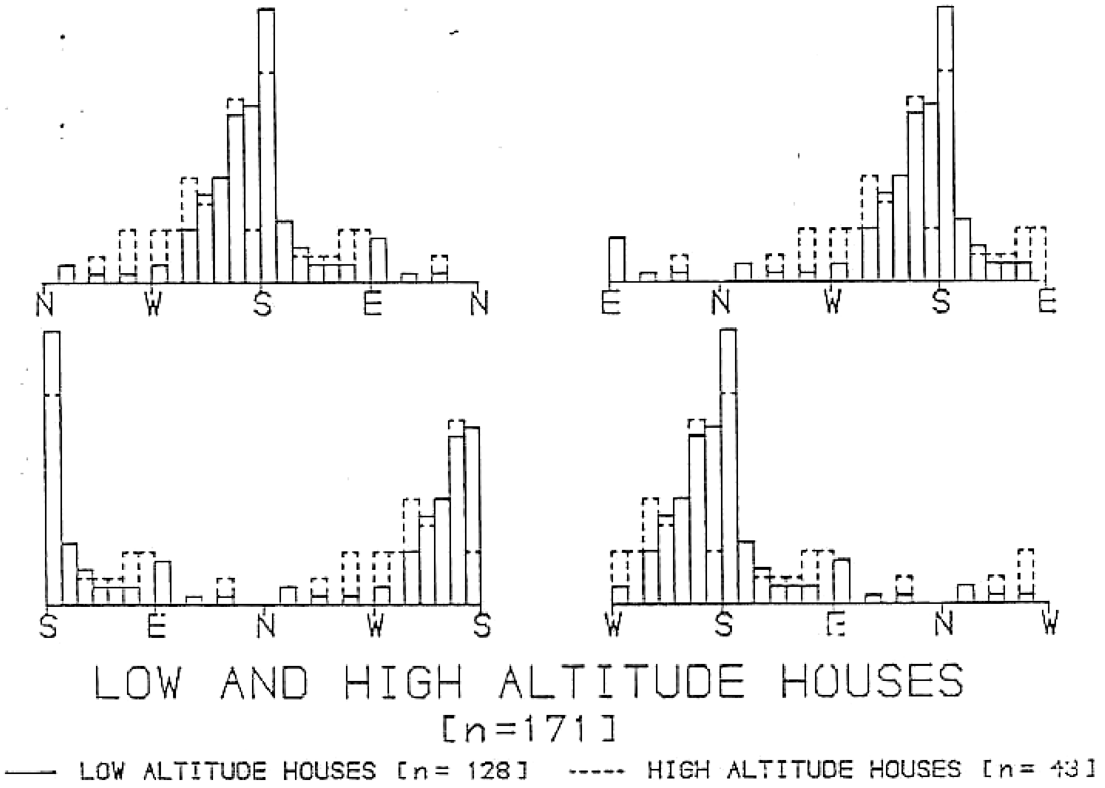
+area
Wall width
Entrance Width
Shape
Aspect
altitude (P/A)
etc
etc



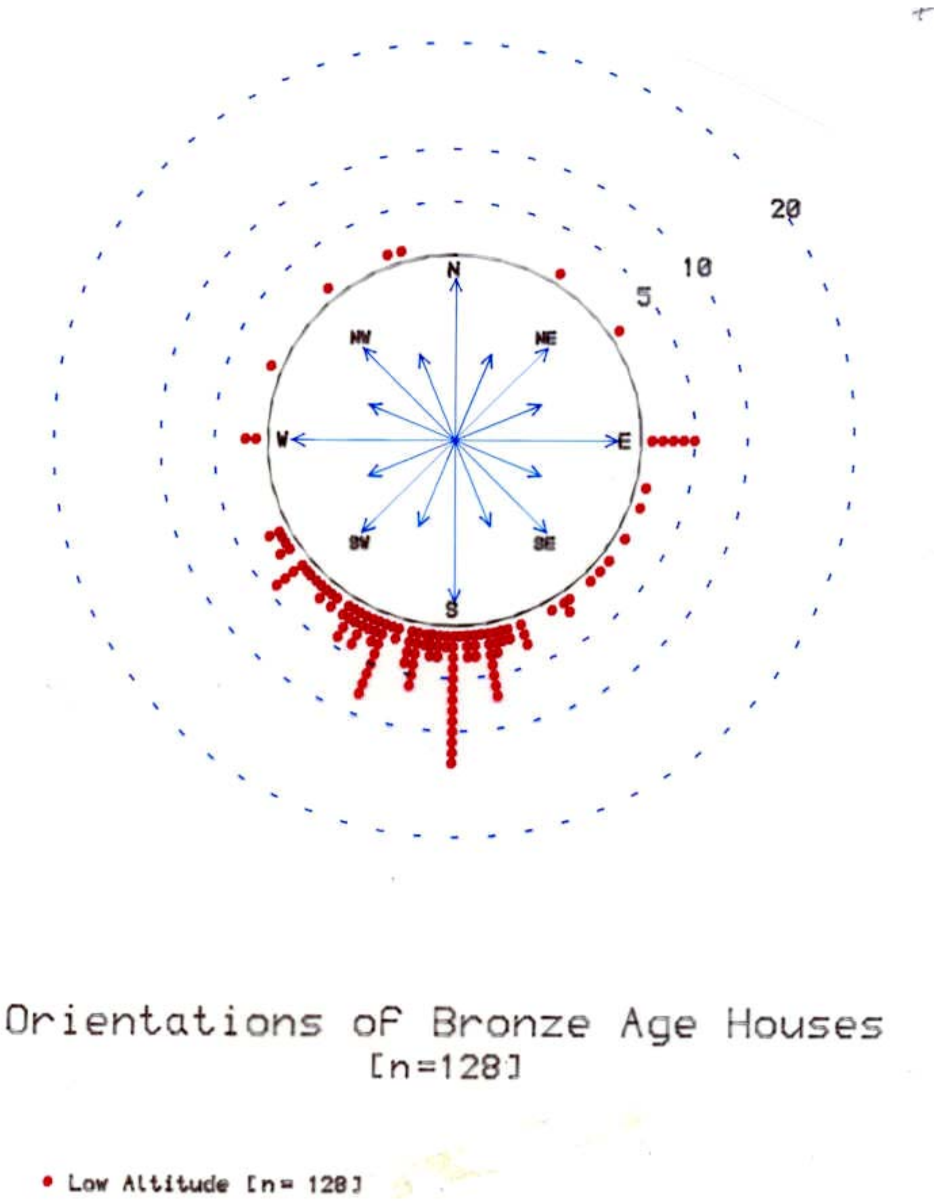
171 houses in upper Plym valley

2 Groups :— "Upper" + "Lower"

is one group more precisely orientated than the other?

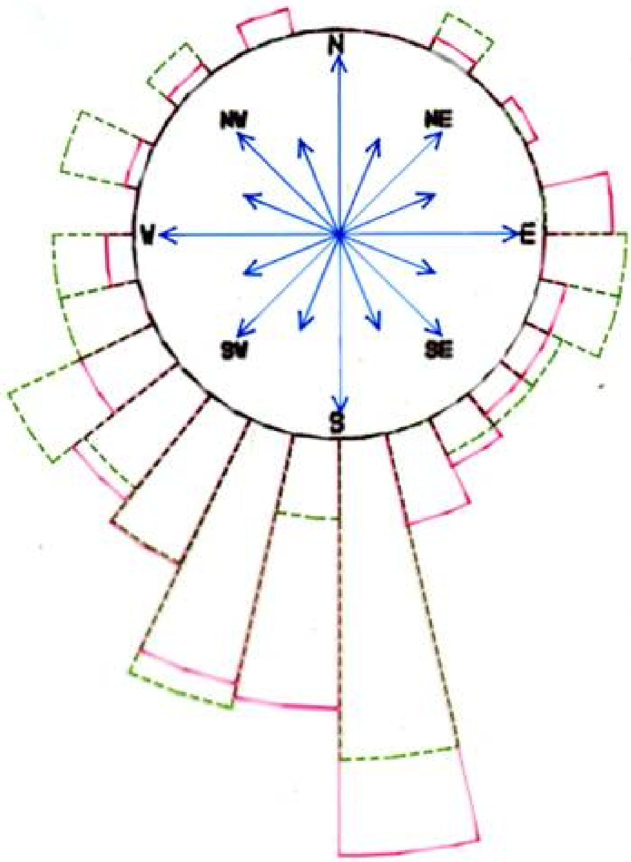


Four possible histograms of the angles as measured from North, East, South and West. Different (erroneous) conclusions might be drawn depending on which base direction is chosen. Better is to preserve the circularity of the data in the display:



A circular dotplot

This shews clearly that the majority of are orientated to the South-West, with a few outliers in other directions.



A circular histogram

The circular histograms have area proportional to ***relative frequency***, this allows comparison of distributions of the two samples even though the sample sizes are different (just as with ordinary histograms).

1.2 2 Dimensions

- Small # points – scatter plots
- Large # points

  – bivariate histograms drawn in perspective

  – bivariate boxplots

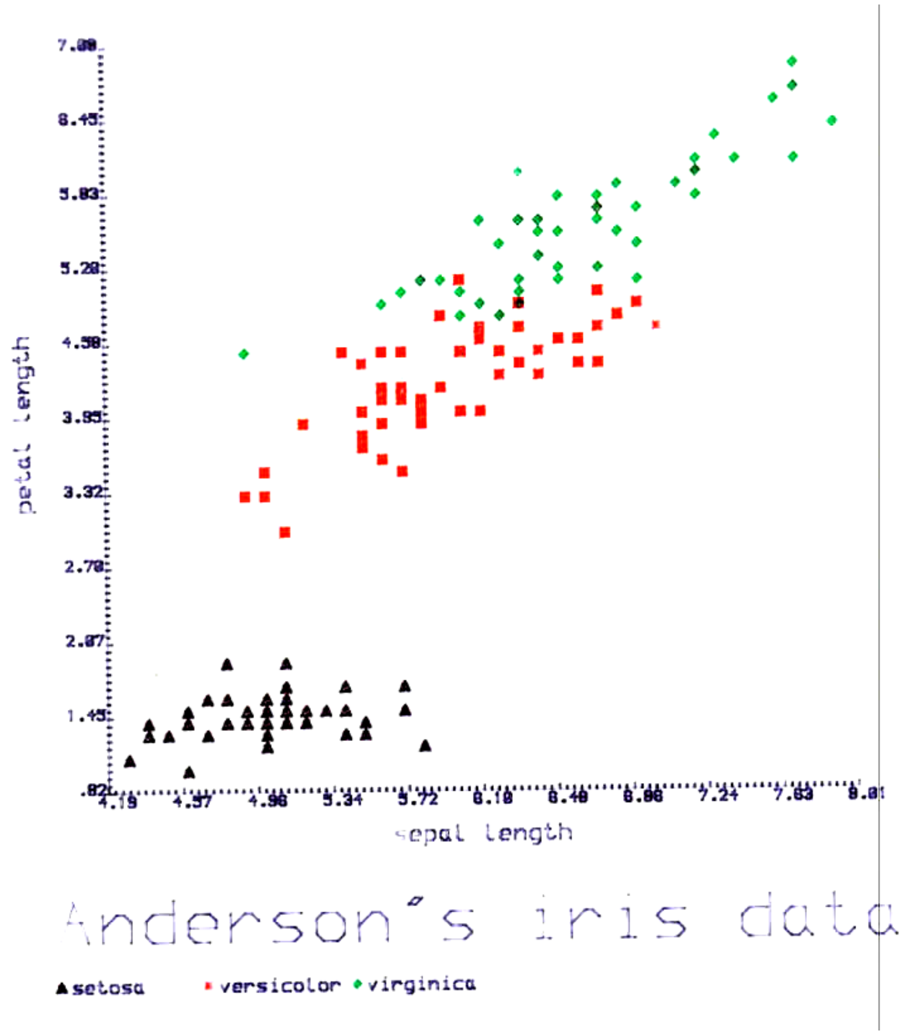
  – 2-dim frequency plots

  – augmented scatter plots

1.2.1 Examples:

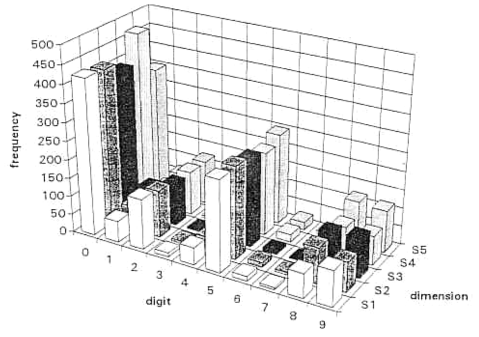
Anderson’s iris data: measurements of sepal and petal length and breadth of 50 of each of 3 varieties of iris.

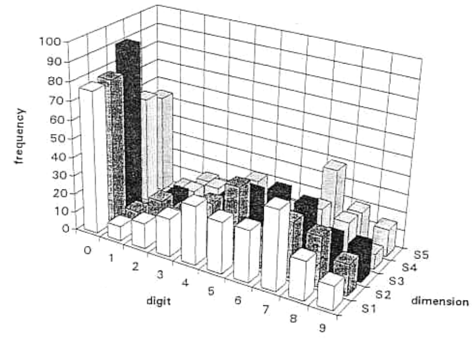
Two-dimensional scatter plot of two components.

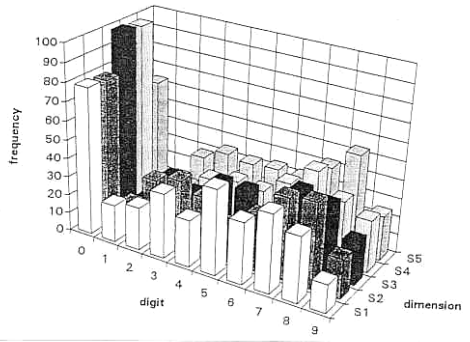


Example of bivariate histograms: Digital fingerprinting data.

Data are final digits in measurements of the same set of dogwhelks by two people, the second of whom measured them twice.







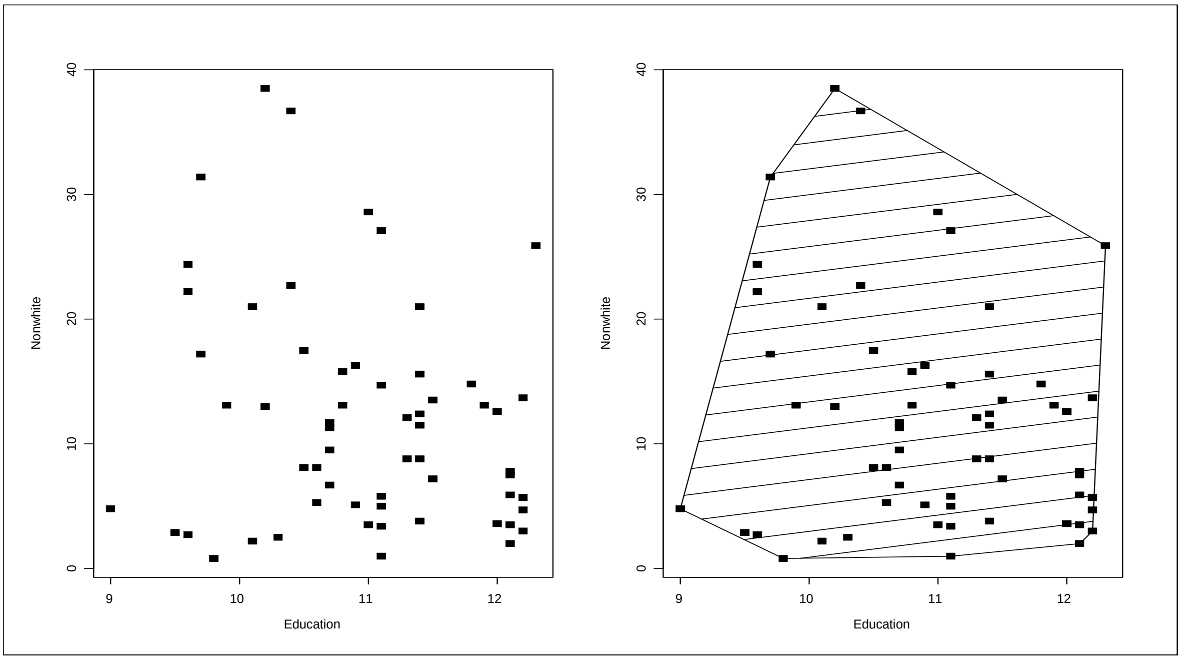
The second two histograms are very similar but quite distinct from the first: this indicates that the two measurers had different subconscious preferences for the final digit to be recorded.

1.2.2 Convex Hulls and Bivariate Boxplots

Everitt provides a function `chull()` which determines which points form the convex hull in a bivariate scatterplot, i.e. those points which form a convex polygon encircling all the points in the two-dimensional scatterplot. For example (using the dataset `airpoll`) :

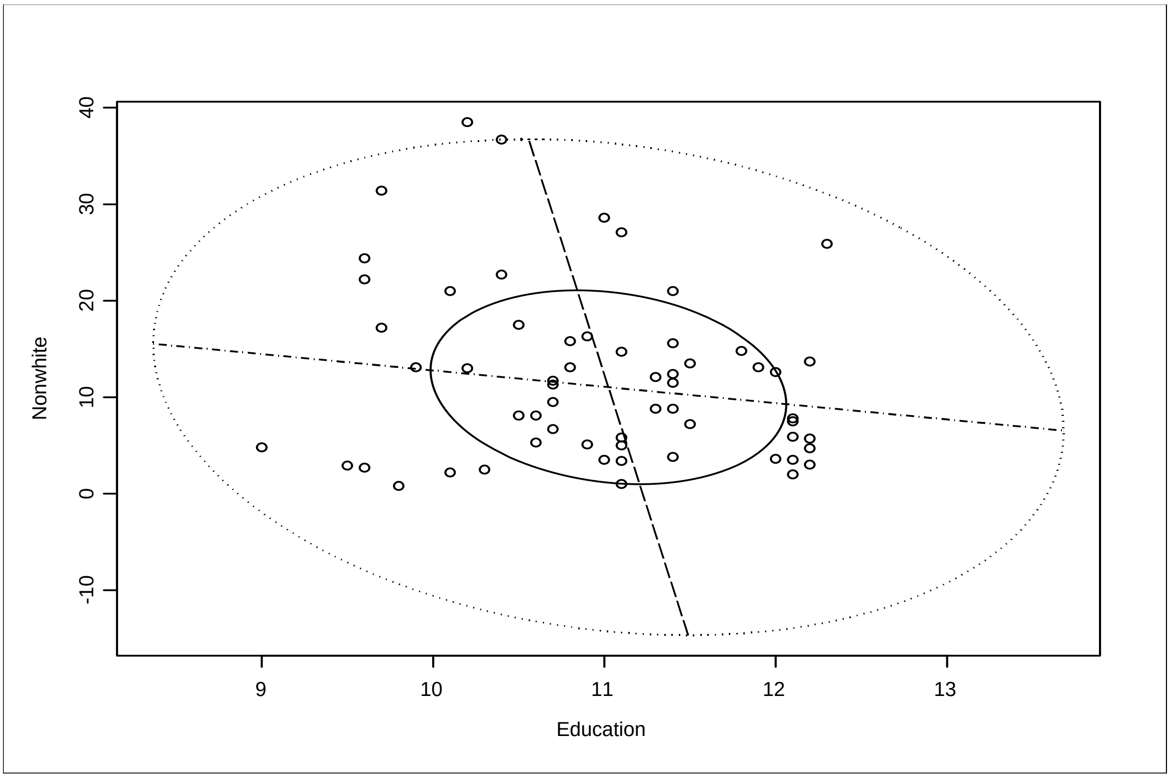
```
> plot(Education,Nonwhite,pch=15)
> plot(Education,Nonwhite,pch=15)
> hull<-chull(Education,Nonwhite)
> polygon(Education[hull],Nonwhite[hull],density=5,angle=10)
>
```

produces:



A further function `bvbox()` (beware of the misspelling in the book) produces a 'bivariate boxplot which is a pair of concentric ellipses; the inner (the "hinge") contains the innermost 50% of the points and the outer (the "fence") excludes outliers:

```
> bvbox(cbind(Education,Nonwhite),xlab="Education",
+ ylab="Nonwhite")
>
```

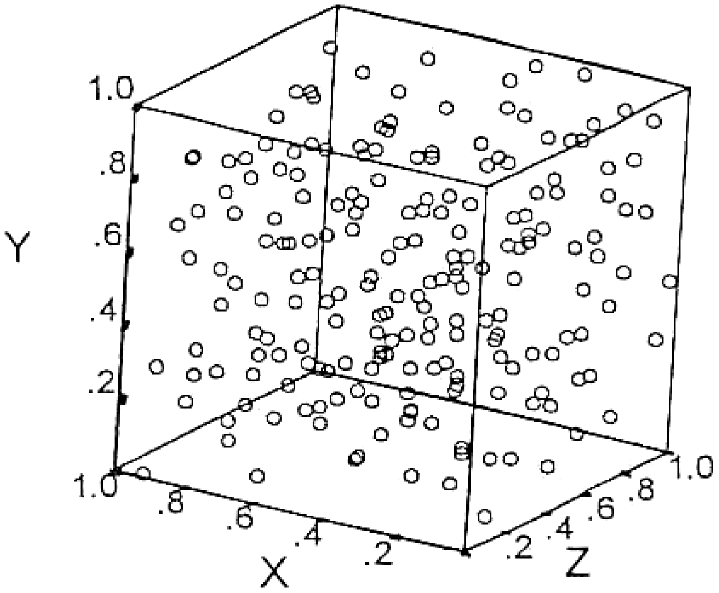


The version provided on the website is perhaps not satisfactory for all purposes but with experience of **R** or S-PLUS it is possible to alter the actual commands in the function to particular needs, e.g. to thicken and darken the lines and alter the symbols.

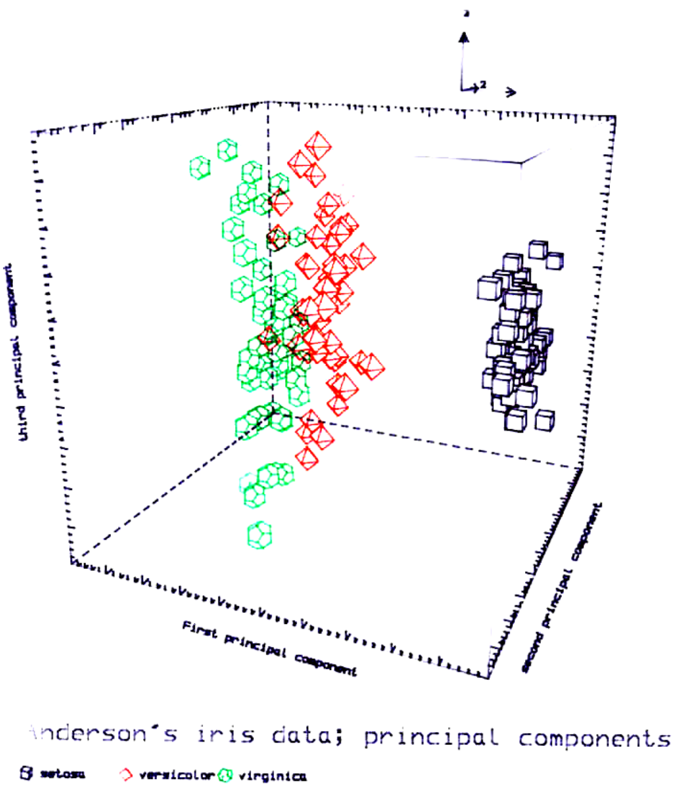
1.3 3 Dimensions

- 2-dim scatter plots of marginal components, (perhaps joined sensibly, see Tukey & Tukey, *in* Interpreting Multivariate Data)
- augmented scatterplots (e.g. code third dimension by size of symbol — see **R** function `symbols()`, or from the menus under `bubbleplot` in `Plot Type` see `help(symbols)`**)**
- 2-dim *random* views.
- 3-dim scatter plots drawn in perspective or rotated interactively, using **R,** S-PLUS or ISP or SAS-insight)

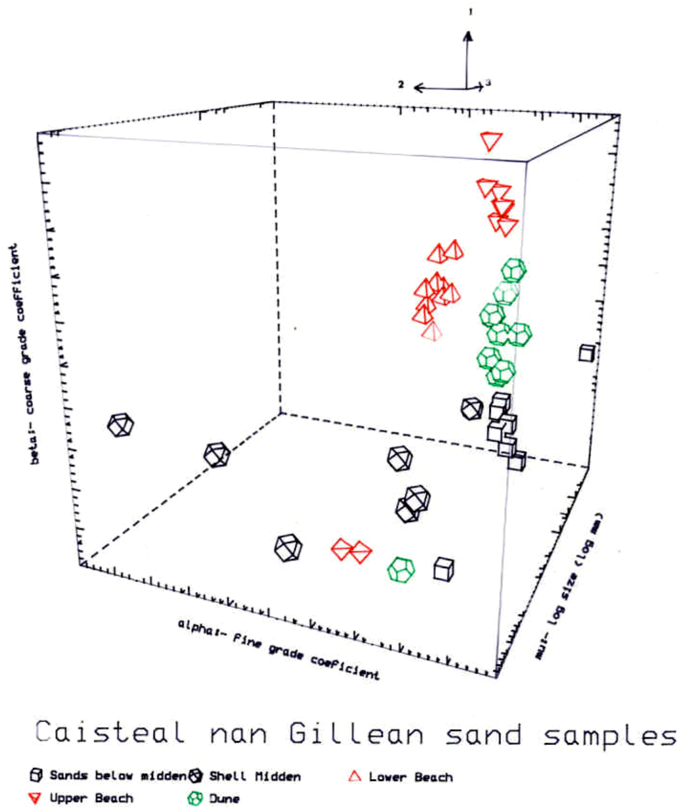
Examples:



Example of a 3-d scatterplot produced by a standard statistical package. This display is not very effective since it is difficult to appreciate the three dimensional structure (if any). Better is to use symbols which can be drawn in perspective:



3-d scatterplot of iris data (actually after transformation to principal components (see later)). Use of perspective and hidden-line removal enhances three-dimensional information.



Another example on particle size data from beaches, dunes and archaeological middens in Oronsay. This example illustrates identification of outliers as well as separation of groups.

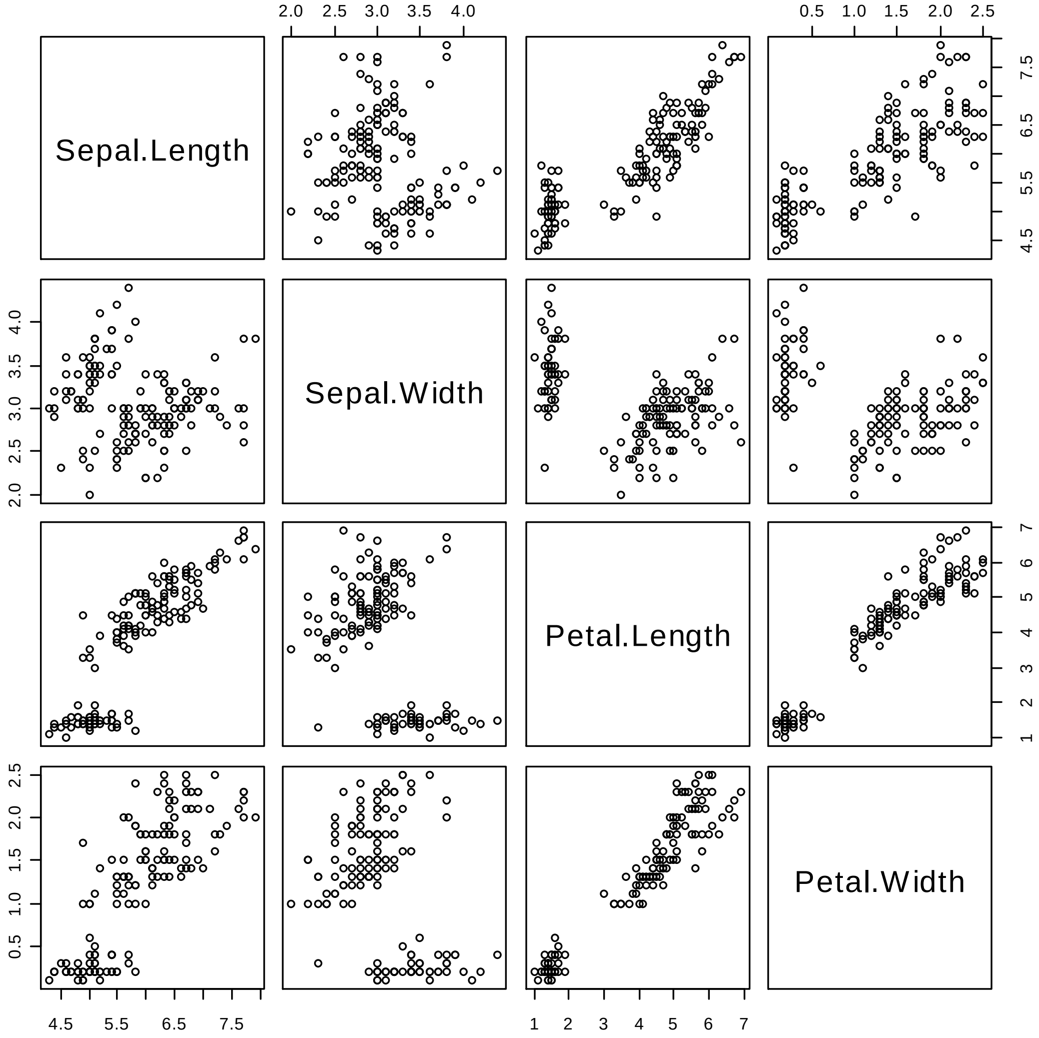
1.4 ≥ 3 Dimensions

1.4.1 Sensible Methods

Matrix plots

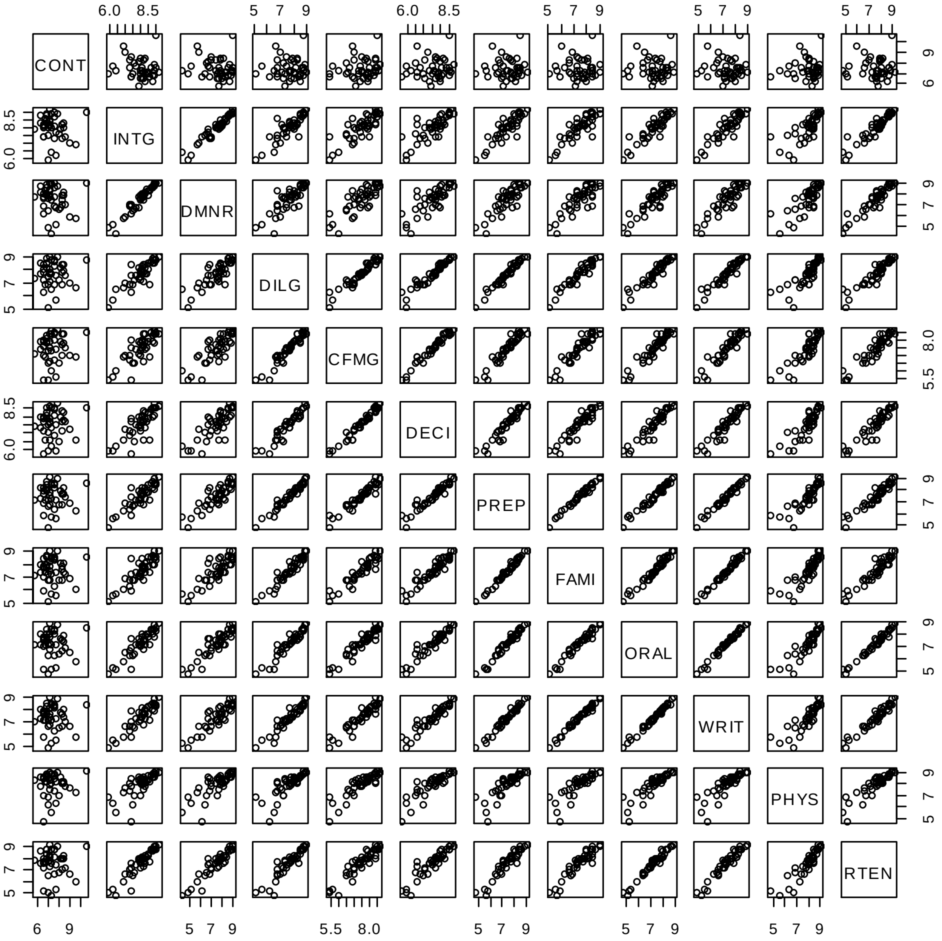
The key tool in displaying multivariate data is scatterplots of pairwise components arranges as a *matrix plot*. Most packages have a facility for producing this easily but the examples below illustrate that as soon as the number of dimensions becomes large (i.e. more than about 5) it is difficult to make sense of the display. If the number of observations is small then some other technique (e.g. star plots, see below) can handle quite large numbers of variables. If there are large numbers of both variables and observations then matrix plots after preliminary analysis has determined the "most interesting" few dimensions to look at (e.g. by principal component analysis, see later). In **R** a matrix plot can be from the command line with the command `pairs()`.

The next two examples give a display of all four components of the iris data where a matrix plot is informative and a matrix plot of 12 dimensional data giving measures of ability and performance of 43 US judges. In the second the number of variables is too large for useful assimilation from the matrix plot.



Matrix plot of Anderson's Iris Data

It is clear that we can see some structure in the data and might even guess that there are three distinct groups of flowers.

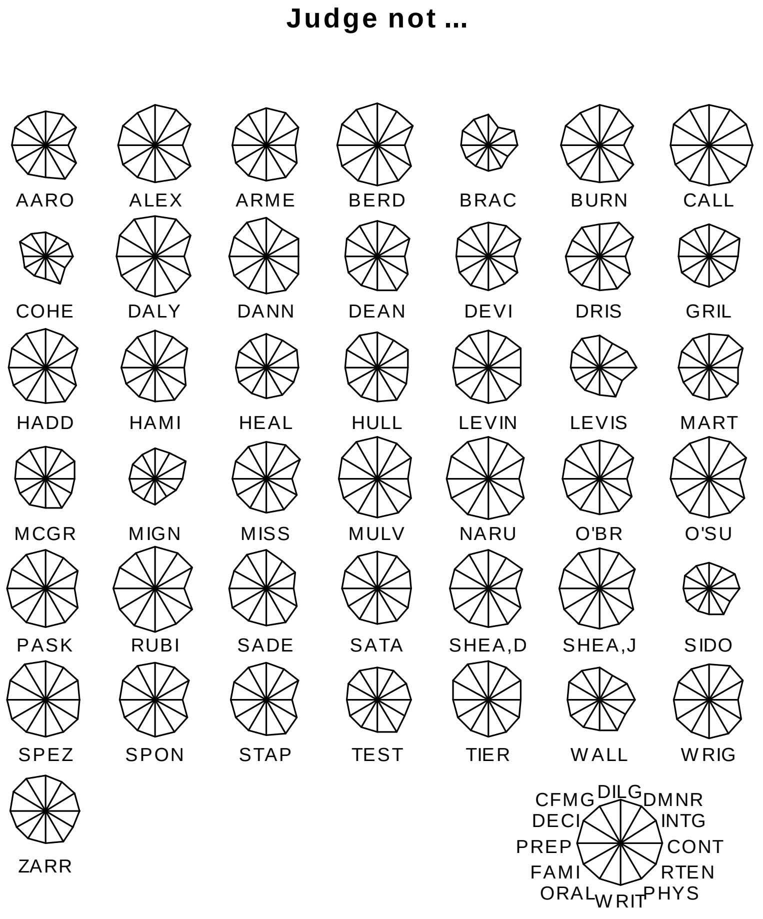


Matrix plot of Judge data

The number of separate plots here is too large for any easy interpretation.

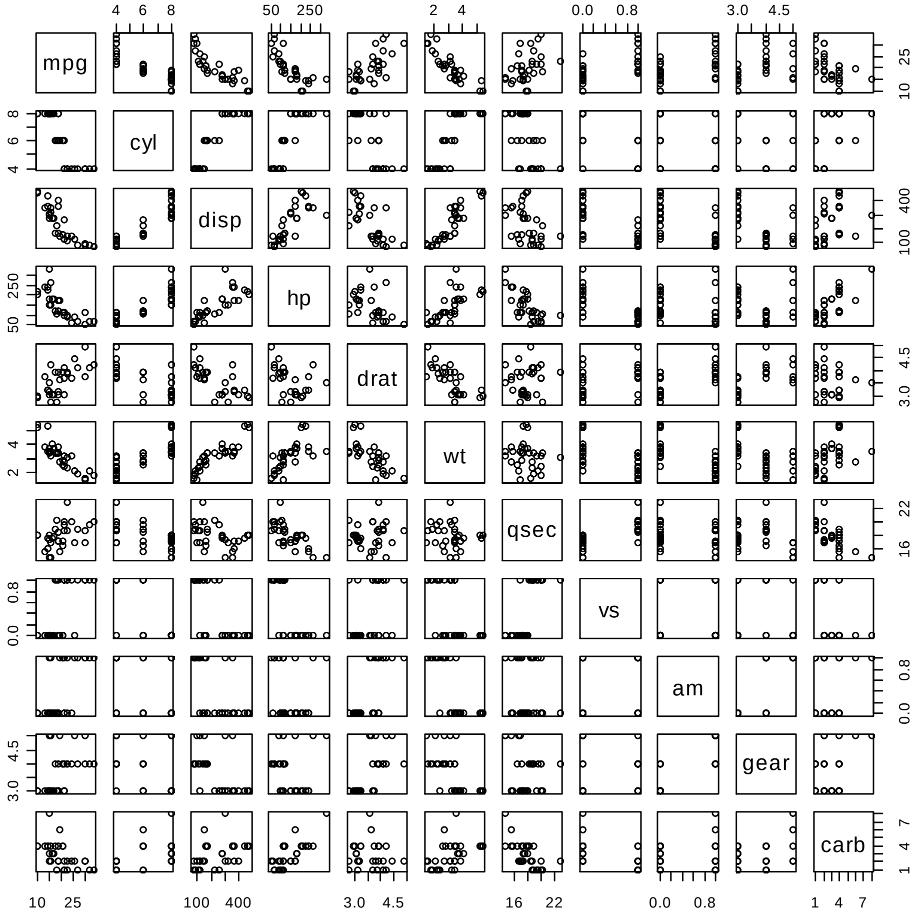
Star Plots

The matrix plot above is difficult to comprehend. An alternative is a *star plot* where each observation is represented by a star or polygon where the length of the vector to each vertex corresponds to the value of a particular variable.

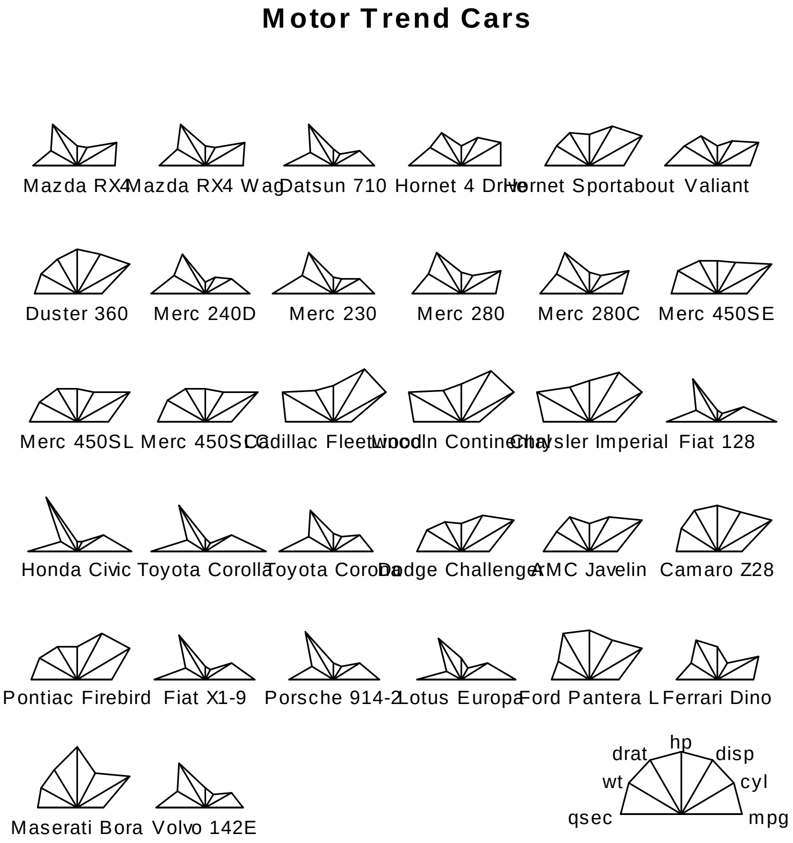


We can begin to see something about similarities and differences between individual observations (i.e. judges) but not any relationships between the variables.

Another example: measurements of 11 properties of 32 cars (fuel consumption, engine size etc).



Again, not informative since there are too many plots.



This is perhaps more informative on individual models of cars but again not easy to see more general structure in the data.

1.4.2 Andrews' Plots

See D.F. Andrews, (1972) *Plots of high dimensional data*, Biometrics, **28,** 125–36.

Data {x_i; i=1,...,n}, (vector observations in p-dimensions so x_{ij} is the jth element of the ith observation).

Define

$$f_{x_i}(t) = \tfrac{1}{\surd 2}x_{i1} + x_{i2}\sin t + x_{i3}\cos t + x_{i4}\sin 2t + x_{i5}\cos 2t + ... + x_{ip}\begin{matrix}\sin\\ \cos\end{matrix}\left(\left[\frac{p}{2}\right]t\right)$$

This maps p-dimensional data {x_i} onto 1-dimensional $\{f_{x_i}(t)\}$ for any t. If we plot $f_{x_i}(t)$ over $-\pi<t<+\pi$ we obtain a 1-dimensional representation of the data with

<u>Properties</u>: (i) preserves means, i.e.

$$f_{\bar{x}}(t) = \tfrac{1}{n}\sum_{i=1}^{n} f_{x_i}(t)$$

(ii) preserves distances; i.e. if we define the square of the 'distance' between two functions as the integrated squared difference:

$$\left\|f_{x_1}(t) - f_{x_2}(t)\right\|^2 = \int_{-\pi}^{+\pi}(f_{x_1}(t) - f_{x_2}(t))^2 dt = \pi\sum\nolimits_{j=1}^{p}(x_{1j} - x_{2j})^2$$

(using properties of the orthogonal functions sin & cos)

= π× square of Euclidean distance between x_1 and x_2.

so close points appear as close functions (though **not** necessarily vice versa).

(iii) yields 1-dimensional views of the data: at $t=t_0$ we obtain the projection of the data onto the vector

$f_1(t_0)=(1/\surd 2, \sin t_0, \cos t_0, \sin 2t_0, ...)'$

(i.e. viewed from some position the data look like the 1-dimensional scatter plot given on the vector $f_1(t_0)$ -- i.e. the line intersecting the Andrews' plot at $t=t_0$).

(iv),(v),.... etc:– tests of significance & distributional results available.

— the plot does depend upon the order in which the components are taken. A preliminary transformation of data may be sensible (e.g. principal components).

— other sets of orthonormal functions are possible

— the 'coverage' decreases rapidly as dimensionality increases, i.e. for high dimensions many 'views' of the data are missed and we miss separation between [groups] of points or other important features.

Computational Note

Many packages give facilities for producing Andrews Plots. The **R** package `andrews` can be installed from the CRAN website.

Examples: First, the example taken from David Andrews' original paper referenced above. Data are 9-dimensional measurements of teeth taken from modern humans of different ethnic origins and various primates. Also, there are similar measurements on early hominids and the objective is to see whether the hominids 'look like' primates or humans. The second example is the iris data [again] shewing the dependence on order of variables.

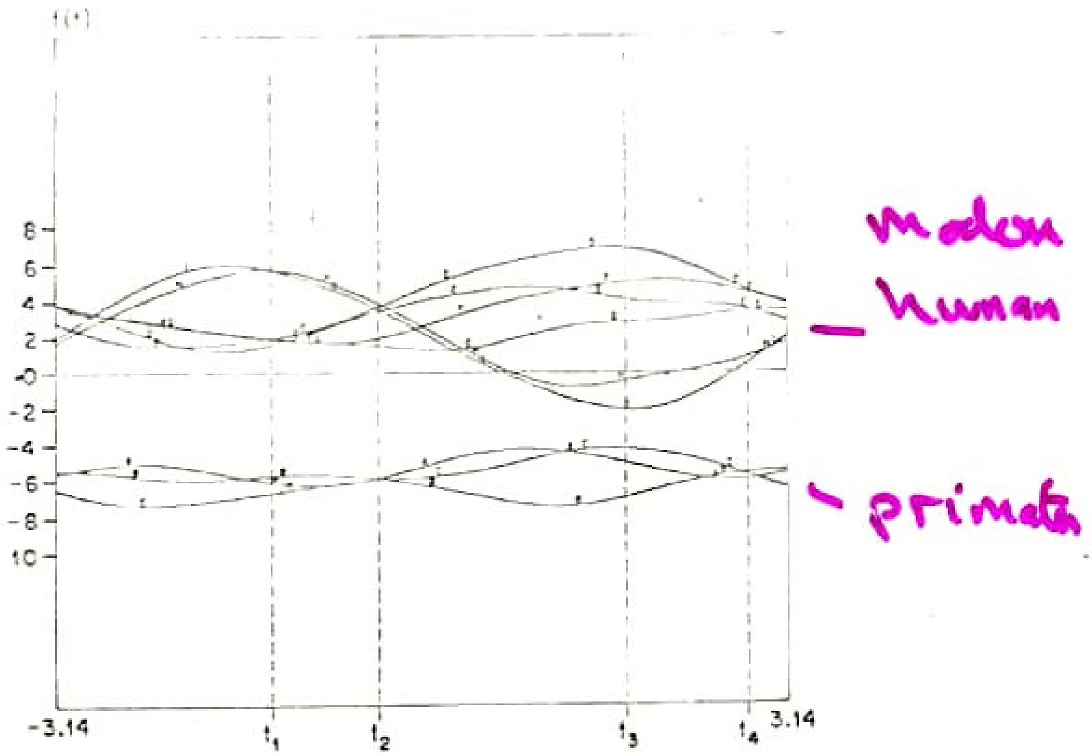


FIGURE 2

8-DIMENSIONAL DATA

PERMANENT FIRST LOWER PREMOLAR GROUP MEANS

A-WEST AFRICAN B-BRITISH C-AUSTRALIAN

D, E-GORILLA F, G-ORANG-OUTANG H, I-CHIMPANZEE

$f(t) = x_1/\sqrt{2} + x_2 \sin(t) + x_3 \cos(t) + \cdots$

Human and primate data shewing clear distinction between humans and primates.

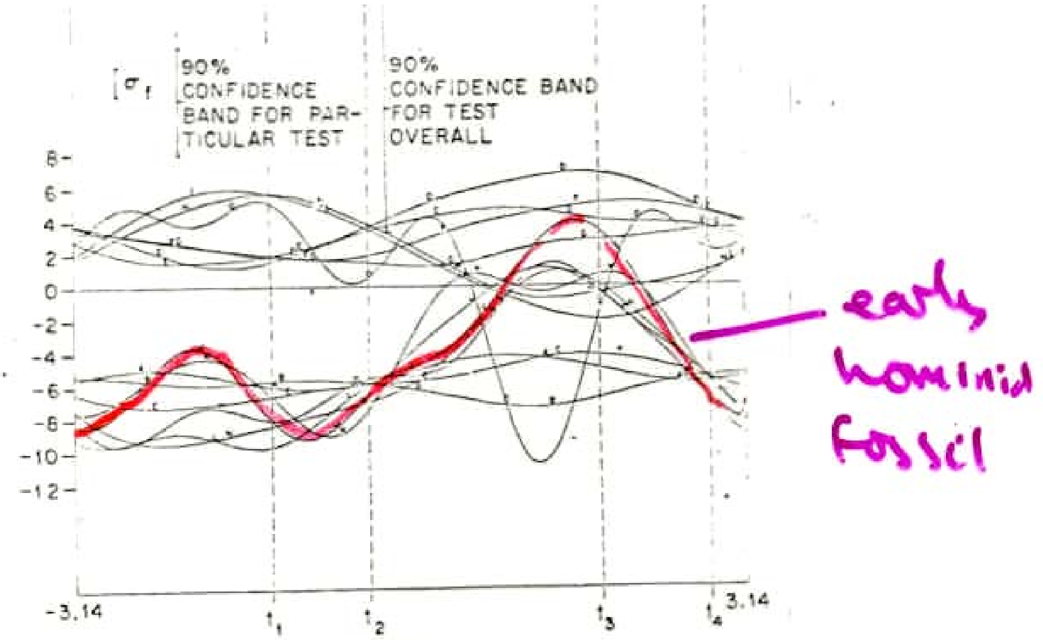


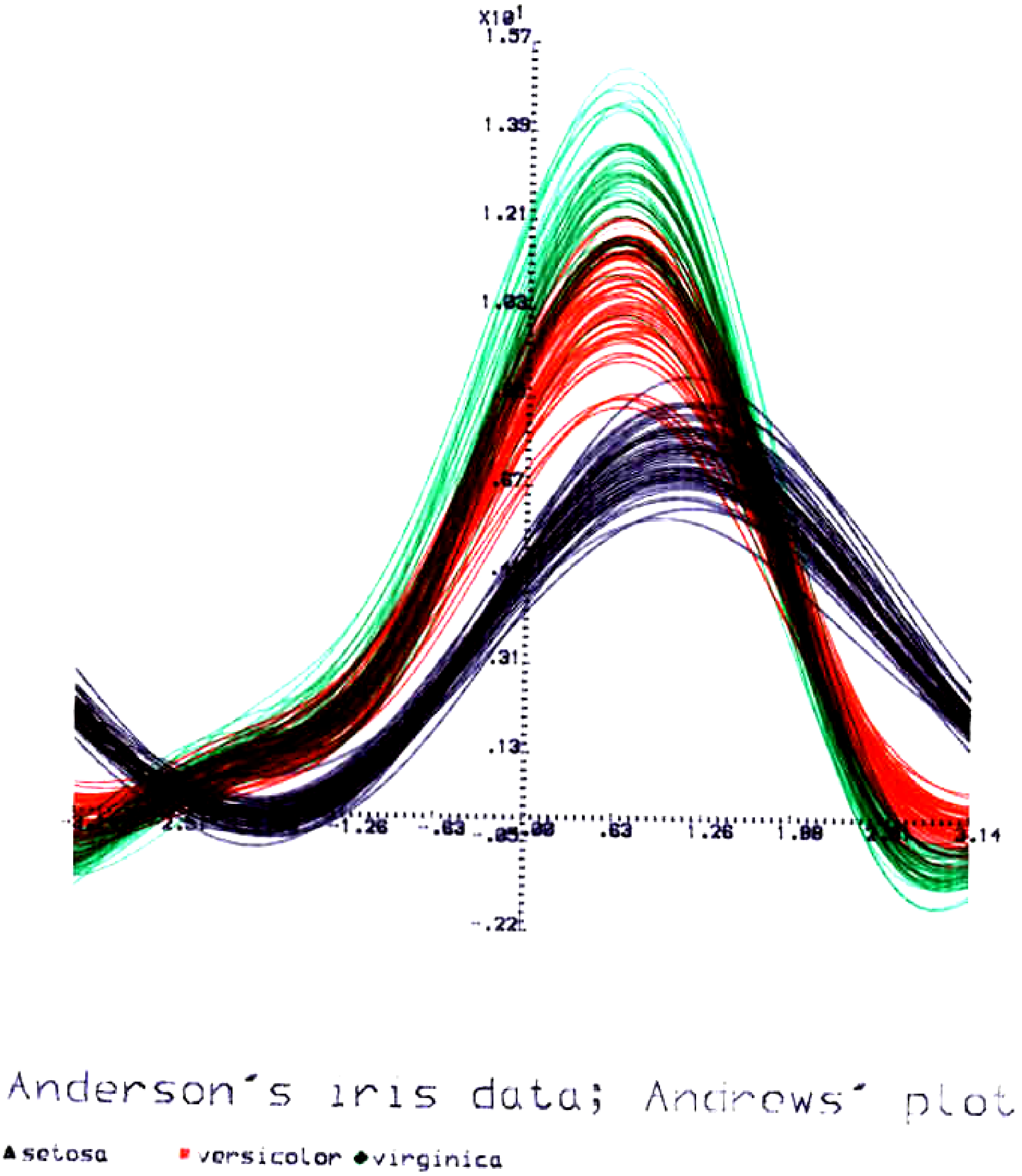
FIGURE 3

8-DIMENSIONAL DATA

AS IN FIGURE 2 BUT WITH FOSSILS ADDED

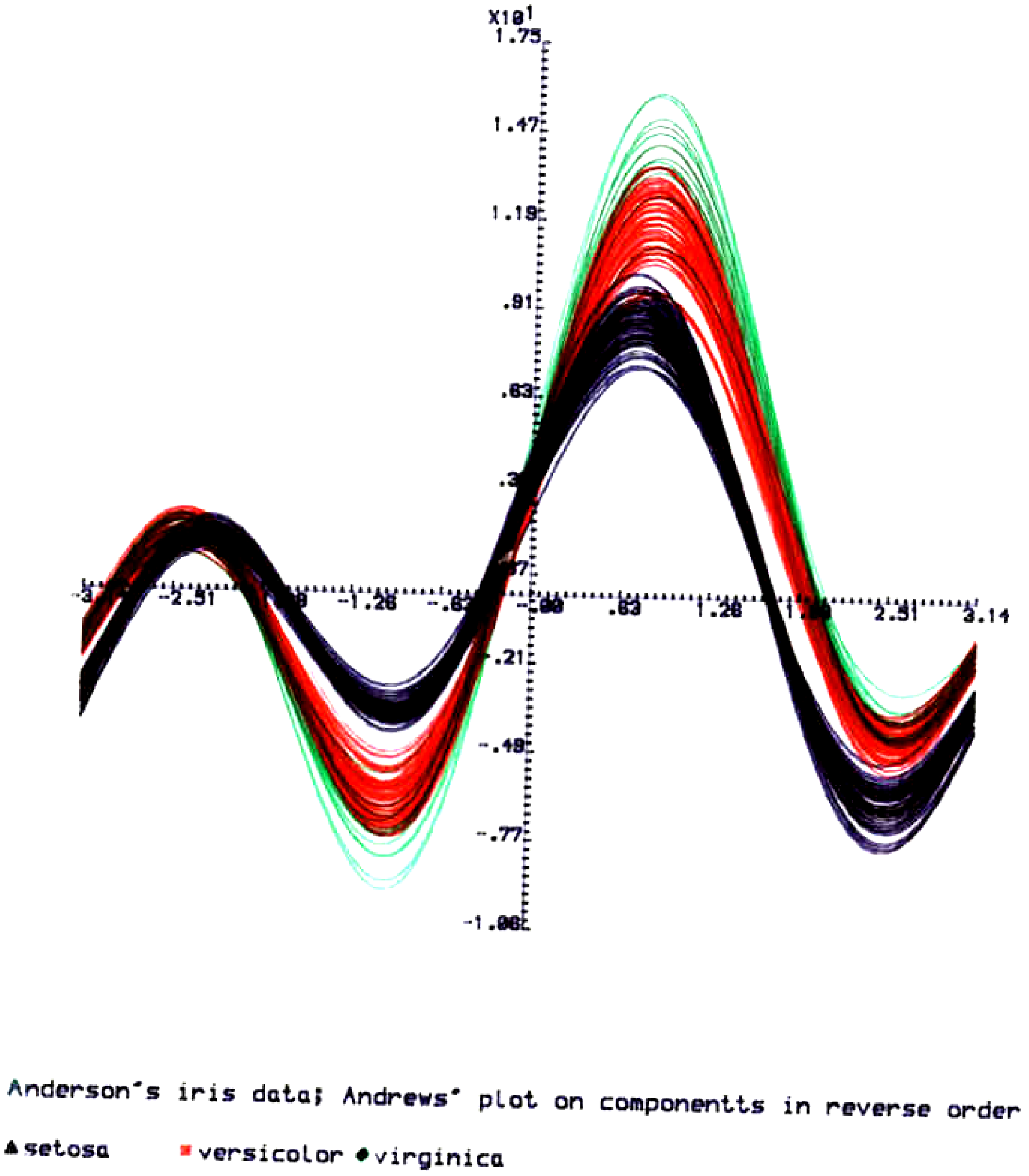
J, K- *Pithecanthropus pekinensis* L-*Paranthropus robustus*
M- *Paranthropus crassidens* N-*Meganthropus palaeojavanicus*
O- *Proconsul africanus*

The diagram shews that the early hominid is mostly like a modern primate but the fact that the curve for the hominid substantially overlaps the human region for part of the range indicates that from some points of view (literally) the tooth looks like a human tooth. Also, the fact that it diverges from the primates means that it does not look entirely like a primate.



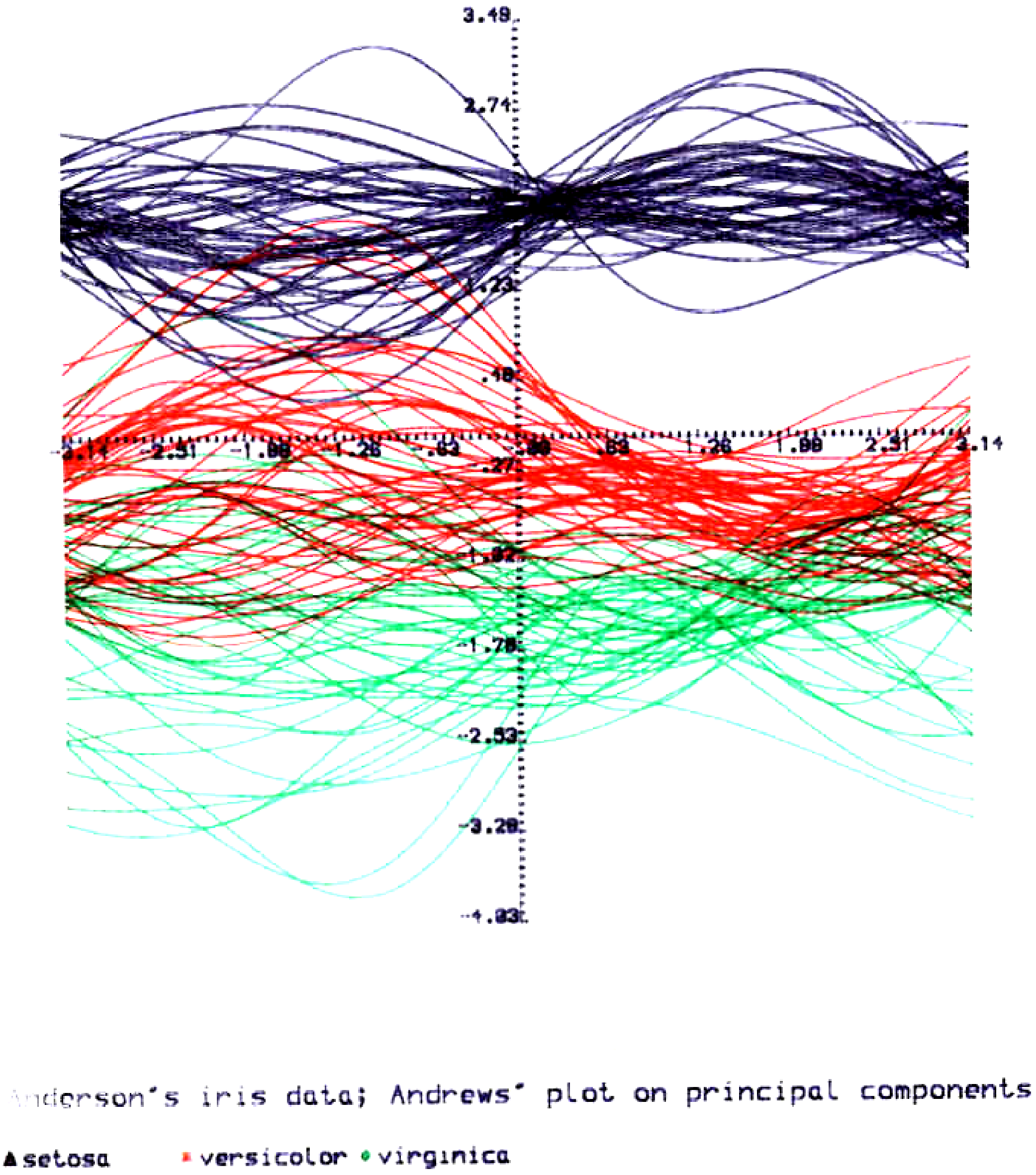
Andrews Plot of iris data with components taken in order

This shews that at least one group separates markedly from the other observations. The fact that the other two groups are only distinguishable in the colour version of this plot (see .pdf version of the notes) means that there is no convincing separation between the versicolor (red) and virginica (green) species.



Andrews Plot with order of components reversed

This shews the dependence in detail of the display on order though the conclusions that can be drawn are perhaps no different.



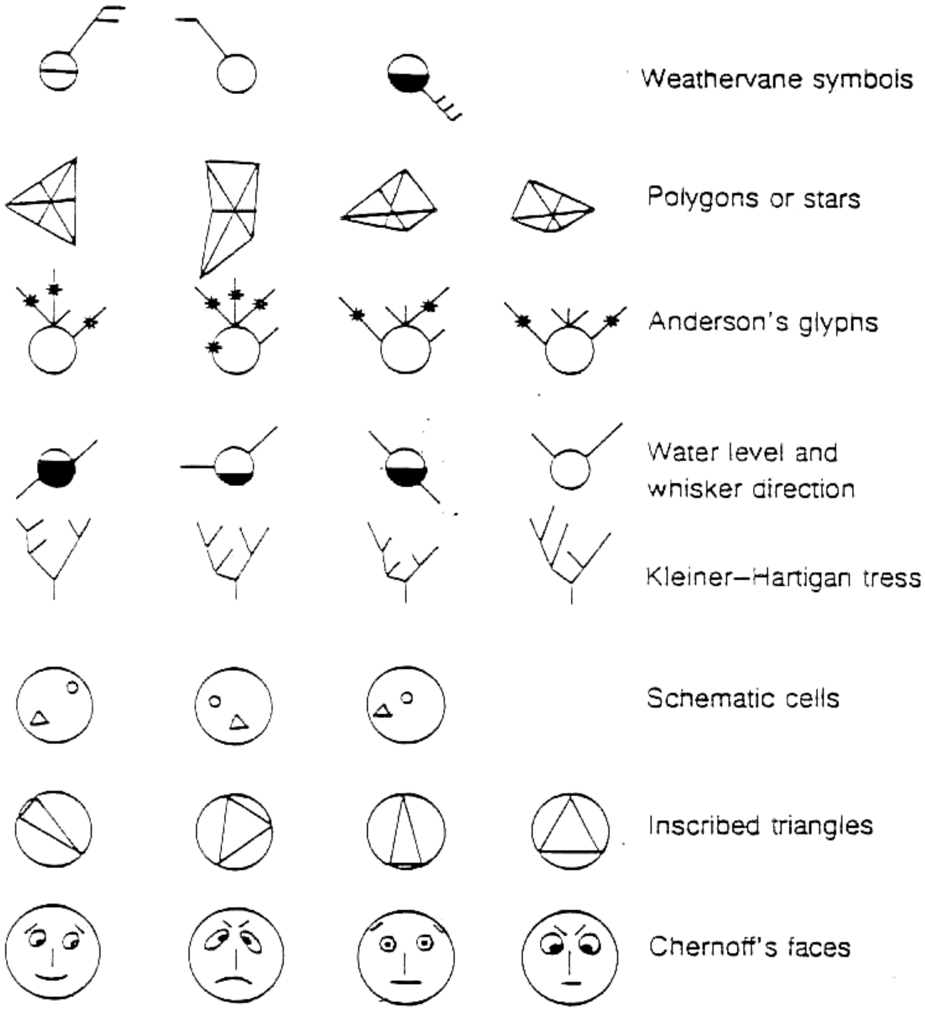
Andrews Plot of Principal Components

This plot shews the iris data after transformation to principal components (see later) where the first component is the ‘most informative. Careful inspection of the colour version shews some separation between the versicolor and virginca species — (especially at around 0.5<t<0.5).

1.4.3 Whimsical Methods

Use of complicated symbols: perhaps 2-dimensional scatter plots of 2 components with other components (back variables) incorporated into a complicated symbol (c.f. weather maps).

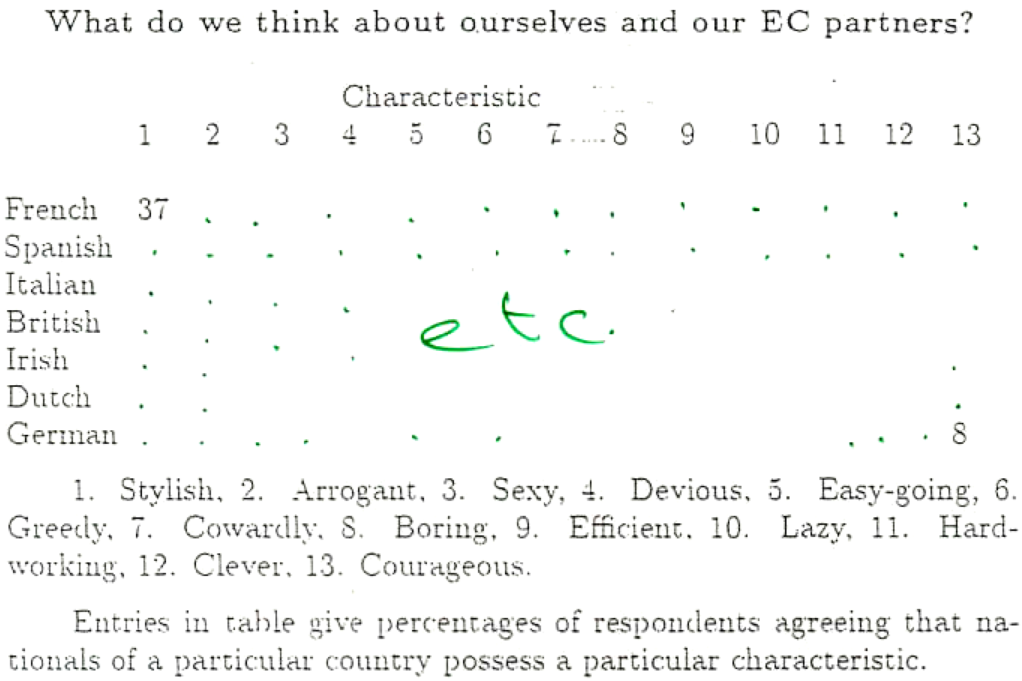
— complicated symbols: e.g. Anderson [Florence Nightingale] Glyphs, Kleiner-Hartigan trees & Chernoff faces (the latter code values of various variables as different facial features). The display below collates some of the suggestions:



1.4.4 Chernoff Faces

This is not a very serious technique but is widely known and available in some packages (e.g. **R** in library `aplpack` with function `faces(.)` and in S-PLUS). The idea is to code each variable in some feature of a face (e.g. length of nose, curvature of mouth). Then you can display each observation as a face (cf using a star or polygon) and look for 'family resemblances'.

The first example (from Everitt, private lecture) gives an illustration of data on perceived national characteristics (data and labels of faces suppressed) and the second illustrates hourly measurements of air pollution during a 24 hour period. In this example the association of variables with features is carefully chosen to give the impression the pollution 'gets worse' by making the faces look increasingly unhappy with rising ozone, sulphur dioxide and radiation.

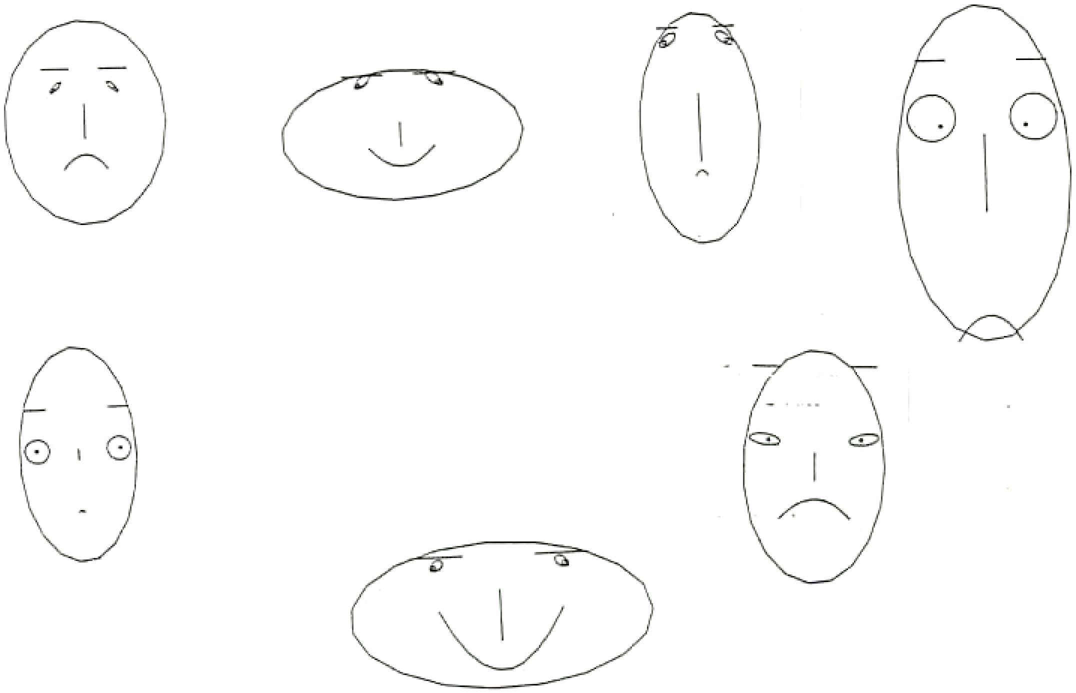


What do we think about ourselves and our EC partners?

	Characteristic												
	1	2	3	4	5	6	7	8	9	10	11	12	13
French	37	.	.	.	.	.	.	.	.	.	.	.	.
Spanish	.	.	.	.	.	.	.	.	.	.	.	.	.
Italian	.	.	.	.									
British	.	.	.	.	etc.								
Irish	.	.	.	.									.
Dutch	.	.											.
German	.	.	.	.	.	.					.	.	8

1. Stylish, 2. Arrogant, 3. Sexy, 4. Devious, 5. Easy-going, 6. Greedy, 7. Cowardly, 8. Boring, 9. Efficient, 10. Lazy, 11. Hard-working, 12. Clever, 13. Courageous.

Entries in table give percentages of respondents agreeing that nationals of a particular country possess a particular characteristic.



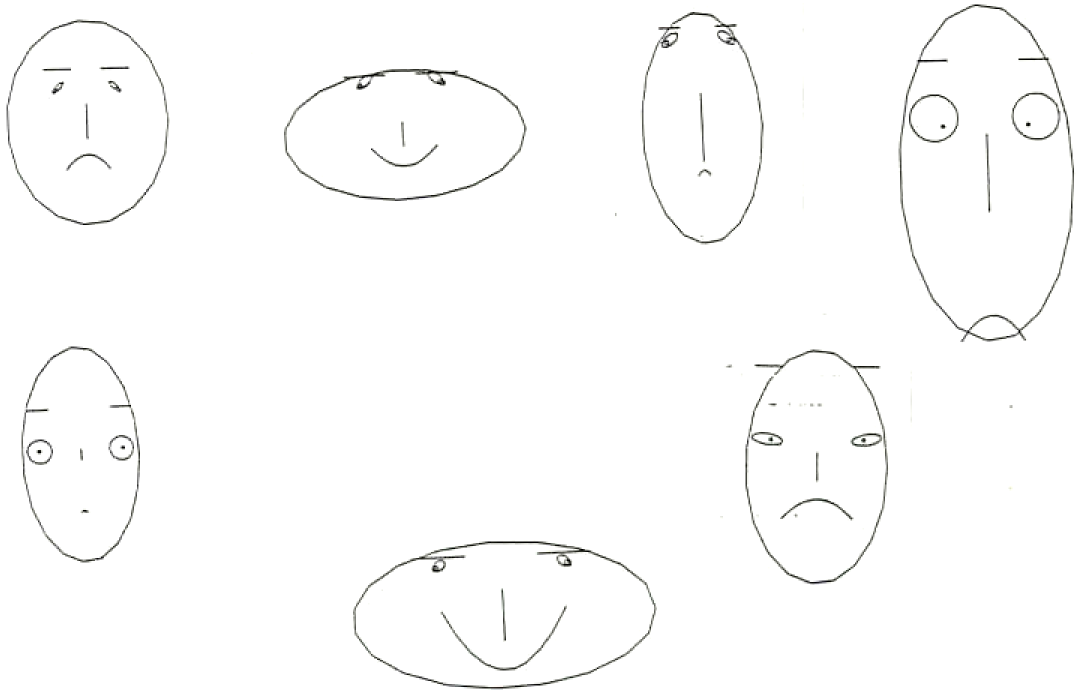
Each face represents a different country as assessed in terms of perceived national characteristics. (Details suppressed).

What do we think about ourselves and our EC partners?

	Characteristic												
	1	2	3	4	5	6	7	8	9	10	11	12	13
French	37	.	.	.	.	.	.	.	.	.	.	.	.
Spanish	.	.	.	.	.	.	.	.	.	.	.	.	.
Italian	.	.	.	.									
British	.	.	.	.		etc.							
Irish	.	.	.	.									.
Dutch	.	.											.
German	.	.	.	.	.	.				.	.	.	8

1. Stylish, 2. Arrogant, 3. Sexy, 4. Devious, 5. Easy-going, 6. Greedy, 7. Cowardly, 8. Boring, 9. Efficient, 10. Lazy, 11. Hard-working, 12. Clever, 13. Courageous.

Entries in table give percentages of respondents agreeing that nationals of a particular country possess a particular characteristic.



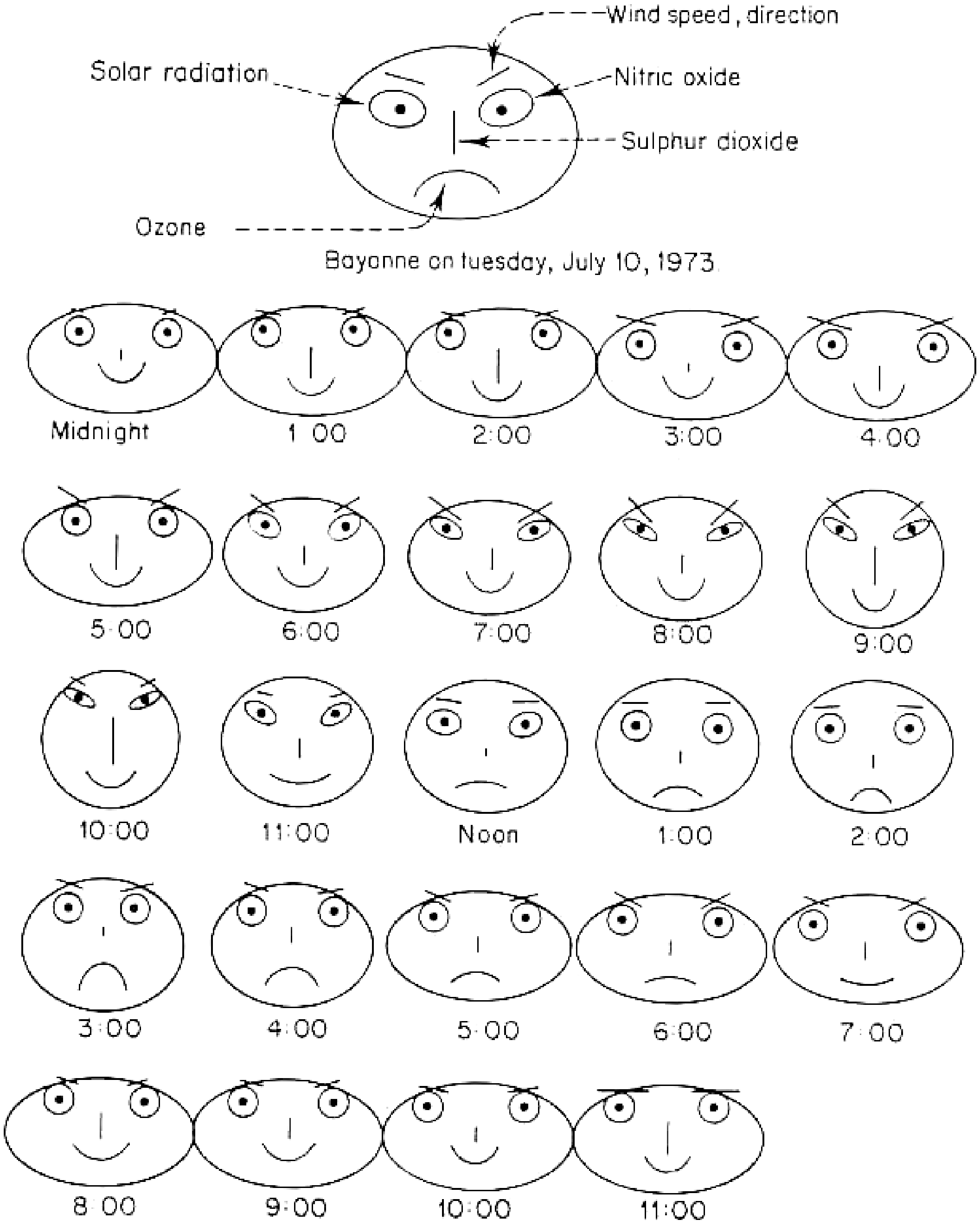


Figure 12.9 Faces used to code air-quality data from Bayonne N.J. on July 10, 1973

Chernoff face display of air quality data.

1.5 Further Reading

The sections above have given only a brief guide to some examples of static graphics. Many more sophisticated displays can be constructed and the book ***R** Graphics* by Paul Murrell (Chapman & Hall, 2006) is a good starting point. More sophisticated interactive and dynamic graphics can be achieved by the add-in package rggobi available from the site http://www.ggobi.org . A good start for working with these facilities is the book *Interactive and dynamic graphics for data analysis* by Dianne Cook and Deborah Swayne (Springer, 2007).

1.6 Conclusions

The simple scatter plots arranged in a matrix work well when there are not too many variables. If there are more than about 5 variables it is difficult to focus on particular displays and so understand the structure. If there are not too many observations then we can construct a symbol (e.g. a star or polygon) for each separate observation which indicates the values of the variables for that observation. Obviously star plots will not be useful if either there are a large number of observations (more than will fit on one page) or if there are a large number of variables (e.g. > 20).

However, many multivariate statistical analyses involve very large numbers of variables, e.g. 50+ is routine, 1000+ is becoming increasingly common in new areas of application.

What is required is a display of ‘all of the data’ using just a few scatterplots (i.e. using just a few variables). That is we need to select ‘the most interesting variables’. This may mean concentrating on ‘the most interesting’ actual measurements or it may mean combining the variables together in some way to create a few new ones which are the ‘most interesting’. That is, we need some technique of ***dimensionality reduction***.

2 Reduction of Dimensionality

2.0 Preliminaries

2.0.1 Warning:

This section requires use of Lagrange Multipliers, eigenanalysis and results on differentiation with respect to vectors. Summaries of these topics are given in the Background Results notes in Appendix 0. Also required is the result that for any vector w then we have

$$\mathrm{var}(X'w)=w'Sw$$

The derivation of principal components below is the first example of a powerful method of maximization which will be used in several different places in later sections and so it is of importance to follow the overall technique.

2.0.2 A Procedure for Maximization

- ♦ 1: Introduce some constraint (either a required constraint or one which is non-operative)
- ♦ 2: Introduce a Lagrange multiplier and define a new objective function
- ♦ 3: Differentiate w.r.t. x and set =0
- ♦ 4: Recognise this is an eigenequation with the Lagrange multiplier as eigenvalue
- ♦ 5: Deduce that there are ONLY a limited number of possible values for this eigenvalue (all of which can be calculated numerically)
- ♦ 6: Use some extra step to determine which eigenvalue gives the maximum (typically use the constraint somewhere)

2.1 Linear Techniques

2.1.0 Introduction

Data $X' = \{x_{ij}\ ;\ i=1,...,n,\ j=1,...,p\} = \{x_i\ ;\ i=1,...,n\}$

Objective is to find a *linear* transformation $X' \rightarrow Y'$ such that

the 1st component of Y′ is the "most interesting",

the 2nd is the "2nd most interesting",

the 3rd .... etc.

i.e. want to choose a new coordinate system so that the data, when referred to this new system, Y′, are such that

the 1st component contains "most information",

the 2nd component contains the next "most information", ... etc.

— & with luck, the first 'few' (2, 3 or 4 say) components contain 'nearly all' the information in the data & the remaining p–2,3,4 contain relatively little information and can be 'discarded' (i.e. the statistical analysis can be concentrated on just the first few components — multivariate analysis is much easier in 2 or 3 dimensions!)

Notes

A *linear* transformation $X' \rightarrow Y'$ is given by $Y'=X'A$ where A is a p×p matrix; it makes statistical sense to restrict attention to *non-singular* matrices A. If A happens to be an *orthogonal* matrix, i.e. $A'A=I_p$ (I_p the p×p identity matrix) then the transformation $X' \rightarrow Y'$ is an orthogonal transformation, (i.e. just a rotation and/or a reflection of the n points in p-dimensional space).

2.1.1. Principal Components

The basic idea is to find a set of orthogonal coordinates such that the sample variances of the data with respect to these coordinates are in decreasing order of magnitude, i.e. the projection of the points onto the 1st principal component has maximal variance among all such linear projections, the projection onto the 2nd has maximal variance subject to orthoganility with the first, projection onto the 3rd has maximal variance subject to orthogonality with the first two, ... etc.

Note

“most interesting” ⇔ “most information” ⇔ **maximum variance**

Definition

The first principal component is the vector a_1 such that the projection of the data X' onto a_1, i.e. $X'a_1$, has maximal variance, subject to the normalizing constraint $a_1'a_1=1$.

Now $\text{var}(X'a_1) = a_1'\text{var}(X')a_1 = a_1'Sa_1$ and note that a_1' is $1\times p$, S is $p\times p$ and a_1 is $p\times 1$ so $a_1'Sa_1$ is $1\times p\times p\times p\times 1=1\times 1$, i.e. a scalar.

So, the problem is

to maximize $a_1'Sa_1$ subject the constraint $a_1'a_1=1$.

Define $\Omega_1 = a_1'Sa_1-\lambda_1(a_1'a_1-1)$, where λ_1 is a Lagrange multiplier, and maximize Ω_1 with respect to both a_1 and λ_1.

Setting $\frac{\partial\Omega_1}{\partial\lambda} = 0$ gives $a_1'a_1=1$.

Differentiating w.r.t. a_1 gives $\frac{\partial\Omega_1}{\partial a_1} = 2Sa_1 - 2\lambda_1 a_1$, and setting equal to zero gives $Sa_1 - \lambda_1 a_1 = 0$ ....$*$

i.e. a_1 is an eigenvector of S corresponding to the eigenvalue λ_1;

S has p non-zero eigenvalues (provided it is non-singular):—

¿ So which eigenvalue is λ_1? →

premultiply equation $*$ by a_1', so $a_1'Sa_1 - a_1'\lambda_1 a_1 = 0$,

so $\text{var}(X'a_1) = a_1'Sa_1 = \lambda_1 a_1'a_1 = \lambda_1$ (since $a_1'a_1=1$),

so to maximize $\text{var}(X'a_1)$ we must take λ_1 to be the ***largest*** eigenvalue of S and a_1 as the corresponding eigenvector, and then the maximum value of this variance is λ_1 (which is also the required value of the Lagrange multiplier).

The 2nd, 3rd,.... principal components are defined recursively:

e.g. a_2: projection of X' onto a_2 has maximal variance subject to $a_2'a_2=1$ and $a_2'a_1=0$ (a_1 and a_2 are orthogonal)

& then a_3: projection of X' onto a_3 subject to $a_3'a_3=1$ and $a_3'a_2=a_3'a_1=0$, etc.

So, for a_2 we require to maximize

$$\Omega_2 = a_2'Sa_2 - \mu a_2'a_1 - \lambda_2(a_2'a_2-1)$$

where μ and λ_2 are Lagrange multipliers.

Differentiating w.r.t. μ and λ_2 and setting to zero just expresses the constraints.

Differentiating w.r.t. a_2 and setting to zero gives

$$2Sa_2 - \mu a_1 - 2\lambda_2 a_2 = 0 \text{ ....}**$$

Premultiplying equation $**$ by a_1' gives

$$2a_1'Sa_2 - \mu a_1'a_1 - 2\lambda_2 a_1'a_2 = 0$$

i.e. $2a_1'Sa_2 = \mu$ (since $a_1'a_2 = 0$)

premultiplying equation $*$ by a_2' gives

$$a_2'Sa_1 - \lambda_1 a_2'a_1 = 0$$

i.e. $a_2'Sa_1 = 0$ (since $a_2'a_1 = 0$), so $\mu = 0$ and so a_2 satisfies

$$Sa_2 - \lambda_2 a_2 = 0$$

i.e. a_2 is an eigenvector of S corresponding to the eigenvalue λ_2

(¿but which eigenvalue?)

premultiplying by a_2' gives $\quad a_2'Sa_2 - \lambda_2 a_2'a_2 = 0$

i.e. $\text{var}(X'a_2) = a_2'Sa_2 = \lambda_2$

and so to maximize var($X'a_2$) we must take λ_2 as the second largest eigenvalue of S (note we cannot take it as the largest eigenvalue since then we would have $a_2=a_1$ but we have $a_2'a_1=0$).

Proceeding recursively we can see that:—

Theorem: The p principal components of data X′ are the p eigenvectors $a_1,a_2,....,a_p$ corresponding to the p ordered eigenvalues $\lambda_1\geq\lambda_2\geq...\geq\lambda_p$ of S, the variance of X′.

NB The above is not a formal mathematical proof but it is easy to construct a proof by induction (i.e. assume that the first k principal components correspond to the first k eigenvectors and then shew that his implies that the (k+1)th is as well. Since we have proved the result for k=1 the result follows by induction.

Notes

- The argument above does not constitute a formal proof of the theorem but it is easy to construct a formal proof by induction: show that assuming the truth of the theorem for i=1,2,...,k implies that it also holds for i=k+1, since it is proved true for i=1 it will follow by induction that it is true for i=1,2,...,p.

- There is a slight difficulty if either

  (i) $\lambda_i=\lambda_{i+1}$ since then a_i and a_{i+1} can be chosen in arbitrarily many ways, i.e. mutually orthogonal but anywhere in the plane orthogonal to a_{i-1}, or

  (ii) $\lambda_p=0$ since a_p is not then uniquely defined.

  However, neither of these is a practical difficulty with real data since it happens only with zero probability if there are more observations than variables and there is no implicit redundancy in the variables (i.e. none is a linear combination of any of the others). If there are fewer observations than variables (i.e. if n<p) then only the first n of the principal components will be of practical interest.

  With the population counterparts of principal components, i.e. the eigenvectors of a population variance matrix (as distinct from those of an observed sample variance matrix), then it could well be that $\lambda_i=\lambda_{i+1}$ or $\lambda_p=0$ (or both) but this is not a practical problem.

2.1.2 Computation

We have $\lambda_1,\lambda_2,...,\lambda_p$ satisfy, for $a=a_1,a_2,...,a_p$,

$Sa - \lambda a = 0$, i.e. $(S - \lambda I_p)a = 0$. This is a system p simultaneous equations in the components of a, so the determinant of the coefficients must be zero for non-trivial solutions to exist, i.e. we must have $\det(S - \lambda I_p) = 0$.

Now $\det(S - \lambda I_p)$ is a polynomial of degree p in λ. In principle, this can be solved for the p values of λ, $\lambda_1,\lambda_2,...,\lambda_p$ and then for each of these p values the simultaneous equations $Sa_i - \lambda_i a_i = 0$ can be solved for the components of a_i, together with the constraint that $a_i'a_i=1$

(i.e. $\sum_{j=1}^{p} a_{ij}^2 = 1$) .

This is only practical for p=2 or perhaps p=3.

For p≥3 there are iterative algorithms based on linear algebra — see Morrison §8.4, pp279.

Easier method: use **R,** S-PLUS, MINITAB, SPSS, Genstat,...

Many of these packages have complete built in ready-made analyses for principal component analysis, or it can be constructed from individual commands.

e.g. in **R** with data in matrix X

```
S<-var(X)
```

`S.eig<-eigen(S)` puts the eigenvectors of S (i.e. the loadings of the principal components) in `s.eig$vectors)` and the eigenvalues (i.e. variance on the principal components) in `S.eig$values.` Technically better is to use the function `svd()` instead of `eigen()` because it is numerically more stable:

`S.svd<-svd(S)`

`S.svd$v` and `S.svd$d` give the eigenvectors and eigenvalues of `S` respectively.

Alternatively

`S.pca<-princomp(X)` puts the eigenvectors of `S` in `S.pca$loadings` and the ***square roots*** of the eigenvalues in `S.pca$sdev`.

`S.pr<-prcomp(X)` puts the eigenvectors of `S` in `S.pr$rotation` and the ***square roots*** of the eigenvalues in `S.pr$sdev`.

These two ready made functions in R, `princomp()` and `prcomp()`, are based on `eigen()` and `svd()` respectively. `prcomp()` is generally preferred. See §2.1.7 for more details.

Tasks 2

(see §2.0–§2.1)

1) Try these simple exercises both 'by hand' and using **R**:

i) Find the eigenvalues and normalized eigenvectors of the 2×2 matrix $\frac{1}{7}\begin{pmatrix} 208 & 144 \\ 144 & 292 \end{pmatrix}$

ii) Find the eigenvalues and one possible set of normalized eigenvectors of the 3×3 matrix $\begin{pmatrix} 2 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 2 \end{pmatrix}$

iii) Find the inverse of $\begin{pmatrix} 2 & 0 & 0 \\ 0 & 2 & 1 \\ 0 & 1 & 2 \end{pmatrix}$

2) (optional — but at least note the results, these are counterexamples to false assumptions that are all to easy to make since they contradict 'ordinary' algebra).

Let $A = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}, B = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}, C = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}, D = \begin{pmatrix} 1 & -1 \\ -1 & -1 \end{pmatrix},$

$E = \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}$ and $F = \begin{pmatrix} 1 & 1 \\ -1 & -1 \end{pmatrix}$ then show:–

i) $A^2 = -I_2$ (so A is 'like' the square root of –1)

ii) $B^2 = 0$ (but $B \neq 0$)

iii) $CD = -DC$ (but $CD \neq 0$)

iv) $EF = 0$ (but $E \neq 0$ and $F \neq 0$)

3) (see 0.10.1) The data file `openclosed.Rdata`★ consists of examination marks in five subjects labelled `mec, vec, alg, ana` and `sta.` Download the datafile to your own hard disk. Using Windows Explorer double click on the file. This will open **R**, change the working directory to that where you have stored the data and read in the data to dataframe `scor.` ★Mardia, Kent & Bibby (1981).

i) Then issue the following commands and read the results

```
ls()      # see what objects are in the works space;
          #  there should be only the dataframe scor

X<-as.matrix(t(scor)) # define X to be the matrix
                      # of the transpose of scor

S<-var(t(X)) # calculate  the variance matrix of X'=scor

A<-eigen(S)$vectors # Calculate the eigenvectors of S
#                     & store them in A
V<-eigen(S)$values # and eigenvalues in V
A  # look at A
V  # look at V
sum(diag(S))# look at trace(S)
sum(V)      # look at sum of eigenvalues in V (they should be the same)

options(digits=4) only print four decimal places

A%*%t(A)    # check that A is an orthogonal matrix
t(A)%*%A    # (as it should be, property of eigenvectors)

round(A%*%t(A)) # easier to see if round to whole numbers
round(t(A)%*%A)

t(A)%*%S%*%A    # calculate A'SA

Y<-t(A)%*%X # let Y=A'X so that Y'=X'A, the data rotated
            # onto the principal components.
var(t(Y))   # the variance of the data on the principal components
            # note these are the same up to rounding errors
round(t(A)%*%S%*%A) # easier to see if round to whole numbers
round(var(t(Y)))
V           # eigenvalues of S, also same.
sum(diag(S)) # find trace(S)
sum(V)       # same as above
```

4) The data file `bodysize.Rdata`★ consists of measurements of the circumferences (in centimetres) of `neck, chest, abdomen, hip, thigh, knee, ankle, biceps, forearm` and `wrist` of 252 men. Download the datafile to your own hard disk. Using Windows Explorer double click on the file. This will open **R**, change the working directory to that where you have stored the data and read in the data to dataframe `bodysize`. Next, download the function `screeplot()` contained in scriptfile `scree.R` to the same directory on you hard disk. Using the menu in **R** open the script file `scree.R` (top left icon in the menu bar), highlight all the lines in the function and click the middle icon to run the selected lines. This will load the function into your current **R** session. ★source: *Journal of Statistics Education* Data Archive

   i) Then issue the following commands and read the results

```
bodysize[1:5,]              # gives first few lines of the data file
diag(var(bodysize))         # gives variances of variables
sqrt(diag(var(bodysize))) # gives standard deviations
# note standard deviations vary by a factor of > 10
# so perform PCA with correlation matrix
body.pc<-princomp(bodysize,cor=T)
body.pc
summary(body.pc)
body.pc$loadings
screeplot(bodysize,T)
print(body.pc$loadings, cutoff=0.01)
```

   ii) How many principal components would you suggest adequately contain the main sources of variation within the data.

   iii) What features of the body sizes do the first three [four?] components reflect?

5) Calculate the principal components of the 4 measurements on Irises: using the ’ready made’ facility for principal component analysis by first calculating the covariance matrix and then looking at the eigenanalysis of the matrix (try `help(eigen)`).

2.1.3 Applications

The principal components a_1,a_2,...a_p provide a useful coordinate system to which to refer the data and for displaying them graphically, i.e. better than the original coordinates

$$\begin{pmatrix}1\\0\\0\\\vdots\\\vdots\\0\end{pmatrix},\begin{pmatrix}0\\1\\0\\\vdots\\\vdots\\0\end{pmatrix},\ldots\ldots,\begin{pmatrix}0\\0\\\vdots\\\vdots\\1\\0\end{pmatrix},\begin{pmatrix}0\\0\\\vdots\\\vdots\\0\\1\end{pmatrix}$$

Let A= (a_1,a_2,....,a_p) — the p×p matrix with j^{th} column=a_j.

(Notice that AA'=I_p , i.e. A is orthogonal)

Then the projection of the data X' onto this coordinate system is

$$Y'=X'A$$

— since A is orthogonal the transformation $X' \rightarrow Y'$ is a rotation/reflection (i.e. no overall change of scale or origin)

If we measure the 'total' variation in the original data as the sum of the variances of the original components,

i.e. $s_{11}+s_{22}+...+s_{pp}$= tr(S)

then the 'total' variation of Y' is $\lambda_1+\lambda_2+...+\lambda_p$

(by construction of A)

and $\Sigma\lambda_i$=tr(S) (since the λ_i are eigenvalues of S) (see property 1 of *Background Results*).

So we can interpret

$$\lambda_1/\Sigma\lambda_i \quad (=\lambda_1/\mathrm{tr}(S)\ )$$

as the proportion of the total variation 'explained' by the first principal component, and

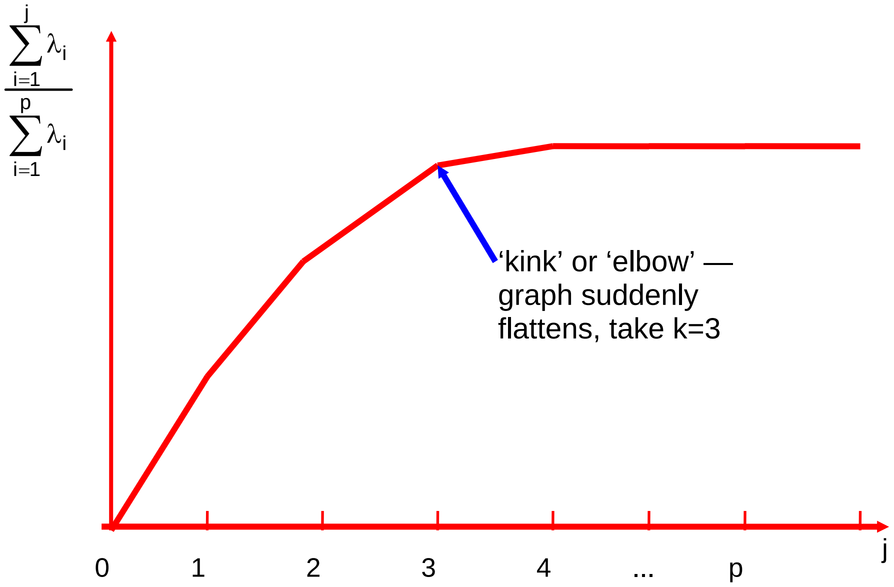
$$(\lambda_1+\lambda_2)/\Sigma\lambda_i$$

as the proportion of the total variation 'explained' by the first two principal components, ... etc.

If the first 'few' p.c.s explain 'most' of the variation in the data, then the later p.c.s are redundant and 'little' information is lost if they are discarded (or ignored).

If e.g. $\frac{\sum_{j=1}^{k}\lambda_i}{\sum_{j=1}^{p}\lambda_i} \approx$ say 80+%, then the $(k+1)^{th}$,...., p^{th} components contain relatively little information and the dimensionality of the data can be reduced from p to k with little loss of information. Useful if k=1, 2, 3, 4?, 5??? The figure of 80% is quite arbitrary and depends really on the type of data being analysed — particularly on the applications area. Some areas might conventionally be satisfied if 40% of the variation can be explained in a few p.c.s, others might require 90%. A figure needs to be chosen as a trade-off between the convenience of a small value of k and a large value of the cumulative relative proportion of variance explained.

If p is large an informal way of choosing a 'good' k is graphically with a scree-plot, i.e. plot $\frac{\sum_{i=1}^{j}\lambda_i}{\sum_{j=1}^{p}\lambda_i}$ *vs* j. The graph will be necessarily monotonic and convex. Typically it will increase steeply for the first few values of j (i.e. the first few p.c.s) and then begin to level off. The point where it starts levelling off is the point where bringing in more p.c.s brings less returns in terms of variance explained.



Scree graphs can be drawn easily in MINITAB, choose Stat>Multivariate>Principal Components ... and then the Graphs... button.

The ready made menu for principal component analysis is useful but limited — there may be some calculations that you need to do with the command language using Eigen, Cova, Mult etc.

Some formal tests of H: λ_p=0 are available (but not used much).

2.1.4 Example (from Morrison).

Measurements in millimetres of length, width and height of shells of 24 female painted turtles (*Chrysemys picta marginata)* collected in a single day in the St Lawrence Valley. This is a rather artificial example since the dimensionality is already low but it illustrates some of the calculations and interpretation.

n=24, p=3

$$S=\begin{pmatrix}451.39 & 271.17 & 168.70\\ * & 171.73 & 103.29\\ * & * & 66.65\end{pmatrix}, \qquad \mathrm{tr}(S)=689.77$$

	Principal Components		
	a₁	**a₂**	**a₃**
1==length	.8126	–.5454	–.2054
2==width	.4955	.8321	–.2491
3=height	.3068	.1008	.9465
variance=λ_j	680.4	6.5	2.86

Checks: (i) $\Sigma\lambda_j = 689.76$

(ii) e.g. $Sa_2–\lambda a_2=0$, i.e. $S\begin{pmatrix}-.5454\\ .8321\\ .1008\end{pmatrix}-6.5\begin{pmatrix}-.5454\\ .8321\\ .1008\end{pmatrix}=0$

The first component accounts for 98.64% of the total variance. This is typical of data on *sizes* of items and it reflects variation in overall sizes of the turtles: The value of it for any particular turtle

(or the *score* on the 1st p.c.) is

$$Y_1=.81\times\text{length}+.50\times\text{width}+.31\times\text{height}$$

— it reflects general size.

The 2nd p.c. is $Y_2=–.55\times\text{length}+.83\times\text{width}+.10\times\text{height}$

Here, the coefficient of height is much smaller in absolute terms than the other two and so the second p.c. is dominated by variations in length and widths and is actually a contrast between them, so high values of Y_2 come from nearly round shells and low values from elongated elliptical shells, i.e. Y_2 is a measure of shape or roundness.

Similarly, Y_3= –.21×length–.25×width+.95×height which reflects height *vs* (length+width) and so is a measure of how peaked or pointed the shell is.

Conclusions are that the shells vary most in overall size, next most in shape and third most in peakedness.

2.1.4.1 R calculations

```
> S<-
      matrix(c(451.39,271.17,168.7,271.17,171.73,103.29,168.7,
       103.29,66.65),nrow=3,ncol=3)
> eigenS<-eigen(S)
> eigenS
$values:
[1] 680.41111   6.50163   2.85726

$vectors:
         [,1]      [,2]      [,3]
[1,] 0.812643 -0.545418  0.205257
[2,] 0.495495  0.832072  0.249282
[3,] 0.306752  0.100874 -0.946429

> eigenS$vectors[,2]
[1] -0.545418  0.832072  0.100874
> S%*%eigenS$vectors[,2]-6.50163*eigenS$vectors[,2]
              [,1]
[1,] -1.86545e-006
[2,]  2.84587e-006
[3,]  3.45010e-007
```

2.1.5 General Comments

- Generally, if data X’ are measurements of p variables all of the same ‘type’ (e.g. all concentrations of amino acids or all linear dimensions in the same units, but ***not*** e.g. age/income/weights) then the coefficients of principal components can be interpreted as ‘loadings’ of the original variables and hence the principal components can be interpreted as contrasts in the original variables, as with the turtle shell example.

  This interpretation is less easy if the data are of mixed ‘types’ or if the variables have widely different variances, even if the eigen analysis is based on the correlation rather than the covariance matrix, see below.

  (The same difficulty arises with the interpretation of regression coefficients in all forms of regression model — linear, logistic, log-linear, Cox proportional hazards,.........)

- Notice that principal components are not invariant under all linear transformations of original variables, in particular separate scaling of the variables. For example, principal components of data on people' heights, weights and incomes measured in inches, pounds and £s are not the same as those on the same data converted to centimetres, kilograms and €s. This means one should be careful in the interpretation of analyses of data on mixed types.

- One way of avoiding this problem is to do principal component analysis on the correlation matrix rather than the covariance matrix. **R** and S-PLUS allow this option and it is the default option in MINITAB & SPSS. This achieves invariance under linear transformations but loses some interpretability in formal statistical inference though it can still achieve the same reduction in dimensionality. This is especially useful if the variables have widely different variances even if they are all of the same type. If one variable has a very large variance then the first principal component will be dominated by this one variable.

2.1.6 Further Example of Interpretation of PCA Coefficients

This data for this example are given in Wright (1954), The interpretation of multivariate systems. In *Statistics and Mathematics in Biology* (Eds. O. Kempthorne, T.A. Bancroft J. W. Gowen and J.L. Lush), 11–33. State university Press, Iowa, USA.

Six bone measurements $x_1,\dots,x_6$ were made on each of 275 white leghorn fowl. These were: x_1 skull length; x_2 skull breadth; x_3 humerus; x_4 ulna; x_5 femur; x_6 tibia (the first two were skull measurements, the third and fourth wing measurements and the last two leg measurements). The table below gives the coefficients of the six principal components calculated from the covariance matrix.

Original	**Principal Components**					
variable	**a_1**	**a_2**	**a_3**	**a_4**	**a_5**	**a_6**
x_1 skull l.	0.35	0.53	0.76	–0.04	0.02	0.00
x_2 skull b.	0.33	0.70	–0.64	0.00	0.00	0.03
x_3 humerus	0.44	–0.19	–0.05	0.53	0.18	0.67
x_4 ulna	0.44	–0.25	0.02	0.48	–0.15	–0.71
x_5 femur	0.44	–0.28	–0.06	–0.50	0.65	–0.13
x_6 tibia	0.44	–0.22	–0.05	–0.48	–0.69	0.17

To interpret these coefficients we 'round' them heavily to either just one digit and ignore values 'close' to zero, giving

Original	**Principal Components**						
variable	$\mathbf{a_1}$	$\mathbf{a_2}$	$\mathbf{a_3}$	$\mathbf{a_4}$	$\mathbf{a_5}$	$\mathbf{a_6}$	
x_1 skull l.	0.4	0.6	0.7	0	0	0	**skull**
x_2 skull b.	0.4	0.6	–0.7	0	0	0	
x_3 humerus	0.4	–0.2	0	0.5	0	0.7	**wing**
x_4 ulna	0.4	–0.2	0	0.5	0	–0.7	
x_5 femur	0.4	–0.2	0	–0.5	0.6	0	**leg**
x_6 tibia	0.4	–0.2	0	–0.5	–0.6	0	

Original	**Principal Components**						
variable	$\mathbf{a_1}$	$\mathbf{a_2}$	$\mathbf{a_3}$	$\mathbf{a_4}$	$\mathbf{a_5}$	$\mathbf{a_6}$	
x_1 skull l.	+	+	+				**skull**
x_2 skull b.	+	+	–				
x_3 humerus	+	–		+		+	**wing**
x_4 ulna	+	–		+		–	
x_5 femur	+	–		–	+		**leg**
x_6 tibia	+	–		–	–		

We can then see that the first component $\mathbf{a_1}$ is proportional to the sum of all the measurements. Large fowl will have all x_i large and so their scores on the first principal component y_1 (=$x'a_1$) will be large, similarly small birds will have low scores of y_1. If we produce a scatter plot using the first p.c. as the horizontal axis then the large birds will appear on the right hand side and small ones on the left. Thus the first p.c. measures ***overall size**.*

The second component is of the form (skull)–(wing & leg) and so high positive scores of y_2 (=$x'a_2$) will come from birds with large heads and small wings and legs. If we plot y_2 against y_1 then the birds with *relatively* small heads for the size of their wings and legs will appear at the bottom of the plot and those with relatively big heads at the top. The second p.c. measures ***overall body shape**.*

The third component is a measure of ***skull shape*** (i.e. skull width *vs* skull length), the fourth is wing size *vs* leg size and so is also a measure of ***body shape*** (but not involving the head). The fifth and sixth are contrasts between upper and lower parts of the wing and leg respectively and so y_5 measures ***leg shape*** and y_6 measures ***wing shape.***

For comparison we see the effect of using the correlation matrix for the principal component analysis instead of the covariance matrix.

	x_1 skull l.	**x_2 skull b.**	**x_3 humerus**	**x_4 ulna**	**x_5 femur**	**x_6 tibia**
x_1 skull l.	1.000	0.505	0.569	0.602	0.621	0.603
x_2 skull b.	0.505	1.000	0.422	0.467	0.482	0.450
x_3 humerus	0.569	0.422	1.000	0.926	0.877	0.878
x_4 ulna	0.602	0.467	0.926	1.000	0.874	0.894
x_5 femur	0.621	0.482	0.877	0.874	1.000	0.937
x_6 tibia	0.603	0.450	0.878	0.894	0.937	1.000

The eigenanalysis of this correlation matrix gives the PCs as:—

Original	**Principal Components**					
variable	**a_1**	**a_2**	**a_3**	**a_4**	**a_5**	**a_6**
x_1 skull l.	-0.35	-0.40	-0.85	-0.05	0.01	0.03
x_2 skull b.	-0.29	-0.81	0.50	-0.02	0.01	0.04
x_3 humerus	-0.44	0.26	0.11	-0.50	0.60	0.33
x_4 ulna	-0.45	0.20	0.10	-0.47	-0.60	-0.41
x_5 femur	-0.45	0.16	0.10	0.50	0.37	-0.58
x_6 tibia	-0.45	0.21	0.07	0.47	-0.38	0.62
eigenvalue	4.46	0.78	0.46	0.17	0.08	0.05

Original	**Principal Components**						
variable	**a_1**	**a_2**	**a_3**	**a_4**	**a_5**	**a_6**	
x_1 skull l.	–	–	–				**skull**
x_2 skull b.	–	–	+				
x_3 humerus	–	+		–	–	+	**wing**
x_4 ulna	–	+		–	+	–	
x_5 femur	–	+		+	+	–	**leg**
x_6 tibia	–	+		+	–	+	

Comparing these results with those from the covariance matrix note that:

- some the + and – signs of the coefficients have swapped (but the signs are arbitrary and a different package might give different signs)

- the overall interpretation of the individual components will be exactly the same, except perhaps the last two

- the sum of the eigenvalues is 6.0 since the trace of the correlation matrix is 6.0 (it is a 6×6 matrix and the diagonal elements are 1.0)

- the first component accounts for almost 75% of the 'variance' of the data — strictly it is 75% of the *standardized data* and not the original data. In fact, if the analysis were based on the covariance matrix then the first p.c. would appear even more dominant — this is typical with measurements of physical sizes. For dimensionality reduction in particular, it is sensible to base assessments on the analysis of the correlation matrix or else look at the proportions of variance explained as a total of all the components ***excluding*** the first.

- the first three components account for 95% of the variance of the standardized data and this might be a satisfactory initial reduction to these three components.

2.1.7 Computation in R

The basic functions for PCA are `prcomp(mydata)` and `princomp(mydata)` where `mydata` is a dataframe. These are generally equivalent but `prcomp()` is generally the preferred one. This is because it will handle cases where the number of variables is larger than the number of observations (`princomp()` fails if this is so) and `prcomp()` is based on the numerically more stable function `svd()` whereas `princomp()` relies on `eigen()`. `princomp()` is provided in **R** for compatibility with **S+.**

`mydata.pca<-princomp(mydata)` puts the principal components of `mydata` in `mydata.pca$loadings` and the ***square roots*** of the eigenvalues in `mydata.pca$sdev` and the scores (i.e. the values rotated onto the principle components) into `mydata.pca$scores`. `plot(mydata.pca)` produces a bar chart of the variances, `plot(mydata.pca$scores[,1:2])`plots the scores on PC 2 against those on PC 1. The basic results (proportions of variance for each principal component etc) can be examined with `summary(mydata.pca)`.

`mydata.pr<-prcomp(mydata)` puts the principal components of `mydata` in `mydata.pr$rotation` and the ***square roots*** of the eigenvalues in `mydata.pr$sdev` and the scores (i.e. the values rotated onto the principle components) into `mydata.pr$x`

`plot(mydata.pr)` produces a bar chart of the variances, `plot(mydata.pr$x[,1:2])`plots the scores on PC 2 against those on PC 1. The basic results (proportions of variance for each principal component etc) can be examined with `summary(mydata.pr)`.

The bar chart produced by `plot(mydata.pca)` or `plot(mydata.pr)` is no substitute for a scree graph of cumulative proportions of variance contained by successive principal components. **R** does not have an inbuilt function to do this but it is simple to write a custom function, see §2.1.7.1 below. This function is on the course web page in script file `scree.R`

Other differences between `prcomp()` and `princomp()`are to perform the analysis on the correlation matrix instead of the default covariance matrix an additional argument needs to be included, either `prcomp(mydata, scale=T)` or `princomp(mydata, cor=T)`. For further details consult the help system.

The examples given below in §2.1.8 and §2.1.10 use `princomp()`. It is suggested that you repeat these using `prcomp()` taking care to change one or two details of the calls involved.

2.1.7.1 R function `screeplot()`

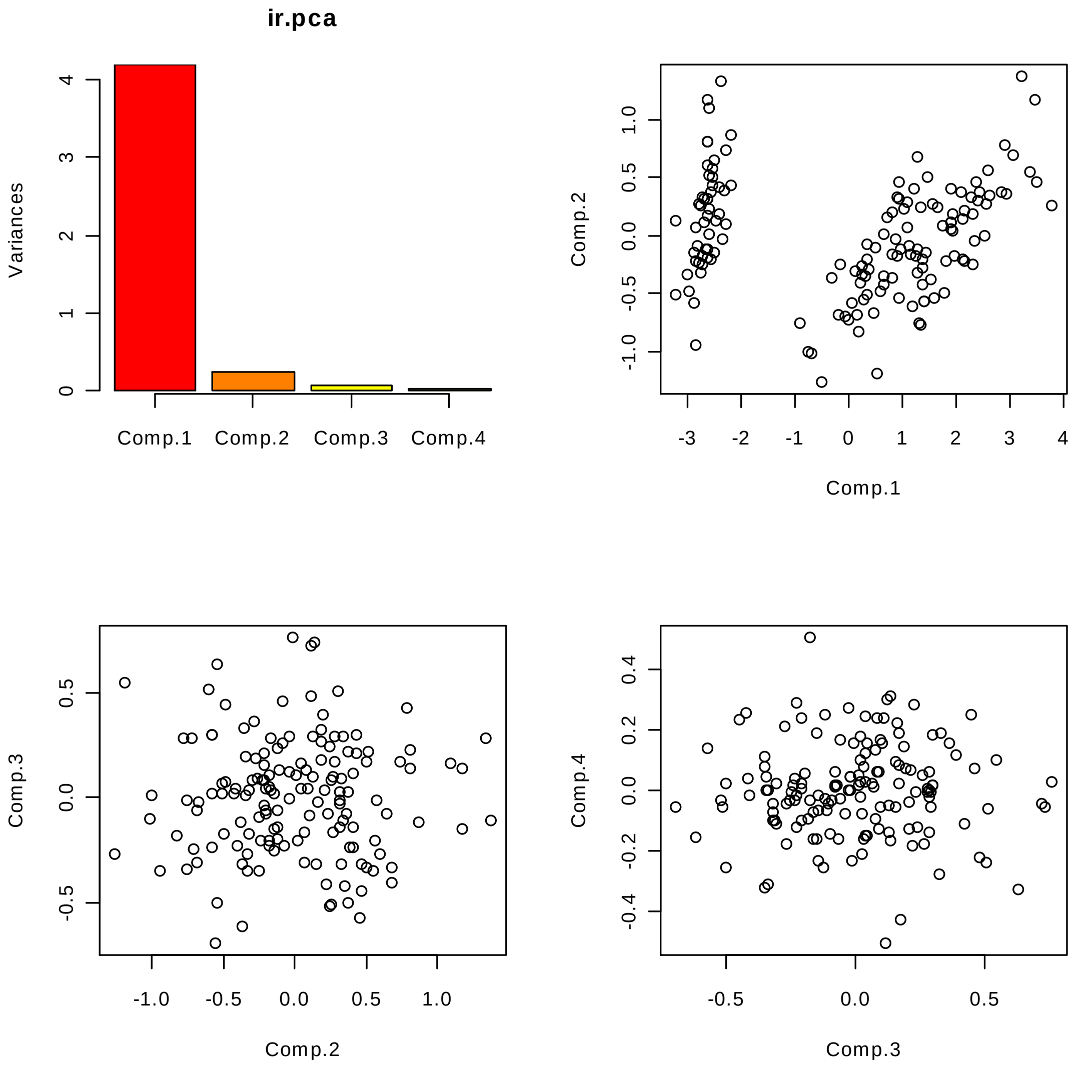
This function produces screeplots of cumulative partial sums of variances on each PC against the number of dimensions. By default the PCs are calculated from the covariance matrix and the number of dimensions used is 10. Both of these can be overridden in the call statement, see the examples. To use in an R session, copy and paste the complete list of commands to a script window and run them. It can be downloaded from the course webpage.

```
screeplot<-function(mydata,cor=F,maxcomp=10) {
my.pc<-prcomp(mydata, scale=cor)
k<-min(dim(mydata),maxcomp)
x<-c(0:k)
y<-my.pc$sdev[1:k]*my.pc$sdev[1:k]
y<-c(0,y)
z<-100*cumsum(y)/sum(my.pc$sdev*my.pc$sdev)

plot(x,z,type="l",xlab="number of dimensions",
      cex.main=1.5, lwd=3, col="red",
      ylim=c(0,100),
      ylab="cumulative percentage of total variance",
      main="Scree plot of variancees",
      xaxt="n", yaxt="n")

axis(1,at=x,lwd=2)
axis(2,at=c(0,20,40,60,80,100),lwd=2)
abline(a=100,b=0,lwd=2,lty="dashed",col="orange")
text(x,z,labels=x,cex=0.8,adj=c(1.2,-.1),col="blue")
}

# Examples of calls to it are
screeplot(mydata) # default uses covariance, maximum 10 components
screeplot(mydata,T) #  uses correlations, maximum 10 components
screeplot(mydata,maxcomp=7) # default use covariance, maximum 7 components
screeplot(mydata,T,maxcomp=8) # use correlations, maximum 8 components
```

2.1.8 Example: Iris data. (Analysis in R)

The following is an example of Principal Component Analysis in the package **R**.

```
> ir<- irisnf[,-5]
> ir.pca<-princomp(ir)
> ir.pca
Call:
princomp(x = ir)

Standard deviations:
   Comp.1    Comp.2    Comp.3    Comp.4 
2.0494032 0.4909714 0.2787259 0.1538707 

 4  variables and  150 observations.
> summary(ir.pca)
Importance of components:
                          Comp.1     Comp.2     Comp.3      Comp.4
Standard deviation     2.0494032 0.49097143 0.27872586 0.153870700
Proportion of Variance 0.9246187 0.05306648 0.01710261 0.005212184
Cumulative Proportion  0.9246187 0.97768521 0.99478782 1.000000000
> plot(ir.pca)
> par(mfrow=c(2,2))
> plot(ir.pca)
> loadings(ir.pca)
                  Comp.1      Comp.2      Comp.3     Comp.4
Sepal.Length  0.36138659  0.65658877 -0.58202985 -0.3154872
Sepal.Width  -0.08452251  0.73016143  0.59791083  0.3197231
Petal.Length  0.85667061 -0.17337266  0.07623608  0.4798390
Petal.Width   0.35828920 -0.07548102  0.54583143 -0.7536574
> ir.pc<- predict(ir.pca)
> plot(ir.pc[,1:2])
> plot(ir.pc[,2:3])
> plot(ir.pc[,3:4])
>
```



Iris data PCA

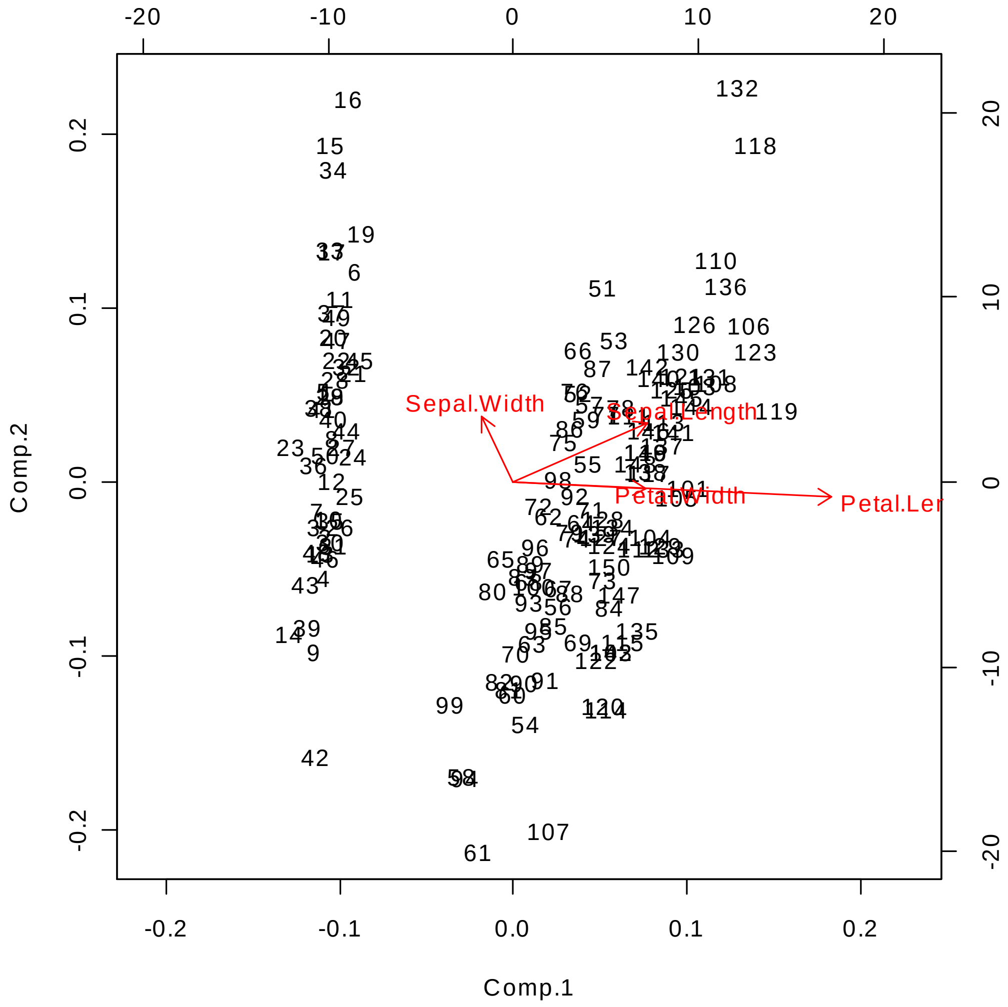
We see that the first PC contrast flowers with big petals and long sepals with those with small petals and short sepals, i.e. big flowers with small flowers. The second contrasts big sepals with small sepals.

The first scatter plot contains 98% of the variation in the data. It be seen that much of this variation is because the data separate into [at least] two groups along the first PC.

2.1.9 Biplots

It is possible to represent the original variables on a plot of the data on the first two principal components. The variables are represented as arrows, with lengths proportional to the standard deviations and angles between a pair of arrows proportional to the covariances. The orientation of the arrows on the plot relates to the correlations between the variables and the principal components and so can be an aid to interpretation.

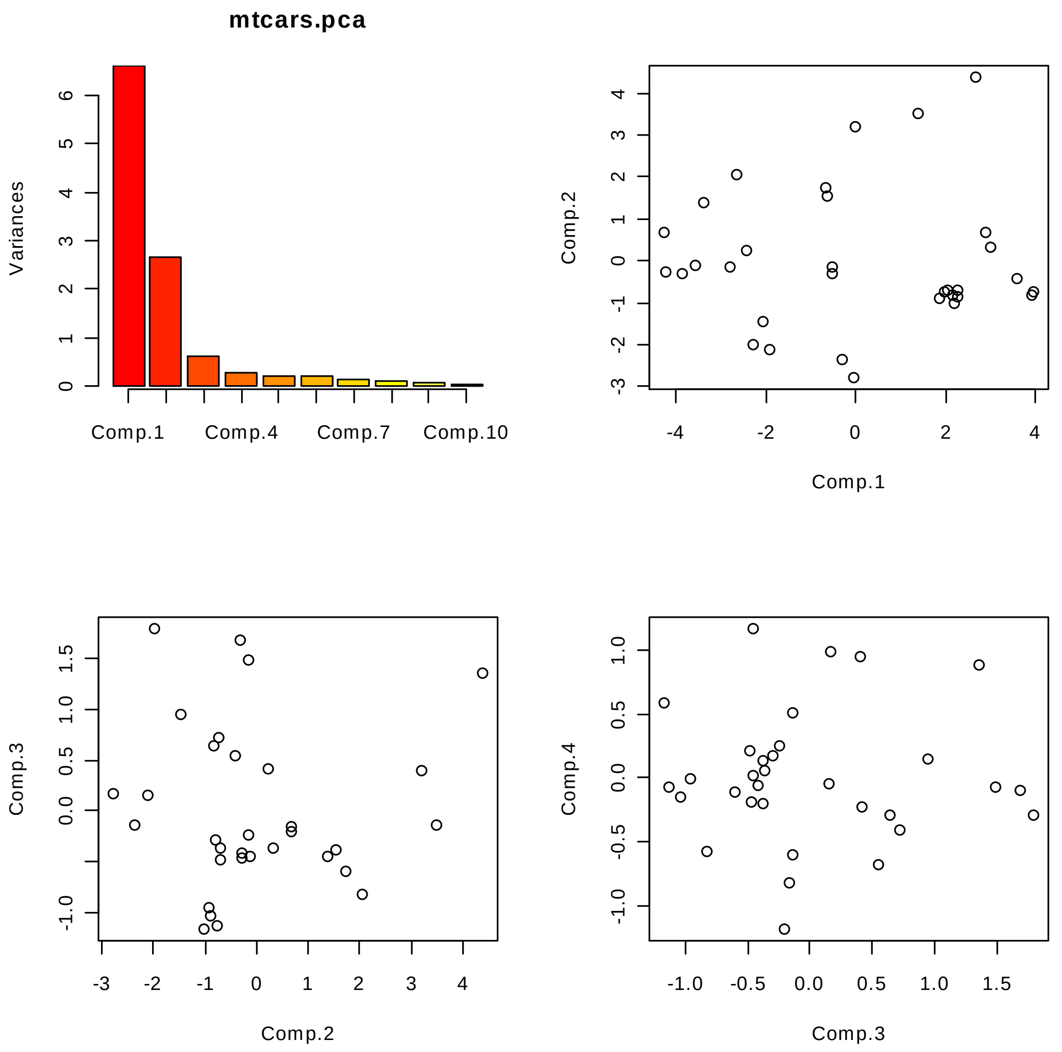
```
> biplot(ir.pca)
```



2.1.10 Cars Example Again, Using Correlation Matrix

(Analysis in R)

```
> mtcars.pca<-princomp(mtcars,cor=T)
> mtcars.pca
Call:
princomp(x = mtcars, cor = T)

Standard deviations:
   Comp.1    Comp.2    Comp.3    Comp.4    Comp.5    Comp.6    Comp.7    Comp.8
2.5706809 1.6280258 0.7919579 0.5192277 0.4727061 0.4599958 0.3677798 0.3505730
   Comp.9   Comp.10   Comp.11
0.2775728 0.2281128 0.1484736

 11  variables and  32 observations.
> summary(mtcars.pca)
Importance of components:
                          Comp.1    Comp.2     Comp.3     Comp.4     Comp.5
Standard deviation     2.5706809 1.6280258 0.79195787 0.51922773 0.47270615
Proportion of Variance 0.6007637 0.2409516 0.05701793 0.02450886 0.02031374
Cumulative Proportion  0.6007637 0.8417153 0.89873322 0.92324208 0.94355581
                           Comp.6     Comp.7     Comp.8      Comp.9     Comp.10
Standard deviation     0.45999578 0.36777981 0.35057301 0.277572792 0.228112781
Proportion of Variance 0.01923601 0.01229654 0.01117286 0.007004241 0.004730495
Cumulative Proportion  0.96279183 0.97508837 0.98626123 0.993265468 0.997995963
                           Comp.11
Standard deviation     0.148473587
Proportion of Variance 0.002004037
Cumulative Proportion  1.000000000
> mtcars.pc<-predict(mtcars.pca)
> plot(mtcars.pca)
> plot(mtcars.pc[,1:2])
> plot(mtcars.pc[,2:3])
> plot(mtcars.pc[,3:4])
```


mtcars.pca
Variances
0
1
2
3
4
5
6
Comp.1
Comp.4
Comp.7
Comp.10
Comp.2
Comp.1
-4
-2
0
2
4
-3
-2
-1
0
1
2
3
4
Comp.3
Comp.2
-3
-2
-1
0
1
2
3
4
-1.0
0.0
0.5
1.0
1.5
Comp.4
Comp.3
-1.0
-0.5
0.0
0.5
1.0
1.5

Car Data PCA

```
> loadings(mtcars.pca)
         Comp.1      Comp.2      Comp.3
mpg      -0.36        0.01       -0.22
cyl       0.37        0.04       -0.17
disp      0.36       -0.04       -0.06
hp        0.33        0.24        0.14
drat     -0.29        0.27        0.16
wt        0.34       -0.14        0.34
qsec     -0.20       -0.46        0.40
vs       -0.30       -0.23        0.42
am       -0.23        0.42       -0.20
gear     -0.20        0.46        0.28
carb      0.21        0.41        0.52
```

```
data(mtcars)

Format:

 A data frame with 32 observations on 11 variables.

  [, 1] mpg  Miles/(US) gallon
  [, 2] cyl  Number of cylinders
  [, 3] disp Displacement (cu.in.)
  [, 4] hp   Gross horsepower
  [, 5] drat Rear axle ratio
  [, 6] wt   Weight (lb/1000)
  [, 7] qsec 1/4 mile time
  [, 8] vs   V/S
  [, 9] am   Transmission (0 = automatic, 1 = manual)
  [,10] gear Number of forward gears
  [,11] carb Number of carburettors
```

```
                  Comp.1        Comp.2        Comp.3
 Miles/(US)        -0.36          0.01         -0.22
 cylinders          0.37          0.04         -0.17
 Displacement       0.36         -0.04         -0.06
 horsepower        0.33          0.24          0.14
 Rear axle ratio   -0.29          0.27          0.16
 Weight             0.34         -0.14          0.34
 ¼ mile time       -0.20         -0.46          0.40
 V/S               -0.30         -0.23          0.42
 Transmission      -0.23          0.42         -0.20
 gears             -0.20          0.46          0.28
 carburettors       0.21          0.41          0.52
```

Interpretation of loadings:

Comp.1 : All coefficients are of comparable size. Positive ones are mainly properties of the car itself (with high values implying high performance), negative ones are measures of actual performance of car. Interpretation: contrast between predicted and actual performance.

Comp.2: largest coefficients are Transmission, gears and carburettors contrasted with ¼ mile time. High scores on this are by cars with a fast ¼ mile time and powerful transmission, i.e. overall power.

Comp.3: (less clear) but highs scores on this component from large heavy cars that do few miles to the gallon and have slow ¼ mile speeds.

The first scatter plot (85% of total variation) reveals some clusters of cars (i.e. they have similar performances etc) as well as several outliers. The second scatterplot (about 30% of total variation) indicates a few outliers. It would be of interest to see how these two plots link together and this can be done interactively with the brush facility in **R,** Minitab or S-Plus. In **R** use `panel.brush.splom(.)` in the `lattice` package.

2.1.11 Notes

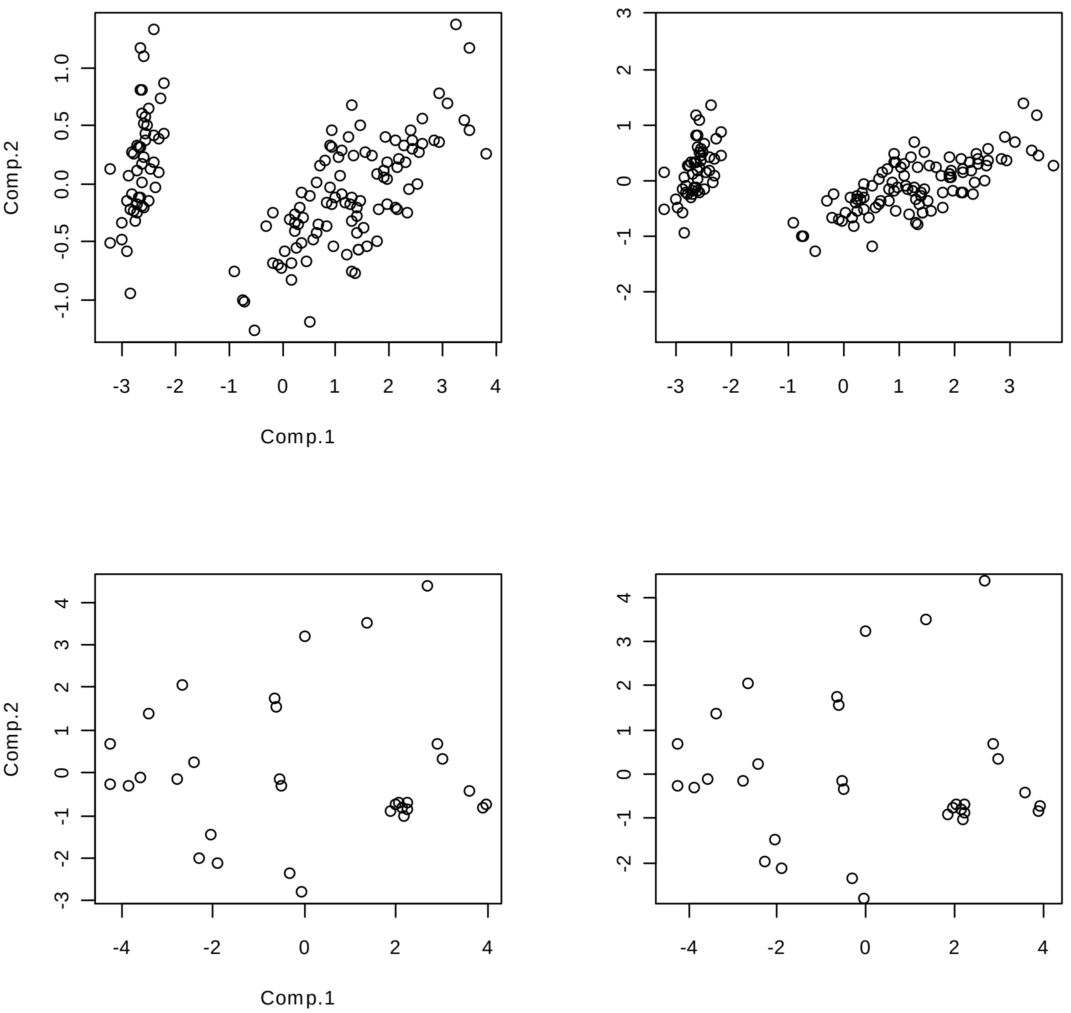
- ♦ The full mathematical/algebraic theory of principal component analysis strictly applies **ONLY** to continuous data on comparable scales of measurements using the covariance matrix. Using the correlation matrix brings the measurements onto a common scale but a little care is needed in interpretation, especially in interpretation of the loadings.

- ♦ The above example has a mixture of continuous and discrete variables on completely different scales of measurements. 'Experience' shows that if you include a few discrete variables with several continuous ones (or if you have a large number of discrete ones without any continuous ones) then principal component analysis based upon the correlation matrix ***'WORKS'***, i.e. you obtain interpretable answers.

- ♦ Strictly, categorical variables with several levels should be recoded into binary dummy variables though in the example above the obvious categorical variables (`gears`, `cylinders` and `carburettors`) are effectively binary (i.e. `gears` either 3 or 4, `cylinders` either 4 or 6, `carburettors` either 1 or 2)

- ♦ Since the key objective of pca is to extract information, i.e. *partition the internal variation* it is sensible to plot the data using equal scaling on the axes. In MINITAB this is cumbersome but can be done using the Min and Max options in Frame of Plot. This is easy in **R** or S-PLUS using the `MASS` function `eqscplot()`:

```
> plot(ir.pc[,1:2])
> eqscplot(ir.pc[,1:2])
> plot(mtcars.pc[,1:2])
> eqscplot(mtcars.pc[,1:2])
```



Note (for **R** and S-PLUS users only) that unlike `plot()` the axes are not automatically labelled by `eqscplot()` and you need to do this by including

```
,xlab="first p.c.",ylab="second p.c."
```

in the call to it.

2.1.12 Miscellaneous comments

- ***supplementary data*** (i.e. further data not included in calculating the principal component transformation but measured on the same variables) can be displayed on a PCA plot (i.e. apply the same transformation to the new data). Idea is particularly of use in related techniques such as discriminant and correspondence analysis. This can be handled in **R** with the function `predict()`, see the help system for details.

- numerical interpretation of loadings is only viable with a small number of original variables. In many modern examples with several hundred examples other techniques have to be used, e.g. plot loadings against variable number (assuming that there is some structure to the variable numbering, e.g. digitising spectra). This may reveal, for example, that the first few PCs are combinations from the middle of the spectra, next few from one third the way along,…. .

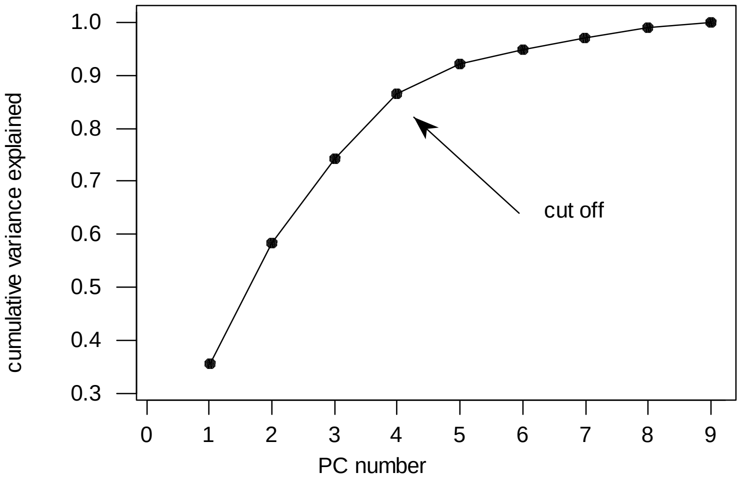
- PCA is obtained by choosing projections to maximize **variances** of projected data. Choosing a different criterion can give 'interesting' views of data :— ***projection pursuit methods***, e.g. maximize projected kurtosis. (See e.g. Venables & Ripley)

- **OUTLIERS**: controversy of which PCs to use for outlier hunting (first few or last few?). **Suggestion**: also look at ***cut-off*** PCs — e.g. plot of 'last interesting PC' *vs* 'first uninteresting one'. However, if searching for outliers then there are better ways of finding them (*outlier displaying components)*.

2.1.13 PCA and Outliers

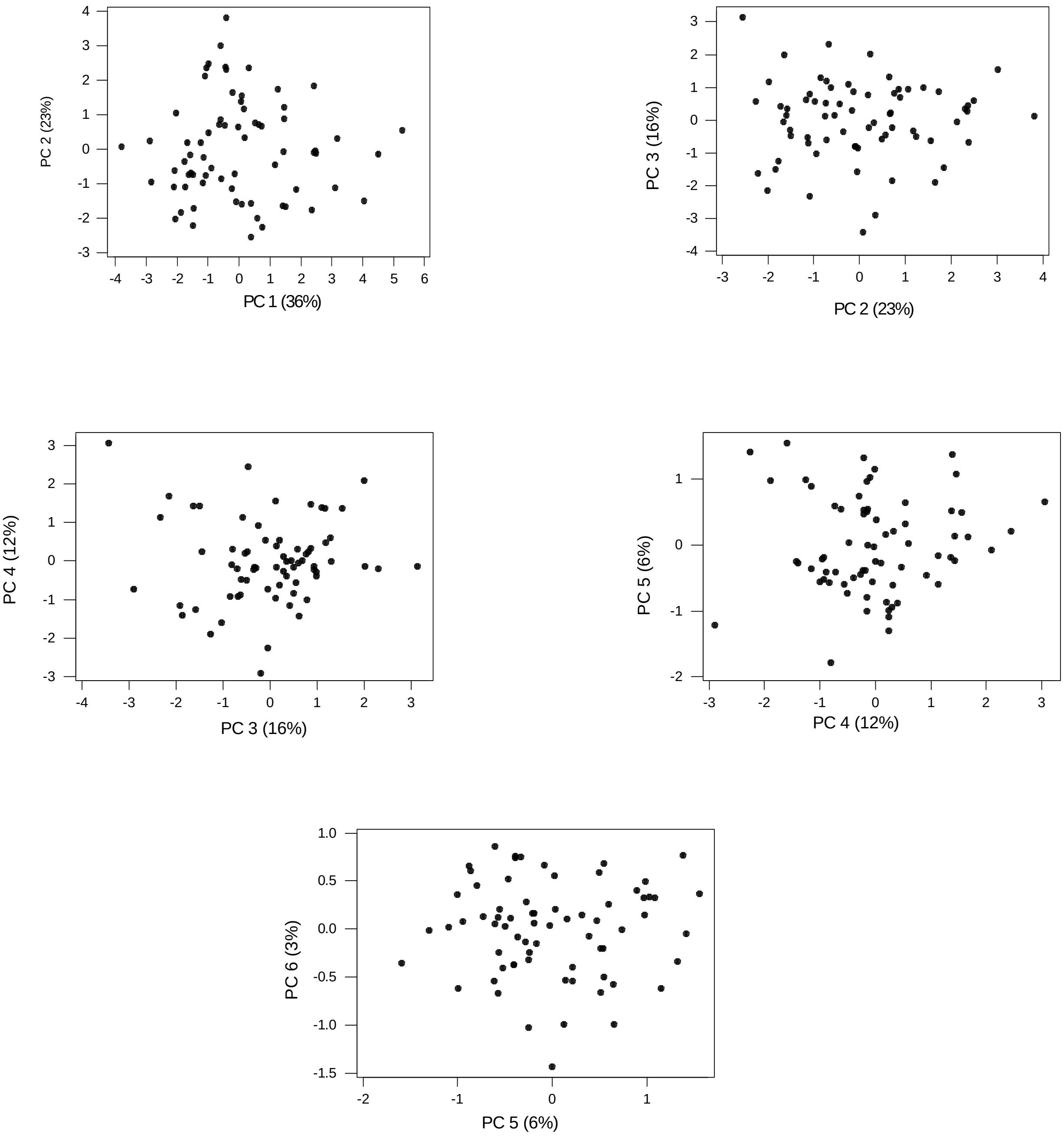
Example: (data extracted from claypots data). Data are concentrations of 9 trace elements in ancient Greek pottery (part of a provenance study). PCA performed on correlation matrix because of widely different standard deviations/scales of measurement.

```
Eigenvalue    3.2170    2.0423    1.4280    1.0995    0.5089    0.2426
Proportion     0.357     0.227     0.159     0.122     0.057     0.027
Cumulative     0.357     0.584     0.743     0.865     0.922     0.949

Eigenvalue    0.1841    0.1761    0.1015
Proportion     0.020     0.020     0.011
Cumulative     0.969     0.989     1.000
```



Scree plot

Scree plot identifies component 4 as cut-off PC — suggests looking at scatter plots of PC4 *vs* PC5 (or *vs* PC3): examination of these plots does shew one or two possible outliers.


PC 2 (23%)
4
3
2
1
0
-1
-2
-3
-4
-3
-2
-1
0
1
2
3
4
5
6
PC 1 (36%)
PC 3 (16%)
3
2
1
0
-1
-2
-3
-4
-3
-2
-1
0
1
2
3
4
PC 2 (23%)
PC 4 (12%)
3
2
1
0
-1
-2
-3
-4
-3
-2
-1
0
1
2
3
PC 3 (16%)
PC 5 (6%)
1
0
-1
-2
-3
-2
-1
0
1
2
3
PC 4 (12%)
PC 6 (3%)
1.0
0.5
0.0
-0.5
-1.0
-1.5
-2
-1
0
1
PC 5 (6%)

2.1.14 Summary of Principal Component Analysis

- Technique for transforming original variables into new ones which are uncorrelated and account for decreasing proportions of variance in the data.
- New variables are ***linear*** combinations of old ones
- Transformation is a rotation/reflection of original points

  ⇒ no essential statistical information is lost (or 'created')
- Can assess the importance of individual new components and assess 'how many are needed' (—scree plots etc)
- Scatterplots of data on first few components contain almost all information so may reveal features such as group structure, outliers, ……
- Can assess the importance of original variables

  (examination of *loadings*)
- It is typical that the first p.c. reflects overall size or level i.e. objects often vary most in overall size or there may be large variation in base level of response (e.g. in ratings data from questionnaires). Suggestion: consider ignoring the first p.c. if it only reveals the obvious (i.e. look at proportions of variance explained excluding the first pc)
- Many other techniques follow **'by analogy'** even though they may not have all the strict mathematical theory underlying them:

  i.e. rotate/transform original data to new variables,

  assess importance of new variables,

  interpret loadings of old variables:

  (includes aspects of projection pursuit, discriminant analysis, canonical correlation analysis, correspondence analysis,……)

Tasks 3

(see §2.1–§2.5 & A0.1)

Note and MEMORIZE the interesting identity

$|I_p + AB| = |I_n + BA|$ where A is p×n and B is n×p. This identity is surprisingly difficult to prove directly so a proof is not given here, however it is extremely useful and will be utilized several times later. The examples below illustrate how effective it can be. In particular, taking the case n=1, if $A=1_p$ and $B=1'_p$ then A is p×1 so BA=p, $I_n=1$ and $|I_n + BA|$ is a scalar and =(1+p). So, noting that $J_p = 1_p1'_p$ (the p×p matrix with all entries = 1) we have that $|I_p + J_p|=(p+1)$. More generally, $|\alpha I_p + \beta J_p| = \alpha^p\,|I_p + (\beta/\alpha)J_p| = \alpha^p(1 + (\beta/\alpha)p) = \alpha^{p-1}(\alpha + \beta p)$

1) Suppose the variance matrix takes the equicorrelation form $\underset{p\times p}{S} = \sigma^2\begin{pmatrix} 1 & \rho & \rho & \cdots & \rho \\ \rho & 1 & \rho & \cdots & \rho \\ \vdots & \vdots & \ddots & \cdots & \vdots \\ \rho & & & \ddots & \rho \\ \rho & \rho & \cdots & \rho & 1 \end{pmatrix}$. By writing $S-\lambda I_p$ in the form $aI_p+b1_p1_p'$ for appropriate a and b and using the result above, shew that if $\rho>0$ then the first principal component accounts for a proportion $(1+\rho(p-1))/p$ of the total variation in the data. What can be said about the other p–1 components? What can be said if $\rho<0$? (but note that necessarily $\rho >$ some constant bigger than –1 which you should determine, noting that S is a correlation matrix).

2) If the variance matrix takes the form $S=\alpha I_p+\beta zz'$ where z is a p-vector, shew that z is an eigenvector of S. Under what circumstances is Z proportional to the *first* principal component of the data?

3) If the variance matrix takes the form (with $\alpha>0$) $S=\begin{pmatrix} 1+\alpha & 1 & \beta \\ 1 & 1+\alpha & \beta \\ \beta & \beta & \alpha+\beta^2 \end{pmatrix}$ find the first principal component and shew that it accounts for a proportion $(\beta^2+\alpha+2)/(\beta^2+3\alpha+2)$ of the total variation. Note that this is similar to the example above with $z=(1,1,\beta)'$

4) Referring to Q3 on Task Sheet 2, examination results in five mathematical papers, some of which were 'open-book' and others 'closed-book', what interpretations can you give to the principal components?

2.2★ Factor Analysis

Factor Analysis (and Latent Variable Analysis) are procedures closely related to PCA but starting from a different 'philosophy'. The idea is that the observed correlations between p variables are the result of their mutual dependence on q factors or latent variables. So the factor model is

$$\mathbf{x=\Lambda f+\varepsilon}$$

where the observations x are of dimension p, Λ is a p×q matrix of factor loadings, and the underlying [unobservable] factors f are of dimension q.

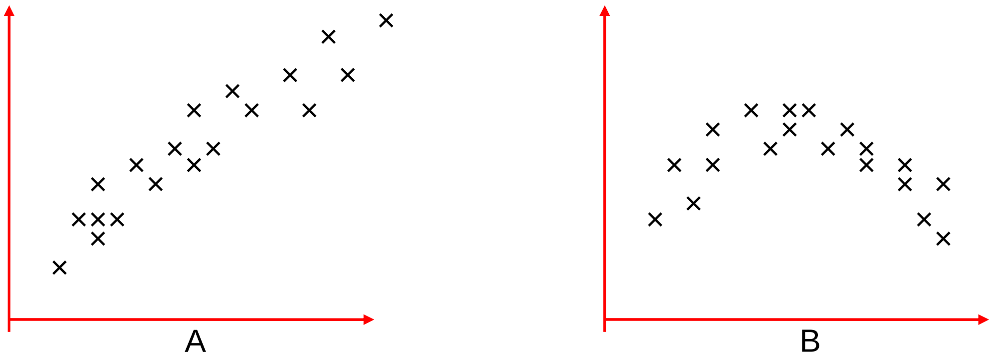
The desire is to determine both f and q. Assumptions may be made on the errors ε (e.g. iid Normal).

This model is inherently unidentifiable — even if q is specified then f is determinable only up to orthogonal transformations (i.e. only the factor space can be determined). Thus one commonly used procedure is to start with a PCA (to determine a suitable q, e.g. with the aid of a scree plot) and then apply orthogonal transformations to the PCs (i.e. rotations) to identify 'factors' which are interpretable. One commonly used procedure is to use *varimax* rotation which aims to maximize the variance (or some other function, e.g. *quartimax*) of the squared factor loadings. The result of doing this is that some factor loadings are 'small' (i.e. negligible) and others are large, thus making the factors easy to interpret.

2.3★ Non-Linear Techniques

2.3.1 Generalized Principal Components

Linear principal component analysis attempts to find a linear coordinate system which is in concordance with the data configuration; it is particularly useful if the data contain a linear singularity or near linear singularity (fig A).



If the data contain a non-linear singularity (fig B) then linear principal component analysis may fail to find it and a more general technique is required. The technique is similar to polynomial regression and its relationship with multiple regression.

e.g. case p=2 and quadratic principal components.

datum $x=(x_1,x_2)'$

— first step is to find $z=ax_1+bx_2+cx_1^2+dx_1x_2+ex_2^2$

such that the variance of z is maximal among all such quadratic functions of x_1, x_2.

Let $x_3=x_1^2$, $x_4=x_1x_2$, and $x_5=x_2^2$

If $x^*=(x_1,x_2,...,x_5)'$ and $a^*=(a,b,...,e)'$ then the problem is to maximize $\text{var}(a^{*\prime}x^*)$ just as in the linear case.

For general p, augment the original p variables by p+½p(p–1) derived ones and perform PCA on the new set of 2p+½p(p–1) variables. This is only practical for small p since you need to have more than p+½p(p–1) negligible eigenvalues for the dimensionality of the original problem to be reduced.

It would in principle be possible to define new augmenting variables such as sin(x_i) or more complicated functions just as with multiple regression. However, determining which such functions are appropriate is not easy, unlike multiple regression with simple plotting of residuals etc available, and it is not a routine technique except in special cases where there is some theoretical suggestion from the nature of the problem. For example, in a study on dimensions of items of timber with circumference of trunk and length of trunk both included as measurements there may be some reason for thinking that the volume of the timber may be roughly constant, so including a new augmenting variable defined as length×circumference^2 could be useful. However, such situations are rare in practice.

2.4 Summary

- ♦ This section has introduced a powerful method for optimisation by introducing a constraint and Lagrange multipliers followed by identification of eigen equations. This technique is used in many other contexts.

- ♦ Principal Component Analysis is the most useful technique for exploratory analysis of multivariate data. It can reveal unsuspected structure (subgroups, outliers etc) as well as giving interpretation of what ‘causes’ the variability (by interpretation of loadings).

- ♦ Many techniques discussed later are interpreted ***by analogy*** with the techniques of PCA, even though strictly the mathematical underpinning is weaker.

- ♦ Other techniques of biplots and factor analysis were introduced.

- ♦ Generalizations to include searches for non-linear structures were introduced but the drawbacks were highlighted

- ♦ Related techniques to PCA are Correspondence Analysis (for contingency tables of frequency data) and Outlier Displaying Components (See Appendices 3 & 4).

Tasks 4

(see §2.1)

1) Suppose X′ (n×p) is a centred data matrix (i.e. each variable has sample mean zero). Then the variance matrix S is given by

$$(n–1)S=XX'$$

Suppose λ_i and a_i are the eigenvalues and eigenvectors of XX′.

   a) What are the eigenvalues and eigenvectors of S?

   b) Shew that the eigenvalues and eigenvectors of the n×n matrix X′X are λ_i and $X'a_i$ respectively.

2) Recently, measurements were made on a total of 26 mummy-pots (which contained mummified birds) excavated from the Sacred Animal Necropolis in Saqqara, Egypt and sent to the British Museum in the last century, see figures 2(a) -2(d). The pots are approximately cylindrical, tapering slightly from the opening. The measurements made (in millimetres) were the overall length, the rim circumference and the base circumference (see Fig 1). Given below is a record of an **R** session analyzing the data.

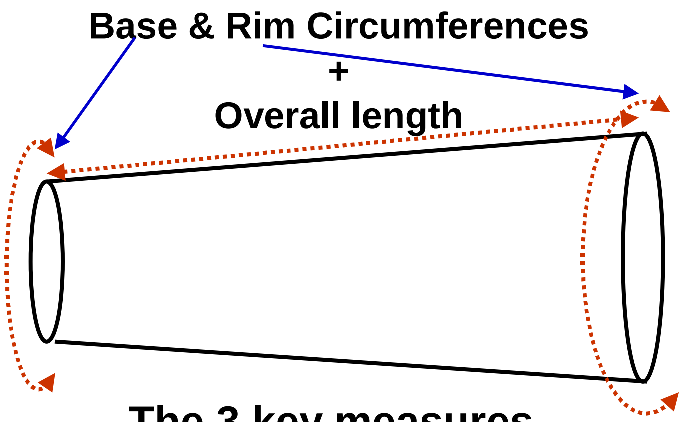
   a) What aspects of the pots do the two derived measurements stored in taper and point reflect?

   b) Principal component analyses have been performed on the correlation matrix of all five variables but on the covariance matrix for just the three linear measurements. Why are these choices to be preferred for these data?

   c) What features of the pots do the first two principal components in each analysis reflect?

Analysis of British Museum Mummy-Pots

```
> attach(brmuseum)
> taper<-(rim.cir-base.circ)/length
> point<-rim.cir/base.circ
> potsize<-cbind(length,rim.cir,base.circ)
> potsize.pca<-princomp(potsize)
> summary(potsize.pca)
Importance of components:
                           Comp.1     Comp.2     Comp.3
    Standard deviation 59.8359424 22.6695236 18.8889569
Proportion of Variance  0.8043828  0.1154578  0.0801594
 Cumulative Proportion  0.8043828  0.9198406  1.0000000

> loadings(potsize.pca)
          Comp.1 Comp.2 Comp.3
   length  0.502 -0.694  0.516
  rim.cir  0.836  0.237 -0.494
base.circ  0.221  0.680  0.699
> potsizeshape<-
cbind(length,rim.cir,base.circ,taper,point)
> potsizeshape.pca<-princomp(potsizeshape, cor=TRUE)
> summary(potsizeshape.pca)
Importance of components:
                          Comp.1    Comp.2    Comp.3
    Standard deviation 1.6082075 1.3352046 0.7908253
Proportion of Variance 0.5172663 0.3565543 0.1250809
 Cumulative Proportion 0.5172663 0.8738206 0.9989015

                             Comp.4      Comp.5
    Standard deviation 0.0698805556 0.024681162
Proportion of Variance 0.0009766584 0.000121832
 Cumulative Proportion 0.9998781680 1.000000000
> loadings(potsizeshape.pca)
          Comp.1 Comp.2 Comp.3 Comp.4 Comp.5
   length  0.428 -0.366 -0.678 -0.316  0.352
  rim.cir  0.548 -0.332  0.207 -0.129 -0.728
base.circ        -0.715  0.371  0.466  0.365
    taper  0.498  0.302  0.561 -0.367  0.461
    point  0.519  0.392 -0.212  0.729
```



The 3 key measures

Fig 1: Definition of

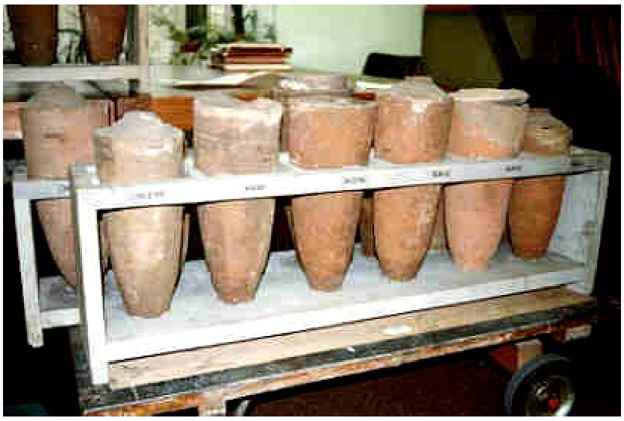


Fig 2a: Mummy-pots in British Museum

**Fig 2b: Measuring mummy-pots in BM
(Dr Julie Hopkins, 25/04/97)**

**Fig 2c: Measuring mummy-pots in BM
(Dr Julie Hopkins, 25/04/97)**

**Fig 2d: Measuring mummy-pots in BM
(Dr Julie Hopkins & Dr Paul Nicholson, 25/04/97)**

Exercises 1

1) Dataset `nfl2000.Rdata`★ gives performance statistics for 31 teams in the US National Football League for the year 2000. Twelve measures of performance were made, six include the syllable `home` in the variable name and six include the syllable `opp`. The measures of performance were

`homedrives50`	drives begun in opponents' territory
`homedrives20`	drives begun within 20 yards of the goal
`oppdrives50`	opponents drives begun in team's territory
`oppdrives20`	opponents drives begun within 20 yards of goal
`hometouch`	touchdowns scored by team
`opptouch`	touchdowns scored against team
`homeyards`	total yardage gained by offence
`oppyards`	total yardage allowed by defence
`hometop`	time of possession by offence (in minutes)
`opptop`	time of possession by opponents' offence
`home1sts`	first downs obtained by offence
`opp1sts`	first downs allowed by defence

The dataset contains a three letter abbreviation for the team as a row name. The coding is

initials	**team**	**initials**	**team**
ARI	Arizona Cardinals	**BAL**	Baltimore Ravens
ATL	Atlanta Falcons	**BUF**	Buffalo Bills
CAR	Carolina Panthers	**CIN**	Cincinnati Bengals
CHI	Chicago Bears	**CLE**	Cleveland Browns
DAL	Dallas Cowboys	**DEN**	Denver Broncos
DET	Detroit Lions	**IND**	Indianapolis Colts
GB	Green Bay Packers	**JAX**	Jacksonville Jaguars
MIN	Minnesota Vikings	**KC**	Kansas City Chiefs
NO	New Orleans Saints	**MIA**	Miami Dolphins
NYG	New York Giants	**NE**	New England Patriots
PHI	Philadelphia Eagles	**NYJ**	New York Jets
SF	San Francisco 49ers	**OAK**	Oakland Raiders
STL	St. Louis Rams	**PIT**	Pittsburgh Steelers
TB	Tampa Bay Buccaneers	**SD**	San Diego Chargers
WAS	Washington Redskins	**SEA**	Seattle Seahawks
		TEN	Tennessee Titans

i) Do the syllables `home` and `opp` most probably refer to when the team was playing at *home* and playing *away* or do the refer to events *by* the team and *against* the team?

ii) Use principal component analysis to identify and describe the main sources of variation of the performances.

iii) Produce a scatter plot of the teams referred to their principal component scores and comment on any features you think worthy of mention.

(**NB:** *You are strongly advised to work through Task Sheet 2, Q3 if you have not already done so*).

⋆source: *Journal of Statistics Education* Data Archive

2) Measurements of various chemical properties were made on 43 samples of soil taken from areas close to motorway bridges suffering from corrosion. The corrosion can be of either of two types and the ultimate aim of the investigation was to see whether these measurements could be used to discriminate between the two types. Before such a full-scale analysis was undertaken some preliminary analyses were performed, using MINITAB. The record of the session (edited in places) is given below.

(a) The principal component analysis has been performed on the correlation matrix rather than the covariance matrix. Why is this to be preferred for these data?

(b) By using some suitable informal graphical technique, how may components would you recommend using in subsequent analyses?

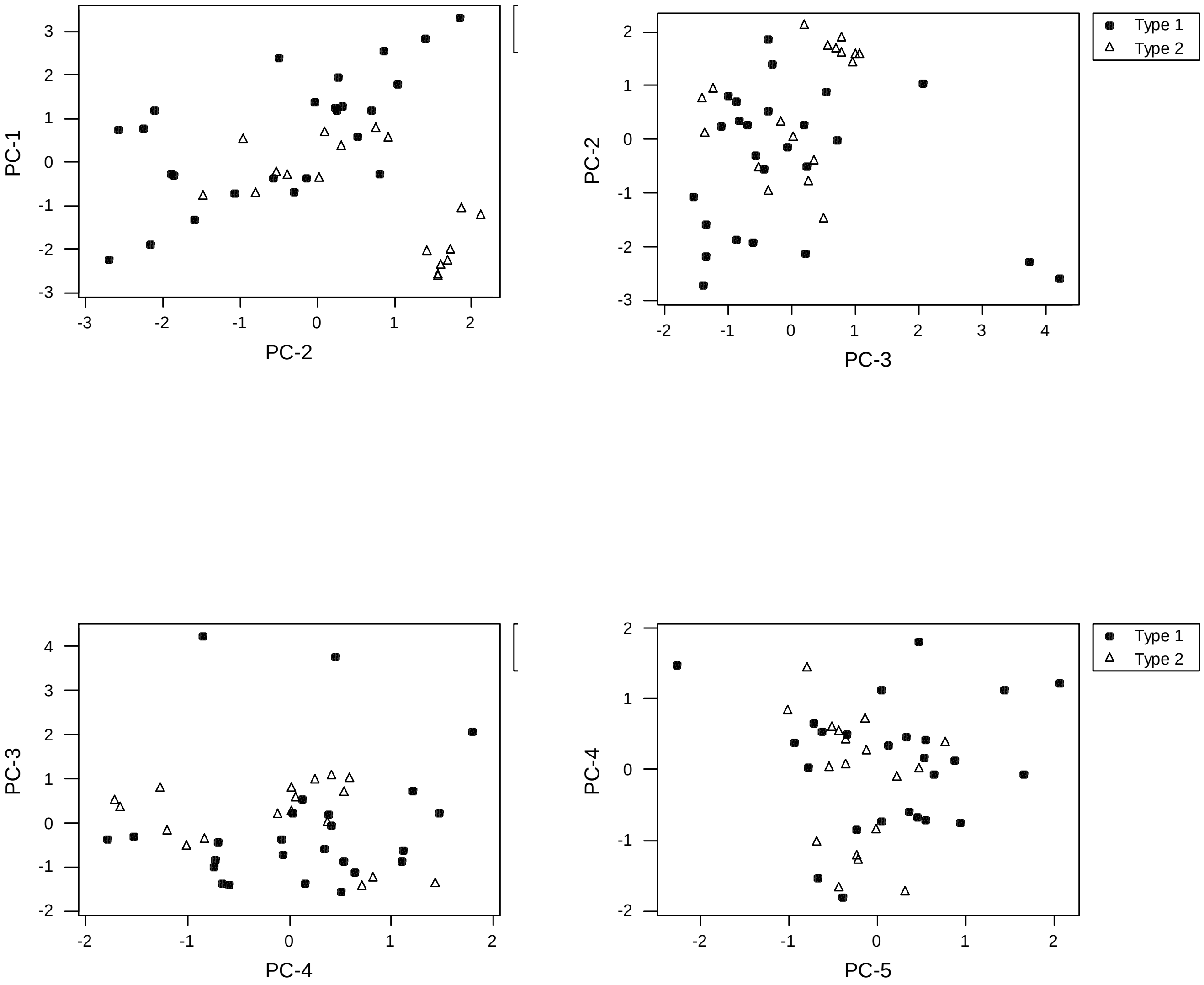
(c) What features of the samples do the first three components reflect?

(d) What, approximately, is the values of the sample correlation between the scores of PC-1 and PC-2?

(e) After looking at the various scatter plots of the principal component scores, what recommendation would you give to the investigator regarding the advisability of continuing with a discriminant analysis?

```
Worksheet size: 100000 cells
MTB > Retrieve  "C:\soil.MTW".

MTB > desc c2-c9;
SUBC> by c1.

Descriptive Statistics
Variable   Type        N       Mean     StDev
pH         Type 1     25      8.416     0.962
           Type 2     18     8.0722    0.3102
Water      Type 1     25      1.693     0.716
           Type 2     18      2.831     1.812
Acid       Type 1     25     0.5672    0.3937
           Type 2     18     0.4322    0.2603
Pyrite     Type 1     25     0.4628    0.2563
           Type 2     18      1.019     0.500
Carbon     Type 1     25     11.251     4.230
           Type 2     18      9.783     1.862
Moisture   Type 1     25     23.712     4.975
           Type 2     18     21.922     2.647
Organic    Type 1     25      2.556     0.720
           Type 2     18      2.272     0.530
MassLos    Type 1     25      5.536     1.575
           Type 2     18      6.833     0.807


MTB > PCA  'pH'-'MassLos';
SUBC>   Coefficients c31-c38;
SUBC>   Scores'PC-1'-'PC-8'.
Principal Component Analysis
Eigenanalysis of the Correlation Matrix
Eigenvalue   2.351   1.862   1.504  0.827  0.612  0.412  0.230  0.197
Proportion   0.294   0.233   0.188  0.103  0.077  0.052  0.029  0.025
Cumulative   0.294   0.527   0.715  0.818  0.895  0.947  0.975  1.000

Variable      PC1     PC2     PC3     PC4     PC5     PC6     PC7     PC8
pH          0.348  -0.032   0.559   0.267  -0.126   0.599   0.334   0.095
Water      -0.455   0.270   0.339   0.219   0.042  -0.460   0.520   0.272
Acid       -0.002  -0.367   0.622  -0.053   0.347  -0.238  -0.520   0.168
Pyrite     -0.351   0.446   0.157   0.417  -0.344   0.157  -0.539  -0.214
Carbon      0.520   0.291  -0.077   0.022  -0.355  -0.285  -0.206   0.624
Moisture   -0.001  -0.582   0.068   0.148  -0.687  -0.318   0.090  -0.231
Organic    -0.204  -0.392  -0.387   0.616   0.181   0.188  -0.067   0.450
MassLos    -0.487  -0.118   0.048  -0.549  -0.336   0.363  -0.049   0.445

MTB > Plot 'PC-1'*'PC-2' 'PC-2'*'PC-3' 'PC-3'*'PC-4''PC-4'*'PC-5';
SUBC>   Symbol 'Type';
SUBC>     Type 6 19;
SUBC>     Size 1.0 1.5;
SUBC>   ScFrame;
SUBC>   ScAnnotation.
```

```
MTB > STOP
```



(This question is taken from the PAS370 1999/2000 examination)

3) ⋆⋆⋆ Suppose X={x_{ij} ; i=1,...,p, j=1,...,n} is a set of n observations in p dimensions with $\sum_{j=1}^{n} x_{ij} = 0$ all i=1,...,p (i.e. each of the p variables has zero mean, so $\bar{x} = 0$) and S=XX′/(n-1) is the sample variance of the data. Let $u_j = x_j{'}S^{-1}x_j$ (j=1,...,n) (so u_j is the squared Mahalanobis distance of x_j from the sample mean 0). Suppose the data are projected into one dimension by Y=β′X (β a p×1 vector). Let $y_j=\beta'x_j$ and define $U_j(\beta)=(n-1)y_j{'}(YY')^{-1}y_j$.

i) Shew that $U_j(\beta)$ is maximized with respect to β by the (right) eigenvector of $S^{-1}x_jx_j{'}$ corresponding to its only non-zero eigenvalue.

ii) If this eigenvector is β_j, shew that this maximum value $U_j(\beta_j)$ is equal to this non-zero eigenvalue.

iii) Shew that $u_j=U_j(\beta_j)$.

iv) Shew that the non-zero eigenvalue of $S^{-1}x_jx_j{'}$ is $x_j{'}S^{-1}x_j$ and the corresponding eigenvector is proportional to $S^{-1}x_j$

(Note that $YY'=\beta'XX'\beta$ is 1×1, i.e. a scalar, so $U_j(\beta) = (n-1)y_j{'}y_j/\beta'XX'\beta = (n-1)x_j{'}\beta\beta'x_j/\beta'XX'\beta = (n-1)\ \beta'x_jx_j{'}\beta/\beta'XX'\beta$ since $\beta'x_j$ & $x_j{'}\beta$ are 1×1 and so commute. Further note that multiplying β by a scalar constant does not alter the value of $U_j(\beta)$ so the problem is not altered if you impose the constraint that the denominator of the expression is 1.)

3 Multidimensional Scaling Techniques

3.0 Introduction

Given a set of n points in Euclidean p-space one can compute the distance between any pair of points and can obtain an n×n distance matrix or dissimilarity matrix (d_{ij}). This section considers the converse problem: given an n×n [symmetrical] matrix (δ_{ij}) of dissimilarities, can a configuration of points be found in Euclidean p-space (p open to choice) such that the calculated distance matrix (d_{ij}) reasonably matches the given dissimilarity matrix (δ_{ij})? The answer is generally *yes* for sufficiently large p (e.g. p=n–1, with some restrictions on the δ_{ij}). The interest is in when it can be done for very small p (e.g. p=1, 2, 3, 4?, 5??,..).

Note that the matrix (δ_{ij}) of dissimilarities can be some general measure of dissimilarity, e.g. train journey times reflect distances between towns, numbers of times one Morse code symbol is mistaken for another reflects how similar/dissimilar they are. The measure may be very coarse, a measure of how ‘similar’ or dissimilar any pair of Departments of France are could be δ_{ij}=1 if they have a common border and δ_{ij}=0 if they have no common border — this is strictly a measure of *similarity* but clearly similarity measures can easily be converted to dissimilarities and vice versa.

Another common measure of similarity is illustrated by an example of prehistoric graves which [potentially] contain artefacts of M types:

Artefact

	1	**2**	**3**	**4**	**5**	**.**	**.**	**.**	**.**	**.**	**.**	**.**	**.**	**.**	**M**
grave i	0	0	1	1	.	.	1	0	1	0	0	.	.	.	1
grave j	0	1	1	0	.	.	0	0	1	0	0	.	.	.	.

0=artifact absent from grave, 1=artifact present in grave.

— define δ_{ij} = #artefacts in common between grave i and grave j

For such presence/absence data there are many (~100+) different measures of similarity/dissimilarity.

The objectives of the multidimensional scaling analysis can be any or some of:

- ♦ to learn more about the measure of dissimilarity itself e.g. the Morse code example — what are the factors which cause the brain to confuse different Morse symbols?

- ♦ to discover some underlying structure in the data themselves e.g. if the dissimilarities can be closely matched by a string of points lying on a line can the distance along the line be interpreted as some variable of interest such as time in the prehistoric graves example?

- ♦ to discover whether the units divide 'naturally' into groups — these techniques are widely used in market research for 'market segmentation' – do the customers divide into different target groups that are reached by different forms of advertising?

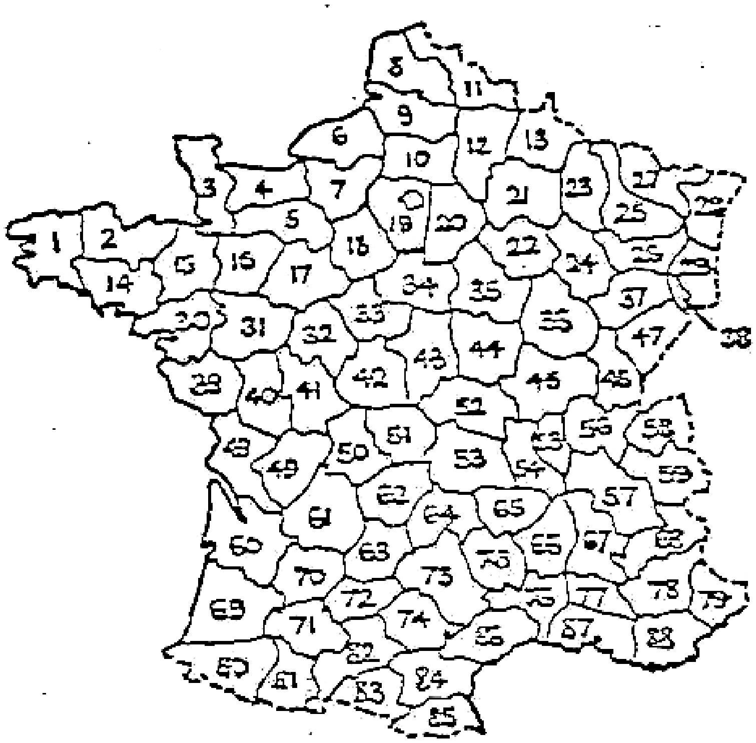
- ♦ to discover some 'gap' in the market than can be filled by a new product.

Applications of the methods to examples such as reconstructing the map of British towns from the railway timetable or the map of France from abuttal data on the Departments are designed to *validate the techniques* not to find out what France looks like. This latter example was used as a preliminary to doing the same analysis on reconstructing a map of ancient towns from inscriptions on linear B clay tablets — two towns scored 1 if their names appeared on the same tablet, 0 otherwise. It was thought that the tablets referred to trading between the towns so if they were mentioned together then they were likely to be geographically close.

The available techniques for multi-dimensional scaling are mostly *ad hoc* methods — i.e. based on intuition rather than solid mathematical theory — and are mostly *non-metric* (i.e. use only the orderings of the given dissimilarities and not their actual absolute values). The one exception is the 'classical solution' which is often used as the initial trial solution for the start of one of the iterative methods.

3.1 Illustrations

3.1.1 Reconstructing France



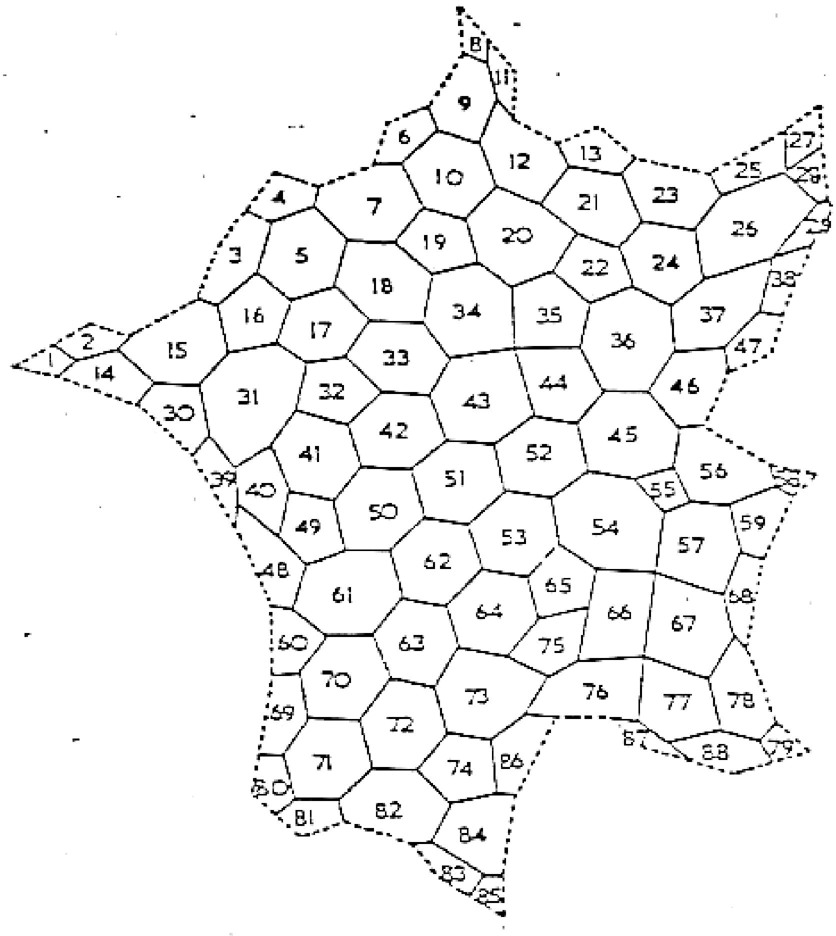
Map of the Departments of France

The measure of similarity used is a binary measure of 1 if two departments abut (i.e. have a common border) and 0 otherwise.

Department										
	1	**2**	**3**	**4**	**……..**	**14**	**15**	**16**	**17**	**…**
1	★	1	0	0	……..	1	0	0	0	…
2	1	★	0	0	……..	1	1	0	0	…
3	0	0	★	1	……..	0	1	1	0	…
4	0	0	1	★	……..	0	0	0	0	…
……………………………………………………………………………………………										
14	1	1	0	0	……..	★	1	0	0	…
15	0	1	1	0	……..	1	★	1	0	…
16	0	0	1	0	……..	0	0	★	1	…
17	0	0	0	0	……..	0	0	1	0	…
…	…	…	…	…	……..	…	…	…	…	…

The similarity matrix

Reconstructing from just this sparse information gives a map with each department positioned on it. The orientation and scaling on the map will be entirely arbitrary. The illustration below has had boundaries drawn in by inserting a Dirichlet tessellation around the points and then the map has been stretched and squashed to make it look as like the overall outline of France (the distinctive hexagon) as possible, but this still preserves the abuttal information.



Reconstructed Map of France

(note that the numbering of departments is roughly in sequence around the coast and working in, not the standard numbering on car number-plates)

3.1.2 British Towns By Road

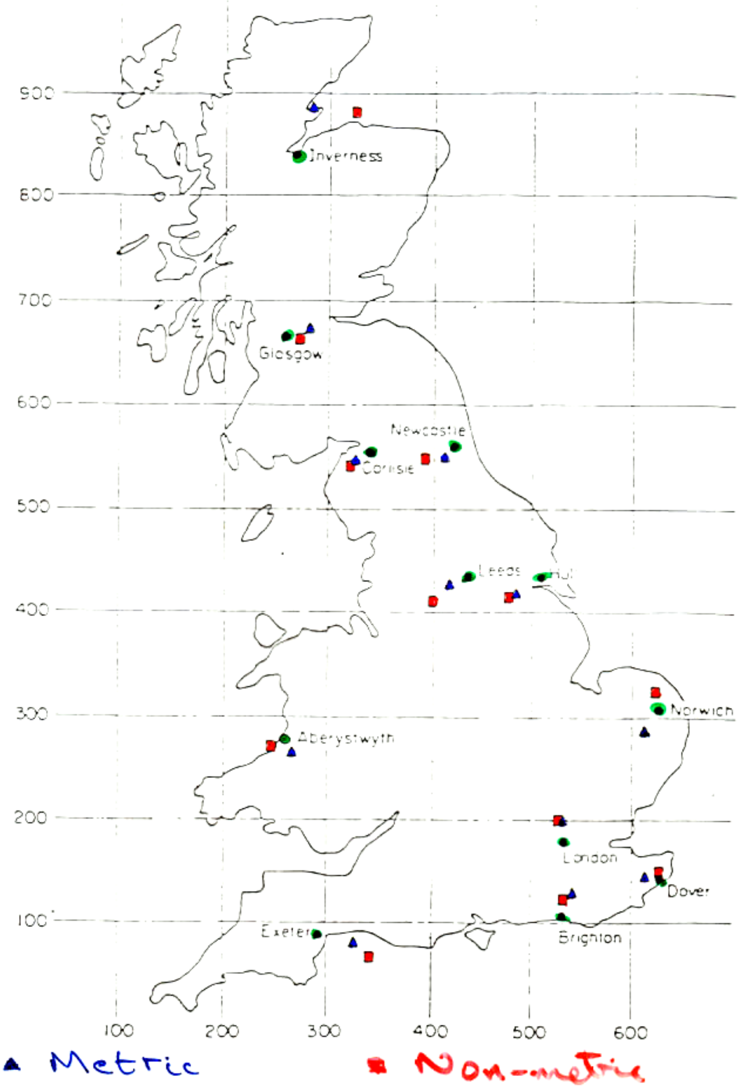


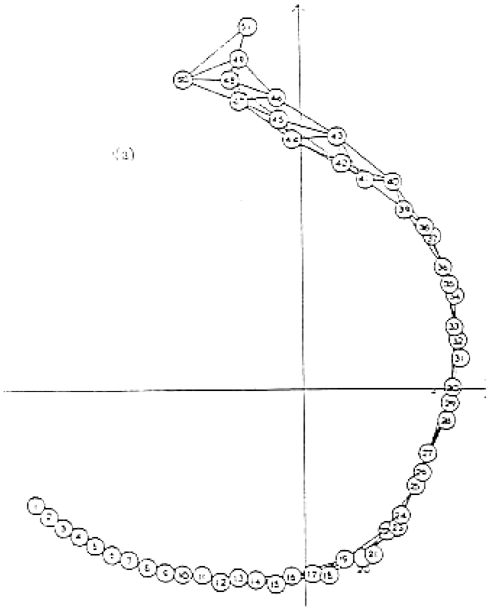
Figure 14.4.1 MDS solutions for the road data in Table 1.2.2. ●, original points; ▲, classical solution; ■, Shepard–Kruskal solution.

Reconstruction of British Towns from Road Distances

Again, the initial solution from the analysis has to be orientated to match the outline of the country but however that is done Inverness is still misplaced — this is because the road distance to Inverness from Glasgow (or Newcastle or Carlisle) is much further than the crow-fly distance, since the route has to pass through the Scottish Highlands.

3.1.3 Time Sequence Of Graves

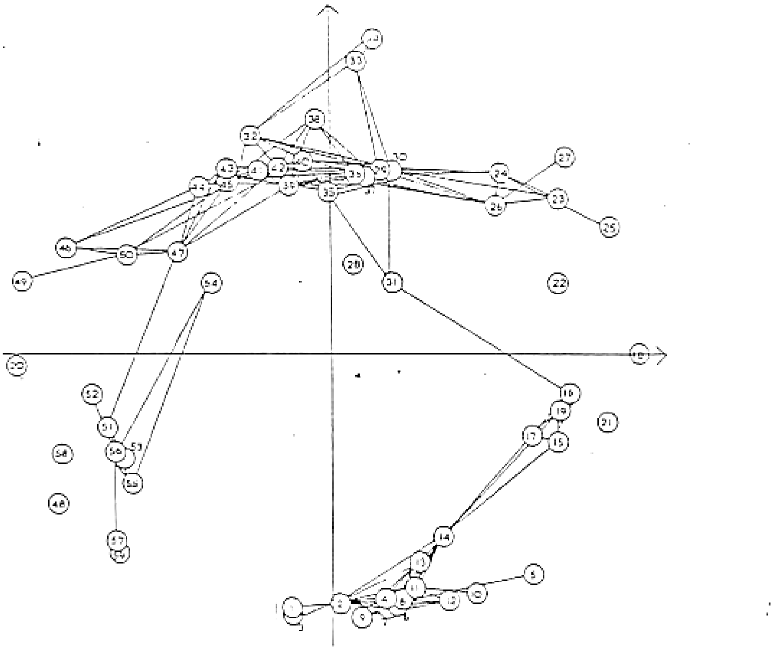
It is believed by archaeologists that the types of objects that are placed in prehistoric graves reflect changing fashion and custom. Thus graves which are close together in time will tend to have a similar collection of objects with many in common. Thus if the similarity between graves is measured by counting the number of objects found in common then this similarity might reflect time and a map constructed from such data might reveal the time sequence of the graves. The Bronze Age cemetery in Munsingen, Austria, has been a source of controversy amongst archaeologists for many years and many attempts have been made to sequence the graves. Below is an attempt using scaling techniques made by Kendall. First is an artificial example where the sequence of ‘graves’ is constructed and ‘objects’ allocated to them as if they appeared in fashion for a few ‘years’ and then disappeared again.



Artificial Example

This reconstructs the original sequence nearly perfectly, reading the sequence counter-clockwise around the 'horseshoe'. Note that the orientation of the map is entirely arbitrary. The *'horsehoe effect'* is typical of this and similar problems — it arises since the similarity is scored as 0 (and the two graves have no goods in common) this could be because they are moderately separated in time or because they are at far ends of the sequence, i.e. the measure cannot separate the ends of the horseshoe adequately so the end of the sequence are pulled together. The other typical feature is the lack of clear sequence at the ends of the horseshoe.

The success of the artificial example lends credence to applying the same technique to the archaeological data:



Reconstruction of Munsingen Graves

The numbers on the map are the ordering determined by Hodson. The correspondence with the ordering implied by this map is generally good. The lack of clear sequence at the top of this display illustrates that these graves are likely to cause controversy however they are placed in sequence.

3.2 Non-metric methods

Such methods use only the relative orderings of the given distances δ_{ij} and is iterative. In outline, the basis of all the techniques is to obtain a trial configuration of points, calculate the distance matrix (d_{ij}) for this configuration and see how well the ordering of the d_{ij} matches that of the given δ_{ij} — then to perturb the trial configuration to try to improve the match of the orderings.

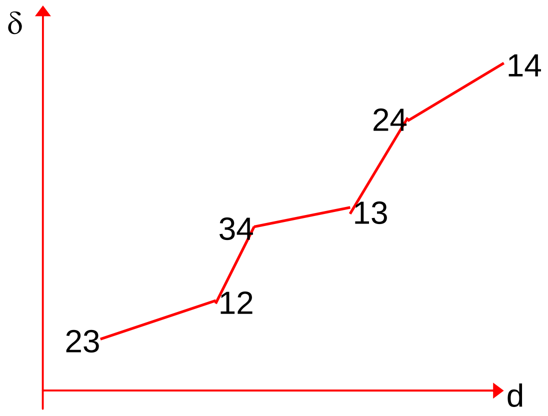
e.g. suppose n=4 and we are given a 4×4 [symmetric] distance matrix (δ_{ij}). Necessarily δ_{ii}=0, i=1,...,4. Suppose the other six distinct δ_{ij} have the ordering

$$\delta_{23}<\delta_{12}<\delta_{34}<\delta_{13}<\delta_{24}<\delta_{14}$$

We want to find a configuration of four points such that the distance matrix (d_{ij}) satisfy

$$d_{23}\leq d_{12}\leq d_{34}\leq d_{13}\leq d_{24}\leq d_{24}\leq d_{14}$$

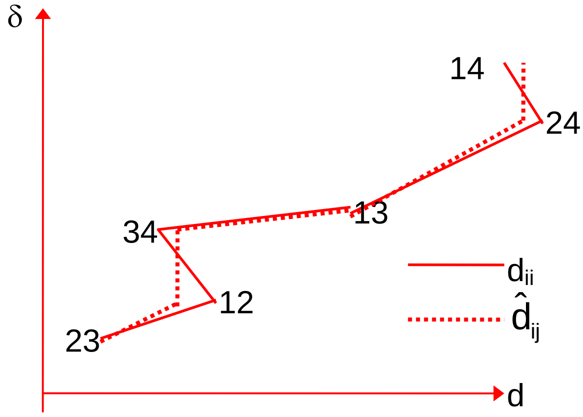
If this is the case, then a plot of δ_{ij} *vs.* d_{ij} looks like



However, suppose the trial configuration turns out to have

$$d_{23}<d_{34}<d_{12}<d_{13}<d_{24}<d_{14}<d_{24}$$

(i.e. d_{34} & d_{12} swapped, also d_{14} & d_{24} swapped)



To assess the extent to which the d_{ij} differ from the δ_{ij} (with respect to the ordering) we fit some set of values $\hat{d}_{ij}$ which satisfy the ordering constraint $\hat{d}_{23}\leq\hat{d}_{12}\leq\hat{d}_{34}\leq\hat{d}_{13}\leq\hat{d}_{24}\leq\hat{d}_{14}$, (i.e. weakly matching the same ordering as the given δ_{ij}). *These need not be a configuration of points with distances $\hat{d}_{ij}$, they are* *__merely a set of values__* *with the same ordering as the δ_{ij}.*

For this example, we could take

$\hat{d}_{12}=\hat{d}_{34}=½(d_{12}+d_{34})$, $\hat{d}_{24}=\hat{d}_{14}=½(d_{24}+d_{14})$,

$\hat{d}_{23}=d_{23}$, $\hat{d}_{3}=d_{13}$

The measure of agreement of the $\hat{d}_{ij}$ with the d_{ij} is a measure of how well the ordering of the d_{ij} matches that of the original given δ_{ij}. We can define first a measure of agreement between the $\hat{d}_{ij}$ and the d_{ij} as

$$S=\left(\frac{\sum\limits_{i<j}(\hat{d}_{ij}-d_{ij})^2}{\sum\limits_{i<j}d_{ij}^2}\right)^{½}$$

S is termed the **‘stress’** of the configuration.

This is the standard choice of stress function but others that are used are a straightforward generalization proposed by

Kruskal: $$S_{\text{Krus}} = \left(\frac{\sum\limits_{i<j} (\theta(d_{ij}) - \hat{d}_{ij})^2}{\sum\limits_{i<j} \hat{d}_{ij}^2} \right)^{1/2}$$ for some monotonic increasing function $\theta(.)$

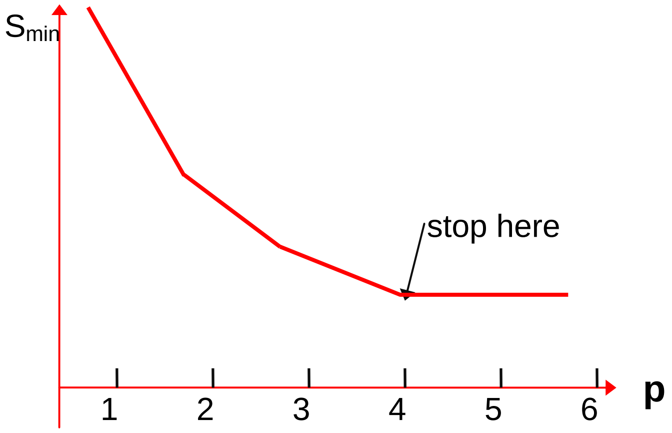
and a rather different one by

Sammon: $$S_{\text{Sam}} = \frac{1}{\sum\limits_{i<j} d_{ij}} \sum_{i<j} \frac{(d_{ij} - \hat{d}_{ij})^2}{d_{ij}}$$

Whichever stress function we select we choose the $\hat{d}_{ij}$ to minimize S subject to the ordering constraint, i.e. the $\hat{d}_{ij}$ are monotonic non-decreasing with the δ_{ij}. Then $S_{\min}$ is a measure of how well the trial configuration matches the original δ_{ij}. If we regard $S_{\min}$ as a function of the coordinates of the n points in p-space defining d_{ij} (i.e. a function of np variables), then we can minimize stress ($S_{\min}$) with respect to these variables and find the least stressful configuration in p-space.

We calculate the minimal stress for p=1, 2, 3, ... and stop when increasing p does not decrease stress very much. This determines the dimensionality of the problem.

We can use a scree plot to help choose a 'good' value of p:



Standard programs are available to minimize S (using monotonic regression algorithms) and hence to determine the optimal configuration, e.g. MDSCAL and GENSTAT. **R** and S-PLUS functions `isoMDS(.)` and `sammon(.)` in the `MASS` library perform variants of non-metric scaling very easily, the distinction between them is in the form of the stress function used.

Some comparative examples are given at the end of this chapter, together with illustrations of classical scaling (see below). The general experience is that Sammon mapping produces configurations that are more evenly spread out and Kruskal or classical scaling may produce ones which are more tightly clustered. Which is preferable is open to choice and depends upon the particular application.

3.3 Metric Methods: The Classical Solution or Principal Coordinate Analysis

This method uses the actual distances and will produce a configuration of points with distance matrix precisely equal to that given if such a configuration actually exists (i.e. if the given distance matrix is indeed a matrix of distances for some configuration, i.e. is *Euclidean*). If such a configuration does not exist then this method will give a configuration with distance matrix approximating that given and this is a useful starting point for the iterative methods outlined in §3.2.

Suppose $D=(d_{ij})$ is an $n\times n$ distance matrix and define $A=(a_{ij})$ by

$$a_{ij}=-\tfrac{1}{2}d_{ij}^2$$

Let $H=I_n-\frac{1}{n}J_n$, the centring matrix, where I_n is the $n\times n$ identity matrix, and J_n is the $n\times n$ matrix with all entries equal to 1, so $J_n=1_n1_n'$, where 1_n is the unit n-vector. So

$$H=\begin{pmatrix} 1-\frac{1}{n} & -\frac{1}{n} & -\frac{1}{n} & \cdots & -\frac{1}{n} \\ -\frac{1}{n} & 1-\frac{1}{n} & & & -\frac{1}{n} \\ \vdots & & \ddots & & \vdots \\ \vdots & & & \ddots & -\frac{1}{n} \\ -\frac{1}{n} & \cdots & & -\frac{1}{n} & 1-\frac{1}{n} \end{pmatrix}$$

H is called the centring matrix because pre- and post-multiplying a matrix by H centres all the elements by the row and column means:

Define $B=HAH$, then $b_{ij}=a_{ij}-\frac{1}{n}\sum_{k=1}^{n}a_{ik}-\frac{1}{n}\sum_{k=1}^{n}a_{kj}+\frac{1}{n^2}\sum_{k=1}^{n}\sum_{l=1}^{n}a_{kl}$

$$= a_{ij}-\bar{a}_{i+}-\bar{a}_{+j}+\bar{a}_{++} \text{ ------------ } *$$

Then:

Theorem: D is a matrix of ***Euclidean*** distances if, and only if, B is positive semi-definite. (i.e. There is a configuration of n points X′ in Euclidean p-space whose interpoint distances are given by D if, and only if, B≥0, i.e. all the eigenvalues of B are non-negative).

Proof: 'only if' (i.e. start with n points and shew B≥0)

Let X′ be an n×p data matrix, and $X=(x_1,x_2,...,x_n)$, x_i a column p-vector. Then if D is the distance matrix derived from X′ we have that

$-2a_{ij}=d_{ij}^2=(x_i-x_j)'(x_i-x_j)$ so that $b_{ij}=(x_i-\bar{x})'(x_j-\bar{x})$

i.e. $B=(X-\bar{X})'(X-\bar{X})$ which is positive semi-definite,

i.e. B≥0.

'if': (i.e. start with B≥0, == all eigenvalues ≥0) and find a configuration of points with distance matrix D):

B≥0, so all eigenvalues $\lambda_i \geq 0$.

Suppose $\lambda_1 \geq \lambda_2 \geq ... \geq \lambda_p > 0 (= \lambda_{p+1} = ... = \lambda_n)$ are the p non-zero eigenvalues. Let the corresponding eigenvectors be $x_{(1)},...x_{(p)}$; where the scale is chosen so that $x_{(i)}'x_{(i)}=\lambda_i$; i=1,...,p.

i.e. $x_{(i)}$ satisfy $Bx_{(i)}-\lambda_i x_{(i)}=0$ & $x_{(i)}'x_{(i)}=\lambda_i$; i=1,...p. - - - - - - - - - - - - - ∗∗

Note that B is n×n and so $x_{(i)}$ is n×1.

Let $X'=(x_{(1)},....,x_{(p)})$ then X′ is n×p.

We have $XX'=(x_{(i)}'x_{(j)})$ and $x_{(i)}'x_{(j)}=\lambda_i$ if i=j, 0 otherwise.

So $XX'=\text{diag}(\lambda_i)=\Lambda$ (say)

(the diagonal matrix with (i,i)th element λ_i)

claim: the n points in p-space given by the n rows of X′ have distance matrix given by D.

since: we have $BX'-X'\Lambda=0$ and $XX'=\Lambda$ (restatement of ∗∗)

Let $M=\begin{pmatrix}\Lambda & 0_{p,n-p}\\ 0_{n-p,p} & 0_{n-p}\end{pmatrix}$ and $N=\begin{pmatrix}\Lambda & 0_{p,n-p}\\ 0_{n-p,p} & I_{n-p}\end{pmatrix}$

and $Y'=(x_{(1)},\ldots,x_{(p)},y_{(p+1)},\ldots,y_{(n)})$ where the y's are chosen almost arbitrarily but such that they are orthogonal to each other and to the $x_{(i)}$ and such that $y_{(i)}'y_{(i)}=1$ (i.e. the $y_{(i)}$ are the eigenvalues of B corresponding to the n–p zero eigenvalues).

Then we can re-write ∗∗ as $BY'-Y'M=0$.

Let $\Gamma=Y'N^{-½}$; where $N^{-½}$ is obtained by taking each of the diagonal elements of N to the power –½, i.e.

$$N^{-\frac{1}{2}}=\begin{pmatrix}\lambda_1^{-\frac{1}{2}} & 0 & \cdots & \cdots & 0\\ 0 & \ddots & & & \vdots\\ \vdots & & \lambda_p^{-\frac{1}{2}} & 0 & 0\\ \vdots & & & 1 & 0\\ 0 & \cdots & \cdots & 0 & 1\end{pmatrix}$$

then $\Gamma\Gamma'=\Gamma'\Gamma=I_n$, so $B\Gamma\text{-}Y'MN^{-½}=0$ and so $B\Gamma=\Gamma M$.

Thus $\Gamma'B\Gamma=\Gamma'\Gamma M^{-½}=M$, so $\Gamma\Gamma'B\Gamma\Gamma'=\Gamma M\Gamma'$,

i.e. $B=Y'N^{-½}MN^{-½}Y=XX'$, i.e. $b_{ij}=x_i^{'}x_j$.

Now the square of the $(i,j)^{th}$ element of the distance matrix of X is

$(x_i\text{–-}x_j)'(x_i-x_j)=x_i'x_i-2x_ix_j+x_j'x_j=b_{ii}-2b_{ij}+b_{jj}$

$= a_{ii}-2a_{ij}+a_{jj}$ (by ∗) $=-2a_{ij}=d_{ij}^2$

(noting that $d_{ii}^2=d_{jj}^2=0$, since D is a distance matrix)□

3.4 Comments on practicalities

The above result gives a practical way of finding a configuration of points with [approximately] a given data matrix. Given a distance matrix D:

1. Find the matrix A=(–½d_{ij}^2)
2. Find the matrix B=HAH
3. Find the eigenanalysis of B
4. Transpose the matrix of eigenvectors
5. Take the columns of this transposed matrix as the principal coordinates of the points.

If all the eigenvalues are non-negative then the distance matrix will be an exact match. If some are negative then [pragmatically] take just the positive ones and the match will be approximate. Eigenvectors taken in order of magnitude of the positive eigenvalues will give increasingly better approximations to the desired configuration of points.

All of this can be done in MINITAB straightforwardly: use the facilities in the Calc>Matrices menus.

If the matrix B is not positive semi-definite (i.e. some eigenvalues are negative) then MINITAB will present the eigenvectors in the order of absolute magnitude (*not* arithmetical magnitude) of the eigenvalues. To sort out the eigenvectors corresponding to just positive eigenvalues, copy the eigenvectors into columns with the Use Rows.... option in the Calc>Matrices menu completed appropriately. Again, if B has negative eigenvalues then assess quality of representation informally by a scree plot based on ***squared*** eigenvalues.

In **R** it is even easier since there is a built-in function `cmdscale(.)` to perform classical scaling.

3.5 Computer implementation

Standard specialist programs are available to minimize S (using monotonic regression algorithms) and hence to determine the optimal configuration: best are on the CD-Rom supplied with Cox, T.F. & Cox, M.A.A (2001) Multidimensional Scaling (2nd Ed.), Chapman & Hall. Other standard programme is MDSCAL, facilities exist in general packages, e.g. SPSS and GENSTAT (but not MINITAB). In **R,** or S-PLUS, (with `MASS` library) classical scaling is provided by function `cmdscale(.)`, Sammon and Kruskal versions are provided by functions `sammmon(.)` and `isoMDS(.)`.

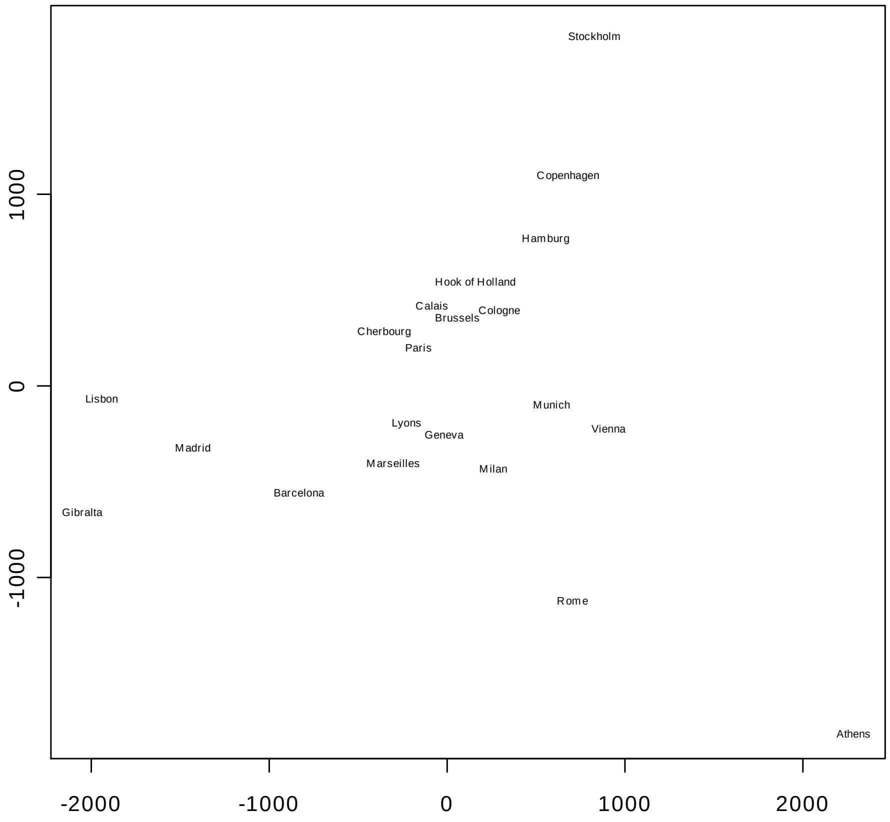
3.6 Examples

3.6.1 Road distances between European cities

The data set `eurodist` gives the distances by road in kilometres between 21 cities in Europe and is given below. Analysis is in **R,** (it is similar in S-PLUS)

```
                Athens Barcelona Brussels Calais Cherbourg Cologne Copenhagen
Barcelona         3313
Brussels          2963      1318
Calais            3175      1326      204
Cherbourg         3339      1294      583    460
Cologne           2762      1498      206    409       785
Copenhagen        3276      2218      966   1136      1545     760
Geneva            2610       803      677    747       853    1662       1418
Gibralta          4485      1172     2256   2224      2047    2436       3196
Hamburg           2977      2018      597    714      1115     460        460
Hook of Holland   3030      1490      172    330       731     269        269
Lisbon            4532      1305     2084   2052      1827    2290       2971
Lyons             2753       645      690    739       789     714       1458
Madrid            3949       636     1558   1550      1347    1764       2498
Marseilles        2865       521     1011   1059      1101    1035       1778
Milan             2282      1014      925   1077      1209     911       1537
Munich            2179      1365      747    977      1160     583       1104
Paris             3000      1033      285    280       340     465       1176
Rome               817      1460     1511   1662      1794    1497       2050
Stockholm         3927      2868     1616   1786      2196    1403        650
Vienna            1991      1802     1175   1381      1588     937       1455
        Geneva Gibralta Hamburg Hook of Holland Lisbon Lyons Madrid
Barcelona
Brussels
Calais
Cherbourg
Cologne
Copenhagen
Geneva
Gibralta          1975
Hamburg           1118     2897
Hook of Holland    895     2428     550
Lisbon            1936      676    2671            2280
Lyons              158     1817    1159             863   1178
Madrid            1439      698    2198            1730    668  1281
Marseilles         425     1693    1479            1183   1762   320   1157
Milan              328     2185    1238            1098   2250   328   1724
Munich             591     2565     805             851   2507   724   2010
Paris              513     1971     877             457   1799   471   1273
Rome               995     2631    1751            1683   2700  1048   2097
Stockholm         2068     3886     949            1500   3231  2108   3188
Vienna            1019     2974    1155            1205   2937  1157   2409
                Marseilles Milan Munich Paris Rome Stockholm
Barcelona
Brussels
Calais
Cherbourg
Cologne
Copenhagen
Geneva
Gibralta
Hamburg
Hook of Holland
Lisbon
Lyons
Madrid
Marseilles
Milan                  618
Munich                1109   331
Paris                  792   856    821
Rome                  1011   586    946  1476
Stockholm             2428  2187   1754  1827 2707
Vienna                1363   898    428  1249 1209      2105
```

```
> data(eurodist)
>      loc <- cmdscale(eurodist)
>      x <- loc[,1]
>      y <- -loc[,2]
>      plot(x, y, type="n", xlab="", ylab="")
>      text(x, y, names(eurodist), cex=0.5)
```



This reproduces the geography of Europe very closely and suggests that the technique itself ‘works’ and so can be applied to other data where we don’t know what the answer is beforehand, e.g. on Morse Code confusion data:

3.6.2 Confusion between Morse code signals

The data are derived from a set presented by Rothkopf(1957). The original data give the percentages of times that 598 subjects responded "Same!" when two symbols were transmitted in quick succession, i.e. when symbol *i* (row) was transmitted first followed by symbol *j* (columns) second. The original matrix for all 26 letters and ten numerals is asymmetric and is given in file morsefullasym.Rdata (and morsefullasym.csv). These are *similarity matrices.* The file morsefull.Rdata gives a symmetric version of a *distance matrix* and indicates the percentages of times that two symbols were declared to be different. It is derived from the original by first taking the symmetric part (i.e. half the sum of the original matrix and its transpose) and then subtracting these from 100. Finally the diagonal elements were set to zero to ensure that it was a true distance matrix.

The file morse.Rdata gives just the submatrix of morsefull.Rdata that contains the digits 0-9.

1: ● – – – –

2: ● ● – – –

3: ● ● ● – –

4: ● ● ● ● –

5: ● ● ● ● ●

6: – ● ● ● ● ●

7: – – ● ● ●

8: – – – ● ●

9: – – – – ●

0: – – – – –

3.6.3 Confusion between symbols for digits: R analysis

```
> load("morse.Rdata")
> attach(morse)
> morse.cmd<-cmdscale(morse,k=9,eig=TRUE)
Warning messages:
1: In cmdscale(morse, k = 9, eig = TRUE) :
  some of the first 9 eigenvalues are < 0
2: In sqrt(ev) : NaNs produced
> morse.cmd$eig
[1]  1.143502e+04  6.354744e+03  4.482303e+03  2.330540e+03
1.758846e+03
[6]  9.104095e+02  1.524900e+01 -1.136868e-12 -7.103261e+02
```

Using the option k=9 means that the eigenvectors corresponding to the first 9 non-negative eigenvalues are recorded and are stored in the matrix `morse.cmd$points`. Omitting the option `k=…` would result in the first two eigenvectors only being calculated and they would be stored in a matrix `morse.cmd[,.,].` Using the `option eig=TRUE` means that the eigenvalues are stored in the vector `morse.cmd$eig.`

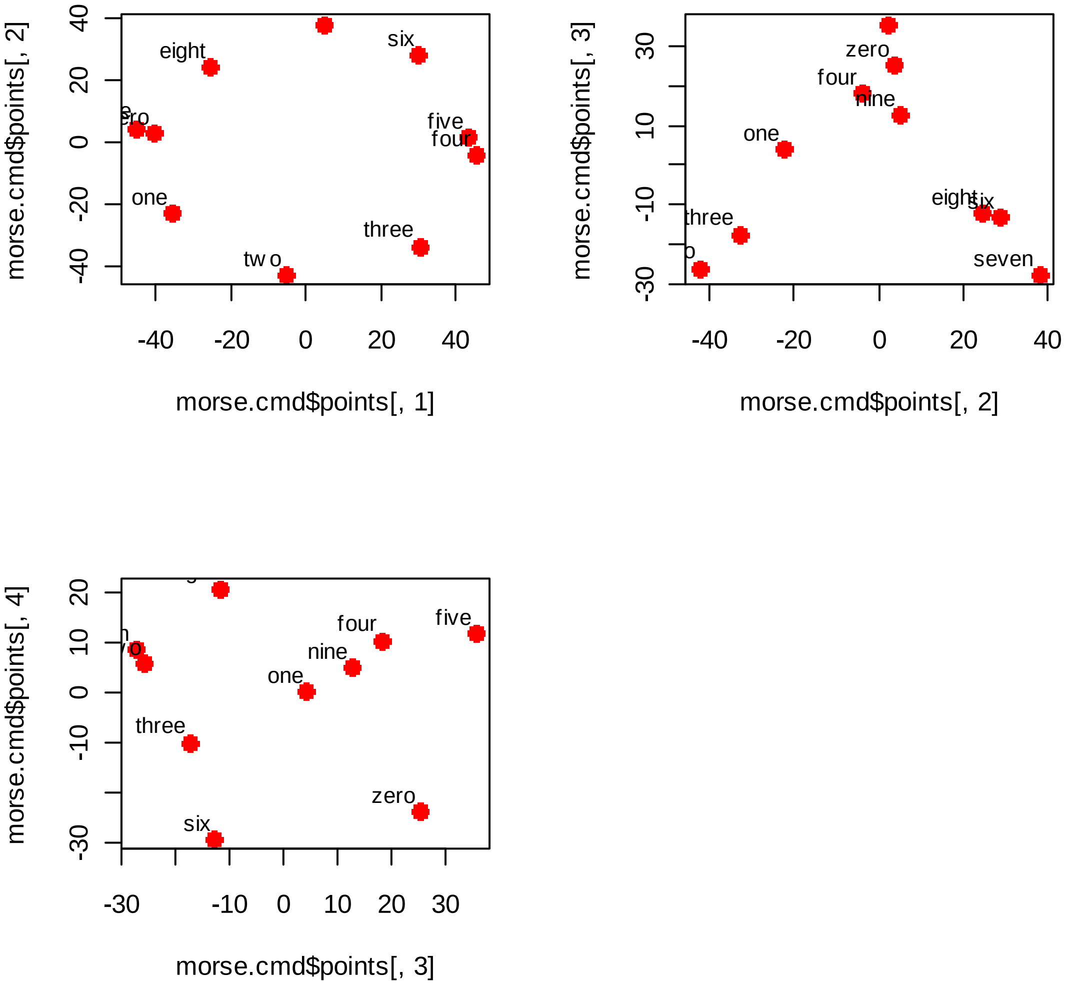
Note the warning message signalling that some eigenvalues are negative, this means it is not possible to find an exact Euclidean solution.

```
>
> plot(morse.cmd$points[,2],morse.cmd$points[,3],pch=16,col="red",
+ cex=1.5)
> text(morse.cmd$points[,2],morse.cmd$points[,3],row.names(morse),
+ cex=0.8,adj=c(1.1,-0.6))
>
> plot(morse.cmd$points[,3],morse.cmd$points[,4],pch=16,col="red",
+ cex=1.5)
> text(morse.cmd$points[,3],morse.cmd$points[,4],row.names(morse),
+ cex=0.8,adj=c(1.1,-0.6))
>
```

After producing the first plot it is useful to make the plot the active window and then click on `History` in the menu bar and select `Recording`. This allows scrolling between all subsequent graphs and the first one with the page up and down keys. Alternatively, issuing the command

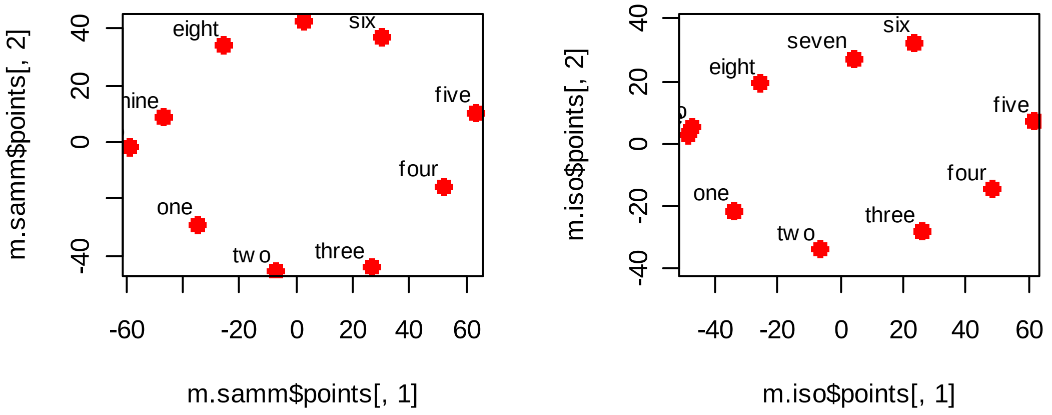
```
> par(mfrow=c(2,2))
>
```

Before the plot commands produces



To compare with non-metric methods we have:

```
> par(mfrow=c(2,2))
> m.samm<-sammon(morse)
Initial stress        : 0.06906
stress after  10 iters: 0.02855, magic = 0.500
stress after  20 iters: 0.02829, magic = 0.500
stress after  30 iters: 0.02829, magic = 0.500
>
eqscplot(m.samm$points[,1],m.samm$points[,2],pch=16,col="red",
cex=1.5)
> text(m.samm$points,names(morse),cex=0.8,adj=c(1.1,-0.6))
>
> m.iso<-isoMDS(morse,cmdscale(morse))
initial  value 13.103235
iter   5 value 9.362759
iter  10 value 7.943274
final  value 7.652321
converged
>
eqscplot(m.iso$points[,1],m.iso$points[,2],pch=16,col="red",ce
x=1.5)
> text(m.iso$points,names(morse),cex=0.8,adj=c(1.1,-0.6))
                                                           >
```



Note that in the call to `isoMDS(.)` a starting configuration was required and this was given by the matrix `cmdscal(morse)`. The object `morse.cmd` could not be used since this was created with the k=9 option (to obtain all nine dimensions and so `morse.cmd` is not a matrix. By default, `cmdscale(.)` with no optional arguments produces a matrix with the two dimensional configuration.

Tasks 5

(see §3.1–§3.5)

1) Continuing Q1 of tasks 4, i.e. X' ($n\times p$) is a centred data matrix:

   i) If D is the $n\times n$ distance matrix of the n p-dimensional observations and A is the matrix given by $a_{ij} = -\tfrac{1}{2}d_{ij}^2$ and B=HAH, where H is the centring matrix, shew that $B = kX'X$ for some suitable scalar k.

   ii) Deduce that deriving a configuration of points from the matrix D by classical scaling is equivalent to referring the original data to principal components

2) If c_{ij} represents the similarity between cases i and j (c_{ij} is a *similarity* if $c_{ij}=c_{ji}$ and $c_{ij} \leq c_{ii}$) then the similarity matrix C can be converted to a *distance* matrix D by defining $d_{ij}=(c_{ii}-2c_{ij}+c_{jj})^{½}$. Define B=HAH where $A=(-\tfrac{1}{2}d_{ij}^2)$

   i) Shew that B=HCH.

   ii) Deduce that you can proceed with classical scaling analysis analyzing C directly instead of converting it to a distance matrix and then calculating A.

3) The table below gives the road distances between 12 UK towns. The towns are Aberystwyth, Brighton, Carlisle, Dover, Exeter, Glasgow, Hull, Inverness, Leeds, London, Newcastle and Norwich.

i) Is it possible to construct an exact map for these distances?

	A	B	C	D	E	G	H	I	Le	Lo	Ne	No
A	0											
B	244	0										
C	218	350	0									
D	284	77	369	0								
E	197	167	347	242	0							
G	312	444	94	463	441	0						
H	215	221	150	236	279	245	0					
I	469	583	251	598	598	169	380	0				
Le	166	242	116	257	269	210	55	349	0			
Lo	212	53	298	72	170	392	168	531	190	0		
Ne	253	325	57	340	359	143	117	264	91	273	0	
No	270	168	284	164	277	378	143	514	173	111	256	0

These data are contained in data set towns.Rdata and in S-PLUS and Minitab format. The Minitab version has the names of the towns in the first column and the data matrix in the next 12 columns. The final 12 columns contain the 12×12 matrix $A=(-\frac{1}{2}d_{ij}^2)$. The **R** and S-PLUS versions give a dataframe with just the symmetric matrix of distances.

To do it in **R** you can use the function cmdscale() and then plot the results by together with as.matrix()which is required for cmdscale to recognise the distance matrix so you need to issue the functions nested: cmdscale(as.matrix(towns)). Note that it is already a distance matrix so you should not use the multidimensional scaling menu in the MASS library which presumes that you have a raw data matrix and calls dist() internally to create a new distance matrix. It is also possible in **R** to try two varieties of non-metric scaling (sammon() and isomds()).

3.6.5 Notes (1): similarity or dissimilarity measures?

- In the examples on the French Departments, Munsingen graves and Morse Code the measures were really similarities, whereas those on distances between towns and the Iris data the measures were dissimilarities. A 'similarity' is easily converted to a 'disimilarity' by changing the sign or taking the reciprocal or subtracting from 100 or many similar devices. In fact, one of the exercises shews that provided you convert between similarities and dissimilarities in a particular way then applying the Classical Solution of §3.3 gives precisely the same answers whether you start with a matrix of dissimilarities (as assumed there) or a matrix of similarities. Generally, it dose not matter very much how dissimilarities are converted into similarities and vice versa and the resulting displays are very similar.

3.6.6 Notes (2): caution on interpretation of close points

- If two points are well separated in a scaling plot then it must be the case that they are dissimilar but ***the converse is not true***. If two points are close together then they ***may or may not*** be similar. Plots on further dimensions might separate them. One way of assessing this is to look at the percentage of stress captured in the plot. If the greater part of the stress is represented in the plot then there 'cannot be enough room' to separate them. Another convenient way to assess this graphically is to superimpose a ***minimum spanning tree*** on the scaling plot. If this is annotate with the actual distances of each branch (only realistic for small numbers of plots) then this gives a full picture for interpretation. Exactly the same is true for PCA plots.

3.7 Minimum Spanning Trees

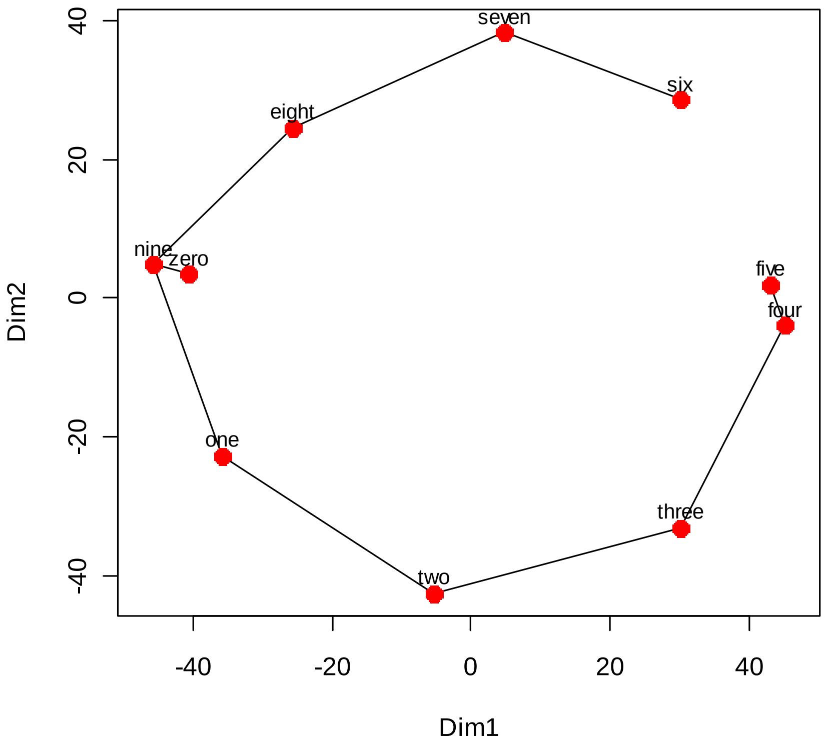
The minimum spanning tree is calculated using the distance information (across all dimensions) using the original dissimilarity matrix (d_{ij}). It provides a unique path such that (i) all points are connected with no circuits; and (ii) has the shortest possible path of all such trees. It is not necessarily unique but in practical applications it is likely to be. It has the property that for nodes *r* and *s* d_{rs} must be greater than the maximum link in the unique path from *r* to s using the edges in the tree. The idea is to give a direct visual impression of each individual's "nearest neighbours" since in general these will be the individuals connected to it by the edges of the tree. So, if two points appear to be close together in a scaling plot but the tree does not join them directly then it can be concluded that in fact they may not be very similar, though note that the converse does not hold (see the example below of "four" and "five" in the Morse confusion data). Nearby points on the plot which ***are not*** joined by edges indicate possible areas of distortion.

3.7.1 Computation in R

R has several facilities for calculating and plotting minimum spanning trees within the contributed packages. A selection of these is **`mst(.)`** in package **`ade4`**; **`mst(.)`** in package **`ape`**; **`dino.mst(.)`** and **`nmds(.)`** in package **`fossil`**; **`mstree(.)`** in package **`spdep`**; **`spantree(.)`** in package **`vegan`**. Only the last of these is illustrated here.

3.7.2 Example: Morse confusions

```
> library(vegan)
> morse.tr<-spantree(morse)
> plot(morse.tr,cmdscale(morse),pch=16,col="red",cex=1.5)
> text(cmdscale(morse),names(morse),cex=0.8,adj=c(0.5,-0.6))
>
```



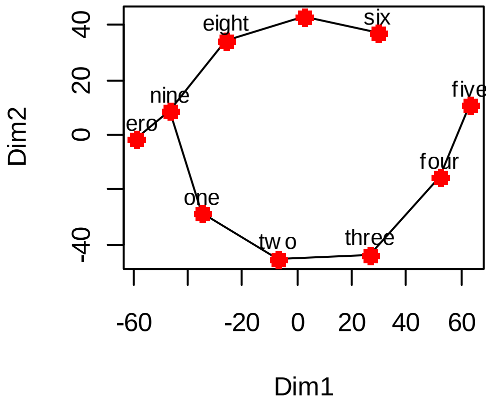
Examining the minimum spanning tree shews that the points nine and zero are joined and also four and five. However, all we can tell from this is these pairs are closer together than to any other point. Of course with this small number of points this could be seen by examining the distance matrix. In this case it shews that sero and nine are indeed close but four and five are well separated (which of course can be seen from the earlier plot of dimensions 2 and 3).To examine the distances on the minimum spanning tree (and so fewer numbers to scan for a large dataset do

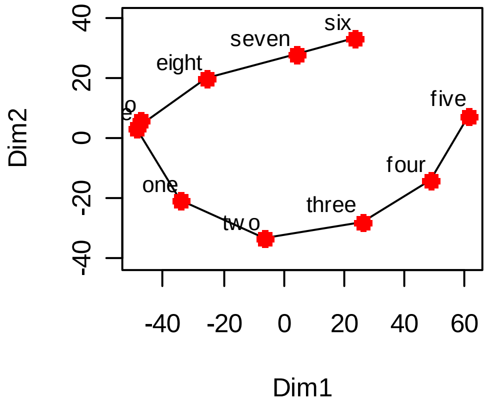
```
> morse.tr$kid
[1] 1 2 3 4 7 8 9 1 9
> morse.tr$dist
[1] 38 41 62 44 35 35 42 43 21
>
```

`morse.tr$kid` gives the child node of the parent, starting from parent number two and `morse.tr$dist` gives the corresponding distances.

The minimum spaning tree can also be added to other scaling plots:

```
> plot(morse.tr,sammon(morse),pch=16,col="red",cex=1.5)
Initial stress          : 0.06906
stress after  10 iters: 0.02855, magic = 0.500
stress after  20 iters: 0.02829, magic = 0.500
stress after  30 iters: 0.02829, magic = 0.500
>    text(sammon(morse)$points,names(morse),cex=0.8,adj=c(0.5,-
0.6))
Initial stress          : 0.06906
stress after  10 iters: 0.02855, magic = 0.500
stress after  20 iters: 0.02829, magic = 0.500
stress after  30 iters: 0.02829, magic = 0.500
>
> m.iso<-isoMDS(morse,cmdscale(morse))
initial  value 13.103235
iter   5 value 9.362759
iter  10 value 7.943274
final  value 7.652321
converged
> plot(morse.tr,m.iso,pch=16,col="red",cex=1.5)
> text(m.iso$points,names(morse),cex=0.8,adj=c(1.1,-0.6))
```





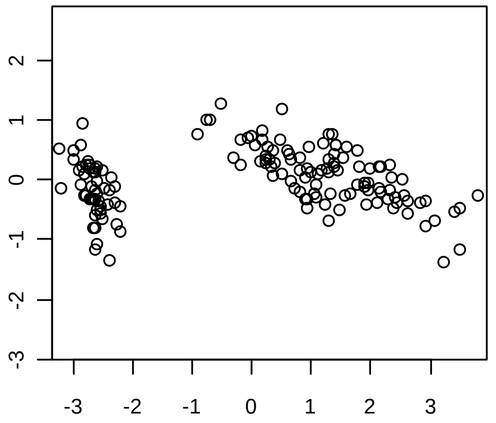
```
>
```

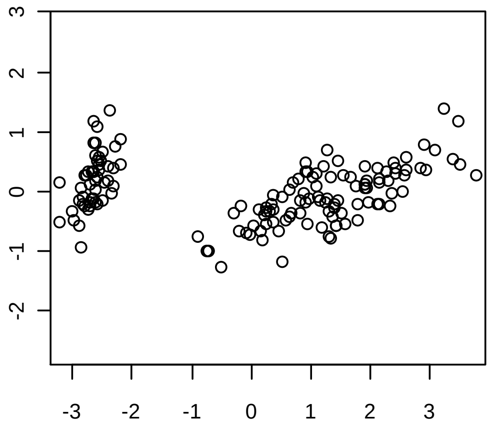
The minimum spaning tree can also be added to plots of any two components of a scaling plots but extreme care is needed in interpretation.

3.8 Duality with PCA

If we start with a data set and then calculate the Euclidean distance matrix for all the points (with function `dist()` in **R** or S-PLUS or in MINITAB with Stat>Multivariate>Cluster Observations) and then apply principal coordinate analysis, i.e. classical metric scaling, (with `cmdscale()` in **R** or S-PLUS) then we obtain precisely the same results as principal component analysis, except perhaps for arbitrary changes of sign of one or more axes, though we do not automatically have the additional information on proportions of variance or loadings available.

```
> ir.scal<- cmdscale(dist(ir))
> eqscplot(ir.scal)
> eqscplot(ir.pc)
```





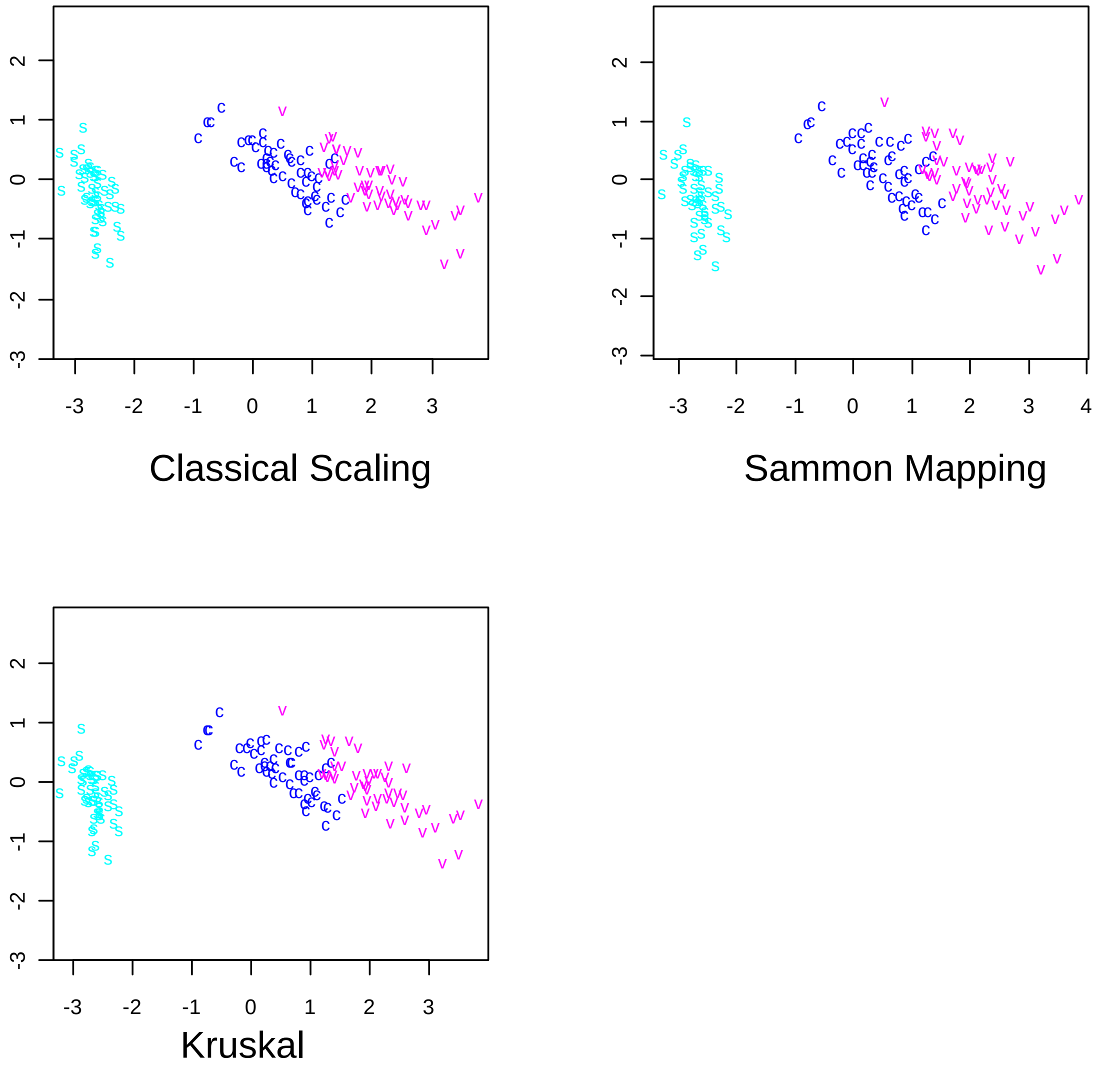
The `cmdscale` plot on the left is identical to the `pca` plot on the right apart from a reflection about the horizontal axis.

3.9 Further Examples

The four examples below compare classical metric scaling (equivalent to PCA for continuous data), Sammon Mapping and Kruskal isotonic regression multidimensional scaling. In the first set of data (the Iris Data) the first two PCs give 98% of the variation so it is not surprising that the three plots are indistinguishable. The second set is on the Swiss demographic data and small differences can be found. The third example is a data set on properties of glass fragments used in a forensic study. The fourth set is the Morse code confusion data used in 4.1. These data sets are standard ones within the **R** and S-PLUS (with `MASS` library) data libraries. Full **R** code (and data set descriptions for the second two examples taken from the **R** documentation) are given.

Very little distinction between the results of the methods is apparent in these examples, especially in the first example. It has been commented that Sammon mapping tends to produce displays with the points more evenly spread out than other methods and there may be a suggestion of this in the examples below, especially in the Morse Code confusion example.

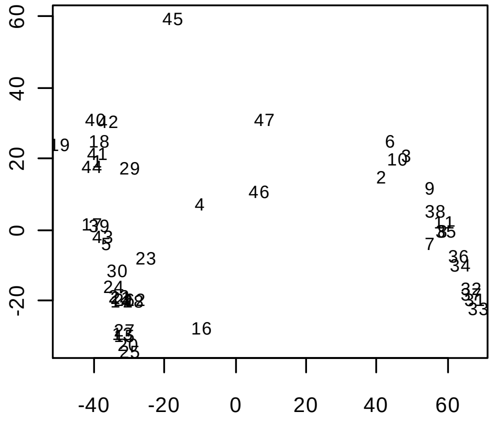
3.8.1 Iris Data



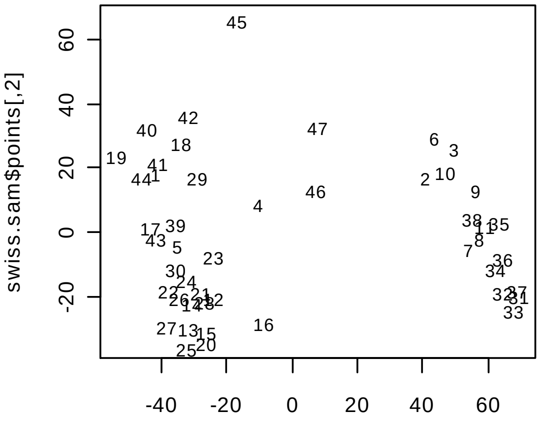
R Code to produce above plots:

```
> library(mass)
>
> data(iris)
> attach(iris)
> ir<-cbind(Sepal.Length,Sepal.Width,Petal.Length,Petal.Width)
> ir.species<- factor(c(rep("s",50),rep("c",50),rep("v",50)))
> par(mfrow=c(2,2))
> ir.scal<-cmdscale(dist(ir),k=2,eig=7)
> ir.scal$points[,2]<--ir.scal$points[,2]
> eqscplot(ir.scal$points,type="n")
> text(ir.scal$points,labels=as.character(ir.species),
  +col=3+codes(ir.species), cex=0.8)
> ir.sam<-sammon(dist(ir[-143,]))
Initial stress         : 0.00678
stress after  10 iters: 0.00404, magic = 0.500
stress after  12 iters: 0.00402
> eqscplot(ir.sam$points,type="n")
> text(ir.sam$points,labels=as.character(ir.species[-143]),
  + col=3+codes(ir.species),cex=0.8)
> ir.iso<-isoMDS(dist(ir[-143,]))
initial  value 3.024856
iter   5 value 2.638471
final  value 2.582360
converged
> eqscplot(ir.iso$points,type="n")
> text(ir.iso$points,labels=as.character(ir.species[-143]),
  +col=3+codes(ir.species),cex=0.8)
```

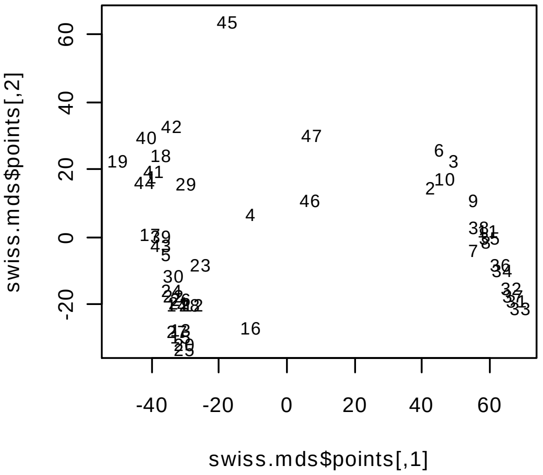
3.8.2 Swiss Demographic Data



Classical Scaling



Sammon Mapping



Kruskal

R Code to produce above plots:

```
> data(swiss)
> attach(swiss)
> swiss.x <- as.matrix(swiss[, -1])
> ir.scal<-cmdscale(dist(swiss.x),k=2,eig=7)
> eqscplot(swiss.scal$points,type="n")
> text(swiss.scal$points,labels=as.character(1:nrow(swiss.x)),
  +cex=0.8)
> swiss.sam <- sammon(dist(swiss.x))
Initial stress        : 0.00824
stress after  10 iters: 0.00439, magic = 0.338
stress after  20 iters: 0.00383, magic = 0.500
stress after  30 iters: 0.00383, magic = 0.500
> plot(swiss.sam$points, type="n")
> text(swiss.sam$points, labels=as.character(1:nrow(swiss.x)),
  +cex=0.8)
> swiss.dist <- dist(swiss.x)
> swiss.iso <- isoMDS(swiss.dist)
initial  value 2.979731
iter   5 value 2.431486
iter  10 value 2.343353
final  value 2.339863
converged
> plot(swiss.iso$points, type="n")
> text(swiss.iso$points, labels=as.character(1:nrow(swiss.x)),
  +cex=0.8)
```

Data Description:

Swiss Fertility and Socioeconomic Indicators (1888) Data

Description:

Standardized fertility measure and socio-economic indicators for each of 47 French-speaking provinces of Switzerland at about 1888.

Usage:

```
data(swiss)
```

Format:

A data frame with 47 observations on 6 variables, each of which is in percent, i.e., in [0,100].

[,1] Fertility Ig, ``common standardized fertility measure''
[,2] Agriculture % involved in agriculture as occupation
[,3] Examination % ``draftees'' receiving highest mark on army examination
[,4] Education % education beyond primary school.
[,5] Catholic % catholic (as opposed to ``protestant'').
[,6] Infant.Mortality live births who live less than 1 year.

All variables but `Fertility' give proportions of the population.

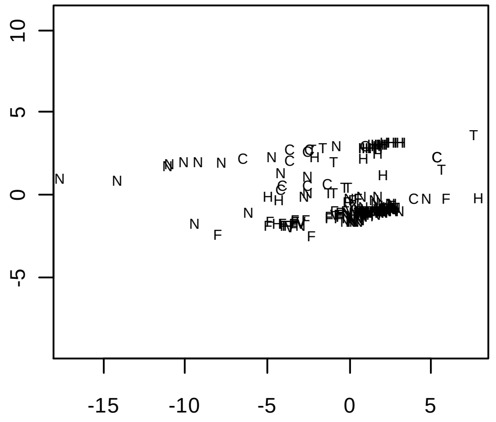
Details:

(paraphrasing Mosteller and Tukey):

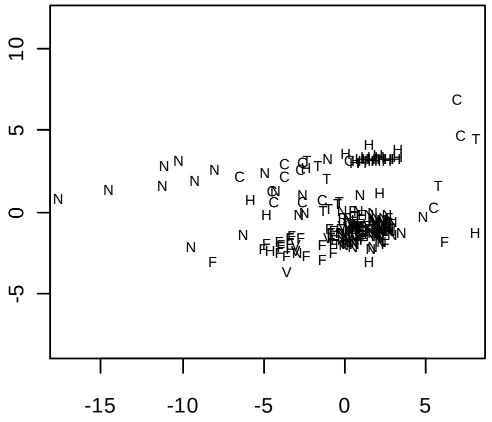
Switzerland, in 1888, was entering a period known as the ``demographic transition''; i.e., its fertility was beginning to fall from the high level typical of underdeveloped countries. The data collected are for 47 seven French-speaking ``provinces'' at about 1888. Here, all variables are scaled to [0,100], where in the original, all but `"Catholic"' were scaled to [0,1].

Source: Project ``16P5'', pages 549-551 in Mosteller, F. and Tukey, J. W. (1977) Data Analysis and Regression: A Second Course in Statistics. Addison-Wesley, Reading, Mass, indicating their source as ``Data used by permission of Franice van de Walle. Office of Population Research, Princeton University, 1976. Unpublished data assembled under NICHD contract number No 1-HD-O-2077.''

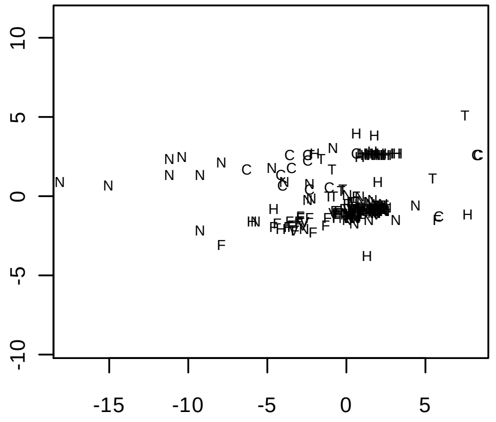
3.8.3 Forensic Glass Data



Classical Scaling



Sammon Mapping



Kruskal

R Code to produce above plots:

```
> data(fgl)
> attach(fgl)
> fgl.dist<-dist(as.matrix(fgl[-40,-10]))
> fgl.scal<-cmdscale(fgl.dist,k=2,eig=7)
> eqscplot(fgl.scal$points,type="n")
> text(fgl.scal$points,labels=c("F","N","V","C","T","H")
+ [fgl$type[-40]],cex=0.7)
> fgl.sam<-sammon(fgl.dist)
Initial stress        : 0.03249
stress after  10 iters: 0.01775, magic = 0.092
stress after  14 iters: 0.01525
> eqscplot(fgl.sam$points,type="n")
> text(fgl.sam$points,labels=c("F","N","V","C","T","H")
+ [fgl$type[-40]],cex=0.7)
> fgl.iso<-isoMDS(fgl.dist)
initial  value 11.518169
iter   5 value 6.353547
iter  10 value 5.993823
iter  15 value 5.913937
final  value 5.888284
converged
> eqscplot(fgl.iso$points,type="n")
> text(fgl.iso$points,labels=c("F","N","V","C","T","H")
+ [fgl$type[-40]],cex=0.7)
```

Data Description:

Measurements of Forensic Glass Fragments

Description:

The `fgl' data frame has 214 rows and 10 columns. It was collected by B. German on fragments of glass collected in forensic work.

Usage:

data(fgl)

Format:

This data frame contains the following columns:

`RI' refractive index; more precisely the refractive index is 1.518xxxx.

The remaining 8 measurements are percentages by weight of oxides.

`Na' sodium
`Mg' manganese
`Al' aluminium
`Si' silicon
`K' potassium
`Ca' calcium
`Ba' barium
`Fe' iron
`type'
The fragments were originally classed into seven types, one of which was absent in this dataset. The categories which occur are window float glass (`WinF': 70), window non-float glass (`WinNF': 76), vehicle window glass (`Veh': 17), containers (`Con': 13), tableware (`Tabl': 9) and vehicle headlamps (`Head': 29).

3.9 Summary and Conclusions

- Multidimensional scaling is concerned with producing a representation of points in low dimensional space starting from a matrix of interpoint distances

- The distances can be a general measure of similarity or equivalently dissimilarity.

- The purpose of multidimensional scaling analyses may be to learn about the measure of (dis)similarity itself or to identify structure in the points themselves.

- Applying the technique to an example where the ‘answer is known’ (e.g. French Departments) gives confidence when applying it to similar but unknown situations.

- The Classical Solution, also known as Principal Coordinate Analysis, gives a method for constructing a display. If the distance matrix is Euclidean it gives an exact representation (up to arbitrary rotations and reflections). Otherwise it can be a starting point for iterative techniques of monotonic regression.

- Scree plots of eigenvalues or stress values can give an informal aid to determining a suitable number of dimensions.

- ♦ Some eigenvalues determined in Principal Coordinate Analysis from non-Euclidean distance matrices will be negative. Care needs to be taken in construction of scree plots in such cases.

- ♦ Principal Coordinate Analysis is the dual of Principal Component Analysis.

- ♦ Other *'unsupervised learning'* techniques for investigating similarities include Cluster Analysis (see Appendix 4) and Kohinen self-organising maps (see Appendix 9)

- ♦ The axes produced by metric or non-metric scaling analysis are ***arbitrary***. It may be possible to assign intuitive interpretations to them by examining the data but these are informal and not a property of the analysis (unlike the axes in PCA or Crimcoords).

- ♦ In **R** and S-PLUS the key commands are `cmdscale()`, `sammon()` and `kruskal()`.

Tasks 6

(see §3.1–§3.5, §5, §6 & A0.4)

1) Continuing Q3 of the tasks 5 (road distances between 12 UK towns)

   i) Determine a configuration of points that will adequately represent the data.

   ii) Construct a two-dimensional map representing the road distances between these towns.

   To do it in **R** you can use the function `cmdscale()` and then plot the results by

   ```
   x<-cmdscale(as.matrix(towns))
   plot(x)
   ```

   The command `as.matrix()` is required for `cmdscale` to recognise the distance matrix. It is also possible in **R** to try two varieties of non-metric scaling (`sammon()` and `isomds()`).

4 Discriminant Analysis

4.0 Summary

A [collection of] technique[s] of use when data are classified into ***known*** groups. Objectives include:

- Effective data display utilising this extra information
- Dimensionality reduction whilst retaining information on differences between groups
- Informal assessment by examination of loadings on nature of differences between groups.
- Classification of future observations into one of the known groups.

Linear Discriminant Analysis can aid all of these objectives but if the primary aim is the final one of classification of future observations then alternatives may do better on particular data sets, e.g. quadratic discriminant analysis or neural networks (see Appendices).

- effectiveness of classification can be assessed by simulation methods (random-relabelling, jackknifing) or classifying further observations of known categories.
- Method requires more observations than variables (i.e. n > p).

4.1 Introduction

So far the data analytic techniques considered have regarded the data as arising from a homogeneous source — i.e. as all one data set. A display on principal components might reveal unforeseen features:– outliers, subgroup structure as well as perhaps singularities (i.e. dimensionality reduction).

Suppose now we know that the data consist of observations classified into k groups (c.f. 1-way classification in univariate data analysis) so that data from different groups might be different in some ways. We can take advantage of this knowledge

- to produce more effective displays
- to achieve greater dimensionality reduction
- to allow informal examination of the nature of the differences between the groups.

Linear Discriminant Analysis finds the best linear combinations of variables for separating the groups, if there are k different groups then it is possible to find k–1 separate linear discriminant functions which partition the *between groups variance* into decreasing order, similar to the way that principal component analysis partitions the *within group variance* into decreasing order. The data can be displayed on these *discriminant coordinates.* ***Boundaries between classification regions are linear.***

4.2 Outline of theory of LDA

The starting point is to define a measure of separation between the known groups:– this is a matrix generalization of the F-statistic used for testing in one-way analysis of variance and is essentially the ratio of *between* group variance to *within* group variance.

Step one is to identify that linear combination of variables which maximizes the F-statistic for testing differences between the groups in one-way analysis of variance on the univariate data. (*cf maximizing variances in derivation of PCA).*

Mathematical theory shews this is achieved by the eigenanalysis of the ratio of *between* group variance to *within* group variance matrix. (*cf the eigenanalysis of the variance matrix in PCA)*

The first eigenvector is the first linear discriminant (also known as Fishers linear discriminant).

Up to this point the mathematical theory is impeccable: subsequent steps and interpretations are ***by analogy with PCA***

> Subsequent eigenvectors are extracted as '2nd, 3rd, …discriminant coordinates (or crimcoords)' and data plotted on these axes though strictly these axes are not orthogonal. ***i.e. the transformation to discriminant coordinates is not just a rotation/reflection.***
>
> Successive eigenvalues (and cumulative proportions etc) are interpreted as 'successive amounts of discrimination achieved' by each axis.

4.3 Preliminaries

4.3.1 Setup:

n_i p-dimensional observations from group G_i; i=1,2,...,k; with $\Sigma n_i = n$.

4.3.2 Notation:

Data matrix for ith group G_i is X_i' (so X_i' is $n_i \times p$), i=1,...,k

(so data are $\{x_{ij}; i=1,...,k;\ j=1,...,n_i\}$, x_{ij} a p×1 vector)

Let $S_i = \frac{1}{n_i-1}(X_i - \overline{X}_i)(X_i - \overline{X}_i)'$ the within group i variance matrix

$$= \frac{1}{n_i-1}\sum_{j=1}^{n_i}(x_{ij} - \overline{x}_i)(x_{ij} - \overline{x}_i)' \quad \text{(so } S_i \text{ is } p \times p\text{).}$$

Let $W = \frac{1}{n-k}\sum_{i=1}^{k}(n_i - 1)S_i$ the [pooled] ***within groups*** variance.

Define $B = \frac{1}{k-1}\sum_{i=1}^{k} n_i(\overline{x}_i - \overline{x})(\overline{x}_i - \overline{x})'$ the ***between groups*** variance.

Here $\overline{x}$ is the overall mean and $\overline{x}_i$ the mean of group G_i, (both p×1) so $X=[X_1:X_2: .... :X_k]$, $\overline{x} = \frac{1}{n}X1_n$ and $\overline{x}_i = \frac{1}{n_i}X_i 1_{n_i}$

$\overline{X}_i = \overline{x}_i 1'_{n_i}$ is the $p \times n_i$ matrix with each row equal to $\overline{x}_i'$

W and B are analogous to the ***within*** and ***between*** groups mean squares in a univariate one-way analysis of variance and if we set p=1 then the usual formulæ for these are retrieved.

Further, it can be shewn that if

$$T=(X-\bar{X})(X-\bar{X})'=\sum_{i=1}^{k}\sum_{j=1}^{n_i}(x_{ij}-\bar{x})(x_{ij}-\bar{x})'$$

(the 'total sum of squares')

then T=(n–k)W+(k–1)B

(which is the 'multivariate analysis of variance' of 'total variance' into 'within' and 'between' components.)

This will be exploited more formally later when considering multivariate analysis of variance (MANOVA) but here we consider just a simple analogy with 1-way analysis of variance.

4.3.3 Motivation:

If we projected all the data into one dimension then we could perform a 1-way analysis of variance and compare the ***between*** groups mean square with the ***within*** groups mean square: we would calculate F-statistic which is the ratio of between to within groups mean squares. If there are large differences between the groups then the between group mean square will be relatively large and so the F-statistic will be large.

As we choose different projections the ratio of *between* to *within* mean squares will vary: from some viewpoints it might look ‘insignificantly small’ and from others it might appear much more appreciable — i.e. from some viewpoints the differences between the groups may not be noticeable and from others the differences may be highlighted.

The objective in determining discriminant coordinates is to choose the projection to highlight the differences between the groups.

4.4 Crimcoords

Project all the p-dimensional data onto vector a_1 (a column p-vector), so the projected data matrix for the ith group G_i is $X_i'a_1$.

Then the within ith group mean square (i.e. the sample variance of the one-dimensional projected data of the ith group) is $a_1'S_ia_1$.

Note that this is a scalar ($1{\times}p{\times}p{\times}p{\times}1=1$)

and $$a_1'S_ia_1 = \tfrac{1}{n_i-1}a_1'\sum_{j=1}^{n_i}(x_{ij}-\bar{x}_i)(x_{ij}-\bar{x}_i)'a_1$$

$$= \tfrac{1}{n_i-1}\sum_{j=1}^{n_i}(a'x_{ij}-a'\bar{x}_i)(a'x_{ij}-a'\bar{x}_i)' = \tfrac{1}{n_i-1}\sum_{j=1}^{n_i}(a'x_{ij}-a'\bar{x}_i)^2$$

Similarly, the within group mean square of the projected data is $a_1'Wa_1$ and the between group mean square is $a_1'Ba_1$.

To highlight the distinction between the groups we want to choose a_1 to maximize $F_1 = \dfrac{a_1'Ba_1}{a_1'Wa_1}$

(i.e. just the usual F-ratio in a classical one-way analysis of variance, though note we have not [as yet] introduced the background of multivariate normality so we will not [as yet] claim that this has a statistical distribution related to the variance-ratio Snedecor F-distribution used to assess significance in a one-way analysis of variance.)

So the problem is to maximize $F_1 = \frac{a_1'Ba_1}{a_1'Wa_1}$ with respect to a_1.

Note that F_1 is a ratio of quadratic forms in a_1, so if a_1 is multiplied by any scalar then the value of F_1 is unaltered. This means that we can impose any scalar constraint on a_1 without altering the maximization problem on F_1, i.e. the maximization problem is invariant with respect to scalar multiplication of a_1.

That is, the problem ‘maximize F_1 with respect to a_1’ is the same as ‘maximize F_1 with respect to a_1 subject to some [convenient to us] scalar constraint on a_1’.

Noting that F_1 is a ratio of 2 quadratic forms, a convenient constraint to impose is that the denominator is one. So, now the problem becomes ‘maximize $F_1 = \frac{a_1'Ba_1}{a_1'Wa_1}$ subject to $a_1'Wa_1=1$,’

i.e. ‘maximize $a_1'Ba_1$ subject to $a_1'Wa_1=1$.’

This introduction of a [non-restrictive] constraint allows us to convert the original maximization problem to a constrained maximization problem which will be solved by using a Lagrange multiplier, thus making it an eigenvalue problem which can be solved easily (numerically at least by **R,** MINITAB, S-PLUS etc).

Introduce a Lagrange multiplier λ_1 and let $\Omega_1 = a_1'Ba_1 - \lambda_1(a_1'Wa_1 - 1)$.

Then $\frac{\partial\Omega_1}{\partial a_1} = 2Ba_1 - 2\lambda_1 Wa_1 = 0$

i.e. $W^{-1}Ba_1 - \lambda_1 a_1 = 0$

i.e. a_1 is an eigenvector of $W^{-1}B$ corresponding to the eigenvalue λ_1, strictly it is a *right* eigenvector since $W^{-1}B$ is not in general symmetric.

To see which eigenvector is needed to maximize F_1,
pre-multiply by $a_1'W$ which gives $a_1'Ba_1 = \lambda_1 a_1'Wa_1 = \lambda_1$

(since $a_1'Wa_1 = 1$ — the original constraint)

so to maximize $a_1'Ba_1$ with $a_1'Wa_1 = 1$ take λ_1 as the *largest* eigenvalue of $W^{-1}B$ and a_1 as the corresponding eigenvector.

Remark: The function $f(x) = a_1'x$ is known as

Fisher's linear discriminant function.

In general $W^{-1}B$ has several non-zero eigenvalues $\lambda_1, \lambda_2, \ldots\ldots \lambda_r$

where $r=\text{rank}(W^{-1}B)=\min(k-1,p)$

(unless there are pathological collinearities in either the data points in one or more groups or in group means)

Now if λ_i and λ_j are distinct eigenvalues of $W^{-1}B$ with [right] eigenvectors a_i and a_j then a_i & a_j are ***not*** necessarily orthogonal (since $W^{-1}B$ is ***not*** necessarily symmetric). However, they have a sort of orthogonality property:

we have $W^{-1}Ba_i - \lambda_i a_i = 0 \Rightarrow Ba_i - \lambda_i Wa_i = 0$

and $W^{-1}Ba_j - \lambda_j a_j = 0 \Rightarrow Ba_j - \lambda_j Wa_j = 0$

so $a_j'Ba_i - \lambda_i a_j'Wa_i = 0$ and $a_i'Ba_j - \lambda_j a_i'Wa_j = 0$

W and B are symmetric; so $a_j'Ba_i = a_i'Ba_j$ and $a_j'Wa_i = a_i'Wa_j$ and $\lambda_i \neq \lambda_j$ so, subtracting the two equations, a_i and a_j have the property that

$$a_j'Wa_i = 0 \text{ for } i \neq j.$$

Definitions: The functions $a_i'x$, i=1,2,...,r are called the **discriminant coordinates** or **crimcoords** and the space spanned by [the first t of] them is called the **discriminant space**. Sometimes these functions are referred to as **canonical variates**, particularly in the context of interpreting the loadings of the variables so as to describe the nature of the difference between the groups, see §6.4.

We can choose a suitable t by examination of the sequence $\lambda_1,\lambda_2,...$ using a scree plot (by analogy with PCA) but the non-orthogonality of the eigenvectors means that the ‘amount of discrimination’ is not partitioned into ‘separate bits’ in contradistinction to principal components and classical scaling.

Examination of the loadings of variables in the crimcoords gives an idea of which variables provide discrimination between groups in an analogous way to the interpretation of principal components.

4.5 Computation

The `MASS` library in **R** provides a function `lda(.)` which provides a full facility, type `help(lda)` to find out more. S-PLUS provides a function `discrim(.)` for performing discriminant analysis but this is not available in **R**. However, the `lda(.)` function in the `MASS` library is in any case superior to this.

MINITAB provides a ready made discriminant analysis module (under Stat>Multivariate>Discriminant Analysis ....) which is very limited in its scope. In particular, it is not possible to produce plots of the data on discriminant coordinates within the module (unlike the equivalent module for principal component analysis). This means that really we need to do the analysis 'from scratch'.

We need the eigenanalysis of $W^{-1}B$, which is not symmetric. MINITAB (and other packages) only compute eigenvalues etc for symmetric matrices but this facility can be used to obtain the anlysis of $W^{-1}B$ since both W and B are symmetric and the eigenanalyses of W and B can be used to derive that for $W^{-1}B$:

Suppose W has eigenvalues $\omega_1,...,\omega_p$

and normalised eigenvectors $v_1,...,v_p$,

Let $\Omega=\text{diag}(\omega_i)=\begin{pmatrix}\omega_1 & & 0\\ & \ddots & \\ 0 & & \omega_p\end{pmatrix}$

and let $V=(v_1,...,v_p)$,

so $v_i'v_j = 0$ if $i\neq j$ and1 if $i=j$,

i.e. $V'V=VV' = I_p$

We have $WV - V\Omega = 0$

so $V'WV = V'V\Omega = \Omega$

and $W=V\Omega V'$ .... the *spectral decomposition* of W…….(**4.5.1**)

Let $\Omega^{½} = \text{diag}(\omega_i^{½}) = \begin{pmatrix}\omega_1^{½} & & 0\\ & \ddots & \\ 0 & & \omega_p^{½}\end{pmatrix}$ (all $\omega_i\geq 0$ since $W\geq 0$)

and $T=V\Omega^{½}V'$ …………………………………………..(**4.5.2**)

(i.e. the 'square root of W', $W^{½}$, so $W=T^2$), then T is symmetric.

Let $B_*=T^{-1}BT^{-1}$ which is symmetric.

Then:

claim: B_* has the same eigenvalues as $W^{-1}B$ and the eigenvectors of $W^{-1}B$ are given by multiplying those of B_* by T^{-1}.

since: Suppose the eigenvalues and eigenvectors of $W^{-1}B$ are λ_i and a_i, i=1,...,p (some of the λ_i may be zero). so $W^{-1}Ba_i - \lambda_i a_i = 0$.

Let the eigenvalues and eigenvectors of B_* be μ_i and b_i (i=1,...,p)

so $B_* b_i - \mu_i b_i = 0$

so $T^{-1}BT^{-1}b_i - \mu_i b_i = 0$

so $T^{-2}BT^{-1}b_i - \mu_i T^{-1}b_i = 0$

so $W^{-1}B(T^{-1}b_i) - \mu_i(T^{-1}b_i) = 0$

which shews that the [right] eigenvalues of $W^{-1}B$ are μ_i and the [right] eigenvectors are $T^{-1}b_i$, i.e. $\lambda_i=\mu_i$ and $a_i=T^{-1}b_i$. □

[This gives a method which can be used in a package which has only basic matrix manipulation facilities such as MINITAB to obtain the eigenanalysis of $W^{-1}B$. In fact, the MANOVA (see a later chapter) routine in MINITAB (ANOVA>Balanced MANOVA... followed by checking the 'Eigen analysis' box on the Results… menu) will produce the required values in the Session window but then they have to be cut&pasted (or read) Stat>Multivariate>Discriminant Analysis into the worksheet].

4.6 Example (Iris Data)

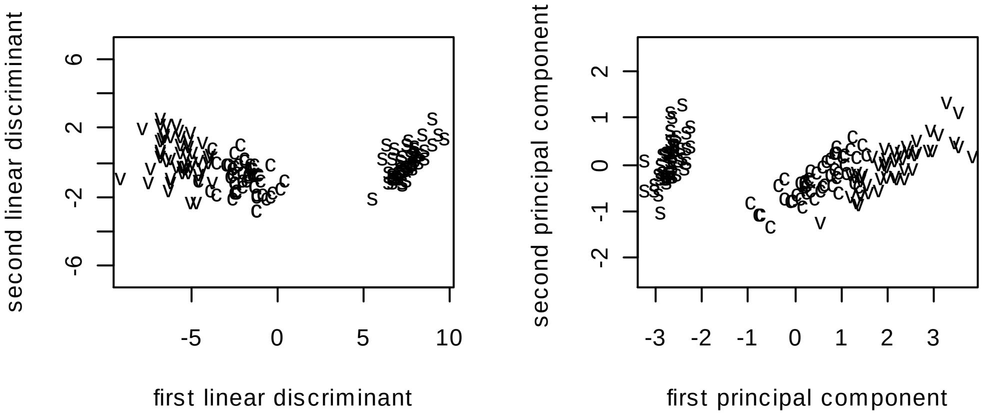
4.6.1 Analysis in R:

Returning yet again to Anderson's Iris data, given below is a record of an **R** session (S-PLUS is essentially the same) to produce a display on crimcoords and compare it with the display on principal components

```
> load("irisnf.Rdata")
> ir<-as.matrix(irisnf[,-5])
> ir.species<- factor(c(rep("s",50),rep("c",50),rep("v",50)))
> ir.lda<-lda(ir,ir.species)
```

Now plot the data on discriminant coordinates and compare with a principal component plot:

```
> ir.ld<-predict(ir.lda,dimen=2)$x
> eqscplot(ir.ld,type="n",
+ xlab="first linear discriminant",
+ ylab="second linear discriminant")
> text(ir.ld,labels=as.character(ir.species))
> eqscplot(ir.pc,type="n",
+ xlab="first principal component",
+ ylab="second principal component ")
> text(ir.pc,labels=as.character(ir.species))
```



There is perhaps a very slightly better separation between the groups labelled `v` and `c` in the left hand plot than in the right hand one.

4.6.3 Prior Probabilities

In **R** and S-PLUS the function `lda()` assumes by default that prior probabilities of group membership are equal to the observed proportions in the training data of the k groups. To over-ride this the call to `lda()` should include a parameter `prior=c(p`$_1$`, p`$_2$`, ..., p`$_k$`)`. In the example on the iris data the training data had 50 observations in each group and so the prior probabilities were taken, by default, to be equal. The effect of taking prior probabilities different from the observed proportions is to alter the estimate of the within-groups variance W defined in §4.3.2 so that instead of using weights $(n_i – 1)/(n – k)$ it uses weights pi, i.e. W is estimated as $W = \sum_{i=1}^{k} p_i S_i$ instead of $\frac{1}{n-k}\sum_{i=1}^{k}(n_i - 1)S_i$.

However the major difference is in the function `predict.lda()` where by default the prior probabilities are taken from those in the original call to `lda()`. This can be over-ridden by specifying `prior` as something different. This can be useful if there is good prior information on group membership probabilities of test data. Similar comments apply to the function `qda()` which performs quadratic discriminant analysis (see Appendices).

4.7 Application to informal (Data Analytic) Classification

Suppose we have k groups of ‘reference’ data and a set of r unknown points u_j; j=1,...,r., and we wish to classify each of the r points into one of the k groups, (e.g. multivariate data from reliably diagnosed patients with one of k different conditions, wish to diagnose r new patients on whom the same measurements are made).

Suppose we have calculated the k reference group means or centroids ($\bar{x}_i$; i=1,...,k). It is natural to assign an observation u to that group to which it is nearest in some sense.

We might measure the distance of u from the group i centroid as D(i) where $D^2(i) = (u - \bar{x}_i)'M(u - \bar{x}_i)$ where M is some suitable weighting matrix which we require to be positive semi-definite (M≥0) to ensure that $D^2(i) \geq 0$.

There are many possible choices for the weight matrix M:

Simple is (i) If we take M=I_p then D is Euclidean distance.

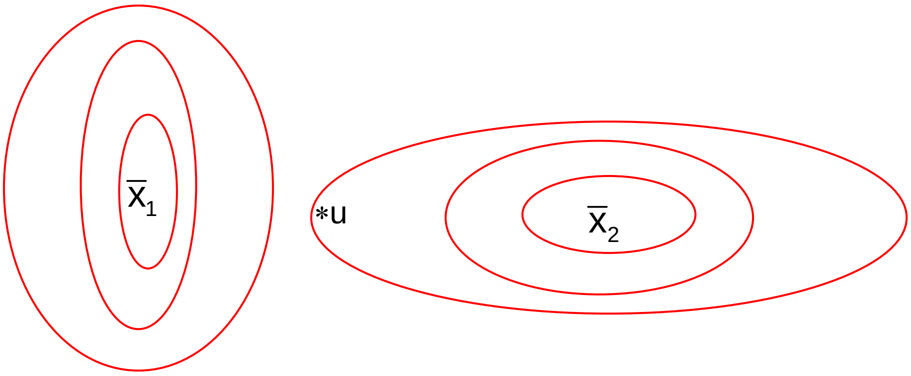
$$D^2(i) = (u - \bar{x}_i)'(u - \bar{x}_i))'$$

Better is (ii) take M=S_i^{-1} giving

$$D^2(i) = (u - \bar{x}_i)'S_i^{-1}(u - \bar{x}_i)$$

(which requires n_i>p to ensure S_i non-singular)

e.g. p = 2, two clusters of points with 'densities' as shewn:



the point u would be assigned to group 1 under (i) Euclidean distance but to 2 under (ii) (the group i Mahalanobis distance). The second choice is more reasonable since u is well within the range of the observed data from group 2 and well away from the data cloud of group 1.

(iii) Take M=A_tA_t' where A_t is the matrix of the first t eigenvectors of $W^{-1}B$, this gives

$$D^2(i) = (u-\overline{x}_i)'A_tA_t'(u-\overline{x}_i) = [A_t'(u-\overline{x}_i)]'[A_t'(u-\overline{x}_i)]$$

and so is equivalent to projecting the data into discriminant space and then using Euclidean distance.

(iv) Take M=W^{-1} giving

$$D^2(i) = (u-\overline{x}_i)'W^{-1}(u-\overline{x}_i)$$

which is the 'average' of the measures in (ii) — sensible if there are good reasons for expecting the covariances in the different groups to be similar. This is known as the Mahalanobis distance of u from the group mean

All of these (and many more) are used in practice and they may give slightly different results in classification. It may be difficult to determine precisely which criterion ready-made analyses in packages actually use. The most commonly used criteria are (iii) (discriminant space distance) and (iv) the Mahalanobis distance. If the different groups have substantially different variances then (ii) is a possibility but extensions of the method to 'quadratic discriminant analysis' may also be useful (see e.g. MINITAB options or function `qda(.)` in **R** or S-PLUS).

4.8 Summary and Conclusions

- Crimcoords (or discriminant coordinates) highlight differences between known groups of observations. Displays on these are strictly distorting the data slightly since the axes are not orthogonal but are conventionally drawn as such.

- The first Crimcoord is also known as Fisher's Linear Discriminant Function

- Interpretations of factor loadings and use of scree plots is performed by analogy with PCA.

- Referral to crimcoords allows informal classification of other points of unknown origin.

- Other informal methods for classification are used, such as classifying by minimum Mahalanobis distance.

- Some further illustrations of discriminant analysis (linear and quadratic) are given in Appendix 1.

- Other techniques for investigating discrimination between known categories are *Logistic Regression*, Classification Trees (see Appendix 7) and Neural Networks (see Appendix 8).

- Method requires more observations than variables (i.e. n > p).

- Also note that equations 4.5.1 and 4.5.2 in §4.5 define the ***square root*** of a symmetric matrix via its ***spectral decomposition***. This is used later in establishing standard properties of the multivariate normal distribution.

Exercises 2

1) The data given in file *dogmandibles.*∗ (in various formats) are extracted, via Manly (1994), from Higham etc (1980), *J.Arch.Sci*, 149–165. The file contains 9 measurements of various dimensions of the mandibles of 5 canine species as well as records of the sex and the species, eleven variables in total. These are

X_1: length of mandible

X_2: breadth of mandible

X_3: breadth of articular condyle

X_4: height of mandible below first molar

X_5: length of 1st molar

$X_{6:}$ breadth of 1st molar

X_7: length between 1st to 3rd molar inclusive (1st to 2nd for Cuons)

X_8: length between 1st to 4th premolar inclusive

X_9: breadth of lower canine

X_{10}: gender (1 ≡ male, 2 ≡ female, 3 ≡ unknown)

X_{11}: species (1 ≡ modern dog from Thailand, 2 ≡ Golden Jackal,
3 ≡ Cuon, 4 ≡ Indian Wolf, 5 ≡ Prehistoric Thai dog)

All measurements are in mm; molars, premolars and canines are types of teeth; an articular condyle is the round knobbly bit in a joint; a Cuon, or Red Dog, is a wild dog indigenous to south east Asia and notable for lacking one pair of molars.

i) Ignoring the group structure, what interpretations can be given to the first two principal components?

ii) Construct a display of the measurements on the first two crimcoords, using different symbols for the five different groups.

iii) If the linear discriminant analysis were performed on the data after transformation to the full set of nine principal components what differences (if any) would there be in the plot on crimcoords and the eigenvalues and eigenvectors of the matrix $W^{-1}B$?

iv) Which group is separated from the other four by the first crimcoord?

v) Which group is separated from the other four by the second crimcoord?

vi) Which group is separated from the other four by the third crimcoord?

vii) What features of the mandibles provide discrimination between the various species?

Notes: if you do this in Minitab then you need to use the slicktrick outlined on p135 of the course notes and use the balanced manova facility within anova to obtain the eigenvectors and then cut&paste these into the worksheet, followed by a couple of steps of matrix manipulation. In S-PLUS the required plots can be obtained from the lda menu. Interpretation of the crimcoords is a little easier after looking at the interpretations of the PCs.

2) ⋆ The question of prime interest in the study of canines was related to an investigation of the origin of the prehistoric dogs. Try calculating the discriminant analysis based on the four groups of modern canines and then plot the prehistoric cases on the same coordinate system a (c.f. informal data classification method (iii) on p140 of course notes) and seeing to which of the modern groups the majority of the prehistoric are closest.

(The interpretation of the results of this exercise are **within** the scope of PAS465; the required computer skills to produce it are useful but a little beyond the scope of PAS465, i.e. if you do not attempt it ensure that you look carefully at the printed solution in due course.)

5★ Multivariate Regression Analysis

5.0 Introduction

This chapter provides a brief introduction to the extension of univariate regression methods to the multivariate case. There is little new in terms of statistical concepts and the univariate mathematical development carries through almost unchanged symbolically to the multivariate case. Formal statistical tests are not described here but are essentially applications of multivariate analysis of variance considered in Chapter 8.

First, recall the univariate case with just one independent variable X. The parameters α and β in the model $E[Y] = \alpha + \beta X$ or $y_i = \alpha + \beta x_i + \varepsilon_i$ where $\varepsilon_i \sim N(0,\sigma^2)$, i.i.d., have least squares estimates (and maximum likelihood estimates) obtained by minimizing $\sum_{i=1}^{n}(y_i - \hat{\alpha} - \hat{\beta}x_i)^2$ which yields $\hat{\alpha} = \bar{y} - \hat{\beta}\bar{x}$ and $\hat{\beta} = \frac{\sum(x_i-\bar{x})(y_i-\bar{y})}{\sum(x_i-\bar{x})^2}$. Note that the sample correlation coefficient is slightly different, $\rho_{XY} = \frac{\sum(x_i-\bar{x})(y_i-\bar{y})}{\sqrt{\sum(x_i-\bar{x})^2\sum(y_i-\bar{y})^2}}$. Note also that the correleation coefficient is symmetric in X and Y but the regression coefficient is not, emphasizing that the correlation coefficient is used for investigating the **relationship** between the X and Y variables whilst regression analysis investigates the **dependence** of the *dependent variable* Y upon the *independent variable* X, (perhaps with the aim of predicting one from the onther).

If there are several independent variables $X_1, X_2, \ldots, X_q$ then the appropriate model can be written as $E[Y] = X\beta$ or $Y = X\beta + \varepsilon$ where $\varepsilon \sim N_n(0, \sigma^2 I_n)$ and $\beta' = (\beta_1, \beta_2, \ldots, \beta_q)$ and X is the $n \times q$ matrix of observations of the q independent variables (and usually the first variable X_1 would be taken to be the unit vector 1_n, so the model includes a constant term). Note here the use of X as the $n \times q$ matrix of observations rather than X′ as in other chapters so as to conform with general usage in presentation of regression.

Minimizing the sum of squares of the residuals, i.e. Ω=(Y – Xβ)′(Y –Xβ), yields, after differentiating Ω with respect to β, 2X′(Y – Xβ) = 0 and thus $\hat{\beta} = (X'X)^{-1}X'Y$ (provided n>p or, more exactly, provided X′X is non-singular). The extension to the case when Y is a matrix of vector observations is now straightforward.

5.1 Multivariate Regression

The p-dimensional multivariate linear model is $E[Y] = X\beta$ or $Y = X\beta + \varepsilon$ where where Y is a $n\times p$ matrix of n observations of p-dimensional varaibles, X is $n\times q$, β is $q\times p$ and ε is an $n\times p$ matrix random variable. The matrix of residual sums of squares and cross products is $\Omega = (Y - X\beta)'(Y - X\beta)$; differentiating Ω with respect to β yields $2X'(Y - X\beta) = 0$ and thus $\hat{\beta} = (X'X)^{-1}X'Y$ (provided n>p or, more exactly, provided $X'X$ is non-singular). Note that Ω is a matrix so its differentiation may need to be taken 'on trust' as also the fact that this minimizes both the determinant and the trace of Ω. This means that the p individual $\hat{\beta}_i$ in $\hat{\beta}$ are identical to those which would be obtained by separate [multiple] regressions of each Y_i on the independent variables $X_1, X_2, \ldots, X_q$. This is true of least squares estimation and of maximum likelihood estimation if the errors are assumed to be Normal with independence from one observation to the next, though they could be correlated between one variable to the next on the same observation, i.e. that the rows of ε are i.i.d $N_p(0, \Sigma)$. Predicted values of Y are given by $\hat{Y} = X\hat{\beta}$ and Σ is estimated as $\hat{\Sigma} = (Y - X\hat{\beta})'(Y - X\hat{\beta})/(n - q - 1)$.

5.2 Example (Beef & Pork Consumption)

This example is discussed by Cox (2005) and the data are taken from the Data and Story Library at

http://lib.stat.cmu.edu/DASL/Datafiles/agecondat.html

The data are

YEAR	PBE	CBE	PPO	CPO	PFO	DINC	CFO	RDINC	RFP
1925	59.7	58.6	60.5	65.8	65.8	51.4	90.9	68.5	877
1926	59.7	59.4	63.3	63.3	68.0	52.6	92.1	69.6	899
1927	63.0	53.7	59.9	66.8	65.5	52.1	90.9	70.2	883
1928	71.0	48.1	56.3	69.9	64.8	52.7	90.9	71.9	884
1929	71.0	49.0	55.0	68.7	65.6	55.1	91.1	75.2	895
1930	74.2	48.2	59.6	66.1	62.4	48.8	90.7	68.3	874
1931	72.1	47.9	57.0	67.4	51.4	41.5	90.0	64.0	791
1932	79.0	46.0	49.5	69.7	42.8	31.4	87.8	53.9	733
1933	73.1	50.8	47.3	68.7	41.6	29.4	88.0	53.2	752
1934	70.2	55.2	56.6	62.2	46.4	33.2	89.1	58.0	811
1935	82.2	52.2	73.9	47.7	49.7	37.0	87.3	63.2	847
1936	68.4	57.3	64.4	54.4	50.1	41.8	90.5	70.5	845
1937	73.0	54.4	62.2	55.0	52.1	44.5	90.4	72.5	849
1938	70.2	53.6	59.9	57.4	48.4	40.8	90.6	67.8	803
1939	67.8	53.9	51.0	63.9	47.1	43.5	93.8	73.2	793
1940	63.4	54.2	41.5	72.4	47.8	46.5	95.5	77.6	798
1941	56.0	60.0	43.9	67.4	52.2	56.3	97.5	89.5	830

where

1. PBE = Price of beef (cents/lb)
2. CBE = Consumption of beef per capita (lbs)
3. PPO = Price of pork (cents/lb)
4. CPO = Consumption of pork per capita (lbs)
5. PFO = Retail food price index (1947-1949 = 100)
6. DINC = Disposable income per capita index (1947-1949 = 100)
7. CFO = Food consumption per capita index (1947-1949 = 100)
8. RDINC = Index of real disposable income per capita (1947-1949 = 100)
9. RFP = Retail food price index adjusted by the CPI (1947-1949 = 100)

For this illustration we will consider the dependence of consumption of beef and pork, i.e. Y=[CBE, CPO], upon the prices of beef and pork and disposable income and include a constant term in the regression, so X=[1_{17}, PBE, PPO, DINC].

So p = 2, q = 4, n=17

$$\mathbf{Y} = \begin{bmatrix} 58.6 & 65.8 \\ 59.4 & 63.3 \\ 53.7 & 66.8 \\ 48.1 & 69.9 \\ 49.0 & 68.7 \\ 48.2 & 66.1 \\ 47.9 & 67.4 \\ 46.0 & 69.7 \\ 50.8 & 68.7 \\ 55.2 & 62.2 \\ 52.2 & 47.7 \\ 57.3 & 54.4 \\ 54.4 & 55.0 \\ 53.6 & 57.4 \\ 53.9 & 63.9 \\ 54.2 & 72.4 \\ 60.0 & 67.4 \end{bmatrix} \qquad \mathbf{X} = \begin{bmatrix} 1 & 59.7 & 60.5 & 51.4 \\ 1 & 59.7 & 63.3 & 52.6 \\ 1 & 63.0 & 59.9 & 52.1 \\ 1 & 71.0 & 56.3 & 52.7 \\ 1 & 71.0 & 55.0 & 55.1 \\ 1 & 74.2 & 59.6 & 48.8 \\ 1 & 72.1 & 57.0 & 41.5 \\ 1 & 79.0 & 49.5 & 31.4 \\ 1 & 73.1 & 47.3 & 29.4 \\ 1 & 70.2 & 56.6 & 33.2 \\ 1 & 82.2 & 73.9 & 37.0 \\ 1 & 68.4 & 64.4 & 41.8 \\ 1 & 73.0 & 62.2 & 44.5 \\ 1 & 70.2 & 59.9 & 40.8 \\ 1 & 67.8 & 51.0 & 43.5 \\ 1 & 63.4 & 41.5 & 46.5 \\ 1 & 56.0 & 43.9 & 56.3 \end{bmatrix}$$

The least squares estimate of β is $\hat{\beta} = \begin{pmatrix} 101.40 & 79.60 \\ -0.753 & 0.153 \\ 0.254 & -0.687 \\ -0.241 & 0.283 \end{pmatrix}$

[**Task:** check these multivariate calculations and also check that the same answers would be obtained by using separate univariate linear regressions.]

The advantage of using the multivariate approach is that the correlation between beef and pork consumption is modelled and this allows construction of simultaneous confidence intervals.

Σ is estimated as $\hat{\Sigma} = \begin{pmatrix} 4.412 & -7.572 \\ -7.572 & 16.835 \end{pmatrix}$.

5.3 Summary and Conclusions

- Multivariate regression models the *dependence* of a p-dimensional random variable Y upon a q-dimensional variable X.
- The individual equations relating the p individual Y variables upon the X variables are identical to those obtained by separate univariate regressions of the Y_i upon the X variables.
- The advantage of the multivariate approach is that the correlation structure of the *dependent variables* is modelled and this allows construction of simultaneous confidence intervals.
- Formal tests of hypothesis (e.g. whether $\beta = 0$) are available.
- It does not matter whether $q < p$ or $q > p$.
- It is required that $n > \max(p, q)$ and that $X'X$ is non-singular.

Tasks 7

1) Retrieve the data on beef and pork consumption referenced in §5.2 and verify the calculations given in §5.2 using R or S-plus. Predict the consumption of beef and pork if the prices in cents/lb are 79.3, 41.2 and the disposable income index is 40.4.

2) Retrieve the dataset chap8headsize referenced in §6.3 and calculate the estimates of the least squares multivariate regression parameters β of length and breadth of heads of first sons upon those of second sons. Is it possible to deduce from these results the estimates for the regression of second sons upon the first?

3) Read the section on Maximum Likelihood Estimation in Background Results (Appendix 0.4). This material will be required and used extensively in Chapter 8.

6★ Canonical Correlation Analysis

6.0 Introduction

This chapter provides a brief introduction to the extension of methods of correlation analysis to the multivariate case. The similarity to and distinction between this and multivariate regression analysis is the same as that in the unviariate case. Canonical correlation analysis aims to investigate the relationship between two sets of variables — here referred to as the X variables and the Y variables rather than the *dependence* of one set upon the other which is the purpose of regression analysis. This means that canonical correlation analysis is symmetric in the X and Y variables though we can accommodate their having different dimensions, say p and q respectively. As with regression methods we require at least more observations than dimensions, i.e. n > max(p, q) and in particular we need various matrices to be non-singular and so possess inverses.

The approach is to find *linear* combinations of the X and Y variables that have maximum correlation with each other amongst all such linear combinations. This is analgous to principal component analysis where linear combinations of variables with maximal variance are sought.

6.1 Derivation of results

Suppose $X=(X_1, X_2,\ldots, X_q)$ and $Y=(Y_1, Y_2,\ldots, Y_p)$ and n observations are available on each (measured simulataneously). Suppose $\mathrm{var}(X) = \Sigma_{XX}$ and $\mathrm{var}(Y) = \Sigma_{YY}$ and $\mathrm{cov}(X,Y) = \Sigma_{XY}$ where these may be taken as either the population [theoretical] values or the sample values based upon the n observations. Let a and b be p- and q-vectors respectively and consider the correlation between $a'X$ and $b'Y$, noting that their variances are $a'\Sigma_{XX}a$ and $b'\Sigma_{YY}b$ respectively. This is $\rho_{XY} = a'\Sigma_{XY}b/\sqrt{}(a'\Sigma_{XX}ab'\Sigma_{YY}b)$.

This is independent of the scale of both a and b so we may impose scale constraints on these without loss of generality, most conveniently to ensure the denominator is unity, i.e. $a'\Sigma_{XX}a = b'\Sigma_{YY}b = 1$.

Maximizing ρ_{XY} subject to these constraints (by introducing λ and μ as Lagrange multipliers) yields $\Sigma_{XY}b - \lambda\Sigma_{XX}a = 0$ and $\Sigma_{XY}a - \mu\Sigma_{YY}b = 0$, after differentiating ρ_{XY} with respect to a and b and setting the results equal to zero. Premultiplying these by a' and b' respectively and recalling the assumed constraints shews that $\lambda = \mu = a'\Sigma_{XY}b = \rho_{XY}$. Premultiply the first equation by $\rho_{XY}(\Sigma_{XX})^{-1}$ and the second by $(\Sigma_{XX})^{-1}\Sigma_{XY}(\Sigma_{YY})^{-1}$ and adding the results gives $(\Sigma_{XX})^{-1}\Sigma_{XY}(\Sigma_{YY})^{-1}\Sigma_{XY}a - (\rho_{XY})^2a = 0$ shewing that a is an eigenvector of $(\Sigma_{XX})^{-1}\Sigma_{XY}(\Sigma_{YY})^{-1}\Sigma_{XY}$ with eigenvalue $(\rho_{XY})^2$. A further step shews that to maximize the correlation we need to take the largest eigenvalue. Similar analysis shews that b is the eigevector corresponding to the largest eigenvalue of $(\Sigma_{YY})^{-1}\Sigma_{XY}(\Sigma_{XX})^{-1}\Sigma_{XY}$ which is also $(\rho_{XY})^2$, (note the symmetry in X and Y).

It is easy to shew that further eigenvectors $a_2, a_3,\ldots$ and $b_2, b_{3,}\ldots$ maximise the correlation between linear functions of the X and Y variables subject to the constraints of orthogonality with earlier ones.

6.3 Example (head sizes of sons)

This example is discussed by Everitt (2005) and the data are available from the website linked from the course webpage and also from these notes in section 0.4. They are dataset `chap8headsize`.

The data give the length (X_1) and breadth (X_2) of first sons and Y_1 and Y_2 the same measurements for second sons.

Analysis gives

$$(\Sigma_{XX})^{-1}\Sigma_{XY}(\Sigma_{YY})^{-1}\Sigma_{XY} = \begin{pmatrix} 0.323 & 0.317 \\ 0.302 & 0.302 \end{pmatrix}$$

$$(\Sigma_{YY})^{-1}\Sigma_{XY}(\Sigma_{XX})^{-1}\Sigma_{XY} = \begin{pmatrix} 0.301 & 0.300 \\ 0.319 & 0.323 \end{pmatrix}$$

Both of these have eigenvalues 0.62 and 0.0029 (**Task:** check this in **R** or S-PLUS), giving canonical correlations of 0.7885 and 0.0537. The eigenvectors are $a_1' = (0.727, 0.687)$, $a_2' = (0.704, -0.710)$, $b_1' = (0.684, 0.730)$ and $b_2' = (0.709, -0.705)$ [noting that Table 8.3 on p164 of Everitt (2005) is almost totally incorrect].

The first two canonical variates are essentially averages of length and breadth and so proportional to circumference, shewing that the major characteristic shared by brothers in this respect is overall head size, with correlation estimated as 0.79. The second canonical variates are contrasts in length and breadth and so reflect shape. The correlation between the shapes of elder and younger brothers is 0.05 and thus whilst being highly correlated in overall size the shapes are virtually unrelated.

6.4 Further example of interpretation of loadings

This example is also discussed by Everitt (2005) and further details of sample correlation matrices etc are given there. The data arise from a study of depression in 294 people in Los Angeles. The variables measured were CESD (a composite score measuring level of depression; high scores indicating low depression), Health (an overall self-perceived health score; high good), Gender (female = 1), Age, Education Level (high meaning highly), Income. The interest is in the relationship between the first two (which are 'health variables) and the final four social-demographic variables. The first pair of canonical variates is

$$a_1 = 0.461\ \text{CESD} - 0.900\ \text{Health}$$

$$b_1 = 0.024\ \text{Gender} + 0.885\ \text{Age} - 0.402\ \text{Education} + 0.126\ \text{Income}$$

which have correlation 0.409 and the second pair are

$$a_2 = -\ 0.95\ \text{CESD} - 0.32\ \text{Health}$$

$$b_2 = 0.62\ \text{Gender} + 0.63\ \text{Age} - 0.65\ \text{Education} + 0.82\ \text{Income}$$

which have correlation 0.26.

The interpretation of these is that relatively older people (both M & F) are associated with low depression levels but perceive their health as relatively poor, while relatively younger people with good education tend to the opposite health perception. Looking at the second pair of canonical variates, the interpretation is that relatively young, uneducated, poor females are associated with higher depression scores and to a lesser extent with poorly perceived health.

6.4 Further comments

- It may be shewn that if one of the variables, say the Y variable, is a group indicator or set of binary dummy variables then the canonical variates of the X variables are precisely the discriminant functions between the groups. This is the reason for the latter sometimes being referred to as canonical variates — they are the linear combinations of the X variables that 'most highly correlate' with the group structure, i.e. discriminate between them. The fact that a different scaling constraint is used in the analysis is immaterial since the result is invariant to scale.

- The choice of sign for the eigenvectors is arbitrary, as with PCA.

- Plots of the data referred to the canonical variates (either a_i *vs* a_j for just the X variables (likewise for the Y variables) or a_i *vs* b_i for all of the variables) may be useful ways of displaying the data to investigate structure. [**Task:** consider what sort of features these two possibilities might reveal, (**hint:** note the link with lda)].

- The number of non-zero eigenvalues (i.e. canonical correlations) is at most min(p, q).

- As with multivariate regression we require various matrices to be non-singular, in particular that n > max(p, q)

- Interpretation of loadings in the canonical variates is similar to that in PCA and LDA and gives insight into aspects of the data structure.

- **R** functions for performing canonical correlation analysis are `cancor(.)` in the `stats`

- library and [better] `cc(.)` in the `CCA` package.

7★ Partial Least Squares

The methods of linear discriminant analysis (LDA), multivariate regression analysis and canonical correlation analysis (CCA) considered in the last three chapters all required more observations than variables measured. This contrasts with the exploratory technique of principal component analysis (PCA) in chapter 2 where there is no such restriction. In PCA if there are fewer observations than variables then the last few eigenvalues (and hence the corresponding eigenvectors or 'aspiring principal components') are non-informative.

In the contexts of the development of these methods and appreciating the computational restrictions of the time in the mid-twentieth century the 'n > p' requirement was not unduly restrictive since typically the data sets considered were small; n a few hundred at most say and p a few tens say (even as many as ten was formidable computationally). It was generally easy to obtain more observations (interview a few more people or measure a few more iris flowers) or else drop a few variables from consideration by using expert knowledge (e.g. that certain variables are more or less useful than others to the purpose at hand).

However, in the past thirty years or so this situation has changed and it is now commonplace to encounter data sets where p >> n and it is not easy to reduce the number of variables or measure more objects. For example in measuring gene expression levels (or abundances of proteins or metabolites or…) it is routine to measure thousands of variables simultaneously on relatively few subjects. Measurement of each variable (i.e. each gene) may be cheap but each high-dimensional observation is expensive. If 13,000 genes are measured on 100 samples then it is not realistic to reduce the number of variables to fewer than 100, especially if the objective of the study is to discover which genes are the most important and there is no existent expert knowledge in the area (hence the reason for the study), nor is it realistic to obtain more samples if each costs some tens of dollars or pounds or euros or … .

However, the underlying questions of what is the relationship between some dependent variable and the set of independent variables and how can we discriminate between groups of subjects are still of interest but if $n < p$ then methods of multivariate regression and LDA are not available. Consequently there is a need for techniques which address these problems directly and which are immune to the singularity of the matrices required in regression and discriminant analysis. This chapter considers such methods which go under the general (but not very descriptive) name of *partial least squares*.

In passing, an obvious approach might be to reduce the dimensionality with PCA as a preliminary step. That is, if $p > n$ then instead of attempting to use all $X_1, X_2, \ldots X_p$ in the regression analysis (noting that this will fail if $p > n$ because $X'X$ is singular) one could regress the dependent variable Y on the first k principal components of the X_i where $k < n < p$. This approach is particularly useful if $X'X$ is singular because of inherent multicollinearities in the X_i rather than because there are too few observations. In such cases the first few PCs corresponding to the non-zero eigenvalues genuinely contain all available information. In more general cases however the drawback is that the PCs are not derived with any regard to the dependence of the Y variable(s) upon the X_i. Similar drawbacks are present in dimensionality reduction by PCA as a preliminary to LDA though in both situations it is a useful technique to try since it is quick and easy and the methods of PLS (Partial Least

Squares) need a little more effort, at least currently and especially in terms of interpretation.

PLS is a dimensionality reduction method with components chosen with the response variable kept in mind, where the response variable may be a continuous dependent variable (possibly multivariate) [*PLS Regression*] or a group indicator [*PLS Classification*]. Essentially, PLS obtains linear components from the high-dimensional data which maximize *covariance* with the response variable. Unlike correlations, covariances require no inversion of matrices and so avoid problems of singularity of matrices calculated when n < p.

Facilities for implementation of PLS are increasingly widely available and MINITAB and SAS provide built-in facilities. In the **R** system there are several contributed packages and these are well-documented with examples and references.

Brief accounts of PLS are given in Cox (2005), p.190 and in Krzanowski & Marriott (1995), Vol. 2, p.111. Some readable references with applications are given in Boulesteix, A.L. (2004) *PLS dimension reduction for classification of microarray data*, **Statistical Applications in Genetics and Molecular Biology, Vol. 3 Issue 1, Article 33,** and Boulesteix, A.L. a& Strimmer, K. (2006), *Partial Least Squares: a versatile tool for the analysis of high-dimensional genomic data*, **Briefings in Bioinformatics.**

Anne-Laure Boulesteix is also the lead author of the **R** package ***plsgenomics*** which is available from the **R** website which provides routines **`pls.regression(.)`** and **`pls.lda(.)`.**

Further details and examples can be found in the refernces above.

8 Statistical Analysis of Multivariate Data

8.0 Introduction

So far, the course has considered only *data-analytic* techniques:– methods of dimensionality reduction — data display, investigation of subgroup structure etc — which depend only upon the structure of the data themselves and not upon assumptions on the form of the generating process of the data. Such methods may be a useful [preliminary??] step in the analysis — they may simplify later analyses and are not prone to failures in assumptions (since none are made!) and they provide an invaluable intuitive understanding of the data. This section considers more formal statistical models and techniques for the analysis of multivariate data.

The section starts with the definition of the p-dimensional multivariate normal distribution and its basic properties (mean, variance and sampling properties such as maximum likelihood estimation). This allows the construction of likelihood ratio tests and thus the extension to several dimensions of the routine one-dimensional tests such as t-tests and analysis of variance.

N. B. If you want to check the basic ideas of maximum likelihood estimation and likelihood ratio tests then you should read **Appendix 0 (Background Results)**.

Also considered are tests of more complex hypotheses, which can only arise in multidimensions, such as whether the population mean is somewhere on the unit sphere. This requires use of Lagrange multipliers to maximize likelihoods subject to constraints. For this section you should read the material on Generalized Likelihood Ratio Tests given in the Background Results

The second important topic of the section is the introduction of a new method for constructing hypothesis tests (*the Union-Intersection principle)* which in some circumstances can give a different form of test from a likelihood ratio test and in others provide a useful additional interpretation of likelihood ratio tests. This topic links in with the idea of projecting data into one dimension, choosing the dimension appropriately, which was encountered in the construction of principal components and crimcoords.

8.1 The Multivariate Normal Distribution

8.1.1 Definition

The random p-vector x has a p-dimensional Multivariate Normal distribution (with mean μ and variance Σ, μ a column p-vector, Σ a $p\times p$ symmetric non-singular positive definite matrix) if the probability density function (p.d.f.) of x is

$$f_x(x)=\frac{1}{(2\pi)^{p/2}|\Sigma|^{1/2}}\exp\{-\tfrac{1}{2}(x-\mu)'\Sigma^{-1}(x-\mu)\} \quad \text{(where } |\Sigma|=\det(\Sigma)\text{)}$$

and we write $x\sim N_p(\mu,\Sigma)$.

8.1.2 Standardization

Suppose $x\sim N_p(\mu,\Sigma)$ and $y=\Sigma^{-1/2}(x-\mu)$, where $\Sigma^{-1/2}$ is as defined earlier (see §4.5, eqs 4.5.1 & 4.5.2), then $(x-\mu)'\Sigma^{-1}(x-\mu)$ = $y'y$ = $\sum_{i=1}^{p} y_i^2$. Now the density of y is

$$f_y(y)=\frac{1}{(2\pi)^{p/2}|\Sigma|^{1/2}}\exp\{-\tfrac{1}{2}y'y\}.J_{xy}$$

where J_{xy} is the Jacobean of the transformation given by $\left|\frac{dx}{dy}\right|$ where $\frac{dx}{dy}$ is the $p\times p$ matrix with (i,j)th element $\frac{dx_i}{dy_j}$.

Now $y=\Sigma^{-1/2}(x-\mu)$ so $x=\Sigma^{1/2}y+\mu$ so $\frac{dx_i}{dy_j}=(\Sigma^{1/2})_{ij}$ and so $\frac{dx}{dy}=\Sigma^{1/2}$

and $J_{xy}=|\Sigma|^{1/2}$ giving $f_y(y)=\frac{1}{(2\pi)^{p/2}}\exp\{-\tfrac{1}{2}\sum_{i=1}^{p}y_i^2\}$

Thus, the y_i's are independent univariate N(0,1);

[& notice therefore that $f_y(y)$ is > 0, integrates to 1 and so is a genuine p.d.f. and so therefore $f_x(x)$ is a p.d.f. also]

8.1.3 Mean & Variance

Now if $x \sim N_p(\mu,\Sigma)$ and $y=\Sigma^{-½}(x-\mu)$ then

$$E[y]=\Sigma^{-½}(E[x]-\mu) \text{ and } var(y)= \Sigma^{-½}var(x)\,\Sigma^{-½}$$

but $E[y]=0$ and $var(y)=I_p$, so $E[x]=\mu$ and $var(x)=\Sigma$.

8.1.4 Random Samples

$x \sim N_p(\mu,\Sigma)$; observations $x_1,x_2,...,x_n$ of x.

Define $\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$

and $S = \frac{1}{(n-1)}(X-\bar{X})(X-\bar{X})' = \frac{1}{(n-1)}\left\{\sum_1^n x_i x_i' - n\bar{x}\bar{x}'\right\}$

Then $E[\bar{x}]=\mu$ and $var(\bar{x}) = \frac{1}{n^2}\sum_1^n var(x_i) = \frac{1}{n}\Sigma$

Also $S=\frac{1}{(n-1)}\left\{(1-\frac{1}{n})\sum_{i=1}^{n}(x_i-\mu)(x_i-\mu)' - \frac{1}{n}\sum\sum_{i\neq j}(x_i-\mu)(x_j-\mu)'\right\}$

(see Notes in §0.9)

and $E[(x_i-\mu)(x_j-\mu)'] = 0$ if $i\neq j$

$= \Sigma$ if $i=j$

and so $E[S] = \Sigma$, i.e. the sample mean and variance are unbiased for the population mean and variance.

(note that as yet we have not used any assumption of normality)

Further, using the fact that if $x \sim N_p(\mu,\Sigma)$ then the characteristic function of x is $\phi_x(t)=E[e^{it'x}]=\exp\{it'\mu-½t'\Sigma t\}$ [where $i=(-1)^{½}$]

we can shew that

$$\boxed{\bar{x} \sim N_p(\mu,\tfrac{1}{n}\Sigma).}$$

8.2★ Maximum Likelihood Estimation

$x_1, x_2, ..., x_n$ independent observations of $x \sim N_p(\mu, \Sigma)$

Then $\mathrm{Lik}(\mu, \Sigma; X) = \frac{1}{(2\pi)^{np/2}|\Sigma|^{n/2}} \exp\left\{-\tfrac{1}{2}\sum_{i=1}^{n}(x_i-\mu)'\Sigma^{-1}(x_i-\mu)\right\}$

so $\ell(\mu, \Sigma; X) = \log_e(\mathrm{Lik}(\mu, \Sigma; X))$

$$= -\tfrac{1}{2}\sum_{i=1}^{n}(x_i-\mu)'\Sigma^{-1}(x_i-\mu) - \tfrac{1}{2}np\log(2\pi) - \tfrac{1}{2}n\log(|\Sigma|)$$

$$= -\tfrac{1}{2}\{\sum_{i=1}^{n}(x_i-\overline{x})'\Sigma^{-1}(x_i-\overline{x}) + n(\overline{x}-\mu)'\Sigma^{-1}(\overline{x}-\mu)\}$$

$$- \tfrac{1}{2}np\log(2\pi) - \tfrac{1}{2}n\log(|\Sigma|)$$

So, $\frac{\partial \ell}{\partial \mu} = n\Sigma^{-1}(\overline{x}-\mu)$ and thus $\hat{\mu} = \overline{x}$.

Further, if we set $T=\Sigma^{-1}$ it can be shewn that

$$\frac{\partial \ell}{\partial T} = \{n\Sigma - (n-1)S - n(\overline{x}-\mu)(\overline{x}-\mu)'\}$$

$$-\tfrac{1}{2}\mathrm{diag}\{n\Sigma - (n-1)S - n(\overline{x}-\mu)(\overline{x}-\mu)'\}$$

(where the derivative of the scalar ℓ with respect to the p×p matrix T is the p×p matrix formed by the partial derivatives of ℓ with respect to each of the elements t_{ij} of T, and where $\mathrm{diag}\{A_{p\times p}\}$ is the diagonal matrix formed by just the diagonal elements of the p×p matrix A, zeroes elsewhere. Some details of this are given in the appendix.)

So $\frac{\partial\ell}{\partial T}$=0 when $\Sigma = \hat{\Sigma} = \frac{n-1}{n}S + (\overline{x} - \hat{\mu})(\overline{x} - \hat{\mu})'$ and when $\hat{\mu} = \overline{x}$ this gives the [unrestricted] maximum likelihood estimates of μ and Σ

$$\boxed{\hat{\mu} = \overline{x}, \quad \hat{\Sigma} = \frac{n-1}{n}S}$$

More generally, whatever the mle of μ is, if $d = \overline{x} - \hat{\mu}$ then we have

$$\boxed{\hat{\Sigma} = \frac{n-1}{n}S + dd'}$$

[This form is sometimes useful in constructing likelihood ratio tests of hypotheses that put some restriction on μ and so under the null hypothesis the maximum likelihood estimate of μ is not $\overline{x}$. In these cases we can easily obtain the maximum likelihood estimate of Σ and thus the value of the maximized likelihood under the null hypothesis.]

8.2.1 The Maximized Log–Likelihood

For the construction of likelihood ratio tests we need the actual form of the maximized likelihood under null and alternative hypotheses. Typically, the alternative hypothesis gives no restrictions on μ and Σ and so the mles under the alternative hypothesis are as given earlier (i.e. $\hat{\mu} = \overline{x}$ & $\hat{\Sigma} = \tfrac{n-1}{n}S$). The null hypothesis will either impose some constraint on Σ (e.g. H_0: $\Sigma=\Sigma_0$) or some constraint on μ (e.g. H_0: $\mu=\mu_0$ or H_0: $\mu\mu'=1$). In these cases we obtain the estimate of μ and then use the more general form given above.

For example, under H_0: $\mu=\mu_0$ we have $\hat{\mu}=\mu_0$ and so this gives

$$\hat{\Sigma} = \tfrac{n-1}{n}S + (\overline{x} - \mu_0)(\overline{x} - \mu_0)' = \tfrac{1}{n}\sum_{i=1}^{n}(x_i - \mu_0)(x_i - \mu_0)' = S^{\dagger}$$

To calculate the actual maximized likelihood in either case usually requires the use of a ***slick trick*** in manipulating matrices. This is the following:

- First note that a vector product such as $y'Ay$ where y is a p-vector and A is $p\times p$ is a scalar (i.e. 1×1)

- Next note that since this is a scalar we have $\text{trace}(y'Ay)=y'Ay$ (only one diagonal element in a 1×1 matrix).

- Next, applying the rule that $\text{trace}(BC)=\text{trace}(CB)$, if both products are defined, gives $y'Ay=\text{trace}(Ayy')$

- Next, noting that $\text{trace}(B+C)=\text{trace}(B)+\text{trace}(C)$ gives

$$\sum_{i=1}^{n} y_i'Ay_i = \text{trace}\{A\sum_{i=1}^{n} y_iy_i'\}$$

The advantage of this is that the matrix product on the right hand side might reduce to the identity matrix whose trace is easy to calculate.

Now we have $\ell(\mu, \Sigma; X) = \log_e(\mathrm{Lik}(\mu, \Sigma; X))$

$$= -\tfrac{1}{2}\sum_{i=1}^{n}(x_i - \mu)'\Sigma^{-1}(x_i - \mu) - \tfrac{1}{2}np\log(2\pi) - \tfrac{1}{2}n\log(|\Sigma|)$$

$$= -\tfrac{1}{2}\{\sum_{i=1}^{n}(x_i - \overline{x})'\Sigma^{-1}(x_i - \overline{x}) + n(\overline{x} - \mu)'\Sigma^{-1}(\overline{x} - \mu)\}$$
$$- \tfrac{1}{2}np\log(2\pi) - \tfrac{1}{2}n\log(|\Sigma|)$$

and $\sum_{i=1}^{n}(x_i - \overline{x})'\Sigma^{-1}(x_i - \overline{x}) = \mathrm{trace}\{\sum_{i=1}^{n}(x_i - \overline{x})'\Sigma^{-1}(x_i - \overline{x})\}$

$$=\mathrm{trace}\{\Sigma^{-1}\sum_{i=1}^{n}(x_i - \overline{x})(x_i - \overline{x})'\} = \mathrm{trace}\{\Sigma^{-1}(n\text{–}1)S\}$$

$$= (n\text{–}1)\mathrm{trace}\{\Sigma^{-1}S\}$$

So $\ell(\mu,\Sigma;X) = -\tfrac{1}{2}\{\sum_{i=1}^{n}(x_i - \overline{x})'\Sigma^{-1}(x_i - \overline{x}) + n(\overline{x} - \mu)'\Sigma^{-1}(\overline{x} - \mu)\}$

$$- \tfrac{1}{2}np\log(2\pi) - \tfrac{1}{2}n\log(|\Sigma|)$$

$$= -\tfrac{1}{2}(n\text{–}1)\mathrm{trace}\{\Sigma^{-1}S\} - \tfrac{n}{2}\mathrm{trace}\{\Sigma^{-1}(\overline{x} - \mu)(\overline{x} - \mu)'\}$$
$$- \tfrac{1}{2}np\log(2\pi) - \tfrac{1}{2}n\log(|\Sigma|)$$

and so $\max_{\mu,\Sigma} \ell(\mu,\Sigma;X) = \ell(\hat{\mu},\hat{\Sigma};X)$

$= –½(n–1)tr\{\Sigma^{–1}S\} – 0 – ½nplog(2\pi) – ½nlog(|\hat{\Sigma}|)$

$= –½(n–1)tr\{nS^{–1}S/(n–1)\}) – ½nlog|(n–1)S/n|$

$– ½nplog(2\pi$

$= –½ntr\{I_p\} – ½nplog({}^{(n-1)}/_n) – ½nlog|S| –½nplog(2\pi)$

$$\boxed{= –½np – ½nplog({}^{(n-1)}/_n) – ½nlog|S| –½nplog(2\pi)}$$

More generally, whatever the mle of μ is,

if $d=\bar{x}-\hat{\mu}$ then we have $\hat{\Sigma} = \tfrac{n-1}{n}S + dd'$ and

$\max_{\mu,\Sigma} \ell(\mu,\Sigma;X) = \ell(\hat{\mu},\hat{\Sigma};X)$

$= –½\,tr\{\hat{\Sigma}^{-1}[(n-1)S + ndd']\} – ½nplog(2\pi) – ½nlog(|\hat{\Sigma}|)$

$= –½np – ½nplog(2\pi) – ½nlog(|\hat{\Sigma}\ \hat{\Sigma}|)$

$$\boxed{= –½np – ½nplog(2\pi) – ½nlog(|\tfrac{n-1}{n}S + dd'|)}$$

8.3 Related Distributions

8.3.0 Introduction

This section introduces two distributions related to the multivariate normal distribution. The densities are not given (they can be found in standard texts) but some basic properties of them are outlined. Their use is in the construction of tests, specifically in determining the distribution of test statistics. They are generalizations of familiar univariate distributions and their properties match those of their univariate special cases.

8.3.1 The Wishart Distribution

If $X=(x_1, x_2, \dots , x_n)$, and if $M_{p\times p}=XX'$ with $x_i\sim N_p(0,\Sigma)$, (i.i.d)

then $M\sim W_p(\Sigma,n)$ — the Wishart distribution with scale matrix Σ and n degrees of freedom.

This is a matrix generalization of the χ^2-distribution :—

$$\text{if } p=1 \text{ then } M=\sum_{i=1}^{n} x_i^2 \ \text{ with } x_i\sim N(0,\sigma^2)$$

Note that it is a ½p(p+1)–dimensional distribution (M is symmetric).

Its ‘standard form’ is when $\Sigma=I_p$.

Its properties are generalizations of those of the χ^2-distribution, e.g. additivity on the degrees of freedom parameter:

if U~ $W_p(\Sigma,m)$ and V~$W_p(\Sigma,n)$ independently

then U+V~$W_p(\Sigma,m+n)$

Its key use is as an intermediate step in deriving the distribution of things of real interest.

In particular, if S= $\frac{1}{n-1}\sum_{i=1}^{n}(x_i-\bar{x})(x_i-\bar{x})'$, the sample variance, then

(n-1)S~$W_p(\Sigma$, n-1) ***independently*** of $\bar{x}$

8.3.2 Hotelling's T^2–Distribution

This is a univariate distribution of a scalar random variable, it is a generalization of student's t-distribution or Snedecor's F-distribution.

Definition: If $d\sim N_p(0,I_p)$ and $M\sim W_p(I_p,n)$ independently then

$$nd'M^{-1}d\sim T^2(p,n)$$

— Hotelling's T^2, parameter p, degrees of freedom n.

In particular, if $x\sim N_p(\mu, \Sigma)$ and $M\sim W_p(\Sigma, n)$ then we have

$$n(x-\mu)'M^{-1}(x-\mu)\sim T^2(p,n)$$

(— prove by writing $d^*=\Sigma^{-½}(x-\mu)$ and $M^*=\Sigma^{-½}M\Sigma^{-½}$)

and especially;

$$\boxed{n(\bar{x}-\mu)'S^{-1}(\bar{x}-\mu)\sim T^2(p,n-1)}$$

(Noting the independence of $\bar{x}$ and S).

This is the basis of one and two sample tests. To evaluate p-values we use the following

Theorem:

$$\boxed{T^2(p,n)\equiv \tfrac{np}{n-p+1}F_{p,n-p+1}}$$

Proof: Not given — see any standard text.

This allows us to calculate a T2 value, multiply by (n–p+1)/np and then refer the result to F–tables with p and (n–p+1) degrees of freedom.

8.4 Simple one– & two–sample tests

8.4.1 One–sample tests

If $x_1, x_2, ..., x_n$ are independent observations of $x \sim N_p(\mu, \Sigma)$

then we have $n(\overline{x}-\mu)'S^{-1}(\overline{x}-\mu) \sim T^2(p,n-1)$,

so $\frac{(n-1)-p+1}{(n-1)p} n(\overline{x}-\mu)'S^{-1}(\overline{x}-\mu) \sim F_{p,n-p}$

i.e. $\frac{n}{n-1}\frac{n-p}{p}(\overline{x}-\mu)'S^{-1}(\overline{x}-\mu) \sim F_{p,n-p}$

So we can test e.g. H_0: $\mu=\mu_0$ vs $\mu\neq\mu_0$ since under H_0 we have $\frac{n}{n-1}\frac{n-p}{p}(\overline{x}-\mu_0)'S^{-1}(\overline{x}-\mu_0) \sim F_{p,n-p}$ and so we reject H_0 when this is improbably large when referred to an F-distribution with (p,n–p) degrees of freedom.

8.4.2 Two–sample tests

The Mahalanobis distance between two populations with means μ_1 and μ_2 and common variance Σ is defined as Δ where

$$\Delta^2 = (\mu_1-\mu_2)'\Sigma^{-1}(\mu_1-\mu_2)$$

If we have samples of sizes n_1 and n_2; means $\overline{x}_1$ and $\overline{x}_2$; variances S_1 and S_2 then we define the sample Mahalanobis distance as

$$D^2 = (\overline{x}_1-\overline{x}_2)'S^{-1}(\overline{x}_1-\overline{x}_2)$$

where $S=[(n_1-1)S_1+(n_2-1)S_2]/(n-2)$ (i.e. the pooled variance [or pooled variance-covariance]); $n=n_1+n_2$

Now if $\mu_1=\mu_2$ then $D^2\sim\frac{n}{n_1n_2}T^2(p,n-2)$

since we have $\overline{x}_i\sim N_p(\mu_i,n_i^{-1}\Sigma)$ and $(n_i-1)S_i\sim W_p(\Sigma,n_i-1)$; $i=1,2$

so $\overline{x}_1-\overline{x}_2\sim N_p(\mu_1-\mu_2,\ \frac{n}{n_1n_2}\Sigma)$ and $(n-2)S\sim W_p(\Sigma,n-2)$

and hence the result follows.

The use is to test H_0: $\mu_1=\mu_2$ since we can reject H_0 if

$\frac{n_1n_2(n-p-1)}{n(n-2)p}D^2$ is improbably large when compared with $F_{p,n-p-1}$.

8.4.3 Notes

- ♦ These one and two sample tests are easy to compute in **R**, S-PLUS or MINITAB by direct calculation using their matrix arithmetic facilities. The two-sample test can be calculated using the general MANOVA facilities, see §8.7.4 below.
- ♦ The library `ICSNP` contains a function `HotellingsT2(.)` which provides one and two sample tests.
- ♦ Note that in one dimension the best practice is always to use the separate variance version of the two-sample t-test. In principle it would be good to do the same in higher dimensions but there is no available equivalent of the Welch approximation to obtain approximate degrees of freedom for the T^2-distribution so the pooled variance version is just a pragmatic expedient.

Tasks 8

(see §8.0–§8.3)

1) Read §8.1 – §8.4 paying particular attention to the results highlighted in boxes as well as §8.3.2 and §8.4.

2) n observations are available on $x \sim N_p(\mu,\Sigma)$ and C is a known $p \times q$ matrix ($p>q$). By finding the distribution of $y=C'x$ (by calculating the mean and variance of y and using the result that [non-singular] linear transformations of Normal random variables are also Normal with appropriate mean, variance & dimension), shew that a test of H_0: $C'\mu=0$ *vs.* H_A: $C'\mu \neq 0$ is given by Hotelling's T^2 with $T^2=n\bar{x}'C(C'SC)^{-1}C'\bar{x}$ ($\bar{x}$ and S are the sample mean and variance). What parameters does the T^2 distribution have?

3) Note: parts (i) & (ii) below should give the same p-value.

   i) A sample of 6 observations on sugar content x_1 and stickiness x_2 of a novel toffee give sample statistics of

   $$\bar{x} = \begin{pmatrix} 8117 \\ 60.33 \end{pmatrix} \text{ and } S = \begin{pmatrix} 27.02 & 7.94 \\ * & 4.26 \end{pmatrix}$$

   Test the hypothesis H_0: $2\mu_1=3\mu_2$ using a Hotelling's T^2-test [Suggestion: consider using the 2×1 matrix $C=(2, -3)'$]

   ii) By noting that if $x = (x_1,x_2) \sim N_2(\mu,\Sigma)$ where $\mu = (\mu_1,\mu_2)'$ and Σ has element σ_{ij} then $2x_1-3x_2 \sim N((2\mu_1-3\mu_2),(4\sigma^2_{11}+9\sigma^2_{22}-12\sigma_{12}))$ test H_0 in i) above using a Student's t-test.

8.5 Likelihood Ratio Tests

8.5.0 Introduction

The one and two sample test statistics for $\mu=\mu_0$ and $\mu_1=\mu_2$ given above are easily shewn to be likelihood ratio statistics (i.e. "optimal''). LRTs are a useful general procedure for constructing tests and can often be implemented numerically using general purpose function maximization routines even when analytic closed forms for maximum likelihood estimates are not obtainable.

Suppose data are available from a distribution depending on a parameter θ, where θ may be a 'vector parameter', i.e. consist of several separate parameters,

(e.g. $\theta=(\mu,\sigma)$, the parameters of a univariate normal distribution which has 2 separate parameters, or e.g. $\theta=(\mu,\Sigma)$, the parameters of a p-dimensional normal distribution has $p+½p(p+1)=½p(p+3)$ separate parameters).

Typically, the null hypothesis H_0 will specify the values of some of these, e.g. in the first case H_0: $\mu=0$ specifies 1 parameter and in the second it specifies p of them.

The general procedure for constructing a likelihood ratio test (i.e. finding the LRT statistic) of H_0 versus H_A is:

1. Find the maximum likelihood estimates of all parameters θ assuming H_0 is true to get $\hat{\theta}_0$, e.g. with $N(\mu,\sigma^2)$ or $N_p(\mu,\Sigma)$, if H_0: $\mu=0$ then estimate σ or Σ assuming $\mu=0$ giving $\hat{\sigma}^2 = \sqrt{\frac{1}{n}\sum_i x_i^2}$ or $\hat{\Sigma} = \frac{1}{n}\sum_i x_i x_i'$ and then $\hat{\theta}_0$ =(0, $\sqrt{\frac{1}{n}\sum_i x_i^2}$) or $\hat{\theta}_0$ =(0,$\frac{1}{n}\sum_i x_i x_i'$)

2. Find the maximum value of the log likelihood under H_0 (i.e. substitute the mles of the parameters under H_0 into the log likelihood function) to get $\ell_{\max}(H_0) = \ell(\hat{\theta}_0)$

3. Find the maximum likelihood estimates of all parameters assuming H_A is true, $\hat{\theta}_A$. Typically these will be the ordinary mles

4. Find the maximum value of the log likelihood under H_A (i.e. substitute the mles of the parameters under H_A into the log likelihood function) to get $\ell_{\max}(H_A) = \ell(\hat{\theta}_A)$

5. Calculate twice the difference in maximized log likelihoods,

   $\lambda = 2\{\ell_{\max}(H_A) - \ell_{\max}(H_0)\}$.

6. Use Wilks' Theorem which says that under H_0 this statistic is approximately distributed as χ^2 with degrees of freedom given by the difference in the numbers of estimated parameters under H_0 and H_A,

   i.e. $\lambda \sim \chi^2_k$ where $k = \dim(H_A) - \dim(H_0)$

   or

7. Find some monotonic function of λ which has a recognisable distribution under H_0.

8.5.1 LRT of H_0: $\mu=\mu_0$ *vs.* H_A: $\mu\neq\mu_0$ with $\Sigma=\Sigma_0$ known

$x_1, x_2, ..., x_n$ independent observations of $x\sim N_p(\mu, \Sigma_0)$.

To test H_0: $\mu=\mu_0$ *vs.* H_A: $\mu\neq\mu_0$ with Σ_0 known (i.e. not estimated).

Now $\ell(\mu; X)$= log lik(μ; X)

$$= -\tfrac{1}{2}np\log(2\pi)-\tfrac{1}{2}n\log|\Sigma_0|-\tfrac{1}{2}(n-1)\mathrm{tr}\{\Sigma_0^{-1}S\} -\tfrac{1}{2}n(\bar{x}-\mu)'\Sigma_0^{-1}(\bar{x}-\mu)$$

$$= K - \tfrac{1}{2}n(\bar{x}-\mu)'\Sigma_0^{-1}(\bar{x}-\mu)$$

So under H_0 we have $\ell(\mu_0; X, H_0) = K - \tfrac{1}{2}n(\bar{x}-\mu_0)'\Sigma_0^{-1}(\bar{x}-\mu_0)$

i.e. $\ell_{max}(H_0) = K - \tfrac{1}{2}n(\bar{x}-\mu_0)'\Sigma_0^{-1}(\bar{x}-\mu_0)$

Under H_A the mle of μ is $\bar{x}$ giving $\ell_{max}(H_A) = K$

So the LRT statistic is

$$\lambda=2\{\ell_{max}(H_A) - \ell_{max}(H_0)\} = n(\bar{x}-\mu_0)'\Sigma_0^{-1}(\bar{x}-\mu_0)$$

and the test is to reject this if it is improbably large when compared with χ^2_p , noting that there are p parameters to be estimated under H_A but none under H_0.

Also note that this is an exact result
(i.e. not a Wilks' Theorem approximation)
since $\lambda=yy'=\Sigma y^2_i$ with $y_i\sim N(0,1)$ where $y=n^{½}\Sigma_0^{-½}(\bar{x}-\mu_0)\sim N_p(0,I_p)$.

8.5.2 LRT of H_0: $\Sigma=\Sigma_0$ *vs* H_A: $\Sigma\neq\Sigma_0$; μ unknown.

$x_1, x_2, \ldots, x_n$ independent observations of $x\sim N_p(\mu, \Sigma)$.

To test H_0: $\Sigma=\Sigma_0$ *vs.* H_A: $\Sigma\neq\Sigma_0$.

Under H_0 we have $\hat{\mu}=\bar{x}$ and $\Sigma=\Sigma_0$.

Under H_A we have $\hat{\mu}=\bar{x}$ and $\hat{\Sigma}=\frac{n-1}{n}S = S^*$ say.

Thus $\ell_{max}(H_0) = –½ntr\{\Sigma_0^{-1}S^*\} – ½nplog(2\pi) – ½nlog(|\Sigma_0|)$

and $\ell_{max}(H_A) = –½np – ½nplog(2\pi) – ½nlog(|S^*|)$

So $\lambda=2\{\ell_{max}(H_A) – \ell_{max}(H_0)\}=ntr\{\Sigma_0^{-1}S^*\} – nlog(|\Sigma_0^{-1} S^*|) – np$

and the test is to reject H_0 if λ is improbably large when compared with a χ^2 distribution on ½p(p+1) degrees of freedom (using the asymptotic result of Wilks' Theorem).

Notice that $tr\{\Sigma_0^{-1}S^*\} = \sum_{i=1}^{p}\lambda_i$ and $|\Sigma_0^{-1} S^*| = \prod_{i=1}^{p}\lambda_i$

where λ_i are the eigenvalues of $\Sigma_0^{-1}S^*$ and so we can express λ as

$\lambda=np(\bar{\lambda} – log(\dot{\lambda}) – 1)$ where $\bar{\lambda}$ and $\dot{\lambda}$ are the arithmetic and geometric means respectively of the λ_i.

8.5.3. LRT of $\mu'\mu$=1 with known Σ=I_p

$x_1, x_2, ..., x_n$ independent observations of $x \sim N_p(\mu, I_p)$.

To test H_0: $\mu'\mu=1$ *vs.* H_A: $\mu'\mu \neq 1$.

Let $\ell(\mu; X)$ be the [unrestricted] likelihood of μ, then

$$\ell(\mu; X) = -½(n-1)\text{trace}(S) - ½n(\overline{x}-\mu)'(\overline{x}-\mu) - ½np\log(2\pi)$$

To maximize $\ell(\mu)$ under H_0 we need to impose the constraint $\mu\mu'=1$ and so introduce a Lagrange multiplier and let $\Omega=\ell(\mu)-\lambda(\mu'\mu-1)$.

Then $\frac{\partial\Omega}{\partial\mu} = n(\overline{x}-\mu) - 2\lambda\mu$ and differentiating w.r.t. λ gives $\mu'\mu=1$.

So we require $\hat{\mu} = \frac{n\overline{x}}{n+2\lambda}$ and then $\mu'\mu=1$ implies $(n+2\lambda)^2=n^2\overline{x}'\overline{x}$

So $\hat{\mu} = \frac{\overline{x}}{\sqrt{\overline{x}'\overline{x}}}$ and

$\ell_{\max}(H_0) = -½(n-1)\text{trace}(S) - ½n(\overline{x}-\frac{\overline{x}}{\sqrt{\overline{x}'\overline{x}}})'(\overline{x}-\frac{\overline{x}}{\sqrt{\overline{x}'\overline{x}}}) - ½np\log(2\pi)$

$= -½(n-1)\text{trace}(S) - ½n\overline{x}'(1-\frac{1}{\sqrt{\overline{x}'\overline{x}}})'(1-\frac{1}{\sqrt{\overline{x}'\overline{x}}})\overline{x} - ½np\log(2\pi)$

$= ½(n-1)\text{trace}(S) - ½n(\sqrt{\overline{x}'\overline{x}}-1)^2 - ½np\log(2\pi)$.

Under H_A we have $\hat{\mu} = \overline{x}$ and so

$\ell_{\max}(H_A) = ½(n-1)\text{trace}(S) - ½np\log(2\pi)$

giving $\lambda = 2\{\ell_{\max}(H_A) - \ell_{\max}(H_0)\} = n(\sqrt{\overline{x}'\overline{x}}-1)^2$

and the test is to reject H_0 when this is improbably large when referred to a χ^2_1 distribution.

Note only 1 degree of freedom since μ has p independent parameters so p are estimated under H_A and under H_0 with one constraint we have effectively p–1 parameters, p–(p–1)=1.

8.5.4 Test of $\Sigma=\lambda\Sigma_0$; $\mu=\mu_0$ known, Σ_0 known.

$x_1, x_2, ..., x_n$ independent observations of $x\sim N_p(\mu_0, \Sigma)$.

To test H_0: $\Sigma=\lambda\Sigma_0$ *vs.* H_A: $\Sigma \neq \lambda\Sigma_0$ where both μ_0 and Σ_0 are known.

(Note that under H_0, λ is the only unknown parameter but under H_A all ½p(p+1) parameters of Σ are unknown).

Under H_0

$\ell(\lambda; X)= -½ntr\{(\lambda\Sigma_0)^{-1}S^{\dagger}\} - ½nlog(|\lambda\Sigma_0|)-½nplog(2\pi)$

(where $S^{\dagger}= \frac{1}{n}\sum_{i=1}^{n}(x_i-\mu_0)(x_i-\mu_0)'$)

$= -½n\lambda^{-1}tr\{\Sigma_0^{-1}S^{\dagger}\}-½nplog(\lambda)-½nlog(|\Sigma_0|)-½nplog(2\pi)$

so $\frac{\partial\ell}{\partial\lambda}=½n\lambda^{-2}tr\{\Sigma_0^{-1}S^{\dagger}\}-½np\lambda^{-1}$

giving $\hat{\lambda}=\frac{1}{p}tr(\Sigma_0^{-1}S^{\dagger})$

so $\ell_{max}(H_0)=-½np-½nplog(\hat{\lambda})-½nlog(|\Sigma_0|)-½nplog(2\pi)$

Under H_A we have $\hat{\Sigma} = S^{\dagger}$ and so

$\ell_{max}(H_A) = -½np - ½nlog(|S^{\dagger}|) - ½nplog(2\pi)$

Then the LRT statistic is $2\{\ell_{max}(H_A) - \ell_{max}(H_0)\}$ and this would be compared with χ_r^2 where r=½p(p+1)–1, using Wilks' Theorem.

Although this is not in a 'simple' algebraic form, it can be calculated numerically in practice and the test evaluated.

8.5.5 Comments

- Examples 8.5.1 and 8.5.2 are multivariate generalizations of equivalent univariate hypotheses, and a useful check is to put p=1 and verify that the univariate test is obtained. (In Example 8.5.2. note that the 'eigenvalue' of a '1×1 matrix' (scalar) is the scalar itself).

- Examples 8.5.3 and 8.5.4 illustrate the more structured hypotheses that can be tested in multivariate problems; they have no counterpart in univariate models.

- Such LRTs are a powerful all-purpose method of constructing tests and can often be implemented numerically even if algebraic analysis cannot produce mles in closed form.

- Further, they are an elegant application of general statistical theory and have various desirable properties — they are guaranteed to be [asymptotically] most powerful, i.e. they are more likely than any other test to be able to detect successfully that the null hypothesis is false provided that the sample is large enough and provided that the parent distribution of the data is indeed that presupposed (e.g. multivariate normal).

- However, a difficulty in multivariate problems involving hypothesis testing is that when a hypothesis is rejected it may not be apparent just 'why' it is false. This is not so in univariate problems; if we have a model that univariate $x \sim N(\mu, \sigma^2)$ and we reject H_0: $\mu=\mu_0$ then we know whether $\overline{x}>\mu_0$ or $\overline{x}<\mu_0$ and hence 'why' H_0 is false. In contrast, if we have a model that multivariate $x \sim N_p(\mu,\Sigma)$ and we reject H_0: $\mu=\mu_0$ then all we know is that there is evidence that

  $$(\mu_1, \mu_2,\ldots,\mu_p)\neq(\mu_{01}, \mu_{02},\ldots,\mu_{0p}).$$

  [For multivariate μ we cannot say $\mu>\mu_0$].

  It may be that only one component in μ_0 is not 'correct' and that $\mu_i=\mu_{0i}$ for all the others. That is we do not know the ***direction*** of departure from H_0.

- That is, a likelihood ratio test may be able to reject a hypothesis but not actually reveal anything interesting about the structure of the data, e.g. knowing that H_0 was 'nearly' correct and only one component was wrong could provide a useful insight into the data but this might be missed by a LRT.

- This leads to considering a different strategy for constructing tests which might provide more information for data analysis, though if they actually produce a different test from the LRT then they may not be so powerful (at least for sufficiently large data sets).

Tasks 9

(see §8.3)

1) Read the solutions to Exercises 2. These contain a detailed guide to the interpretation of principal components and of crimcoords by examining the loadings of the variables in the PCs and Crimcoords and so provide further practice at this important aspect.

2) Referring to the data set *dogmandibles.*∗ **excluding the Prehistoric Thai dogs (group 5 on X_{11})** test the hypotheses that Male and Female dogs have

   i) equally sized mandibles (i.e. variables X_1 & X_2 together)

      a) equally long mandibles (variable X_1)

      b) equally broad mandibles (variable X_2)

   ii) equal overall mandible characteristics (i.e. variables X_1–X_9)

3) Test the hypotheses that Iris Versicolor and Iris Virginica have
   i) equally sized sepals
   ii) equally sized petals
   iii) equally sized sepals & petals.

8.6 Union–Intersection Tests

8.5.0 Introduction

Union-Intersection Tests (UITs) provide a different strategy for constructing multivariate tests. They are not available in all situations (unlike LRTs), they do not have any general statistical optimal properties (again unlike LRTs) and sometimes they produce test statistics that can only be assessed for statistical significance by simulation or Monte Carlo or Bootstrap procedures. However, they will automatically provide an indication of the ***direction of departure*** from a hypothesis (just as in univariate problems it is apparent whether the sample mean is too big or too small).

The method is to project the data into one dimension (just as with many multivariate exploratory data analytic techniques) and test the hypothesis in that one dimension. The particular dimension chosen is that which shews the greatest deviation from the null hypothesis, again there are close analogies with multivariate EDA.

The validity of the procedure relies on the Cramér–Wold Theorem which establishes the connection between the set of all one-dimensional projections and the multivariate distribution.

8.6.1 The Cramér–Wold Theorem

The distribution of a p-vector x is completely determined by the set of all 1-dimensional distributions of 1-dimensional projections of x, t′x, where t∈{all fixed p-vectors}

Proof: Let y=t′x, then, for any t, the distribution (and hence the characteristic function) of y is known and is, say,

$$\phi_y(s) = E[e^{isy}] = E[e^{ist'x}]$$

Putting s=1 gives $\phi_y(1) = E[e^{it'x}]$ is known for all $t\in\Re^p$.

But $E[e^{it'x}]=\phi_x(t)$, the characteristic function of x,

i.e. $\phi_x(t)$ is known for all $t\in\Re^p$,

i.e. the distribution of x is determined by specifying the distributions of t′x for all $t\in\Re^p$.

Importance: is that any multivariate distribution can be defined by specifying the distribution of **all** of its linear combinations (not just the p marginal distributions), e.g. if we specify that the mean of **all** one-dimensional projections of x is 0, then necessarily the mean of the p-dimensional distribution must be 0 (the converse is true also of course). Note that specifying that the p *marginals* have a zero mean is not sufficient to ensure that the p-dimensional distribution is zero.

8.6.2 An Example of a UIT

Suppose $x\sim N_p(\mu,I_p)$. Then for any p-vector β we have that if $y_\beta=\beta'x$ then $y_\beta\sim N(\beta'\mu,\ \beta'I_p\beta)$, i.e. $y_\beta\sim N(\beta'\mu,\ \beta'\beta)$

[and note that the C–W theorem shews the converse is true.]

Suppose that we want to test the hypothesis H0: μ=0, based just on the single observation x.

Then, under H_0, we have that for all β, $H_{0\beta}$: $y_\beta\sim N(0,\beta'\beta)$ is true.

i.e. H_0 true $\Rightarrow$ $H_{0\beta}$ true for all β

and (by the C–W theorem) $H_{0\beta}$ true for all $\beta \Rightarrow H_0$ true.

$$\text{i.e. } H_0 = \bigcap_\beta H_{0\beta}$$

H_0 is the 'intersection' of all univariate hypotheses $H_{0\beta}$.

For any β, $H_{0\beta}$ is a '*component'* of H_0.

Now for any specific β, $H_{0\beta}$ is the hypothesis that the mean of a normal distribution with known variance $\sigma^2=\beta'\beta$ is zero, and we would reject $H_{0\beta}$ at level α if $\left|\frac{y_\beta}{\sqrt{\beta'\beta}}\right| > c$, for some suitable c

(actually the upper 100×½α% point of N(0,1).)

i.e. the rejection region for $H_{0\beta}$ is

$\{y_\beta$: $\left|\frac{y_\beta}{\sqrt{\beta'\beta}}\right| > c\}$ = $\{x$: $\left|\frac{\beta'x}{\sqrt{\beta'\beta}}\right| > c\}$= R_β say

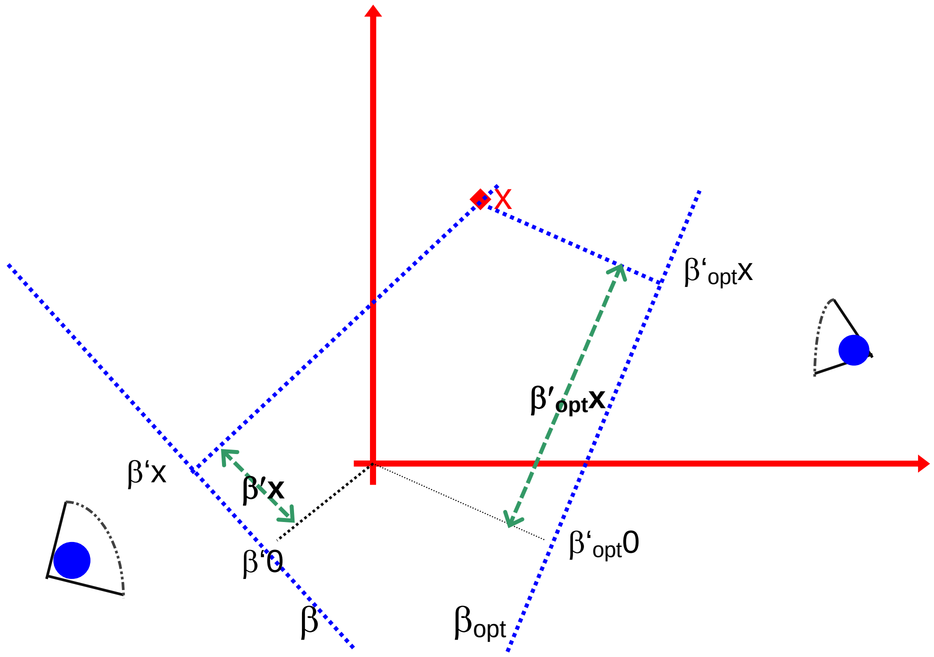
and we reject $H_{0\beta}$ if $x\in R_{\beta.}$

Further: H_0 is true if and only if **every** $H_{0\beta}$ is true.

i.e. if **any** of the $H_{0\beta}$ is false then H_0 is false.

So a sensible rejection region for H_0 is the union of all the rejection regions for the component hypotheses $H_{0\beta,}$ i.e. reject H_0 if $x\in\bigcup_{\beta} R_\beta$.

i.e. reject H_0: $\mu=0$ if in **any** one-dimensional projection of x, $\beta'x$, is sufficiently different from 0.



— if H_0 is rejected then we know which β (or βs) “cause” the rejection, and hence the **direction** of deviation from H_0.

[c.f. a 2-sided test in the univariate case, then we know whether the mean is large or small]

8.6.3 Definition

A union–intersection test of a multivariate hypothesis is a test whose rejection region can be written as a union of rejection regions R_β, where R_β is the rejection region of a component hypothesis $H_{0\beta}$, where H_0 is the intersection of the $H_{0\beta}$.

Ex 8.6.2. continued

In the above case we reject H_0 if for any β we have $\left|\frac{y_\beta}{\sqrt{\beta'\beta}}\right| > c$.

i.e. H_0 is **not** rejected (i.e. 'accepted') iff $\left|\frac{y_\beta}{\sqrt{\beta'\beta}}\right| < c$ for all β,

i.e. iff $\max\limits_\beta \left|\frac{y_\beta}{\sqrt{\beta'\beta}}\right| < c$,

i.e. iff $\max\limits_\beta \frac{y_\beta^2}{\beta'\beta} < c^2$

i.e. iff $\max\limits_\beta \frac{y_\beta' y_\beta}{\beta'\beta} < c^2$

i.e. iff $\max\limits_\beta \frac{\beta' xx'\beta}{\beta'\beta} < c^2$

Now $\frac{\beta'xx'\beta}{\beta'\beta}$ is invariant under scalar multiplication of β, so we can impose the [non-restrictive] constraint $\beta'\beta=1$ and maximize $\beta'xx'\beta$ subject to this constraint.

Introducing a Lagrange multiplier gives the problem:

maximize $\Omega=\beta'xx'\beta - \lambda(\beta'\beta-1)$ w.r.t. β and λ.

Differentiating w.r.t. β gives $xx'\beta - \lambda\beta = 0$
so β is an eigenvector of xx'
Now xx' is of rank 1 and so has only one non-zero eigenvalue.
This eigenvector of xx' is x with eigenvalue $x'x$:

Check: $(xx')x - \lambda x=0$ if $\lambda=x'x$
(since $(x'x)x=x(x'x)$, noting $x'x$ is a scalar).

i.e. $\max\limits_{\beta} \frac{\beta'xx'\beta}{\beta'\beta} = \frac{x'xx'x}{x'x} = x'x$

So the UIT of H_0 is to reject H_0 if $x'x>c$, c chosen to give the desired size of test. Now $x'x \sim \chi^2_p$ under H_0, so for a size α test take c=upper 100α% point of χ^2_p.

This is actually the same as the LRT. For this problem and clearly telling the *direction of deviation* from $\mu=0$ is not difficult with just a single observation. The following examples illustrate cases where more information is obtained from the UIT over and above that gained from the LRT.

8.6.4 UIT of H_0: $\mu=\mu_0$ *vs.* H_A: $\mu \neq \mu_0$, Σ unknown.

$x_1, x_2, ..., x_n$ independent observations of $x\sim N_p(\mu, \Sigma)$.

To test H_0: $\mu=\mu_0$ *vs.* H_A: $\mu\neq\mu_0$ with Σ unknown (i.e. to be estimated).

Let β be any p-vector, and $y_\beta=\beta'x$ then $y_\beta\sim N(\beta'\mu, \beta'\Sigma\beta)$,

i.e. $y_\beta\sim N\mu_y, \sigma^2{}_y)$ say.

A component hypothesis is $H_{0\beta}$: $\mu_y=\mu_{0y}$ $\quad(\mu_{0y}=\beta'\mu_0)$

This needs a test of a univariate normal mean, with unknown variance → usual one-sample t-test and we look at

$$t_\beta = \frac{\bar{y}-\mu_{0y}}{\sqrt{\frac{1}{n}s_y^2}} \quad \text{where } s_y^2 = \tfrac{1}{n-1}\sum_{i=1}^{n}(y_i-\bar{y})^2 = \tfrac{1}{n-1}\sum_{i=1}^{n}(\beta'x-\beta'\bar{x})^2$$

$$= \tfrac{1}{n-1}\sum_{i=1}^{n}\beta'(x_i-\bar{x})(x_i-\bar{x})'\beta = \beta'S\beta$$

Also $\bar{y}-\mu_{0y}=\beta'(\bar{x}-\mu_0)$ and

$$(\bar{y}-\mu_{0y})^2 = \{\beta'(\bar{x}-\mu_0)\}^2 = \beta'(\bar{x}-\mu_0)(\bar{x}-\mu_0)'\beta$$

so $t_\beta^2 = \dfrac{n\beta'(\bar{x}-\mu_0)(\bar{x}-\mu_0)'\beta}{\beta'S\beta}$ and the component hypothesis $H_{0\beta}$ is rejected if this is large.

The union–intersection test statistic is obtained by maximizing t_β^2 with respect to β:

i.e. it is $t^2 = \max\limits_\beta t_\beta^2 = \max\limits_\beta \dfrac{n\beta'(\bar{x}-\mu_0)(\bar{x}-\mu_0)'\beta}{\beta'S\beta}$

Now t^2 is invariant under scalar multiplication of β so impose the [non-restrictive] constraint $\beta'S\beta=1$ and maximize instead

$\Omega = n\beta'(\bar{x}-\mu_0)(\bar{x}-\mu_0)'\beta - \lambda(\beta'S\beta-1)$ w.r.t. β and λ.

Differentiating w.r.t. β shews that β satisfies

$$n(\bar{x}-\mu_0)(\bar{x}-\mu_0)'\beta - \lambda S\beta = 0$$

i.e. $nS^{-1}(\bar{x}-\mu_0)(\bar{x}-\mu_0)'\beta - \lambda\beta = 0$

i.e β is the eigenvector of the [rank 1 $p\times p$ matrix]

$nS^{-1}(\bar{x}-\mu_0)(\bar{x}-\mu_0)'$ corresponding to the only non-zero eigenvalue.

Now this eigenvector is $S^{-1}(\bar{x}-\mu_0)$

(or more exactly a scalar multiple of it to satisfy $\beta'S\beta=1$)

Check: $[nS^{-1}(\bar{x}-\mu_0)(\bar{x}-\mu_0)'].[S^{-1}(\bar{x}-\mu_0)] - \lambda\, S^{-1}(\bar{x}-\mu_0) = 0$

for $\lambda = n(\bar{x}-\mu_0)'S^{-1}(\bar{x}-\mu_0)$

So $t^2 = n(\bar{x}-\mu_0)'S^{-1}(\bar{x}-\mu_0)$ which is Hotelling's T^2 and thus the UIT is identical to the LRT.

Further, if H_0 is rejected then this shews that the *direction of deviation* is along the vector $S^{-1}(\bar{x}-\mu_0)$, and we can interpret this direction by looking at the magnitude of the loadings on the individual components, just as in PCA and LDA.

i.e. not just along the difference $(\bar{x}-\mu_0)$ but adjusted to take account of the differing variances of the components of S.

If $S=\sigma^2 I_p$ then the direction of deviation is along $(\bar{x}-\mu_0)$.

8.6.5 UIT of H_0: $\Sigma=\Sigma_0$ *vs* H_A: $\Sigma \neq \Sigma_0$; μ unknown.

$x_1, x_2, \ldots, x_n$ independent observations of $x \sim N_p(\mu, \Sigma)$.

To test H_0: $\Sigma=\Sigma_0$ *vs.* H_A: $\Sigma\neq\Sigma_0$.

[N.B. The LRT for this problem was considered in 4.5.2]

$H_{0\beta}$: $\sigma^2_y = \sigma^2_{0y}$, tested by $U_\beta=(n-1)s^2_y/\sigma^2_{0y}$ $(\sim\chi^2_{n-1}$ under $H_{0\beta})$

rejecting if either $U_\beta<c_{1,\beta}$ or $U_\beta>c_{2,\beta}$.

So, the UIT is obtained by rejecting H_0

if $\min_\beta\{U_\beta\} < c_1$ or if $\max_\beta\{U_\beta\} > c_2$

(where c_1 and c_2 are chosen to give the test the desired size).

Now $U_\beta=(n-1)\beta'S\beta/\beta'\Sigma_0\beta$ which is max/minimized when

$(n-1)\Sigma_0^{-1}S\beta - \lambda\beta = 0$ and $\beta'\Sigma_0\beta=1$

We have that if $(n-1)\Sigma_0^{-1}S\beta - \lambda\beta = 0$ and $\beta'\Sigma_0\beta=1$ then (pre-multiplying by $\beta'\Sigma_0$) $(n-1)\beta'S\beta = \lambda\beta'\Sigma_0\beta = \lambda$

And so $\max_\beta\{U_\beta\}=\lambda_1$ and $\min_\beta\{U_\beta\}=\lambda_p$ where $\lambda_1>\lambda_2>\ldots>\lambda_p$ are the eigenvalues of $\Sigma_0^{-1}S$.

Thus the test is:

- not the same as the LRT
- indicates that the *direction of deviation* is along one or other of the first or last eigenvectors (and which it is will be evident from whether it is λ_1 that is too big or λ_p that is too small)
- requires simulation to apply in practice since there are no general results for UITs comparable to Wilks' Theorem for LRTs

8.7 Multisample Tests — Multivariate Analysis of Variance

Setup: k independent samples from $N_p(\mu_i, \Sigma)$ of sizes n_i

To test H_0: $\mu_1=\mu_2= . . . =\mu_k$ (=μ say) *vs* H_A: at least one $\mu_i \neq \mu$.

8.7.1★ Likelihood Ratio Approach:– Wilks Λ–test

$\ell(\mu_1,\mu_2,...,\mu_k,\Sigma;\, X) =$

$$\sum_{i=1}^{k}\left\{-\tfrac{n_i}{2}\log|\Sigma| - \tfrac{n_ip}{2}\log(2\pi) - \tfrac{1}{2}(n_i-1)\mathrm{tr}(\Sigma^{-1}S_i) - \tfrac{n_i}{2}\mathrm{tr}\left(\Sigma^{-1}(\overline{x}_i-\mu_i)(\overline{x}_i-\mu_i)'\right)\right\}$$

(i.e. the sum of the k separate log-likelihoods of the individual samples)

Under H_0 we have a sample of size $n=\sum_{i=1}^{k} n_i$ from $N_p(\mu,\Sigma)$, so mles are $\hat{\mu}=\overline{x}$, $\hat{\Sigma}=\tfrac{n-1}{n}S$ and so $\ell_{\max}(H_0) = -\tfrac{n}{2}\log(\tfrac{n-1}{n}|S|) - \tfrac{np}{2}\log(2\pi) - \tfrac{np}{2}$,

noting $\sum_{i=1}^{k}\left\{-\tfrac{1}{2}(n_i-1)\mathrm{tr}(\hat{\Sigma}^{-1}S_i) - \tfrac{n_i}{2}\mathrm{tr}\left(\hat{\Sigma}^{-1}(\overline{x}_i-\hat{\mu})(\overline{x}_i-\hat{\mu})'\right)\right\}$

$$=\sum_{i=1}^{k}\left\{-\tfrac{1}{2}\mathrm{tr}(\tfrac{n}{n-1}S^{-1}\sum_{j=1}^{n_i}(x_{ij}-\overline{x}_i)(x_{ij}-\overline{x}_i)' - \tfrac{n_i}{2}\mathrm{tr}\left(\tfrac{n}{n-1}S^{-1}(\overline{x}_i-\overline{x})(\overline{x}_i-\overline{x})'\right)\right\} \text{ and}$$

$$\sum_{i=1}^{k}\left\{\sum_{j=1}^{n_i}(x_{ij}-\overline{x}_i)(x_{ij}-\overline{x}_i)' + n_i(\overline{x}_i-\overline{x})(\overline{x}_i-\overline{x})'\right\} = (n-1)S$$

Under H_A we have $\hat{\mu}_i=\overline{x}_i$ the i^th^ sample mean,

and $\hat{\Sigma}=\tfrac{n-k}{n}W = \tfrac{1}{n}\sum_{i=1}^{k}(n_i-1)S_i$ (W as defined in §3.0)

and so $\ell_{\max}(H_A) = -\tfrac{n}{2}\log\left(\tfrac{n-k}{n}|W|\right) - \tfrac{np}{2}\log(2\pi) - \tfrac{np}{2}$, noting

$\hat{\Sigma}^{-1}(\overline{x}_i-\hat{\mu}_i)(\overline{x}_i-\hat{\mu}_i)' = \hat{\Sigma}^{-1}(\overline{x}_i-\overline{x}_i)(\overline{x}_i-\overline{x}_i)' = 0$

and thus $2[\ell_{\max}(H_A) - \ell_{\max}(H_0)] = n\log\left(\tfrac{n-1}{n-k}\tfrac{|S|}{|W|}\right)$

and so a likelihood ratio test statistic for H_0 is $\tfrac{|S|}{|W|} = |W^{-1}S|$, rejecting H_0 if this is improbably large.

Now $(k-1)B = \sum_{i=1}^{k} n_i(\bar{x}_i - \bar{x})(\bar{x}_i - \bar{x})' = (n-1)S - (n-k)W$

and so an equivalent test statistic is $|W^{-1}[(k\text{-}1)B+(n\text{-}k)W]|$

or equivalently $|I_p+\frac{k-1}{n-k}W^{-1}B|$ rejecting if this is large

or equivalently $\Lambda=|I_p+\frac{k-1}{n-k}W^{-1}B|^{-1}$, rejecting if this is small.

Λ is said to have a Wilks' Λ-distribution $\Lambda(p,n\text{–}k,k\text{–}1)$ which for some values of p, n, k (in particular k=2 or 3) is closely related to an F-distribution. Additionally, for other values of p, n and k, F and χ^2 approximations are available and Biometrika Tables, vol 2, give percentage points. For k=2 this test reduces to the 2-sample Hotelling's T^2 test (see §8.3.2).

8.7.2 Computational Note

- In **R** and S-PLUS the function `manova(.)` provides facilities for multivariate analysis of variance.
- MINITAB provides Wilks' test (complete with p-values, exact for k≤3, approximate otherwise) for one-way multivariate analysis of variance in the menu Stat>ANOVA>Balanced MANOVA...)
- In MINITAB the menu Stat>ANOVA>General MANOVA... provides the same facility.

8.7.3 The Union-Intersection Test

Following the usual procedure, if β is any vector then the test statistic for testing $H_{0\beta}$ is $F_\beta=\beta'B\beta/\beta'W\beta$ whose maximum value is the largest eigenvalue of $W^{-1}B$ (see §4.3 on Crimcoords).

For k=2 this reduces to the 2-sample Hotelling's T^2 test (which is the same as the LRT) but for k>2 the UIT and LRT are different.

The null distribution of this largest eigenvalue is closely related to Roy's Greatest Root Distribution, see Biometrika Tables Vol. 2.

8.7.4 Further Notes

♦ **R**, MINITAB and S-PLUS provide Roy's statistic as well as Wilks' statistic in the routines referred to in §8.7.2.

♦ In addition they produce two further statistics:– Pillai's Trace and the Lawley-Hotelling Trace. The first of these is the trace of the matrix $B(B+W)^{-1}$ and the second is the trace of $W^{-1}B$.

♦ Wilks' test statistic can be expressed as the product of all the eigenvalues of $W(B+W)^{-1}$.

♦ All four of these statistics measure or reflect the 'magnitude' of the matrix $W^{-1}B$ which is the obvious multivariate generalization of the F-statistic in univariate 1-way analysis of variance. Generally, all four tests should lead to equivalent conclusions — if they do not then there is something very unusual about the data which needs further investigation.

- Hotelling's T^2 statistic is most easily computed as (n–2)× Lawley-Hotelling Trace, using the MANOVA option described above, n the total number of observations.

- In principle further extensions of MANOVA (e.g. 2-way or General Multivariate Linear Model) are possible. MINITAB does not provide these — if you specify a two factor model in the Balanced Anova or General Linear Model menu and then ask for multivariate tests it will give you only two separate 1-way MANOVAs, even though it gives the full 2-way univariate ANOVAs for each component.

- MANOVA is rarely the only stage in the analysis, not least because the interpretation of the results is often difficult. It is always useful to look at the separate univariate ANOVAs, supplemented by the first eigenvector of $W^{-1}B$.

- A key advantage of MANOVA over p separate univariate ANOVAs is when an experiment consists of measuring lots of variables on the same individuals in the hope that at least one (or even some) will shew differences between the groups, but it is not known which of the p variables will do so. This is a multiple comparison problem which is partially overcome by performing an initial MANOVA to see whether there are any differences at all between the groups. If the MANOVA fails to reveal any differences then there is little point in investigating differences on separate variables further. If there is some overall difference between the groups then examination of the coefficients in the first eigenvector of $W^{-1}B$, together with informal examination of the individual ANOVAs will indicate which variables or combination of variables (i.e. *directions*) contribute to the differences.

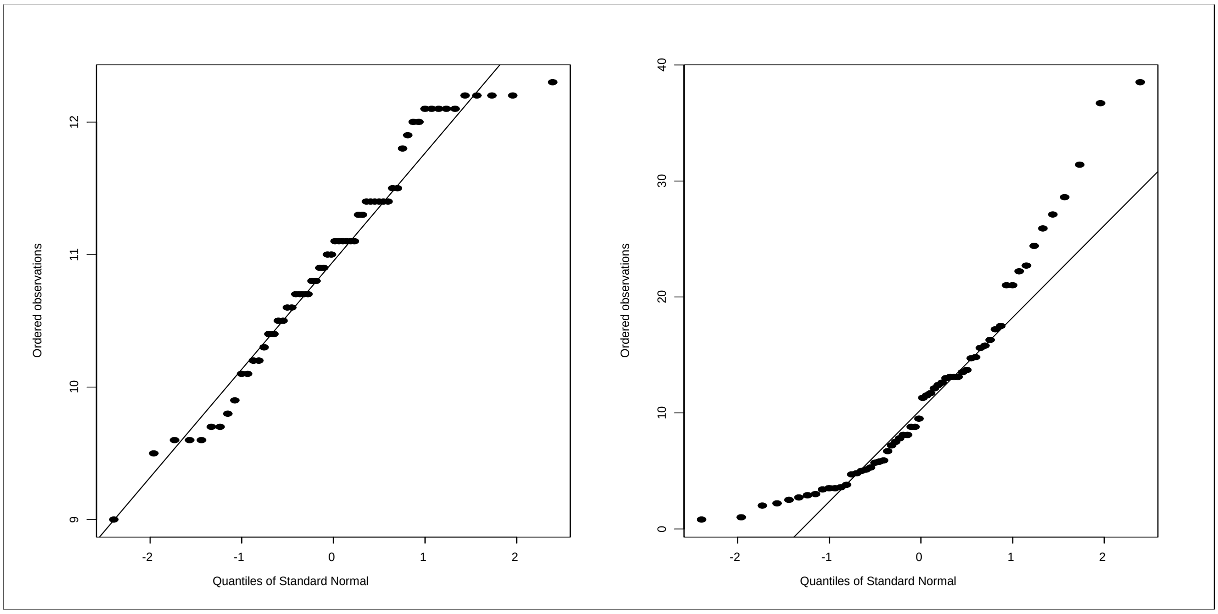
8.8 Assessing Multivariate Normality

If a p-dimensional random variable has a multivariate normal distribution then it follows that the p one dimensional marginal components must be univariate normal. However, the converse does not follow, it is possible that a p-dimensional viable has univariate Normal components but is not *multivariate* normal. This peculiarity means that although it is sensible to check each marginal component of sample data for Normality (e.g. by probability plotting) it does not follow that the multivariate data are satisfactorily multivariate Normally distributed for the statistical tests and other procedures to be appropriate. A further check is provided by the fact that the squared Mahalanobis distances of each observation from the mean

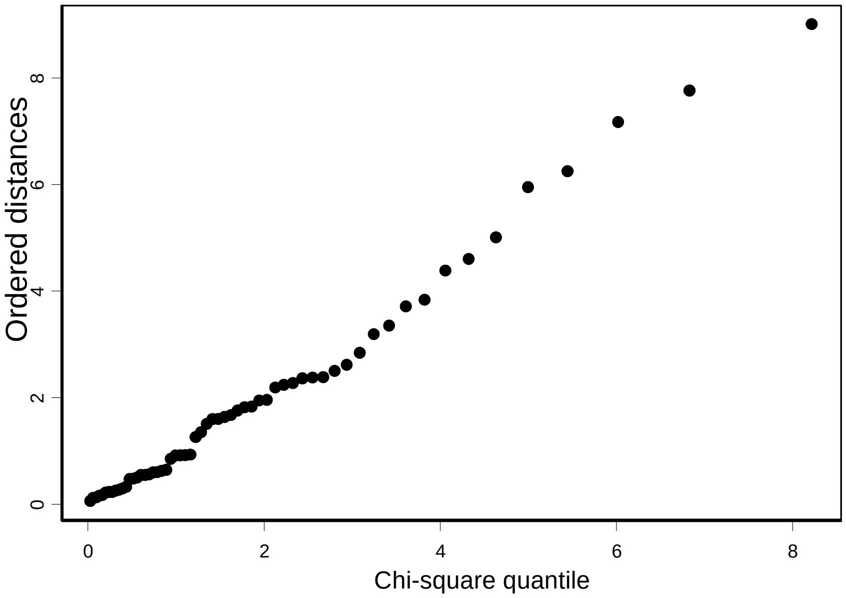
$$D_i^2 = (x_i - \overline{x})'S^{-1}(x_i - \overline{x})$$

have approximately a chi-squared distribution with p degrees of freedom, χ_p^2. These distances will not actually be independent but are nearly so, consequently a test of Normaility is provided by assessing the D_i^2 as a sample of observations from a χ_p^2-distribution. Everitt provides a function chisplot() for producing a chi-squared probability plot (i.e. ordered observations against quantiles of χ_p^2). As an example, consider Everitt's air pollution data airpoll and the variables Education and Nonwhite considered in §0.8. First there are the two Normal probability plots of the marginal components (which give clear cause for concern) followed by the chisquared plot which is also not very satisfactory:

```
> attach(airpoll)
> par(mfrow=c(1,2))
> X<-cbind(Education,Nonwhite)
> qqnorm(X[,1],ylab="Ordered observations")
> qqline(X[,1])
> qqnorm(X[,2],ylab="Ordered observations")
> qqline(X[,2])
>
```



```
> par(mfrow=c(1,1))
> chisplot(X)
```



8.9 Summary and Conclusions

- This chapter has illustrated the extension of basic univariate results to multivariate data.
- Multivariate Normal, Wishart and Hotelling's T^2-distributions were introduced.
- The sample mean and variance are unbiased estimates of the population mean and variance.
- If additionally, the data are Multivariate Normal then the sample mean is also Normal, the variance is Wishart and they are **independent.**
- One and two-sample T^2-tests are direct generalizations of univariate t-tests.
- Generalized likelihood ratio tests can be constructed of hypotheses which cannot arise in one dimension.
- Union-Intersection tests provide an alternative strategy for constructing tests. These have similarities with multivariate EDA techniques such as PCA and LDA in construction and interpretation of 'directions'.
- All standard tests can be performed in standard packages.

Tasks 10

(see §8.6)

1) Suppose we have samples of sizes n_1 and n_2 with means $\overline{x}_1$ and $\overline{x}_2$ and variances S_1 and S_2 from populations $N_p(\mu_1,\sigma_1^2)$ and $N_p(\mu_2,\sigma_2^2)$, let $S=[(n_1–1)S_1+(n_2–1)S_2]/(n–2)$ where $n=n_1+n_2$.

   i) Shew that the UIT of H_0: $\mu_1 = \mu_2$ *vs* H_A: $\mu_1 \neq \mu_2$ is given by Hotelling's $T^2 = \frac{n_1 n_2}{n}(\overline{x}_1 - \overline{x}_2)'S^{-1}(\overline{x}_1 - \overline{x}_2)$

   ii) Deduce that the greatest difference between the two populations is exhibited in the direction $S^{-1}(\overline{x}_1 - \overline{x}_2)$.

   [Suggestion: adapt the argument of §8.6.4]

2) Referring to the data set *dogmandibles.*∗ **excluding the Prehistoric Thai dogs (group 5 on X_{11})**

   i) **W**hat combination of length and breadth of mandible exhibits the greatest difference between Males and Females?

   ii) **W**hat combination of length and breadth of mandible exhibits the greatest difference between the four species?

9 Statistical Discriminant Analysis

9.0 Introduction

Suppose we collect data on ‘objects’ which can be classified into one or other of k *known* categories (k$\geq$2).

> e.g. ‘objects’ $\equiv$ patients and the k=3 categories are Rheumatoid Arthritis / Psoratic Arthritis / Psoriasis
>
> e.g. ‘objects’ $\equiv$ iris flowers and the k=3 categories are setosa / versicolor / virginica.

Suppose further that we measure p variates on each ‘object’ to obtain a p-dimensional datum x, so $x\in\Re^p$ — p-dimensional space.

A *discriminant rule*, d, is a partition of $\Re^p$ into k regions R_1, R_2, ..., R_k (so $\bigcup_{i=1}^{k} R_i = \Re^p$ and $R_i \cap R_j = \varnothing$) such that if $x \in R_j$ then we classify x (i.e. the ‘object’ with measurements x) as in category j.

This is a formal way of saying that a discriminant rule, d say, is a way of deciding unambiguously which category an object x belongs to, just on the basis of the measurements x.

The informal or *data-analytic* methods of classification considered in §4.6 were proposed on intuitive or *ad hoc* grounds but will all lead to discriminant rules, which potentially are different from each other.

Statistical discriminant analysis is concerned with

- Regarding the measurements x as observations of some random variable whose distribution depends upon which category x belongs to, i.e. if $x \in$category j then the density of x is $f_j(.)$
- How can we construct a discriminant rule from data x_1, x_2, ..., x_n known to belong to specified categories
- What are the [statistical] properties of the rule and in particular how can we evaluate one rule d in relation to another d^*.

Two cases need to be distinguished:

(i) All $f_j(.)$ known completely

(ii) $f_j(.)$ not known but observations from known categories are available

Case (ii) can be subdivided into

$(ii)_a$ $f_j(.)$ assumed to be of known parametric form depending on unknown but estimable parameters

$(ii)_b$ no such assumption made

Generally, the most exact statistical theory is available only for the case (i) which is unrealistic in practice — extensions to case (ii) are made on the basis of appeals to asymptotic theory of large samples etc and largely consist of estimating any unknown parameters (or densities) and then proceeding as if the densities were known, as in case (i).

9.1 Discrimination with k known category densities

(i.e. all $f_j(.)$ known completely)

9.1.1 The Maximum Likelihood Discriminant Rule

The maximum likelihood discriminant rule is to allocate x to that category which gives the greatest likelihood to x.

i.e. allocate x to category j where $f_j(x)=\max_{i=1}^{k}\left(f_i(x)\right)$

> [Note that technically we should only consider likelihoods of *parameters* for given data, not of data themselves. Here we are effectively considering the index j of the categories as the parameter and so are considering the likelihood of j for data x].

Ex 9.1.1.1 Two one-dimensional Normal populations

p=1, k=2, $x \sim N(u_i, \sigma_i^2)$ if in category i, i=1,2.

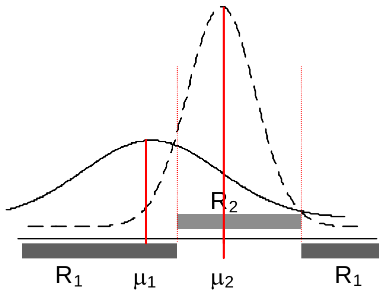
We allocate x to category 1 if $f_1(x)>f_2(x)$. i.e. if

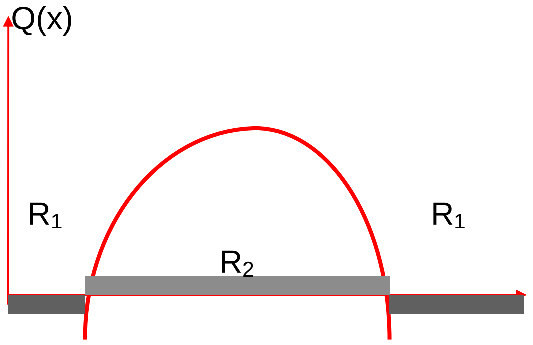
$$\tfrac{\sigma_2}{\sigma_1}\exp\left\{-\tfrac{1}{2}\left(\tfrac{x-\mu_1}{\sigma_1}\right)^2+\tfrac{1}{2}\left(\tfrac{x-\mu_2}{\sigma_2}\right)^2\right\}>1$$

i.e. if $Q(x)=x^2\left(\tfrac{1}{\sigma_1^2}-\tfrac{1}{\sigma_2^2}\right)-2x\left(\tfrac{\mu_1}{\sigma_1^2}-\tfrac{\mu_2}{\sigma_2^2}\right)+\left\{\left(\tfrac{\mu_1^2}{\sigma_1^2}-\tfrac{\mu_2^2}{\sigma_2^2}\right)-2\log(\tfrac{\sigma_2}{\sigma_1})\right\}<0$

Suppose $\sigma_1>\sigma_2$ and the coefficient of x^2 in Q(x) is negative.

Then Q(x) < 0 if x is sufficiently small or sufficiently big.

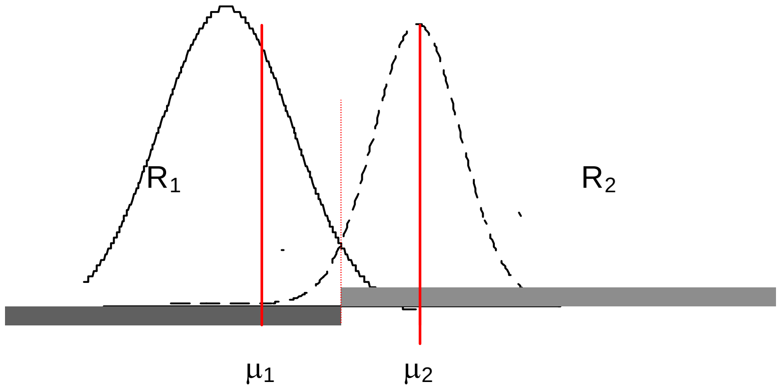


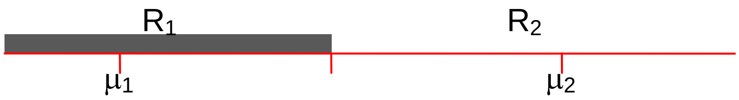


If $\sigma_1=\sigma_2$ then $\log(\sigma_2/\sigma_1)=0$ and the rule becomes allocate to 1 if

$$|x-\mu_1| < |x-\mu_2|$$

i.e. if $\mu_1<\mu_2$ then allocate to 1 if $x<½(\mu_1+\mu_2)$





Ex 9.1.1.2 p dimensions, k Normal populations

means μ_i, common variance Σ.

$$f_i(x)=(2\pi)^{-p/2}|\Sigma|^{-½}\exp\{-½(x-\mu_i)'\Sigma^{-1}(x-\mu_i)\}$$

which is maximized when $(x-\mu_i)'\,\Sigma^{-1}(x-\mu_i)$ is minimized,

i.e. allocate x to the category whose mean has the smallest Mahalanobis distance from x.

Comments 9.1.1.3

When k=2, Ex 8.1.1.2 reduces to the rule: allocate x to category 1

$$\text{if } (\mu_1-\mu_2)'\Sigma^{-1}(x-\mu)>0, \text{ where } \mu=½(\mu_1+\mu_2).$$

i.e. the dividing point (p=1), line (p=2), plane (p=3), hyperplane (p>3) between populations 1 & 2 with the *same* variances is

$$(\mu_1-\mu_2)'\Sigma^{-1}(x-\mu)=0$$

i.e. the Maximum Likelihood Discriminant Function for two Normal populations with the ***same*** variances is **linear.**

i.e. the boundary between the allocation regions is a hyperplane passing through μ, the mid-point of the line joining μ_1 and μ_2 (though is not necessarily perpendicular to this line).

9.1.2 Bayes Discriminant Rules

In some circumstances it is sensible to recognise that there are differing *a priori* probabilities of membership of categories. For example, in medical diagnosis some conditions may be very rare and others very common although the symptoms (i.e. measurements available) may be very similar for the differing conditions (e.g. 'flu and polio). In these cases it is reasonable to make some allowance for this and shift the balance of classifying towards the more common category.

If the k categories have prior probabilities π_1, π_2, ..., π_k then the **Bayes Discriminant Rule** is to allocate x to that category for which $\pi_i f_i(x)$ is greatest.

The Maximum Likelihood Rule is equivalent to a Bayes Discriminant Rule if $\pi_1=\pi_2= ... =\pi_k=k^{-1}$ i.e. the prior probabilities are equal.

9.2 Discrimination Under Estimation

9.2.1 Sample ML Discriminant Rule

Suppose we know the form of the distributions $f_j(.)$ up to a few unknown parameters and that we have n_i observations ***known*** to be from category i, each i=1, 2, ..., k. The idea is to replace unknown parameters in 8.1 by their mles. The approach is pragmatic and the theoretical justification is an appeal to the consistency of mles (i.e. for 'large' samples).

In particular, in the case k=2, two normal populations $N_p(\mu_i,\Sigma)$ (common variance) then estimate μ_1 and μ_2 by the sample means and Σ by the pooled sample variance and then the rule becomes to allocate x to category 1 if $(\overline{x}_1-\overline{x}_2)'W^{-1}\{x-{}^1\!/\!_2(\overline{x}_1+\overline{x}_2)\}>0$

where $W=\frac{1}{n-2}\sum_{i=1}^{2}(n_i-1)S_i$, the pooled sample variance.

Ex 9.2.1.1 Iris Setosa and Versicolor

Taking just sepal length and width gives

$$\overline{x}_1=\begin{pmatrix}5.01\\3.43\end{pmatrix},\ \overline{x}_2=\begin{pmatrix}5.94\\2.77\end{pmatrix},\ W=\begin{pmatrix}0.195 & 0.092\\0.092 & 0.121\end{pmatrix}$$

so $(\overline{x}_1-\overline{x}_2)'W^{-1}=(-11.44,14.14)$ and the rule is to

allocate x=(x_1, x_2)′ if $(-1144,1414)\begin{pmatrix}x_1-\frac{1}{2}(5.01+5.94)\\x_2-\frac{1}{2}(3.43+2.77)\end{pmatrix}>0$

i.e. if $-11.44x_1+14.14x_2+18.74>0$.

9.2.2 The Likelihood Ratio Discriminant Rule

This is a subtly different generalization of §8.1.

Let H_i : x and the n_i observations *known* to be from category i are all from category i, all others from known categories.

The rule is to allocate x to category j where $\ell_{\max}(H_j) = \max\limits_{i=1}^{k}\{\ell_{\max}(H_i)\}$

The distinction is that x is included in the ML estimation.

Ex 9.2.2.1 Two Normal populations $N_p(\mu_i, \Sigma)$

n_i observations from category i with means $\bar{x}_i$, variances S_i, i=1,2.

Let H_1: x from population 1. Then under H_1 we have n_1+1 observations from population 1 which have mean $\frac{n_1\bar{x}_1+x}{n_1+1}$ and n_2 from population 2.

So under H_1 we have $\hat{\mu}_1 = \frac{n_1\bar{x}_1+x}{n_1+1}$ and $\hat{\mu}_2 = \bar{x}_2$ and the m.l.e of Σ is

$$\hat{\Sigma}_1 = \tfrac{1}{n_1+n_2+1}\left\{ (n_1-1)S_1 + (n_2-1)S_2 + \tfrac{n_1}{n_1+1}(x-\bar{x}_1)(x-\bar{x}_1)' \right\}$$

and under H_2 we have $\hat{\mu}_1 = \bar{x}_1$ and $\hat{\mu}_2 = \frac{n_2\bar{x}_2+x}{n_2+1}$ and the m.l.e of Σ is

$$\hat{\Sigma}_2 = \tfrac{1}{n_1+n_2+1}\left\{ (n_1-1)S_1 + (n_2-1)S_2 + \tfrac{n_2}{n_2+1}(x-\bar{x}_2)(x-\bar{x}_2)' \right\}$$

Now $\ell_{\max}(H_1) - \ell_{\max}(H_2) = \tfrac{1}{2}(n_1 + n_2 + 1)\{\log|\hat{\Sigma}_2| - \log|\hat{\Sigma}_1|\}$

and $|\hat{\Sigma}_i| = \left| \tfrac{1}{n_1+n_2+1}\left\{ (n_1 - 1)S_1 + (n_2 - 1)S_2 + \tfrac{n_i}{n_i+1}(x - \overline{x}_i)(x - \overline{x}_i)' \right\} \right|$

$$=(n_1+n_2+1)^{-p}|T+\tfrac{n_i}{n_i+1}(x - \overline{x}_i)(x - \overline{x}_i)'|$$

where $T=(n_1-1)S_1+(n_2-1)S_2$

$$=(n_1+n_2+1)^{-p}|T|.|I_p+\tfrac{n_i}{n_i+1}T^{-1}(x - \overline{x}_i)(x - \overline{x}_i)'|$$

$$=(n_1+n_2+1)^{-p}|T|.|1+\tfrac{n_i}{n_i+1}(x - \overline{x}_i)'T^{-1}(x - \overline{x}_i)|$$

(using result that $|I_p+A_{p\times n}B_{n\times p}|=|I_n+B_{n\times p}A_{p\times n}|$)

so $\ell_{\max}(H_1) > \ell_{\max}(H_2)$ and x is allocated to population 1 if

$$\tfrac{n_2}{n_2+1}(x - \overline{x}_2)'T^{-1}(x - \overline{x}_2) > \tfrac{n_1}{n_1+1}(x - \overline{x}_1)'T^{-1}(x - \overline{x}_1)$$

If $n_1=n_2$ then this is the same as the sample ML rule, and if n_1 and n_2 are both large then it is almost so. However if either sample size is small and $n_1 \neq n_2$ then the allocation is different.

9.3 Fisher's Linear Discriminant Function

In §4 we shewed that if the data are projected into one dimension, the projection which maximizes the ratio of the between to within groups sums of squares, $a'Ba/a'Wa$ is the [right] eigenvector of $W^{-1}B$ corresponding to its largest eigenvalue.

New observations, x, can be classified according to any of the criteria in §4.7. The most common practice is to calculate the discriminant score, $a'x$, and allocate to that group j where

$$\min_{i=1}^{k} | a'x - a'\bar{x}_i | = a'x - a'\bar{x}_j$$

In the case k=2, B has rank 1 and $B = \frac{n_1 n_2}{n}(\bar{x}_1 - \bar{x}_2)(\bar{x}_1 - \bar{x}_2) = \frac{n_1 n_2}{n} dd'$ (say) and $W^{-1}B$ has only one non-zero eigenvalue which is $\frac{n_1 n_2}{n} d'W^{-1}d$ and eigenvector $W^{-1}d$ and the rule is to allocate to 1 if

$$(\bar{x}_1 - \bar{x}_2)'W^{-1}\{x - \tfrac{1}{2}(\bar{x}_1 + \bar{x}_2)\} > 0$$

i.e. the same as the sample discriminant rule.

The function $h(x)=a'x$ where a is the first eigenvector of $W^{-`}B$, is called Fisher's Linear Discriminant Function.

9.4 Probabilities of Misclassification

9.4.1 Introduction

Let p_{ij} = P[a type j object is classified as type i]

— the ***performance*** of a discriminant rule is described by the p_{ij}. Good rules have p_{ij} small for $i \neq j$ and big for $i=j$ (i.e. low probabilities of *misclassification* and high probabilities of *correct* classification).

If d and d^* are two discriminant rules with classification probabilities p_{ij} and p^*_{ij} then we say d is ***better*** than d^* if

$$p_{ii} \geq p^*_{ii} \text{ for all } i=1,2,...,k \text{ and } p_{jj} > p^*_{jj} \text{ for at least one } j,\ 1 \leq j \leq k.$$

If d is a discriminant rule for which there is no better rule then d is ***admissible***. If d is better than d^* then d^* is ***inadmissible***.

Note that it may be that it is not possible to compare two rules d and d^* since, for example, it could be that $p_{11} > p^*_{11}$ but $p_{22} < p^*_{22}$.

Suppose d is defined by the partition $\{R_i\}$, i.e. x is classified as category j if $x \in R_j$, where $R_i \cap R_j = \varnothing$ if $i \neq j$ and $\bigcup_{i=1}^{k} R_i = \Re^p$, and suppose the density of x is $f_j(.)$ when x is of type j,

i.e. if x belongs to category j then x has density $f_j(.)$.

Now $\quad p_{ij} = P[x \in R_i$ when x is of type j]

$= P[x \in R_i$ when x has density $f_j(.)]$

$$= \int_{R_i} f_j(x)dx$$

$$= \int_{R_i} \phi_i(x)f_j(x)dx \text{ where } \phi_i(x) \text{ is the indicator function of } R_i$$

(i.e. $\phi_i(x)=1$ if $x \in R_i$ and $\phi_i(x)=0$ if $x \notin R_i$)

$$= \int_{\Re^p} \phi_i(x)f_j(x)dx$$

9.4.2 Good Discriminant Rules

Consider the case k=2 suppose we consider the probabilities of misclassification p_{12} and p_{21} and try to find a rule, d, which minimizes these.

We could do this by minimizing p_{21} for fixed p_{12} (c.f. minimizing β, type II error, for fixed α, type I error in Neyman-Pearson theory of hypothesis testing).

To preserve symmetry of the overall setup we consider instead a weighted sum of p_{12} and p_{21},

say $L_d=\omega_2 p_{12}+\omega_1 p_{21}$

$$=\int_{\Re^p} \phi_1(x)\omega_2 f_2(x)+\phi_2(x)\omega_1 f_1(x)dx$$

$$=\int_{\Re^p} \tau_d(x)dx \text{ say.}$$

Now $\min_d\{L_d\} \geq \int_{\Re^p} \min_d\{\tau_d(x)\}dx$ with equality holding if for each [fixed] x we take d=d*, where $\tau_{d^*}(x)=\min_d\{\tau_d(x)\}$

i.e. minimize L_d by constructing d so that τ_d is minimized for each fixed x — which is done by taking

$\phi_1(x)= 1$ if $\omega_2 f_2(x) < \omega_1 f_1(x)$ and $\phi_2(x)= 1$ if $\omega_1 f_1(x) < \omega_2 f_2(x)$

(note that defining $\phi_1(.)$ and $\phi_2(.)$ defines R_1 and R_2 which defines the rule d).

Notice that if $\omega_1=\omega_2$ then L_d = sum of misclassification probabilities =P[incorrect classification] and this construction gives the maximum likelihood rule.

Notice also that taking $\omega_i = \pi_i$ the prior probability of category i gives the Bayes Discriminant Rule.

In fact, all Bayes discriminant rules are ***admissible***:

Suppose d^* is a Bayes discriminant rule with respect to prior probabilities $\pi_1, \pi, ..., \pi_k$ and that d is a *better* rule than d^*. Then $p_{ii} \geq p^*_{ii}$ with strict inequality for at least one i.

So $\Sigma\pi_i p_{ii} > \Sigma\pi_i p^*_{ii}$.

But $\Sigma\pi_i p_{ii} = \Sigma\int\pi_i\phi_i(x)f_i(x)$

$\leq \sum_i \int \phi_i(x)\max_j\{\pi_j f_j(x)\}dx$

$= \int\{\Sigma\phi_i(x)\}\max\{\pi_j f_j(x)\}dx$

$= \int\max\{\pi_j f_j(x)\}dx$ (since$\Sigma\phi_i(x)=1$ for any x)

$= \int\Sigma\phi^*_i(x)\pi_i f_i(x)dx$ (since Bayes discriminant rules maximize $\pi_i f_i(x)$ thus defining $\phi^*_i(.)$ and in the summation only one term is non-zero for any x)

$= \Sigma\pi_i p^*_{ii}$ which is a contradiction.

Note that in particular the maximum likelihood rule is admissible (since taking $\pi_i = k^{-1}$ for each i=1,2,...,k gives this rule).

Note also that this shews that there are arbitrarily many admissible rules since there are arbitrarily many different prior distributions over the k categories.

9.4.3 Particular Cases

Ex 9.4.3.1 Two Normal Populations $N_p(\mu_i, \Sigma)$

Let $\mu=½(\mu_1+\mu_2)$ then when $x\in$ Pop^n 1

we have $\alpha'(x-\mu)\sim N(½\alpha'(\mu_1-\mu_2), \alpha'\Sigma\alpha)$ for any vector α.

When we classify x on the value of the discriminant function $h(x)=\alpha'(x-\mu)$ with $\alpha=\Sigma^{-1}(\mu_1-\mu_2)$ (classifying as Pop^n 1 if $h(x)>0$) then $h(x)\sim N(½\Delta^2,\Delta^2)$ where $\Delta^2=(\mu_1-\mu_2)'\Sigma^{-1}(\mu_1-\mu_2)$.

Similarly if $x\in$Pop^n 2, $h(x)\sim N(-½\Delta^2,\Delta^2)$.

So, $p_{12}=P[h(x)>0|\ x\in\text{Pop}^n\ 2] = \Phi\ = \Phi\left(\frac{-½\Delta^2}{\sqrt{\Delta^2}}\right)=\Phi(-½\Delta)$

and similarly $p_{21}=\Phi(-½\Delta)$. Thus, for two Normal populations with a common variance ***the misclassification probabilities are equal.***

Ex. 9.4.3.2 Correlation in the Bivariate Normal

Consider the case when p=2, with Normal populations $N_2(0,\Sigma)$ and $N_2(\mu,\Sigma)$ where $\mu=(\mu_1,\mu_2)'$ and suppose $\mu_1,\mu_2>0$ and $\Sigma=\begin{pmatrix}1 & \rho\\ \rho & 1\end{pmatrix}\sigma^2$

Then $\Delta^2=\mu'\Sigma^{-1}\mu=\frac{\mu_1^2+\mu_2^2-2\rho\mu_1\mu_2}{\sigma^2(1-\rho^2)}$

If the variables were uncorrelated then we would have

$$\Delta^2=\frac{\mu_1^2+\mu_2^2}{\sigma^2}=\Delta_0^2$$

Now the correlation will 'improve' the discrimination (i.e. reduce p_{12}) if $\Delta^2>\Delta_0^2$, i.e. if $\frac{\mu_1^2+\mu_2^2-2\rho\mu_1\mu_2}{(1-\rho^2)}>\mu_1^2+\mu_2^2$,

i.e. if $\rho((1+(\mu_2/\mu_1)^2)\rho-2(\mu_2/\mu_1))>0$, i.e. if $\rho<0$ or $\rho>\frac{2\mu_1\mu_2}{\mu_1^2+\mu_2^2}$,

so if $\mu_1=\mu_2$ then any positive correlation reduces the power of discrimination.

Ex 9.4.3.3 One variable or two?

Continuing the above example, if the rule were based just on measurements of the first variable then $p_{12}=\Phi(-½\Delta^{*})$ where $\Delta^{*2}=\frac{\mu_1^2}{\sigma^2}$

and so using both variables instead of just one improves the discrimination if $\frac{\mu_1^2+\mu_2^2-2\rho\mu_1\mu_2}{\sigma^2(1-\rho^2)} > \mu_1^2$

i.e. if $\rho^2\mu_1^2 + \mu_2^2 - 2\rho\mu_1\mu_2 > 0$

i.e. if $(\rho\mu_1-\mu_2)^2 > 0$,

i.e. **always.**

9.4.4 Misclassification Probabilities under Estimation

If parameters are estimated then these can be used to estimate the p_{ij}.

For example, in Ex. 9.4.3.1 we can obtain $\hat{p}_{12} = \Phi(-½\hat{\Delta})$ where

$$\hat{\Delta}^2 = (\overline{x}_1 - \overline{x}_2)'S^{-1}(\overline{x}_1 - \overline{x}_2)$$

In example 9.2.1.1 of discriminating between iris setosa and iris versicola this gives $p_{12}=0.013$ (i.e. 1.3%).

Generally, such estimates tend to be over-optimistic — i.e. biased downwards.

Alternatively, given any discriminant rule $\{R_i\}$ let

$n_{ij} = \#\{x\in Pop^n{}_{j,}\ x\in R_i\}$ then we can estimate p_{ij} by

$\hat{p}_{ij} = \frac{n_{ij}}{\sum_i n_{ij}} = \frac{n_{ij}}{n_j}$ = proportion from pop^n j misclassified as i.

Again, tends to be over-optimistic — improve by jackknifing or permutation tests etc.

9.4.4.1 Jackknifing

The discriminant rule is calculated by leaving out each observation in turn and then using that rule to classify the omitted observation. The overall correct classification rate is then a *jackknife estimate* of the true probability of correct classification.

9.4.4.2 Permutation and Randomization Tests

To assess whether the observed correct classification rate is ‘significantly’ higher than would be achieved by chance it is possible to perform the complete discrimant analysis on the same observations but by permuting the labels of group membership and calculating the correct classification rate for this permuted set of labels. Observing where the rate obtained from the correct labels falls in this *permutation distribution* provides the *permutation test:* if it is in the upper tail of the distribution then there is evidence that the rate obtained is higher than would be achieved by chance.

Typically, the number of permutations is too large to compute the complete permutation distribution and so a reasonable number (e.g. 99 or 999) random permutations are used to construct a *randomisation test.* Random permutations can be obtained by sampling with replacement from the label variable.

9.5 Summary and Conclusions

- This chapter has considered the formal problem of classifying new observations on the basis of rules constructed from training data.
- The ideal *Maximum Likelihood Rule* which assumes that all densities are known has two sample versions:– the *sample discriminant rule* and the *likelihood ratio discriminant rule* which can give different results if the samples sizes are small and very different.
- Probabilities of misclassification were considered and it was shewn that the ideal Maximum Likelihood Rule minimized the total probability of misclassification.
- Admissible and inadmissible rules were defined. Bayes rules are always admissible.
- Methods for estimating misclassification probabilities were outlined, in particular by jackknifing and by using randomisation tests.
- Some illustrations of the use of randomisation tests are given in Appendices 1 & 2. Appendix 8 on Neural Networks gives some examples of simulation methods similar to (but more general than) jackknifing.

Tasks 11

(see §9.0–§9.5 & revision of §4.4–§4.7)

[Note that these questions are more substantial than on previous task sheets. Question 1 is a past examination question. Question 3 is only of benefit to those wanting more practice on PCA interpretation and practical data anlysis]

1) An archaeologist wishes to distinguish pottery from two different sources on the basis of its chemical composition. Measurements by Neutron Activation Analysis of the concentrations in parts per million of trace elements Cr and V in 19 samples of pottery from Tell el-Amarna gave mean results of 2.3 and 6.7, respectively, with sample variances 0.62 and 1.41 and covariance 0.09. Similar measurements on 23 samples from Memphis gave mean results of 2.9 and 5.9 with sample variances 0.7 and 1.36 and sample covariance 0.08.

i) Assuming that these measurements are adequately modelled by bivariate Normal distributions with a common variance, calculate the linear discriminant rule for distinguishing Amarna from Memphis pottery on the basis of the concentrations of Cr and V.

ii) Prove that the estimated probabilities of misclassifying Memphis pottery as Amarna and *vice versa* are the same using this rule.

iii) By how much is this misclassification probability an improvement over those using each of the elements separately?

iv) What advice would you give to the archaeologist in the light of these results?

2) Referring to the data set *dogmandibles.*∗ (including the Prehistoric Thai dogs (group 5 on X_{11}))

   i) Using lda() in **R** look at the discrimination between the 5 species (using the nine measurements) and estimate the classifcation rate. [In **R** it is easy to find the cross-validation (or jackknife) estimate of classification rate using the CV=T option].

   ii) Perform the discriminant analysis just on the first four [modern] species and then use this to classify the prehistoric Thai dogs.

   iii) Compare the results of these analyses with the results of the more informal exploratory analyses with Crimcoords in Exercises 2.

3) The datafile CLAYPOTS has 272 observations on the trace element content of clay samples from pots found at various archaeological sites around the Aegean. Column 1 gives the group number (i.e. archaeological site for most of the pots) and columns 2–9 give the amounts of 9 trace elements (which have been labelled A to I) found in samples of clay from the pots. It is suggested that before investigating the specific questions below it is advisable to do some exploratory analysis with PCA etc. Groups 1, 3 and 4 are from known sources; groups 2 and 5 are from unknown sources but are believed to come from one or other of 1,3 or 4.

   i) Construct a display on crimcoords of groups 1,3 and 4 and add in the points from groups 2 and 5.

Which are the best classifications of these pots?

Exercises 3

1)

i) Measurements of cranial length x_{11} and cranial breadth x_{12} on 35 female frogs gave $\overline{x}_1'$=(22.860, 24.397) and

$$S_1 = \begin{pmatrix} 17.683 & 20.290 \\ * & 24.407 \end{pmatrix}.$$ Test the hypothesis that $\mu_{11}=\mu_{12}$.

ii) Similar measurements on 14 male frogs gave

$\overline{x}_2'$=(21.821, 22.843) and $$S_2 = \begin{pmatrix} 18.479 & 19.095 \\ * & 20.756 \end{pmatrix}.$$

Calculate the pooled variance matrix for male & female frogs and test the hypothesis that female & male frogs come from populations with equal mean vectors.

2) Using you favourite computer package, access the British Museum Mummy Pots data (see task sheet for week 4) and calculate the two shape variables ‘`taper`’ and ‘`point`’.

i) Do the two batches of pots differ in overall shape as reflected by the calculated shape measures ‘`taper`’ and ‘`point`’?

ii) Do the two batches of pots differ in overall size?

iii) Without doing any calculations,

a) would your answer to (ii) be different in any respect if you used the scores on the three PCs calculated from the size variables?

b) would it make any difference were you to calculate the PCs using the correlation matrix instead of the covariance matrix?

[Suggestion: Read §8.7.4 of the lecture notes]

3) ⋆ $x_1,\ldots,x_n$ are independent measurements of $N_p(\mu,\sigma^2 I_p)$

i) Shew that the maximum likelihood estimate of μ, subject to $\mu'\mu = r_0^2$ (a known constant) is the same whether σ is known or unknown.

ii) Find the maximum likelihood estimate of σ when neither μ nor σ are known.

iii) Hence, in the case when $\sigma = \sigma_0$ (a known constant) consruct the likelihood ratio test of $H_0 : \mu'\mu = r_0^2$ *vs* $H_A : \mu'\mu \neq r_0^2$ based on n independent observations of $N_p(\mu,\sigma_0^2 I_p)$.

iv) In an experiment to test the range of a new ground-to-air missile thirty-nine test firings at a tethered balloon were performed and the three dimensional coordinates of the point of ignition of the missile's warhead measured. These gave a mean result of $(0.76, 0.69, 0.66)'$ relative to the site expressed in terms of the target distance. Presuming that individual measurements are independently normally distributed with unit variance, are the data consistent with the theory that the range of the missile was set correctly?

Notes & Solutions for Tasks 1

1) *Read the Study Guide for this course if you have not already done so*

   Trust you have done this by now

2) *If A is any* $p\times q$ *matrix then* $var(X'A)=A'var(X')A=A'SA$,

   This actually follows directly from the expression for var(Y) putting $y_i=A'x_i$ etc and is essentially identical to the special case when q=1 and A is a vector.

3) *Access the Iris Dataset.*

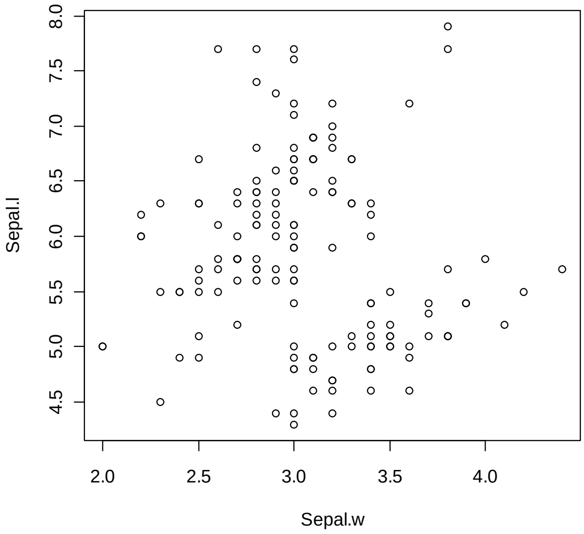
   i) *Find the 4-vector which is the mean of the four dimensions* `Sepal.l, Sepal.w, Petal.l, Petal.w` *and the 4×4 matrix which is their variance.*

```
> attach(irisnf)
> apply(cbind(Sepal.l,Sepal.w,Petal.l,Petal.w),2,mean)
 Sepal.l Sepal.w Petal.l Petal.w
  5.8433  3.0553   3.758  1.1993
> var(cbind(Sepal.l,Sepal.w,Petal.l,Petal.w))
          Sepal.l   Sepal.w  Petal.l  Petal.w
Sepal.l  0.685694 -0.040736  1.27432  0.51627
Sepal.w -0.040736  0.193629 -0.32873 -0.12124
Petal.l  1.274315 -0.328734  3.11628  1.29561
Petal.w  0.516271 -0.121238  1.29561  0.58101
>
```

ii) *. Plot sepal length against sepal width using:*

a) the default choices

```
> plot(Sepal.w,Sepal.l)
```

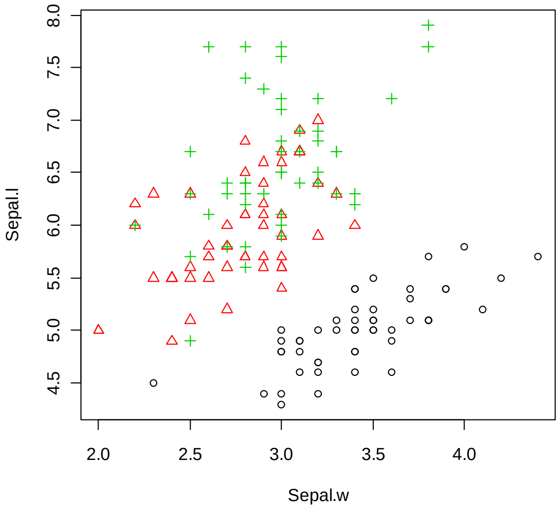


```
>
```

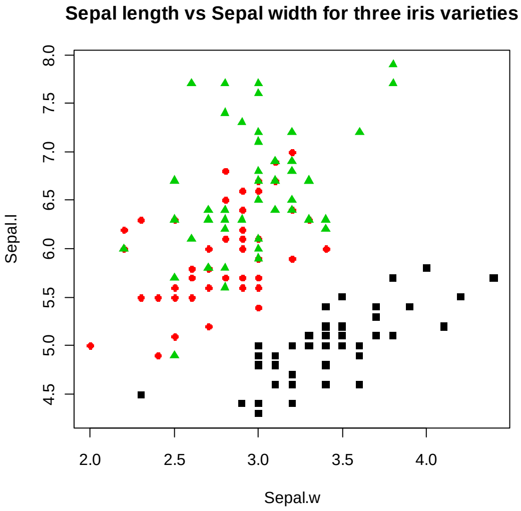
Note that to plot length ***against*** width you ***must*** have length on vertical and width on horizontal axis.

b) using different symbols for each variety (explore the menus and panels, and maybe the help system to find out how to do this). Also try adding titles etc.

```
> plot(Sepal.w,Sepal.l, pch=unclass(Variety),
+ col=unclass(Variety))
```

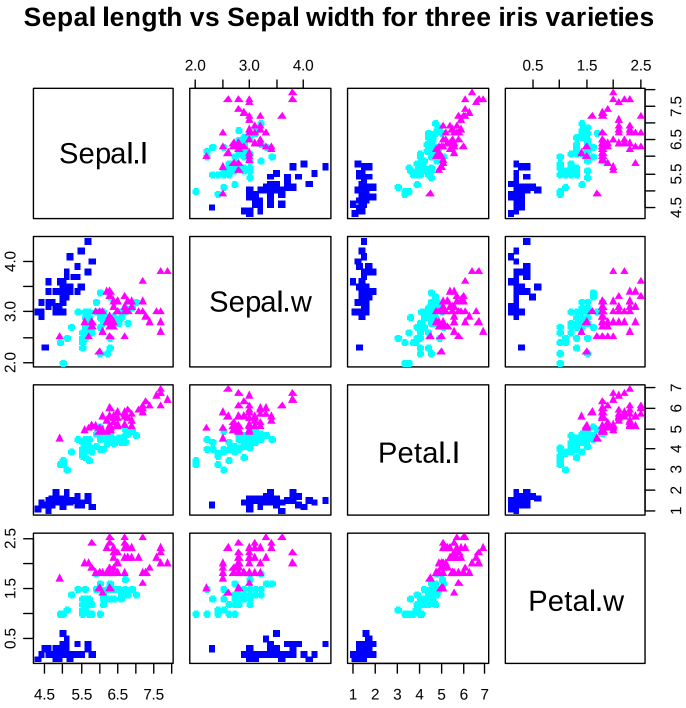


```
> plot(Sepal.w,Sepal.l, pch=unclass(Variety)+14,
+ col=unclass(Variety),
+ main="Sepal length vs Sepal width for three iris varieties")
```



iii) Construct a matrix plot of all four dimensions, using first the default choices and then enhancing the display as above.

```
> pairs(cbind(Sepal.l,Sepal.w,Petal.l,Petal.w),
+ pch=unclass(Variety)+14, col=unclass(Variety)+3,
+ main="Sepal length vs Sepal width for three iris varieties")
```



iv) *Try the commands*

```
       var(irisnf)
       diag(var(irisnf))
```

```
> options(digits=3)
> var(irisnf)
        Sepal.l Sepal.w Petal.l Petal.w Variety
Sepal.l  0.6857 -0.0407   1.274   0.516   0.531
Sepal.w -0.0407  0.1936  -0.329  -0.121  -0.152
Petal.l  1.2743 -0.3287   3.116   1.296   1.372
Petal.w  0.5163 -0.1212   1.296   0.581   0.597
Variety  0.5309 -0.1523   1.372   0.597   0.671
> diag(var(irisnf))
Sepal.l Sepal.w Petal.l Petal.w Variety
  0.686   0.194   3.116   0.581   0.671
>
```

4) *Try these simple exercises both 'by hand' and R.*

i) *Let* $a=\begin{pmatrix}1\\2\\3\end{pmatrix}$, $A=\begin{pmatrix}1&2&3\\4&5&6\end{pmatrix}$, $B=\begin{pmatrix}1&2\\3&4\\5&6\end{pmatrix}$,

Find AB, B′A′, BA, a′A, a′Aa

```
> a<-matrix(c(1,2,3),3,1)
> A<-matrix(c(1,2,3,4,5,6),2,3,byrow=T)
> B<-matrix(c(1,2,3,4,5,6),3,2,byrow=T)
> A%*%B
     [,1] [,2]
[1,]   22   28
[2,]   49   64
> t(B)%*%t(A)
     [,1] [,2]
[1,]   22   49
[2,]   28   64
> B%*%A
     [,1] [,2] [,3]
[1,]    9   12   15
[2,]   19   26   33
[3,]   29   40   51
> t(a)%*%A
Error in t(a) %*% A : non-conformable arguments
> t(a)%*%A%*%a
Error in t(a) %*% A : non-conformable arguments
>
```

Note that a′A and a′Aa are not defined since the dimensions do not match.

5) *Read through the Sections on eigenvalues and eigenvectors, differentiation w.r.t. vectors and use of Lagrange Multipliers in the Background Results booklet.* I trust that you have done this by now and would have contacted me if there were any problems.

6) *Read the Study Guide for this course [again] if you have not already done so [or have done so only once]…*and this also [again].

Notes & Solutions for Tasks 2

1)

i) *Find the eigenvalues and normalized eigenvectors of the 2×2 matrix*

$$\frac{1}{7}\begin{pmatrix} 208 & 144 \\ 144 & 292 \end{pmatrix}$$

Solving $\begin{vmatrix} \frac{208}{7}-\lambda & \frac{144}{7} \\ \frac{144}{7} & \frac{292}{7}-\lambda \end{vmatrix}=0$ gives λ^2–$500/7\lambda+40000/7^2=0$ so $\lambda_1=400/7$ and $\lambda_2=100/7$. Putting $(S–\lambda_1I_3)a_1=0$ gives

$–192a_{11}+144a_{12}=0$ and $144\,a_{11}–108a_{12}=0$.

(Note that these two equations are essentially identical). Using the normalizing constraint that $a^2_{11}+a^2_{12}=1$ gives $a_{11}=3/5=0.6$ and $a_{12}=0.8$. Similarly $a_{21}=0.8$ and $a_{22}=–0.6$

(note that $a_1=(–0.6,–0.8)'$ and/or $a_2=(–0.8,0.6)'$ are equally acceptable solutions for a_1 and a_2 since the signs of eigenvectors are arbitrary.

```
> s<-matrix(c(208,144,144,292),nrow=2,ncol=2)/7
> eigen(s)
$values
[1] 57.14286 14.28571

$vectors
     [,1] [,2]
[1,]  0.6 -0.8
[2,]  0.8  0.6

>
```

ii) *Find the eigenvalues and one possible set of normalized eigenvectors of the 3×3 matrix* $\begin{pmatrix} 2 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 2 \end{pmatrix}$

$|S\text{-}\lambda I_3|=\lambda^3-6\lambda^2+9\lambda-4=(\lambda-4)(\lambda-1)^2$, so $\lambda_1=4$ and $\lambda_2=\lambda_3=1$.

$(S-\lambda I_3)a_1=0 \Rightarrow a_{12}+a_{13}=2a_{11}$, $a_{11}+a_{13}=2a_{12}$ and $a_{11}+a_{12}=2a_{13}$ so $a_{11}=a_{12}=a_{13}$ and since $a_1'a_1=1$ we have $a_1=3^{-½}(1,1,1)'$. For a_2 and a_3 we need **any** two normalized orthogonal vectors which are also orthogonal to the unit vector:

e.g. $6^{-½}(1,1,-2)'$ and $2^{-½}(1,-1,0)'$ or equally well $38^{-½}(2,3,-5)'$ and $114^{-½}(-8,7,1)$ **or infinitely many other possibilities.**

```
> options(digits=3)
> t<-matrix(c(2,1,1,1,2,1,1,1,2),nrow=3,ncol=3)
> eigen(t)
$values
[1] 4 1 1

$vectors
       [,1]   [,2]   [,3]
[1,] -0.577  0.816  0.000
[2,] -0.577 -0.408 -0.707
[3,] -0.577 -0.408  0.707
```

iii) *Find the inverse of* $\begin{pmatrix} 2 & 0 & 0 \\ 0 & 2 & 1 \\ 0 & 1 & 2 \end{pmatrix}$. $|S|=6$; $S^{-1}=\frac{1}{6}\begin{pmatrix} 3 & 0 & 0 \\ 0 & 4 & -2 \\ 0 & -2 & 4 \end{pmatrix}$

```
> solve(matrix(c(2,0,0,0,2,1,0,1,2),nrow=3,ncol=3))
     [,1]   [,2]   [,3]
[1,]  0.5  0.000  0.000
[2,]  0.0  0.667 -0.333
[3,]  0.0 -0.333  0.667
```

2) *(optional — but at least note the results, these are counterexamples to false assumptions that are all to easy to make since they contradict 'ordinary' algebra).*

Let $A = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}$, $B = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}$, $C = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$, $D = \begin{pmatrix} 1 & -1 \\ -1 & -1 \end{pmatrix}$,

$E = \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}$ and $F = \begin{pmatrix} 1 & 1 \\ -1 & -1 \end{pmatrix}$ *then show:–*

```
> A<-matrix(c(0,1,-1,0),2,2,byrow=T)
> B<-matrix(c(0,1,0,0),2,2,byrow=T)
> C<-matrix(c(1,1,1,-1),2,2,byrow=T)
> D<-matrix(c(1,-1,-1,-1),2,2,byrow=T)
> E<-matrix(c(1,1,1,1),2,2)
> F<-matrix(c(1,1,-1,-1),2,2,byrow=T)
```

i) $A^2 = -I_2$ (so A is 'like' the square root of –1)

```
> A%*%A
     [,1] [,2]
[1,]   -1    0
[2,]    0   -1
```

ii) $B^2 = 0$ (but $B \neq 0$)

```
> B%*%B
     [,1] [,2]
[1,]    0    0
[2,]    0    0
```

iii) $CD = -DC$ (but $CD \neq 0$)

```
> C%*%D
     [,1] [,2]
[1,]    0   -2
[2,]    2    0
> D%*%C
     [,1] [,2]
[1,]    0    2
[2,]   -2    0
```

iv) $EF = 0$ (but $E \neq 0$ and $F \neq 0$)

```
> E%*%F
     [,1] [,2]
[1,]    0    0
[2,]    0    0
```

3) *(see 0.10.1) The data file* `openclosed.Rdata*` *consists of examination marks in five subjects labelled* `mec, vec, alg, ana` *and* `sta.` *Download the datafile to your own hard disk. Using Windows Explorer double click on the file. This will open* ***R****, change the working directory to that where you have stored the data and read in the data to dataframe* `scor.` **Mardia, Kent & Bibby (1981).*

i) Then issue the following commands and read the results

```
> ls()   # see what objects are in the works space;
[1] "scor"
>              #   there should be only the dataframe scor
>
> X<-as.matrix(t(scor)) # define X to be the matrix
>                       # of the transpose of scor
>
> S<-var(t(X)) # calculate  the variance matrix of X'=scor
>
> A<-eigen(S)$vectors # Calculate the eigenvectors of S
> #                   & store them in A
> V<-eigen(S)$values # and eigenvalues in V
> A  # look at A
           [,1]        [,2]       [,3]         [,4]        [,5]
[1,] -0.5054457  0.74874751 -0.2997888  0.296184264 -0.07939388
[2,] -0.3683486  0.20740314  0.4155900 -0.782888173 -0.18887639
[3,] -0.3456612 -0.07590813  0.1453182 -0.003236339  0.92392015
[4,] -0.4511226 -0.30088849  0.5966265  0.518139724 -0.28552169
[5,] -0.5346501 -0.54778205 -0.6002758 -0.175732020 -0.15123239
> V  # look at V
[1] 686.98981 202.11107 103.74731  84.63044  32.15329
> sum(diag(S))# look at trace(S)
[1] 1109.632
> sum(V)     # look at sum of eigenvalues in V (they should
be the same)
[1] 1109.632
>
> options(digits=4) # only print four decimal places
>
> A%*%t(A)   # check that A is an orthogonal matrix
           [,1]       [,2]       [,3]       [,4]       [,5]
[1,]  1.000e+00  1.476e-16 -2.964e-17  4.014e-17 -1.586e-17
[2,]  1.476e-16  1.000e+00 -1.441e-16 -2.639e-16  3.010e-16
[3,] -2.964e-17 -1.441e-16  1.000e+00 -1.121e-16 -3.787e-16
[4,]  4.014e-17 -2.639e-16 -1.121e-16  1.000e+00 -3.263e-16
[5,] -1.586e-17  3.010e-16 -3.787e-16 -3.263e-16  1.000e+00
```

```
> t(A)%*%A   # (as it should be, property of eigenvectors)
          [,1]       [,2]       [,3]       [,4]       [,5]
[1,]  1.000e+00 -6.101e-17  1.099e-16 -2.397e-16  1.118e-16
[2,] -6.101e-17  1.000e+00 -1.115e-16  1.241e-16  1.837e-16
[3,]  1.099e-16 -1.115e-16  1.000e+00  8.888e-16  1.701e-16
[4,] -2.397e-16  1.241e-16  8.888e-16  1.000e+00 -1.225e-16
[5,]  1.118e-16  1.837e-16  1.701e-16 -1.225e-16  1.000e+00
>
> round(A%*%t(A)) # easier to see if round to whole numbers
     [,1] [,2] [,3] [,4] [,5]
[1,]    1    0    0    0    0
[2,]    0    1    0    0    0
[3,]    0    0    1    0    0
[4,]    0    0    0    1    0
[5,]    0    0    0    0    1
> round(t(A)%*%A)
     [,1] [,2] [,3] [,4] [,5]
[1,]    1    0    0    0    0
[2,]    0    1    0    0    0
[3,]    0    0    1    0    0
[4,]    0    0    0    1    0
[5,]    0    0    0    0    1
>
> t(A)%*%S%*%A    # calculate A'SA
          [,1]       [,2]       [,3]       [,4]       [,5]
[1,]  6.870e+02  2.381e-13 -1.029e-13  6.612e-14  4.718e-14
[2,]  2.595e-13  2.021e+02  3.081e-15 -3.109e-15 -2.730e-15
[3,] -1.219e-13 -1.259e-14  1.037e+02  5.388e-14 -8.734e-15
[4,]  7.972e-14  1.552e-14  4.434e-14  8.463e+01  3.257e-14
[5,]  3.606e-14  5.202e-15 -3.147e-15  3.728e-14  3.215e+01
>
> Y<-t(A)%*%X # let Y=A'X so that Y'=X'A, the data rotated
>             # onto the principal components.
>  var(t(Y))         #  the  variance  of  the  data  on  the
principal components
          [,1]      [,2]       [,3]      [,4]      [,5]
[1,]  6.870e+02 2.678e-13 -1.291e-13 9.386e-14 2.932e-14
[2,]  2.678e-13 2.021e+02  4.731e-15 1.460e-14 1.758e-15
[3,] -1.291e-13 4.731e-15  1.037e+02 3.553e-14 5.889e-15
[4,]  9.386e-14 1.460e-14  3.553e-14 8.463e+01 3.651e-14
[5,]  2.932e-14 1.758e-15  5.889e-15 3.651e-14 3.215e+01
>        # note these are the same up to rounding errors
>  round(t(A)%*%S%*%A)  #  easier  to  see  if  round  to  whole
numbers
     [,1] [,2] [,3] [,4] [,5]
[1,]  687    0    0    0    0
[2,]    0  202    0    0    0
[3,]    0    0  104    0    0
[4,]    0    0    0   85    0
[5,]    0    0    0    0   32
> round(var(t(Y)))
     [,1] [,2] [,3] [,4] [,5]
[1,]  687    0    0    0    0
[2,]    0  202    0    0    0
```

```
[3,]    0    0  104    0    0
[4,]    0    0    0   85    0
[5,]    0    0    0    0   32
> V           # eigenvalues of S, also same.
[1] 686.99 202.11 103.75  84.63  32.15
> sum(diag(S)) # find trace(S)
[1] 1110
> sum(V)        # same as above
[1] 1110
>
```

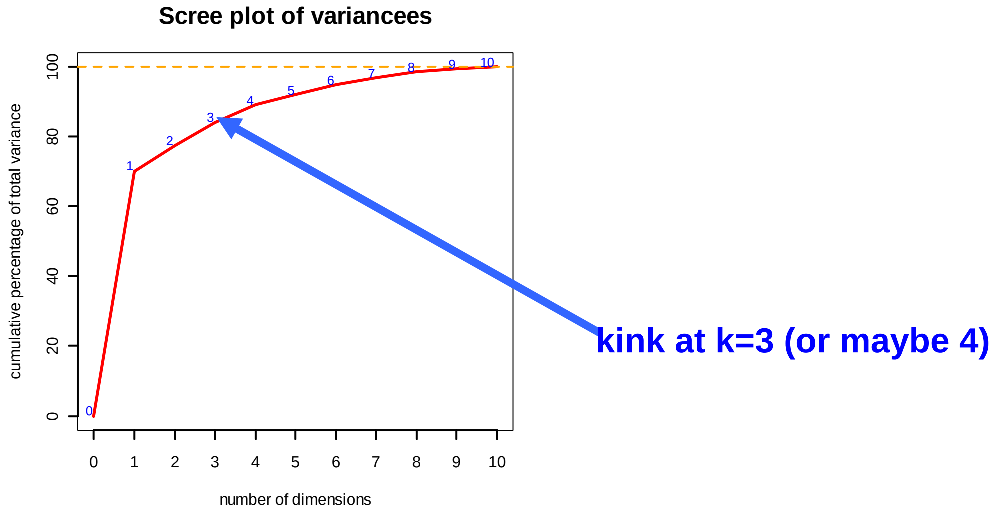
4) *The data file `bodysize.Rdata`⋆ consists of measurements of the circumferences (in centimetres) of `neck, chest, abdomen, hip, thigh, knee, ankle, biceps, forearm` and `wrist` of 252 men. Download the datafile to your own hard disk. Using Windows Explorer double click on the file. This will open **R**, change the working directory to that where you have stored the data and read in the data to dataframe `bodysize`. Next, download the function `screeplot()` contained in scriptfile `scree.R` to the same directory on you hard disk. Using the menu in **R** open the script file `scree.R` (top left icon in the menu bar), highlight all the lines in the function and click the middle icon to run the selected lines. This will load the function into your current **R** session. ⋆source: Journal of Statistics Education Data Archive*

i) Then issue the following commands and read the results

```
bodysize[1:5,]               # gives first few lines of the data file
diag(var(bodysize))          # gives variances of variables
sqrt(diag(var(bodysize))) # gives standard deviations
# note standard deviations vary by a factor of > 10
# so perform PCA with correlation matrix
body.pc<-princomp(bodysize,cor=T)
body.pc
summary(body.pc)
body.pc$loadings
screeplot(bodysize,T)
print(body.pc$loadings, cutoff=0.01)
```

```
> # function to draw screeplots of cumulative
> # eigenvalues in principal component analysis
>
> screeplot<-function(mydata,cor=F,maxcomp=10) {
+ my.pc<-princomp(mydata, cor=cor)
+ k<-min(dim(mydata),maxcomp)
+ x<-c(0:k)
+ y<-my.pc$sdev[1:k]*my.pc$sdev[1:k]
+ y<-c(0,y)
+ z<-100*cumsum(y)/sum(my.pc$sdev*my.pc$sdev)
+
+ plot(x,z,type="l",xlab="number of dimensions",
+  cex.main=1.5, lwd=3, col="red",
+  ylim=c(0,100),
+  ylab="cumulative percentage of total variance",
+   main="Scree plot of varianceees",
+  xaxt="n", yaxt="n")
+
+ axis(1,at=x,lwd=2)
+ axis(2,at=c(0,20,40,60,80,100),lwd=2)
+ abline(a=100,b=0,lwd=2,lty="dashed",col="orange")
+ text(x,z,labels=x,cex=0.8,adj=c(1.2,-.1),col="blue")
+ }
>
> bodysize[1:5,] # gives first few lines of the data file
  neck chest abdomen   hip thigh knee ankle biceps forearm wrist
1 36.2  93.1    85.2  94.5  59.0 37.3  21.9   32.0    27.4  17.1
2 38.5  93.6    83.0  98.7  58.7 37.3  23.4   30.5    28.9  18.2
3 34.0  95.8    87.9  99.2  59.6 38.9  24.0   28.8    25.2  16.6
4 37.4 101.8    86.4 101.2  60.1 37.3  22.8   32.4    29.4  18.2
5 34.4  97.3   100.0 101.9  63.2 42.2  24.0   32.2    27.7  17.7
>  diag(var(bodysize)) # gives variances of variables
   neck   chest abdomen     hip   thigh    knee   ankle  biceps forearm
wrist
  5.909  71.073 116.275  51.324  27.562   5.817   2.873   9.128   4.083
0.872
>  sqrt(diag(var(bodysize))) # gives standard deviations
   neck   chest abdomen     hip   thigh    knee   ankle  biceps forearm
wrist
  2.431   8.430  10.783   7.164   5.250   2.412   1.695   3.021   2.021
0.934
>  # note standard deviations vary by a factor of > 10
>  # so perform PCA with correlation matrix
>  body.pc<-princomp(bodysize,cor=T)
>  body.pc
Call:
princomp(x = bodysize, cor = T)

Standard deviations:
 Comp.1  Comp.2  Comp.3  Comp.4  Comp.5  Comp.6  Comp.7  Comp.8  Comp.9
Comp.10
  2.650   0.853   0.819   0.701   0.547   0.528   0.452   0.405   0.278
0.253

 10  variables and  252 observations.
>  summary(body.pc)
Importance of components:
                       Comp.1 Comp.2 Comp.3 Comp.4 Comp.5 Comp.6 Comp.7 Comp.8
Standard deviation      2.650 0.8530 0.8191 0.7011 0.5471 0.5283 0.4520 0.4054
Proportion of Variance  0.702 0.0728 0.0671 0.0492 0.0299 0.0279 0.0204 0.0164
Cumulative Proportion   0.702 0.7749 0.8420 0.8912 0.9211 0.9490 0.9694 0.9859
                        Comp.9 Comp.10
Standard deviation     0.27827  0.2530
Proportion of Variance 0.00774  0.0064
Cumulative Proportion  0.99360  1.0000
>  body.pc$loadings

Loadings:
        Comp.1 Comp.2 Comp.3 Comp.4 Comp.5 Comp.6 Comp.7 Comp.8 Comp.9 Comp.10
neck    -0.327        -0.259  0.339         0.288  0.719  0.318              
chest   -0.339  0.273         0.243 -0.447        -0.235  0.127 -0.543 -0.419
abdomen -0.334  0.398         0.216 -0.310 -0.147 -0.134         0.303  0.669
hip     -0.348  0.255  0.210 -0.119                      -0.349  0.551 -0.563
thigh   -0.333  0.191  0.180 -0.411  0.255  0.105  0.289 -0.404 -0.524  0.234
knee    -0.329         0.273 -0.135  0.446 -0.442 -0.118  0.624              
ankle   -0.247 -0.625  0.583        -0.416  0.168                            
biceps  -0.322        -0.256 -0.304         0.671 -0.471  0.197  0.130       
forearm -0.270 -0.363 -0.590 -0.404 -0.262 -0.440                            
```

```
wrist   -0.299 -0.377 -0.141  0.568  0.429        -0.271 -0.396

               Comp.1 Comp.2 Comp.3 Comp.4 Comp.5 Comp.6 Comp.7 Comp.8 Comp.9
SS loadings       1.0    1.0    1.0    1.0    1.0    1.0    1.0    1.0    1.0
Proportion Var    0.1    0.1    0.1    0.1    0.1    0.1    0.1    0.1    0.1
Cumulative Var    0.1    0.2    0.3    0.4    0.5    0.6    0.7    0.8    0.9
               Comp.10
SS loadings        1.0
Proportion Var     0.1
Cumulative Var     1.0
> screeplot(bodysize,T)
```



kink at k=3 (or maybe 4)

```
> print(body.pc$loadings, cutoff=0.01)

Loadings:
        Comp.1 Comp.2 Comp.3 Comp.4 Comp.5 Comp.6 Comp.7 Comp.8 Comp.9 Comp.10
neck    -0.327        -0.259  0.339  0.054  0.288  0.719  0.318  0.079 -0.023
chest   -0.339  0.273 -0.059  0.243 -0.447 -0.081 -0.235  0.127 -0.543 -0.419
abdomen -0.334  0.398  0.066  0.216 -0.310 -0.147 -0.134 -0.061  0.303  0.669
hip     -0.348  0.255  0.210 -0.119  0.059 -0.070  0.071 -0.349  0.551 -0.563
thigh   -0.333  0.191  0.180 -0.411  0.255  0.105  0.289 -0.404 -0.524  0.234
knee    -0.329 -0.022  0.273 -0.135  0.446 -0.442 -0.118  0.624 -0.011  0.013
ankle   -0.247 -0.625  0.583 -0.022 -0.416  0.168  0.066  0.016  0.022  0.047
biceps  -0.322 -0.022 -0.256 -0.304  0.094  0.671 -0.471  0.197  0.130  0.031
forearm -0.270 -0.363 -0.590 -0.404 -0.262 -0.440  0.087 -0.092  0.068  0.029
wrist   -0.299 -0.377 -0.141  0.568  0.429 -0.073 -0.271 -0.396 -0.076  0.033
```

```
               Comp.1 Comp.2 Comp.3 Comp.4 Comp.5 Comp.6 Comp.7 Comp.8 Comp.9
SS loadings       1.0    1.0    1.0    1.0    1.0    1.0    1.0    1.0    1.0
Proportion Var    0.1    0.1    0.1    0.1    0.1    0.1    0.1    0.1    0.1
Cumulative Var    0.1    0.2    0.3    0.4    0.5    0.6    0.7    0.8    0.9
               Comp.10
SS loadings        1.0
Proportion Var     0.1
Cumulative Var     1.0
>
```

ii) *How many principal components would you suggest adequately contain the main sources of variation within the data.*

Looking at the scree plot, 3 or maybe 4 components (accounting for 84% or 89% of total variation). Ignore obvious kink at k=1

iii) *What features of the body sizes do the first three [four?] components reflect?*

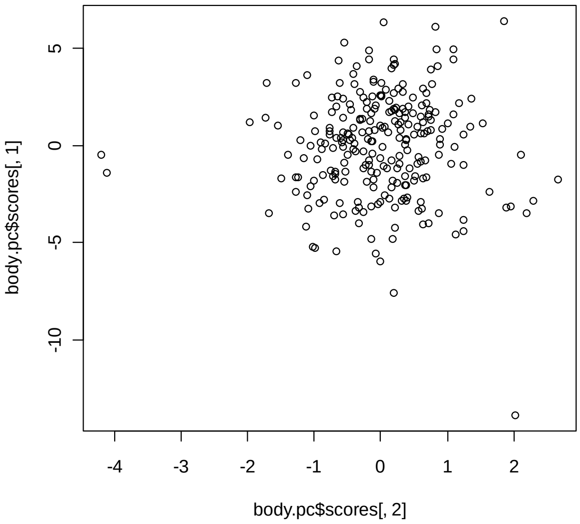
It is maybe clearer to see what is going on if we suppress as many decimal places as possible and use a fairly high cutoff value (it isn't possible to round to zero digits so try with just one):

```
> print(body.pc$loadings, cutoff=0.1,digits=1)
```

```
Loadings:
        Comp.1 Comp.2 Comp.3 Comp.4 Comp.5 Comp.6 Comp.7 Comp.8 Comp.9 Comp.10
neck    -0.3          -0.3    0.3           0.3    0.7    0.3
chest   -0.3    0.3           0.2   -0.4          -0.2    0.1   -0.5   -0.4
abdomen -0.3    0.4           0.2   -0.3   -0.1   -0.1           0.3    0.7
hip     -0.3    0.3    0.2   -0.1                        -0.3    0.6   -0.6
thigh   -0.3    0.2    0.2   -0.4    0.3    0.1    0.3   -0.4   -0.5    0.2
knee    -0.3           0.3   -0.1    0.4   -0.4   -0.1    0.6
ankle   -0.2   -0.6    0.6          -0.4    0.2
biceps  -0.3          -0.3   -0.3           0.7   -0.5    0.2    0.1
forearm -0.3   -0.4   -0.6   -0.4   -0.3   -0.4
wrist   -0.3   -0.4   -0.1    0.6    0.4          -0.3   -0.4
```

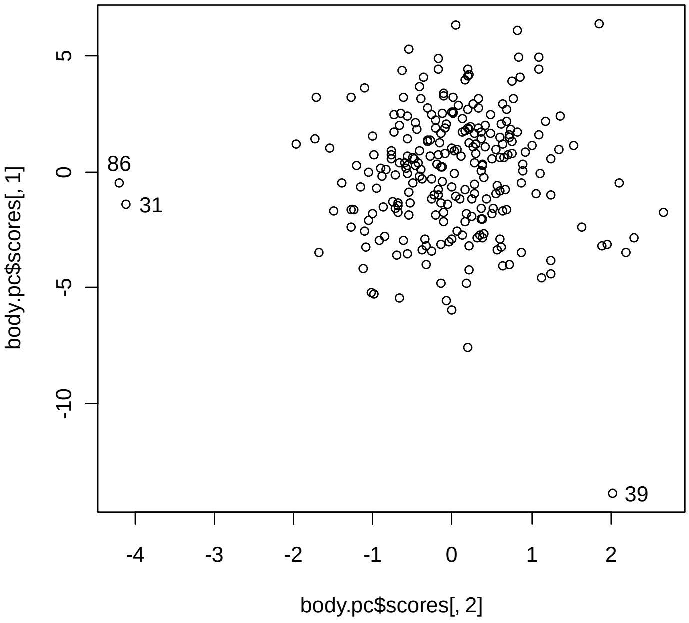
Now it is easy to see that the first PC reflects variations in overall size of body, the second contrasts arm size (mostly) with body and leg size, the third contrasts leg with rest of the body and the fourth is body versus limbs. If we plot PC1 against PC2 (i.e. the scores of on first principal component against those on the second) with

```
> plot(body.pc$scores[,2],body.pc$scores[,1])
```



we can see one outlier at the bottom of the plot and two to the left. Noting the signs of the loadings on the first PC (vertical axis) we can see that the outlier att he bottom arises from a subject with large measurements (i.e. a large person). The two outliers to the left are from people of average size but with proportionately well-developed arms by comparison with their legs. Using `identify()` gives

```
> identify(body.pc$scores[,2],body.pc$scores[,1])
[1] 31 39 86
```



which reveals that these outliers are observations 39 (lower), 31 & 86 (rightmost).

If it is preferred to plot with the large people at the top of the plot then do

```
> plot(body.pc$scores[,2],-body.pc$scores[,1])
```

5) *Calculate the principal components of the four measurements on Irises:*
   - *i) using the 'ready made' facility for principal component analysis*
   - *ii) by first calculating the covariance matrix and then looking at the eigenanalysis of the matrix.*

```
> attach(irisnf)
> options(digits=2)
> iris.pc<-princomp(cbind(Sepal.l,Sepal.w,Petal.l,Petal.w))
> iris.pc
Call:
princomp(x = cbind(Sepal.l, Sepal.w, Petal.l, Petal.w))

Standard deviations:
Comp.1 Comp.2 Comp.3 Comp.4
  2.05   0.49   0.28   0.15

 4  variables and  150 observations.
> summary(iris.pc)
Importance of components:
                       Comp.1 Comp.2 Comp.3 Comp.4
Standard deviation       2.05  0.494  0.280 0.1542
Proportion of Variance   0.92  0.054  0.017 0.0052
Cumulative Proportion    0.92  0.978  0.995 1.0000
> iris.pc$loadings

Loadings:
        Comp.1 Comp.2 Comp.3 Comp.4
Sepal.l  0.361 -0.650  0.590  0.316
Sepal.w        -0.737 -0.592 -0.314
Petal.l  0.857  0.171        -0.480
Petal.w  0.358        -0.543  0.756

               Comp.1 Comp.2 Comp.3 Comp.4
SS loadings      1.00   1.00   1.00   1.00
Proportion Var   0.25   0.25   0.25   0.25
Cumulative Var   0.25   0.50   0.75   1.00
> screeplot(cbind(Sepal.l,Sepal.w,Petal.l,Petal.w))
> iris.cov<-var(cbind(Sepal.l,Sepal.w,Petal.l,Petal.w))
> iris.cov
        Sepal.l Sepal.w Petal.l Petal.w
Sepal.l   0.686  -0.041    1.27    0.52
Sepal.w  -0.041   0.194   -0.33   -0.12
Petal.l   1.274  -0.329    3.12    1.30
Petal.w   0.516  -0.121    1.30    0.58
> eigen(iris.cov)
$values
[1] 4.228 0.246 0.079 0.024

$vectors
       [,1]   [,2]  [,3]  [,4]
[1,]  0.361 -0.650 -0.59  0.32
[2,] -0.084 -0.737  0.59 -0.31
[3,]  0.857  0.171  0.08 -0.48
[4,]  0.358  0.073  0.54  0.76
```

Note that instead of `cbind(Sepal.l,Sepal.w,Petal.l,Petal.w)` we could use `irisnf[,-5]` which is the data set without column 5 which contains variety.

```
> cov(irisnf[,-5])
        Sepal.l Sepal.w Petal.l Petal.w
Sepal.l   0.686  -0.041    1.27    0.52
Sepal.w  -0.041   0.194   -0.33   -0.12
Petal.l   1.274  -0.329    3.12    1.30
Petal.w   0.516  -0.121    1.30    0.58
>
```

Note that one of the covariances is negative and thus the first PC does not have loadings all of the same sign, though the negative covariance is very small by comparison with the others and so the corresponding coefficient is negligible and thus we can regard the first PC as reflecting variations in overall size.

Notes & Solutions for Tasks 3

Note and MEMORIZE the interesting identity

$|I_p + AB| = |I_n + BA|$ where A is p×n and B is n×p.

A key application of this result, which is used extensively later in this course, is when n=1. To evaluate $|I_p+xx'|$ where x is p×1 we have that this = $|I_1+x'x|$ which is the determinant of a 1×1 matrix (i.e. a scalar) and so $=1+x'x=1+\Sigma x_i^2$.

A variant on the result is the following (where c and d are scalars):
$|cI_p+dAB|=c^p|I_p+dAB/c|=c^p|I_n+dBA/c|=c^{p-n}|cI_n+dBA|$
(noting that if Z is a p×p matrix and c a scalar then $|cZ|=c^p|Z|$)
In particular, $|cI_p+dxx'|=c^{(p-1)}(c+d\Sigma x_i^2)$ and **especially**, if $x=1_p$ then $|cI_p+d1_p1_p'|=c^{(p-1)}(c+pd)$ since $1_p'1_p=p$.

5) Suppose the variance matrix takes the equicorrelation form $\underset{p\times p}{S}=\sigma^2\begin{pmatrix}1 & \rho & \rho & \cdots & \rho\\ \rho & 1 & \rho & \cdots & \rho\\ \vdots & \vdots & \ddots & \cdots & \vdots\\ \rho & & & \ddots & \rho\\ \rho & \rho & \cdots & \rho & 1\end{pmatrix}$*. By writing S is the form S=aI$_p$+b1$_p$1$_p$' for appropriate a and b and using result above, shew that if ρ>0 then the first principal component accounts for a proportion (1+ρ(p–1))/p of the total variation in the data. What can be said about the other p–1 components? What can be said if ρ<0? (but note that necessarily ρ > some constant bigger than –1 which you should determine, noting that S is a correlation matrix)*

We can see that $S=\sigma^2[(1-\rho)I_p+\rho J_p]$ where J_p is the p×p matrix with all entries =1 and easy to see that $J_p=1_p1_p'$ where 1_p is the unit p-vector with all entries=1. Then, to obtain the eigenvalues we need $|S-\lambda I_p|$ and obtain the roots of this p-degree polynomial in λ. We could use row and column manipulation of the determinant but using the result above we have

$|S-\lambda I_p|=|\sigma^2[(1-\rho)I_p+\rho 1_p1_p']-\lambda I_p|$
$=|[(1-\rho)\sigma^2-\lambda]I_p+\rho\sigma^2 1_p1_p'|=[(1-\rho)\sigma^2-\lambda]^{(p-1)}[(1-\rho)\sigma^2-\lambda+\rho\sigma^2 1_p'1_p]$
$=[(1-\rho)\sigma^2-\lambda]^{(p-1)}[(1-\rho)\sigma^2-\lambda+ p\rho\sigma^2]$ (noting that $1_p'1_p=p$)

Thus the eigenvalues of S are $(1+(p-1)\rho)\sigma^2$ and $(1-\rho)\sigma^2$ (the latter with multiplicity (p–1)). If $\rho>0$ then the first of these is the largest (i.e. $\lambda_1=(1+(p-1)\rho)\sigma^2$)).

If $\rho<0$ then we must have $\rho>-(p-1)^{-1}$ since we must have $|S|>0$:
if $\rho<-(p-1)^{-1}$ then one and only one eigenvalue is negative and since $|S|=\Pi\lambda_i$ this would give $|S|<0$.

When $\rho>0$, the first principal component is a_1 where $Sa_1-\lambda_1 a_1=0$,
i.e. where $Sa_{1=}(1+(p-1)\rho)\sigma^2\ a_1$ and $a_1'a_1=1$.

Easily seen that $a_1=p^{-½}1_p$ (i.e. proportional to the unit vector).
The other (p–1) p.c.s are solutions of $Sa-(1-\rho)a=0$ with $a'a=1$ (normalizing constraint) and $a'1_p=0$ (orthogonality with a_1) (i.e. $\Sigma a_j^2=1$ and $\Sigma a_j=0$) and there are **infinitely many possibilities.** One possible set is proportional to (1,–1,0,0,…,0)'; (1,1,–2,0,0,…0)'; (1,1,…,1,–(p–1))'.

Note: This example explains intuitively why the first principal component of a data set consisting of dimensional measurements on physical objects is often a measure of overall size: generally, if one of the objects is big then ***all*** of its dimensions will be big (presuming that the objects are more or less the same shape). This means that, generally, the measurements of all the dimensions will be *positively correlated* with each other. Consequently, the correlation (or covariance) matrix will be approximately like the equicorrelation matrix and so the first p.c. will be approximately proportional to the unit p-vector and so the score of any datum on the first p.c. will be proportional (approx) to the sum of its individual components). In fact the Perron-Frobenius theorem states that if all the elements of a (not necessarily symmetric) matrix are strictly positive then there is a unique positive eigenvalue corresponding to an eigenvector which can be chosen to have all positive elements and so could be interpreted as a weighted average of all measurements. The closer the correlations are in value the closer the coefficients of the first eigenvector are to a common multiple of the unit p-vector. If a small number of correlations are negative then it is often the case that the 2nd or 3rd (or…..) PC is size measure. Note also that there can be only one PC at most which is a weighted average of all variables since PCs are necessarily orthogonal.

6) *If the variance matrix takes the form $S=\alpha I_p+\beta zz'$ where z is a p-vector, shew that z is an eigenvector of S. Under what circumstances is Z proportional to the first principal component of the data?*

$S=\alpha I_p+\beta zz'$ so $Sz=(\alpha I_p+\beta zz')z=\alpha z+\beta zz'z=\alpha z+\beta\ z(z'z)$

then, noting z'z is a scalar and so commutes with z,

$=(\alpha+\beta z'z)z$ $=\lambda z$ where $\lambda=\alpha+\beta z'z$. So z is an eigenvector of S with eigenvalue $\alpha+\beta z'z$. Thus $z/(z'z)^{½}$ is the first p.c. if $\alpha+\beta z'z$ is the largest eigenvalue. The other p–1 eigenvalues are easily seen to be α and so for z to be the first we need $\beta>0$ (since $z'z=\Sigma z_i^2>0$).

7) *If the variance matrix takes the form (with $\alpha>0$)* $S=\begin{pmatrix}1+\alpha & 1 & \beta\\ 1 & 1+\alpha & \beta\\ \beta & \beta & \alpha+\beta^2\end{pmatrix}$ *find the first principal component and shew that it accounts for a proportion $(\beta^2+\alpha+2)/(\beta^2+3\alpha+2)$ of the total variation.*

$S=\alpha I_3+\gamma\gamma'$ where $\gamma=(1,1,\beta)'$ (notice that the diagonal of S contains $+\alpha$ in each entry so subtracting αI_3 leaves a matrix which is easier to make an intelligent guess at factorizing).

$|S-\lambda I_3|=|(\alpha-\lambda)I_3+\gamma\gamma'|=(\alpha-\lambda)^2(\alpha-\lambda+\gamma'\gamma)=(\alpha-\lambda)^2(\alpha-\lambda+2+\beta^2)$ and so the eigenvalues of S are $\beta^2+\alpha+2$ and α (twice). The largest must be the first (since $\beta^2+2>0$) and so accounts for a proportion $(\beta^2+\alpha+2)/(\beta^2+3\alpha+2)$ of the total variation.

8) *Referring to Q3 on Task Sheet 2, examination results in five mathematical papers, some of which were 'open-book' and others 'closed-book', what interpretations can you give to the principal components? .*

The principal components can be read from the eigenvectors calculated in Q3 or easily from

```
> options(digits=1)
> prcomp(scor)$rotation
     PC1   PC2  PC3    PC4   PC5
mec -0.5 -0.75  0.3 -0.296 -0.08
vec -0.4 -0.21 -0.4  0.783 -0.19
alg -0.3  0.08 -0.1  0.003  0.92
ana -0.5  0.30 -0.6 -0.518 -0.29
sta -0.5  0.55  0.6  0.176 -0.15
>
```

PC1 is a measure of overall ability across the five mathematical subjects, with low scores indicating high marks (not signs of PCs are arbitrary) . PC2 is a contrast between Pure&Statistics versus Applied Mathematics, with high scores indicating higher marks in Pure and Statistics than in Applied. PC3 is a contrast of the more applied subjects of Statistics and Mechanics versus the more theoretical Pure and Vectors, with high scores indicating preference for the applied. PC4 is primarily vectors versus analysis and PC5 is primarily ability at Algebra.

Notes & Solutions for Tasks 4

1) *uppose X' ($n\times p$) is a centred data matrix (i.e. each variable has sample mean zero). Then the variance matrix S is given by*

$$(n-1)S=XX'$$

Suppose λ_i and a_i are the eigenvalues and eigenvectors of XX'.

a) *What are the eigenvalues and eigenvectors of S?*

We have $XX'a_i = \lambda_i a_i$ so $Sa_i = [\lambda_i/(n-1)]a_i$ and so the eigenvalues and eigenvectors of S are $\lambda_i/(n-1)$ and a_i.

b) *Shew that the eigenvalues and eigenvectors of the $n\times n$ matrix $X'X$ are λ_i and $X'a_i$ respectively.*

We $XX'a_i = \lambda_i a_i$ so $X'XX'a_i = \lambda_i X'a_i$ i.e. $X'X(X'a_i) = \lambda_i(X'a_i)$ and result follows.

2) *Recently, measurements were made on a total of 26 mummy-pots (which contained mummified birds) excavated from the Sacred Animal Necropolis in Saqqara, Egypt and sent to the British Museum in the last century. The pots are approximately cylindrical, tapering slightly from the opening. The measurements made (in millimetres) were the overall length, the rim circumference and the base circumference. The rim of one pot was slightly damaged. Given below is a record of a S-Plus session analyzing the data.*

a) *What aspects of the pots do the two derived measurements stored in taper and point reflect?*

They both reflect the shape of the pots.

b) *Principal component analyses have been performed on the correlation matrix of all five variables but on the covariance matrix for just the three linear measurements. Why are these choices to be preferred for these data?*

The five variables include three on linear dimensions in centimetres and two dimensionless shape variables and so are on quite different scales. The three linear variables are all linear dimensions and so the covariance analysis is to be preferred.

c) *What features of the pots do the first two principal components in each analysis reflect?*

First analysis: Not that both taper and point are large if the rim-circumference is large and so the first pc reflects overall size. The second pc is clearly size vs shape, specifically large straight plots vs small pointed ones.

Analysis of British Museum Mummy-Pots

```
> attach(brmuseum)
> taper<-(rim.cir-base.circ)/length
> point<-rim.cir/base.circ
> potsize<-cbind(length,rim.cir,base.circ)
> potsize.pca<-princomp(potsize)
> summary(potsize.pca)
Importance of components:
                           Comp.1     Comp.2     Comp.3
    Standard deviation 59.8359424 22.6695236 18.8889569
Proportion of Variance  0.8043828  0.1154578  0.0801594
 Cumulative Proportion  0.8043828  0.9198406  1.0000000

> loadings(potsize.pca)
          Comp.1 Comp.2 Comp.3
   length  0.502 -0.694  0.516
  rim.cir  0.836  0.237 -0.494
base.circ  0.221  0.680  0.699
> potsizeshape<-cbind(length,rim.cir,base.circ,taper,point)
> potsizeshape.pca<-princomp(potsizeshape, cor=TRUE)
> summary(potsizeshape.pca)
Importance of components:
                          Comp.1    Comp.2    Comp.3
    Standard deviation 1.6082075 1.3352046 0.7908253
Proportion of Variance 0.5172663 0.3565543 0.1250809
 Cumulative Proportion 0.5172663 0.8738206 0.9989015

                             Comp.4      Comp.5
    Standard deviation 0.0698805556 0.024681162
Proportion of Variance 0.0009766584 0.000121832
 Cumulative Proportion 0.9998781680 1.000000000
> loadings(potsizeshape.pca)
          Comp.1 Comp.2 Comp.3 Comp.4 Comp.5
   length  0.428 -0.366 -0.678 -0.316  0.352
  rim.cir  0.548 -0.332  0.207 -0.129 -0.728
base.circ        -0.715  0.371  0.466  0.365
    taper  0.498  0.302  0.561 -0.367  0.461
    point  0.519  0.392 -0.212  0.729
```

Notes & Solutions for Tasks 5

1) Continuing Q1 of tasks 4, i.e. X' (n×p) is a centred data matrix:

i) *If D is the n×n distance matrix of the n p-dimensional observations and A is the matrix given by $a_{ij} = -½ d_{ij}^2$ and B=HAH, where H is the centring matrix, shew that B=kX′X for some suitable scalar k.*

$d_{ij}^2 = (x_i - x_j)'(x_i - x_j) = x_i'x_i - 2x_i'x_j + x_j'x_j$ so $b_{ij} = a_{ij} - \bar{a}_{i+} - \bar{a}_{+j} + \bar{a}_{++} = x_i'x_j$ noting that $\bar{x}_i = 0$ so B = X′X.

ii) Deduce that deriving a configuration of points from the matrix D by classical scaling is equivalent to referring the original data to principal components

If the data are referred to principal components then the coordinates of the rotated points are X′A where A=(a_i) are the eigenvectors of $S=(n-1)^{-1}XX'$. If we calculate the distance matrix directly from X′, then the principal coordinates from this distance matrix are given by the eigenvectors of X′X which are (from Q1 of week 4) $X'a_i$, i.e. X′A, thus showing that deriving a configuration of points from the matrix D by classical scaling is equivalent to referring the original data to principal components.

2) *If c_{ij} represents the similarity between cases i and j (c_{ij} is a similarity if $c_{ij}=c_{ji}$ and $c_{ij} \leq c_{ii}$) then the similarity matrix C can be converted to a distance matrix D by defining $d_{ij}=(c_{ii}-2c_{ij}+c_{jj})^{½}$. Define B = HAH where A=$(-½\,d_{ij}^2)$*

i) *Shew that B=HCH.*

If $d_{ij} = (c_{ii}-2c_{ij}+c_{jj})^{½}$ then $a_{ij} = -½\,d_{ij}^2 = -½(c_{ii}-2c_{ij}+c_{jj})$ so $b_{ij} = a_{ij} - \bar{a}_{i+} - \bar{a}_{+j} + \bar{a}_{++} = c_{ij} - \bar{c}_{i+} - \bar{c}_{+j} + \bar{c}_{++}$ and so B = HCH.

ii) *Deduce that you can proceed with classical scaling analysis analyzing C directly instead of converting it to a distance matrix and then calculating A.*

So, we deduce that you can proceed with classical scaling analysis using C directly in place of the matrix A instead of converting C to a distance matrix and then calculating A from it.

3) *The table below gives the road distances between 12 UK towns. The towns are Aberystwyth, Brighton, Carlisle, Dover, Exeter, Glasgow, Hull, Inverness, Leeds, London, Newcastle and Norwich.*

i) *Is it possible to construct an exact map for these distances?*

	A	B	C	D	E	G	H	I	Le	Lo	Ne	No
A	0											
B	244	0										
C	218	350	0									
D	284	77	369	0								
E	197	167	347	242	0							
G	312	444	94	463	441	0						
H	215	221	150	236	279	245	0					
I	469	583	251	598	598	169	380	0				
Le	166	242	116	257	269	210	55	349	0			
Lo	212	53	298	72	170	392	168	531	190	0		
Ne	253	325	57	340	359	143	117	264	91	273	0	
No	270	168	284	164	277	378	143	514	173	111	256	0

These data are contained in data set TOWNS. The Minitab version has the names of the towns in the first column and the data matrix in the next 12 columns. The final 12 columns contain the 12×12 matrix A=(–½ d_{ij}^2*). The R and S-plus versions give a dataframe with just the symmetric matrix of distances.*

The key to this is to calculate the eigenanalysis of the centred version of A (i.e. B=HAH). This gives the values 394473, 63634, 13544, 10246, –7063, 2465, 1450, –1141, 500, –214, –17 and since some of these are negative it means no it is not possible to construct an **exact** map. A transcript of an **R** session to do this is given below.

In **R** use the function `cmdscale()` (use the help system to find out how). A transcript is given below. Note that `towns.Rdata` is a dataframe and so the function `as.matrix()` is required to convert this to a matrix.

```
> options(digits=3)
> x<-cmdscale(as.matrix(towns),k=11,eig=TRUE)
Warning messages:
1: In cmdscale(as.matrix(towns), k = 11, eig = TRUE) :
  some of the first 11 eigenvalues are < 0
2: In sqrt(ev) : NaNs produced
> x$eig
 [1]  3.94e+05  6.36e+04  1.35e+04  1.02e+04  2.46e+03  1.45e+03
 [7] 5.01e+02  -9.09e-13 -1.69e+01 -2.14e+02 -1.14e+03
>
```

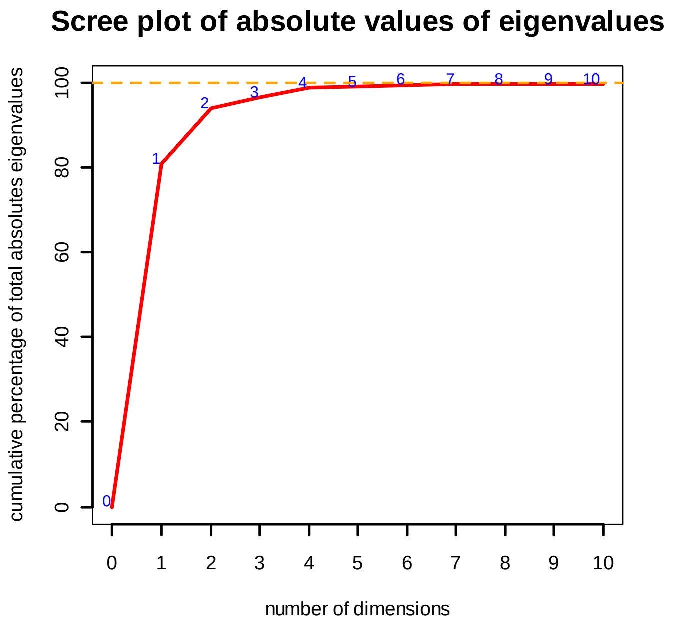
Note that it warns you of negative eigenvalues.

Notes & Solutions for Tasks 6

1) *Continuing Q3 of the tasks for week 5 (road distances between 12 UK towns) Determine a configuration of points that will adequately represent the data.*

First construct a scree plot (or else eyeball the [positive] eigenvalues and observe the first two dominate, so 2 dimensions is enough).

```
> library(MASS)
> options(digits=3)
> x<-cmdscale(as.matrix(towns),k=11,eig=TRUE)
Warning messages:
1: In cmdscale(as.matrix(towns), k = 11, eig = TRUE) :
   some of the first 11 eigenvalues are < 0
2: In sqrt(ev) : NaNs produced
>
> x$eig
 [1]  3.94e+05  6.36e+04  1.35e+04  1.02e+04  2.46e+03  1.45e+03
 [7]  5.01e+02 -9.09e-13 -1.69e+01 -2.14e+02 -1.14e+03
>
> CMDscreeplot(towns,raw=FALSE,abs=TRUE,maxcomp=10)
Warning messages:
1: In cmdscale(as.matrix(mydata), k = n, eig = TRUE) :
   some of the first 11 eigenvalues are < 0
2: In sqrt(ev) : NaNs produced
```



i) *Construct a two-dimensional map representing the road distances between these towns.*

To do it in **R** you can use the function `cmdscale()` with

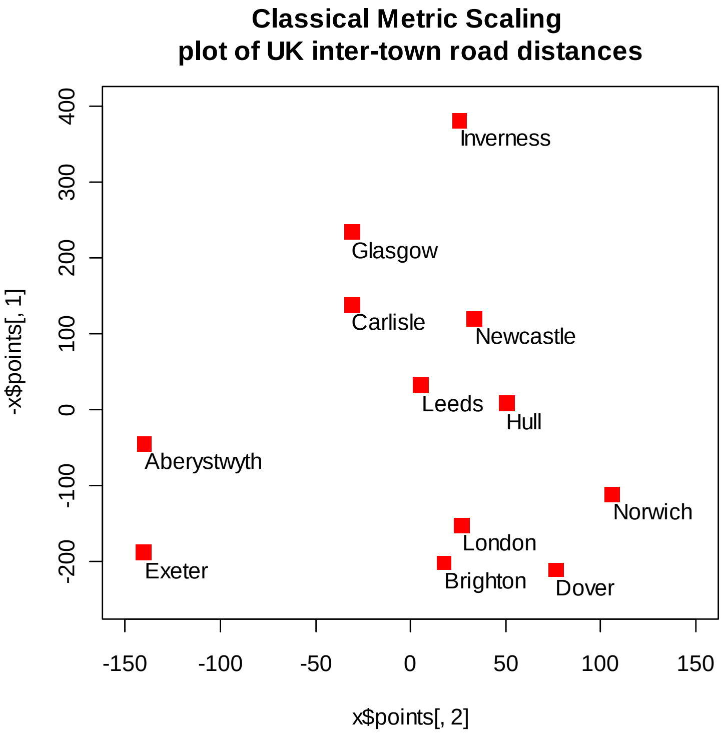
```
x<-cmdscale(as.matrix(towns))
```

and then plot the results by the coordinates of the points are in `x$` points and can be plotted. Note that `towns.Rdata` is already a distance matrix so you should not use the multidimensional scaling menu in the MASS library which presumes that you have a raw data matrix and calls `dist()` internally to create a new distance matrix. The command `as.matrix()` is required for `cmdscale` to recognise the distance matrix. It is also possible in **R** to try two varieties of non-metric scaling (`sammon()` and `isomds()`).

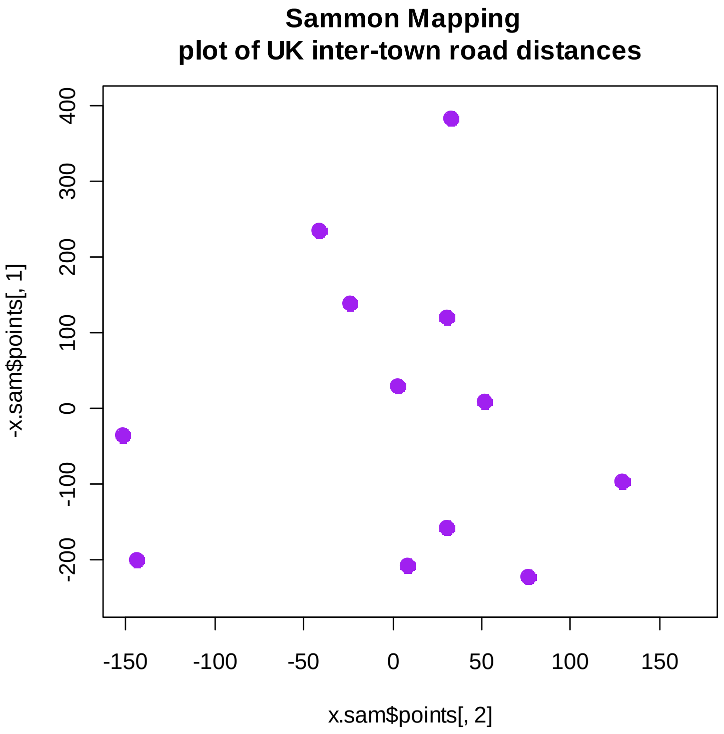
A record of an **R** session to perform these analyses is given below:

```
> plot(x$points[,2],-x$points[,1],pch=15,col="red",
+ xlim=c(-150,150),ylim=c(-250,400),cex=1.5,
+ main="Classical Metric Scaling
+ plot of UK inter-town road distances")
>
> text(x$points[,2],-
x$points[,1],row.names(towns),adj=c(0,1.3),
+ xlim=c(-150,150),ylim=c(-250,400))
>
```

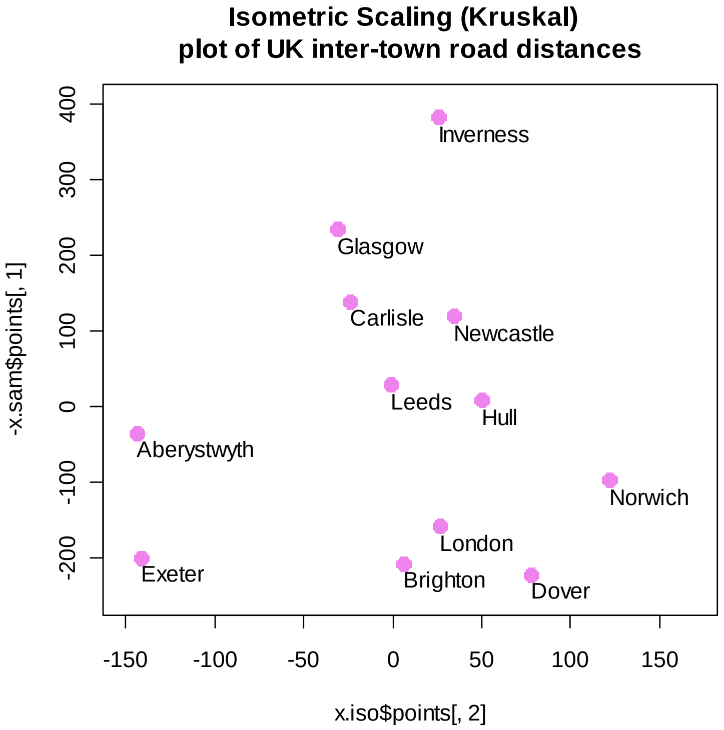
Note the reversal of sign of the vertical axis — an initial plot revealed that Inverness appeared on the lower edge of the plot so re-plotting with the sign changed produces a plot more aligned to the geography of the UK. Fortunately the east-west axis came out correctly but it would be possible to rotate or flip the plot in any way that is required.



```
>
plot(x.sam$points[,2],
+ -x.sam$points[,1],pch=19,col="purple",
+ xlim=c(-150,170),ylim=c(-250,400),cex=1.5,
+ main="Sammon Mapping
+ plot of UK inter-town road distances")
>
```



```
>  plot(x.iso$points[,2],-
x.sam$points[,1],pch=16,col="violet",
+ xlim=c(-150,170),ylim=c(-250,400),cex=1.5,
+ main="Isometric Scaling (Kruskal)
+ plot of UK inter-town road distances")
>
>
> text(x.iso$points[,2],-x.sam$points[,1],
+ labels=row.names(towns),adj=c(0,1.3),xlim=c(-
150,170),ylim=c(-250,400))
>
```



Notes & Solutions for Tasks 7

1) *Retrieve the data on beef and pork consumption referenced in §5.2 and verify the calculations given in §5.2 using **R** or S-PLUS. Predict the consumption of beef and pork if the prices in cents/lb are 79.3, 41.2 and the disposable income index is 40.4.*

   First, create a data set with the name `meat` containing the six columns needed (including a column named `constant` containing just 1s), then:

```
> attach(meat)
> y<- cbind(cbe,cpo)
> x<- cbind(constant, pbe,ppo,dinc)
> betahat<- solve(t(x)%*%x)%*%t(x)%*%y
> betahat
              cbe    cpo
constant 101.448 79.569
pbe       -0.753  0.153
ppo        0.254 -0.687
dinc      -0.241  0.283>
> sigmahat<-t(y-x%*%betahat)%*%(y-x%*%betahat)
> sigmahat<-sigmahat/(17-3-1)
> sigmahat
      cbe   cpo
cbe  4.41 -7.57
cpo -7.57 16.83>
 > x0<- c(1,79.3,41.2,40.4)
> ypred<-x0%*%betahat
> ypred
      cbe  cpo
[1,] 42.5 74.9
>>
```

   So predicted consumption of beef is 42.5 and of pork 74.9 pounds.

2) *Retrieve the dataset chap8headsize referenced in §6.3 and calculate the estimates of the least squares multivariate regression parameters β of length and breadth of heads of first sons upon those of second sons. Is it possible to deduce from these results the estimates for the regression of second sons upon the first?*

   (Note that the individual data files seem no longer to be available but you should be able to download the complete set of files from Brian Everitt’s webpage as a zipped archive.)

```
> "headsize" <-
+  matrix(c(191, 195, 181, 183, 176, 208, 189, 197, 188, 192, 179, 183,
174, 190, 188, 163, 195, 186, 181, 175, 192, 174,
+  176, 197, 190, 155, 149, 148, 153, 144, 157, 150, 159, 152, 150, 158,
147, 150, 159, 151, 137, 155, 153,
+  145, 140, 154, 143, 139, 167, 163, 179, 201, 185, 188, 171, 192, 190,
189, 197, 187, 186, 174, 185, 195,
+  187, 161, 183, 173, 182, 165, 185, 178, 176, 200, 187, 145, 152, 149,
149, 142, 152, 149, 152, 159, 151,
+  148, 147, 152, 157, 158, 130, 158, 148, 146, 137, 152, 147, 143, 158,
150)
+ , nrow = 25, ncol = 4 ,  dimnames = list(character(0)
+ , c("head1", "breadth1", "head2", "breadth2")))
> attach(data.frame(headsize))
```

The calculations for the regression of first son sizes on second son sizes are:-

```
> y<-cbind(head1,breadth1)
> x<-cbind(rep(1,25),head2,breadth2)
> betahat<- solve(t(x)%*%x)%*%t(x)%*%y
> betahat
          head1 breadth1
         34.282   35.802
head2     0.394    0.245
breadth2  0.529    0.471
```

and this would allow prediction of first son head sizes from second, though it would be more plausible to predict second from first and then the regression analysis would need to be done the other way around. It is not possible to deduce one from the other in the same way as in univariate regression regressing y on x does not give all the information needed for the regression of x on y. Note however that you can get the estimates of the regression coefficients but not the variance matrix) from univariate [multiple] regressions:-

```
> ll<-lm(head1~head2+breadth2)
> ll
Call:
lm(formula = head1 ~ head2 + breadth2)
Coefficients:
(Intercept)        head2     breadth2
     34.282        0.394        0.529
> bb<-lm(breadth1~head2+breadth2)
> bb
```

```
Call:
lm(formula = breadth1 ~ head2 + breadth2)
Coefficients:
(Intercept)        head2     breadth2
     35.802        0.245        0.471
```

3) *Read the section on Maximum Likelihood Estimation in Background Results. This material will be required and used extensively in Chapter 8.*

Trust you have done this by now.

Notes & Solutions for Tasks 8

1) *Read §8.1 – §8.4 paying particular attention to the results highlighted in boxes as well as §8.3.2 and §8.4.*

   Trust you have done this by now.

2) *n observations are available on $x \sim N_p(\mu,\Sigma)$ and C is a known $p\times q$ matrix (p>q). By finding the distribution of $y=C'x$, shew that a test of H_0: $C'\mu=0$ vs. H_A: $C'\mu\neq 0$ is given by Hotelling's T^2 with $T^2=n\bar{x}'C(C'SC)^{-1}C'\bar{x}$ ($\bar{x}$ and S are the sample mean and variance). What parameters does the T^2 distribution have?*

   If $x \sim N_p(\mu, \Sigma)$ then $y=C'x \sim N_q(\mu_y,\Sigma_y)$ where $\mu_y=C'\mu$ and $\Sigma_y=C'\Sigma C$, further $S_y=C'SC$ and $\bar{y}=C'\bar{x}$ and so the T^2 statistic for testing $\mu_y=0$ which is $n\bar{y}'S_y^{-1}\bar{y} = n\bar{x}'C(C'SC)^{-1}C'\bar{x}$ and the null distribution is $T^2(q,n-1)$.

3) *Note: parts (i) & (ii) below should give the same p-value.*

   i) *A sample of 6 observations on sugar content x_1 and stickiness x_2 of a novel toffee give sample statistics of*

   $$\bar{x}=\begin{pmatrix}81.17\\60.33\end{pmatrix} \text{ and } S=\begin{pmatrix}27.02 & 7.94\\ * & 4.26\end{pmatrix}.$$

   Test the hypothesis H_0: $2\mu_1=3\mu_2$

   [Suggestion: consider using the 2×1 matrix $C=(2, -3)'$]

   We have n=6 and H_0: $2\mu_1-3\mu_2=0$ i.e. $(2,-3)\begin{pmatrix}\mu_1\\\mu_2\end{pmatrix}=0$,

   i.e. $C=(2,-3)'$. $C'\bar{x}=-18.65$, $C'SC=51.14$,

   so $T^2=40.81$: $\frac{n-1}{1}\frac{1}{n-1}T^2 \equiv F_{1,5}$ and so we compare 40.81 with $F_{1,5}$ and then conclude that this is highly significant and so reject the null hypothesis.

   ii) *By noting that if $x = (x_1,x_2) \sim N_2(\mu,\Sigma)$ where $\mu = (\mu_1,\mu_2)'$ and Σ has element σ_{ij} then $2x_1-3x_2 \sim N((2\mu_1-3\mu_2),(4\sigma^2_{11}+9\sigma^2_{22}-12\sigma_{12}))$ test H_0 in i) above using a Student's t-test.*

$2\overline{x}_1 - 3\overline{x}_2 = -18.65$; $4s_{11}^2 + 9s_{22}^2 - 12s_{12} = 51.14$, n=6

so $t = -\frac{18.65}{\sqrt{51.14/6}} = 6.39$ and compare with t_5 giving same p-value as in part (i) noting that $6.39^2 = 40.8$ and $t_5{}^2 \equiv F_{1,5}$.

Notes & Solutions for Tasks 9

1) *Read the solutions to Exercises 2. These contain a detailed guide to the interpretation of principal components and of crimcoords by examining the loadings of the variables in the PCs and Crimcoords and so provide further practice at this important aspect.*

   Trust you have done this by now.

2) *Referring to the data set dogmandibles.∗* ***excluding the Prehistoric Thai dogs (group 5 on X_{11})*** *test the hypotheses that Male and Female dogs have*

   i) equally sized mandibles (i.e. variables X_1 & X_2 together)

   This calls for a Hotelling's T^2-test with (X_1,X_2). The easiest way of doing this is to use a MANOVA facility in. Values of T^2 can be obtained as (n–2)×Lawley-Hotelling statistic.

```
> options(digits=7)
>
> mf.manova<-manova(cbind(length, breadth) ~ gender)
>
> summary.manova(mf.manova,test="H")
          Df Hotelling-Lawley approx F num Df den Df  Pr(>F)
gender     1          0.08025  2.56806      2     64 0.08457
Residuals 65
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
>
```

The T^2 is then (32 + 35 –2)×0.08025 = 5.21625, converting this to an F-value gives 2.568 (as in table above) and p-value 0.085 (also as in table above) and so we conclude that there is only weak evidence of a difference in mean sizes of mandibles between male and female.

a) equally long mandibles (variable X_1)

b) equally broad mandibles (variable X_2)

These can be done using two separate univariate student t-tests:

```
> options(digits=3)
> t.test(length ~ gender)

        Welch Two Sample t-test

data:  length by gender
t = 1.74, df = 64.9, p-value = 0.08606
alternative hypothesis: true difference in means is not
equal to 0
95 percent confidence interval:
 -1.12 16.42
sample estimates:
mean in group 1 mean in group 2
            133             126

> t.test(breadth ~gender)

        Welch Two Sample t-test

data:  breadth by gender
t = 2.26, df = 64.8, p-value = 0.02699
alternative hypothesis: true difference in means is not
equal to 0
95 percent confidence interval:
 0.092 1.475
```

and we conclude that there is good evidence of a difference in mean breadths between males and females. Note that the apparent contradiction between the multivariate and univariate tests is in part because of the slight loss of power in the multivariate test caused by estimating more parameters and partly because these are different hypotheses.

ii) equal overall mandible characteristics (i.e. variables X_1–X_9)

We have

```
> xx.manova=manova(cbind(length, breadth, condyle.breadth,
height,
+ molar.length, molar.breadth, first.to.3rd.length,
+ first.to.4th.length, canine.breadth) ~ gender)
>
> summary(xx.manova, "H")
          Df Hotelling-Lawley approx F num Df den Df Pr(>F)
gender     1            0.137    0.871      9     57   0.56
Residuals 65
```

(so T^2 = 8.9375, p=0.556) and we conclude there is no evidence in a difference in mean characteristics as measured by these variables. Note that there is no need to calculate the p-value again from the T^2 statistics: it is necessarily the same as that given already.

3) Test the hypotheses that Iris Versicolor and Iris Virginica have

i) equally sized sepals

ii) equally sized petals

iii) equally sized sepals & petals.

This question calls for several two-sample Hotellings T^2tests; for (i) we need p=2 with elements sepal lengths & widths, for (ii) we need p=2 with elements petal lengths & widths, for (iii) we need p=4 with elements sepal lengths & widths, petal lengths & widths. The easiest way of doing this is to use a MANOVA facility. Values of T^2 can be obtained as (n–2)×Lawley-Hotelling statistic. Doing this in **R** has no new features and so it is not given here.

To do this 'by hand' (and it is strongly recommended that you do try it by hand for at least one case) you need to calculate the means and covariances of the four measurements separately for the varieties.

Notes & Solutions for Tasks 10

1) *Suppose we have samples of sizes n_1 and n_2 with means $\overline{x}_1$ and $\overline{x}_2$ and variances S_1 and S_2 from populations $N_p(\mu_1,\sigma_1^2)$ and $N_p(\mu_2,\sigma_2^2)$, let $S=[(n_1-1)S_1+(n_2-1)S_2]/(n-2)$ where $n=n_1+n_2$.*

   i) *Shew that the UIT of H_0: $\mu_1 = \mu_2$ vs H_A: $\mu_1 \neq \mu_2$ is given by Hotelling's* $T^2 = \frac{n_1n_2}{n}(\overline{x}_1-\overline{x}_2)'S^{-1}(\overline{x}_1-\overline{x}_2)$

   ii) *Deduce that the greatest difference between the two populations is exhibited in the direction* $S^{-1}(\overline{x}_1-\overline{x}_2)$.

   [Suggestion: adapt the argument of §5.6.4]

   When the data are projected into one dimension we require a two-sample t-test for the equality of two Normal means and we use the statistic $t_\beta^2 = \dfrac{n_1n_2\beta'(\overline{x}_1-\overline{x}_2)(\overline{x}_1-\overline{x}_2)'\beta}{n\beta'S\beta}$ and maximizing this wrt β means we require that β is the eigenvector of the [rank 1 p×p matrix] $n_1n_2S^{-1}(\overline{x}_1-\overline{x}_2)(\overline{x}_1-\overline{x}_2)'/n$ corresponding to the only non-zero eigenvalue. Easily seen by the usual procedure that this maximum value is $T^2 = \frac{n_1n_2}{n}(\overline{x}_1-\overline{x}_2)'S^{-1}(\overline{x}_1-\overline{x}_2)$ and that the eigenvector is proportional to $S^{-1}(\overline{x}_1-\overline{x}_2)$ which therefore exhibits the maximum deviation between the two populations.

2) *Referring to the data set dogmandibles.* ***excluding the Prehistoric Thai dogs (group 5 on X_{11})***

   i) *What combination of length and breadth of mandible exhibits the greatest difference between Males and Females?*

In **R**:

```
> library(MASS)
> attach(dogmandibles)
> moddogs=dogmandibles[1:67,]
> detach(dogmandibles)
> attach(moddogs)
> lda(gender~length+breadth)
Call:
lda(gender ~ length + breadth)

Prior probabilities of groups:
        1         2 
0.5223881 0.4776119 

Group means:
  length   breadth
1 133.40 10.274286
2 125.75  9.490625

Coefficients of linear discriminants:
                LD1
length   0.01938347
breadth -0.90204609
Warning message:
In lda.default(x, grouping, ...) : group 3 is empty
>
```

So the linear combination is `0.01938×length −0.90205×breadth`, or rescaling `breadth – 0.0214 × length` as before *ii) What combination of length and breadth of mandible exhibits the greatest difference between the four species?*

In **R**:

```
>  lda(species~length+breadth)
Call:
lda(species ~ length + breadth)

Prior probabilities of groups:
        1         2         3         4 
0.2388060 0.2985075 0.2537313 0.2089552 

Group means:
    length  breadth
1 125.3125  9.70625
2 111.0000  8.18000
3 133.2353 10.72353
4 157.3571 11.57857

Coefficients of linear discriminants:
               LD1        LD2
length  0.08941756 -0.1189762
breadth 0.60275578  1.5483264

Proportion of trace:
   LD1    LD2 
0.9333 0.0667 
Warning message:
In lda.default(x, grouping, ...) : group 5 is empty
```

So the linear combination is `0.08942×length + 0.6028×breadth`, or rescaling `breadth + 0.0000674 × length` (i.e. primarily breadth).

Notes & Solutions for Tasks 11

Notes & Solutions

[Note that these questions are more substantial than on previous task sheets. Question 1 is a past examination question. Question 3 is only of benefit to those wanting more practice on PCA interpretation and practical data analysis]

1) *An archaeologist wishes to distinguish pottery from two different sources on the basis of its chemical composition. Measurements by Neutron Activation Analysis of the concentrations in parts per million of trace elements Cr and V in 19 samples of pottery from Tell el-Amarna gave mean results of 2.3 and 6.7, respectively, with sample variances 0.62 and 1.41 and covariance 0.09. Similar measurements on 23 samples from Memphis gave mean results of 2.9 and 5.9 with sample variances 0.7 and 1.36 and sample covariance 0.08.*

   i) *Assuming that these measurements are adequately modelled by bivariate Normal distributions with a common variance, calculate the linear discriminant rule for distinguishing Amarna from Memphis pottery on the basis of the concentrations of Cr and V.*

   First, calculate pooled variance matrix as W = $[18S_{TA}+22S_M]/40$

   $$=\begin{pmatrix}0.664 & 0.085\\ * & 1.383\end{pmatrix} \text{ and so } W^{-1}=\begin{pmatrix}1.518 & -0.093\\ * & 0.729\end{pmatrix}$$

   If $x=(x_{Cr},x_V)'$ then the linear discriminant rule is to classify x as from Amarna if $((2.3,6.7) – (2.9,5.9))W^{-1}(x – ½((2.3,6.7) + (2.9,5.9))')>0$,

   i.e. if $(–0.6, 0.8)W^{-1}(x – (2.6,6.3)') > 0$,

   i.e. if $(–0.985,0.639)(x – (2.6,6.3)') > 0$

   i.e. if $– 0.985x_{Cr} + 0.639x_V – 1.465 > 0$

ii) Prove that the estimated probabilities of misclassifying Memphis pottery as Amarna and vice versa are the same using this rule.

Classification rule is to allocate to Amarna if

$h(x) = -0.985x_{Cr} + 0.639x_V - 1.465 > 0$

Now if x is from Memphis then

$E[h(x)] = -0.985\times 2.9 + 0.639\times 5.9 - 1.465 = -0.551$ and $\mathrm{var}(h(x)) = (-0.985)^2(0.664) + (0.639)^2(1.383) + 2(-0.985)(0.639)(0.085) = 1.102$,

so P[classify as Amarna |from Memphis]

$= P[h(x) > 0 \mid h(x)\sim N(-0.551, 1.102)] = 1 - \Phi(0.551/\sqrt{}1.102)$

$= 1- \Phi(0.525) = 1 - 0.700 = 0.300.$

If x is from Amarna then

$E[h(x)] = -0.985\times 2.3 + 0.639\times 6.7 - 1.465 = 0.551$ and $\mathrm{var}(h(x)) = (-0.985)^2(0.664) + (0.639)^2(1.261) + 2(-0.985)(0.639)(0.085) = 1.102.$

So P[classify as Memphis |from Amarna]

$= P[h(x) < 0 \mid h(x)\sim N(-0.551, 1.102)]$

$= \Phi(-0.551 /\sqrt{}1.102) = \Phi(-0.525) = 0.30$

Thus the two misclassification probabilities are equal.

iii) By how much is this misclassification probability an improvement over those using each of the elements separately?

If only Cr is used then rule is to classify as Amarna if $h_{Cr}(x_{Cr})=(2.3 - 2.9)(0.664^{-1}(x_{Cr} - 2.6) > 0$, i.e. if $x_{Cr} < 2.6$.
If x is from Memphis, then $x_{Cr}\sim N(2.9,0.664)$ and
so P[classify as Amarna | from Memphis] = $\Phi(-0.3/0.664^{½})$
= $\Phi(-0.368) = 0.356$
Similarly if only V is used then rule is to classify as Amarna if $x_V >$ 6.3 and so P[classify as Amarna | from Memphis]
= $1 - \Phi(0.4/1.383^{½}) = 1 - \Phi(0.340) = 1 - 0.633 = 0.367$.
Thus the improvement in misclassification probability over using just Cr is 5.6% and over using just V it is 6.7%

iv) What advice would you give to the archaeologist in the light of these results?

The archaeologist needs to be warned that the error rate will be at least 30% if (s)he uses either or both of the elements. This is substantial and may give cause for not proceeding with the study. Measuring more trace elements will certainly not worsen the situation.

2) *Referring to the data set dogmandibles.∗ (including the Prehistoric Thai dogs (group 5 on X_{11}))*

   i) *Using `lda()` in **R** look at the discrimination between the 5 species (using the nine measurements) and estimate the classifcation rate. [In **R** it is easy to find the cross-validation (or jackknife) estimate of classification rate].*

NB The computer analysis in the solutions given here and in Q3 have been produced using Minitab. The **R** analysis is considerably easier and has not been given but if there are any difficulties then this can be provided

```
MTB > Discriminant 'species' 'length'-'canine breadth';
SUBC>   Predict C51-C59.
```

Discriminant Analysis: species versus length, breadth, ...

```
Linear Method for Response:   species
Predictors:  length  breadth  condyle  height  molar le  molar br  first to
             first to  canine b

Group        1        2        3        4        5
Count       16       20       17       14       10

Summary of Classification

Put into     ....True Group....
Group            1        2        3        4        5
1               15        0        0        0        2
2                0       20        0        0        0
3                0        0       17        0        0
4                0        0        0       14        0
5                1        0        0        0        8
Total N         16       20       17       14       10
N Correct       15       20       17       14        8
Proportion   0.938    1.000    1.000    1.000    0.800

N =   77     N Correct =   74     Proportion Correct = 0.961
```

So estimated classification rate without using cross-validation is 96%: With cross-validation gives

```
MTB > Discriminant 'species' 'length'-'canine breadth';
SUBC>   XVal;
SUBC>   Predict C51-C59.
```

Discriminant Analysis: species versus length, breadth, ...

```
Linear Method for Response:   species
Predictors:  length  breadth  condyle  height  molar le  molar br  first to
             first to  canine b

Group        1        2        3        4        5
Count       16       20       17       14       10
```

```
Summary of Classification

Put into     ....True Group....
Group           1        2        3        4        5
1              15        0        0        0        2
2               0       20        0        0        0
3               0        0       17        0        0
4               0        0        0       14        0
5               1        0        0        0        8
Total N        16       20       17       14       10
N Correct      15       20       17       14        8
Proportion   0.938    1.000    1.000    1.000    0.800

N =   77     N Correct =   74     Proportion Correct = 0.961

Summary of Classification with Cross-validation

Put into     ....True Group....
Group           1        2        3        4        5
1              14        1        0        0        3
2               0       19        0        0        0
3               0        0       17        0        0
4               0        0        0       13        0
5               2        0        0        1        7
Total N        16       20       17       14       10
N Correct      14       19       17       13        7
Proportion   0.875    0.950    1.000    0.929    0.700

N =   77     N Correct =   70     Proportion Correct = 0.909
```

so an estimate of 91%.

ii) *Perform the discriminant analysis just on the first four [modern] species and then use this to classify the prehistoric Thai dogs.*

```
MTB > COPY C1-C11 C101-C111;
SUBC> USE C11 = 5.
MTB > copy c1-c11 c1-c11 ;
SUBC> omit c11 = 5.
MTB > Discriminant 'species' 'length'-'canine breadth';
SUBC>   Predict c101-c109.
```

Discriminant Analysis: species versus length, breadth, ...

```
Linear Method for Response:   species
Predictors:  length  breadth  condyle  height  molar le  molar br  first to
             first to  canine b

Group        1        2        3        4
Count       16       20       17       14

Summary of Classification

Put into     ....True Group....
Group           1        2        3        4
1              16        0        0        0
2               0       20        0        0
3               0        0       17        0
4               0        0        0       14
Total N        16       20       17       14
N Correct      16       20       17       14
Proportion   1.000    1.000    1.000    1.000

N =   67     N Correct =   67     Proportion Correct = 1.000

Prediction for Test Observations
 Observation  Pred Group  From Group Sqrd Distnc Probability
           1           1
```

```
                         1      14.722       0.987
                         2      23.353       0.013
                         3      71.906       0.000
                         4      63.387       0.000
 2           1
                         1      14.226       1.000
                         2      36.185       0.000
                         3      88.289       0.000
                         4      49.060       0.000
 3           1
                         1      17.875       1.000
                         2      52.703       0.000
                         3      93.366       0.000
                         4      37.821       0.000
 4           1
                         1       8.635       0.998
                         2      21.142       0.002
                         3      66.208       0.000
                         4      69.793       0.000
 5           1
                         1      39.810       1.000
                         2      83.256       0.000
                         3     113.618       0.000
                         4      75.934       0.000
 6           1
                         1      27.584       1.000
                         2      69.580       0.000
                         3      65.908       0.000
                         4      66.938       0.000
 7           1
                         1      39.170       1.000
                         2      59.289       0.000
                         3     126.981       0.000
                         4      74.862       0.000
 8           1
                         1       8.226       1.000
                         2      33.646       0.000
                         3      78.406       0.000
                         4      55.935       0.000
 9           1
                         1      16.727       1.000
                         2      42.650       0.000
                         3     108.792       0.000
                         4      73.268       0.000
10           1
                         1      12.329       1.000
                         2      40.900       0.000
                         3      75.376       0.000
                         4      57.443       0.000
```

Note the apparent 100% correct classification on just the modern species and that with ‘near certainty’ all the prehistoric are classified as ‘modern’, even though

iii) Compare the results of these analyses with the results of the more informal exploratory analyses with Crimcoords in Exercises 2.

The plots on crimcoords revealed a consistent slight difference from modern dogs.

3) The datafile CLAYPOTS has 272 observations on the trace element content of clay samples from pots found at various archaeological sites around the Aegean. Column 1 gives the group number (i.e. archaeological site for most of the pots) and columns 2—9 give the amounts of 9 trace elements (which have been labelled A to I) found in samples of clay from the pots. It is suggested that before investigating the specific questions below it is advisable to do some exploratory analysis with PCA etc. Groups 1, 3 and 4 are from known sources; groups 2 and 5 are from unknown sources but are believed to come from one or other of 1,3 or 4.

i) Construct a display on crimcoords of groups 1,3 and 4 and add in the points from groups 2 and 5.

ii) Which are the best classifications of these pots?

*In **R** you can use the function `lda()` and the generic function `predict()` — use the Help system to find out the details.*

```
Welcome to Minitab, press F1 for help.
MTB > RETR "C:\TEACHING\MVA\CLAYPOTS.MTW"
Retrieving worksheet from file: C:\TEACHING\MVA\CLAYPOTS.MTW
# Worksheet was saved on 22/11/01 11:10:19
```

Results for: CLAYPOTS.MTW

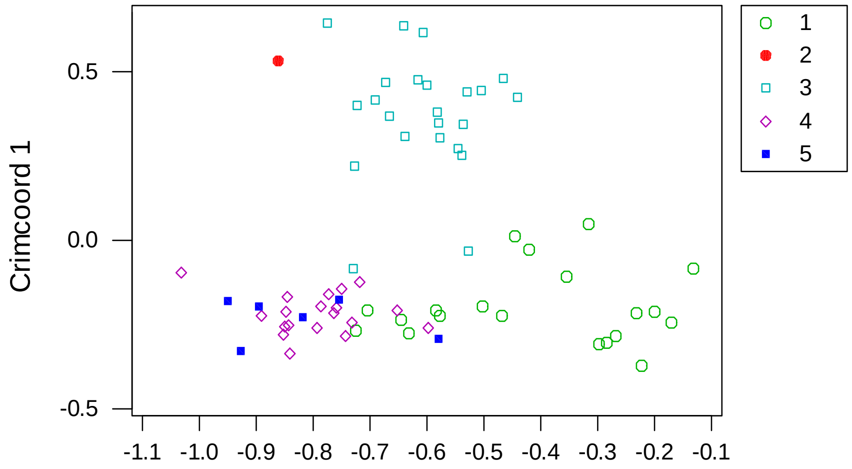
```
# First remove all data from groups 6 and above, done using
the Copy columns in Manip with the mit rows option
MTB > Copy 'Group'-'I' 'Group'-'I';
SUBC>   Omit 'Group' = 6:99.
# Next, copy out the data for groups 2 and 5, with the Use
rows option, not forgetting to cancel the omit rows option
from last time
MTB > Copy 'Group'-'I' c101-c110;
SUBC>   Use 'Group' = 2 5.
# Next, copy remove the data from groups 2 and 5 from the main
body by copying the columns into themselves, with the omit
rows option, not forgetting to cancel the use rows option from
last time
MTB > Copy 'Group'-'I'  'Group'-'I';
SUBC>   Omit 'Group' = 2 5.
# Now do the MANOVA to get the W**(-1)B matrix. This is in
Balanced Manova within ANOVA in version 13 but in version 12
it is in Multivariate, Balanced manova and some details of
output may be different.
MTB > ANOVA 'A'-'I' = Group;
SUBC>   MANOVA;
SUBC>   Eigen;
SUBC>   NoUnivariate.
```

ANOVA: A, B, C, D, E, F, G, H, I versus Group

```
MANOVA for Group               s =  2    m =  3.0    n =
24.5

Criterion        Test Statistic          F            DF       P
Wilk's                  0.06928     15.862   ( 18,   102)  0.000
Lawley-Hotelling        6.35730     17.659   ( 18,   100)  0.000
Pillai's                1.42097     14.179   ( 18,   104)  0.000
Roy's                   4.91873
```

```
EIGEN Analysis for Group

Eigenvalue    4.9187    1.4386   0.00000   0.00000   0.00000   0.00000
Proportion    0.7737    0.2263   0.00000   0.00000   0.00000   0.00000
Cumulative    0.7737    1.0000   1.00000   1.00000   1.00000   1.00000

Eigenvector        1         2         3         4         5         6
A             -0.010    -0.001    -0.007    -0.003     0.026    -0.031
B             -0.027     0.197    -0.099    -0.103    -0.086     0.036
C             -0.012     0.030     0.154     0.051    -0.006     0.085
D              0.433    -0.332     0.260    -0.284     0.043    -0.018
E             -3.026    -3.674    -4.324    -1.873    -1.920     0.182
F             12.912    -8.621     0.784    -3.279    -2.330    -2.055
G             -0.006    -0.017     0.010     0.005    -0.002    -0.001
H             -0.060     0.054    -0.037     0.151    -0.218    -0.168
I             -7.888     0.678    -5.970    12.367     9.882    -1.723

Eigenvalue   0.00000   0.00000   0.00000
Proportion   0.00000   0.00000   0.00000
Cumulative   1.00000   1.00000   1.00000

Eigenvector        7         8         9
A             -0.033    -0.044    -0.045
B             -0.005    -0.001     0.107
C             -0.016     0.012     0.095
D              1.537    -0.207    -0.025
E             -3.408     2.042    -7.957
F             -1.129    -0.909    -4.936
G              0.021     0.023    -0.002
H              0.017     0.037     0.024
I             -0.001    -0.041    -3.756

# Now highlight the first two eigenvectors by positioning the
cursor just to the left of the -0.010 of eigenvector 1,
holding down the ALT key and then with the left button
depressed moving the cursor just to the right of 0.678, (this
selects just the columns rather than all the  rows). Now click
the copy icon to copy into the clipboard, then move to the
data window and click the cell in row 1 of C12, then click the
paste icon to paste the two eigenvectors into C12 and C13.
#
# Now move back the data for groups 2 and 5 to be in the same
columns as the 'training data' for groups 1,3 and 4. Use
Manip>Stack>Stack Blocks of columns to do this.
MTB > Stack ('Group'-'I') (C101-C110) ('Group'-'I').

# Next, copy training data (groups 1,3, 4) and 'new cases'
(groups 2 and 5) into a matrix m1.
MTB > Copy 'A'-'I' m1.
# copy the two eigenvectors (which have just been pasted into
the data sheet) into matrix m2
MTB > Copy  C12 C13 m2.
# rotate all the data (training and new) onto the crimcoords.
MTB > Multiply m1 m2 m3.

# and copy the resulting matrix back into columns for
plotting, naming them sensibly.
MTB > Copy m3 c15 c16.
MTB > name c15 'Crimcoord 1' c16 'Crimcoord 2'
# Now produce the plots with Graph>Plot and under Data display
change the default in ‘For each’ from ‘Graph’ to ‘Group’ and
give Group as the name of the variable, then go into Edit
Attributes to choose pretty symbols for the plot
(alternatively use the lines of code given below).
MTB > Plot 'Crimcoord 1'*'Crimcoord 2';
SUBC>   Symbol 'Group';
SUBC>     Type 1 6 11 15 12;
SUBC>     Color 9 2 11 12 4;
SUBC>   ScFrame;
SUBC>   ScAnnotation.
```

Plot Crimcoord 1 * Crimcoord 2



It is clear from the plot that most of group 5 are like group 4 with one in the overlap region with group 1, the group 2 pot is most like group 3 out of the three possibilities on offer but is not a ‘typical’ group 3 pot.

iii) Compare your opinions with the results from the ready-made analysis in STATS>MULTIVARIATE>DISCRIMINANT using the options to predict the membership of groups 2 and 5.

First without cross-validation

```
MTB > Discriminant 'Group' 'A'-'I';
SUBC>   Predict C102-C110.
```

Discriminant Analysis: Group versus A, B, C, D, E, F, G, H, I

```
Linear Method for Response:   Group
Predictors:  A  B  C  D  E  F  G  H  I

Group        1        3        4
Count       20       23       19

Summary of Classification

Put into     ....True Group....
Group            1        3        4
1               16        1        1
3                0       21        0
4                4        1       18
Total N         20       23       19
N Correct       16       21       18
Proportion   0.800    0.913    0.947

N =   62     N Correct =   55     Proportion Correct = 0.887

Squared Distance Between Groups
                1        3        4
1          0.0000  21.6392   8.7280
3         21.6392   0.0000  22.8474
4          8.7280  22.8474   0.0000

Linear Discriminant Function for Group
                1        3        4
Constant   -64.24   -72.82   -78.14
A            1.90     1.55     1.86
B           -8.32   -13.37   -12.23
C            5.51     5.88     8.46
D           56.90    80.71    65.72
E           77.11    10.79    33.65
F          196.22   750.93   248.60
G            0.59     0.83     0.99
H            6.44     2.74     4.81
```

```
I           37.33  -290.41   -66.66

Summary of Misclassified Observations

Observation     True     Pred     Group      Squared   Probability
                Group    Group               Distance
    5 **           1        4         1        18.52         0.090
                                      3        34.33         0.000
                                      4        13.89         0.910
    9 **           1        4         1        7.631         0.265
                                      3       26.576         0.000
                                      4        5.591         0.735
   12 **           1        4         1       10.001         0.313
                                      3       31.864         0.000
                                      4        8.431         0.687
   14 **           1        4         1       11.013         0.054
                                      3       30.185         0.000
                                      4        5.277         0.946
   24 **           3        4         1        8.920         0.061
                                      3       15.680         0.002
                                      4        3.444         0.937
   34 **           3        1         1        12.22         0.854
                                      3        19.98         0.018
                                      4        16.01         0.129
   56 **           4        1         1        6.681         0.512
                                      3       28.222         0.000
                                      4        6.776         0.488
```

Prediction for Test Observations

```
 Observation  Pred Group  From Group Sqrd Distnc Probability
           1           3
                                   1      68.554       0.000
                                   3      29.937       1.000
                                   4      58.479       0.000
           2           4
                                   1      25.307       0.032
                                   3      37.330       0.000
                                   4      18.517       0.967
           3           4
                                   1      78.880       0.000
                                   3      86.709       0.000
                                   4      63.197       1.000
           4           4
                                   1      92.554       0.001
                                   3     102.880       0.000
                                   4      79.229       0.999
           5           4
                                   1      80.461       0.001
                                   3      98.965       0.000
                                   4      65.380       0.999
           6           4
                                   1     138.275       0.007
                                   3     152.710       0.000
                                   4     128.415       0.993
           7           1
                                   1     202.608       0.600
                                   3     226.511       0.000
                                   4     203.419       0.400
```

Note that these classifications agree with those obtained informally with the crimcoord plot but that the classification of the group 2 plot as type 3 has been made with 'near certainty' making no allowance for the fact that it does not look like a typical group 3 pot.

```
MTB > Discriminant 'Group' 'A'-'I';
SUBC>   XVal;
SUBC>   Predict C102-C110.
```

Discriminant Analysis: Group versus A, B, C, D, E, F, G, H, I

```
Linear Method for Response:   Group
Predictors:  A  B  C  D  E  F  G  H  I

Group        1        3        4
Count       20       23       19

Summary of Classification

Put into     ....True Group....
Group            1        3        4
1               16        1        1
3                0       21        0
4                4        1       18
Total N         20       23       19
N Correct       16       21       18
Proportion   0.800    0.913    0.947

N =   62     N Correct =   55     Proportion Correct = 0.887

Summary of Classification with Cross-validation

Put into     ....True Group....
Group            1        3        4
1               12        1        1
3                1       21        0
4                7        1       18
Total N         20       23       19
N Correct       12       21       18
Proportion   0.600    0.913    0.947

N =   62     N Correct =   51     Proportion Correct = 0.823

Squared Distance Between Groups
                1        3        4
1          0.0000  21.6392   8.7280
3         21.6392   0.0000  22.8474
4          8.7280  22.8474   0.0000
Summary of Misclassified Observations
```

```
Observation    True     Pred    X-val    Group    Squared Distance
Probability
              Group    Group    Group               Pred     X-val    Pred
X-val
   5 **           1        4        4        1     18.52     30.13    0.09
0.00
                                             3     34.33     40.13    0.00
0.00
                                             4     13.89     17.32    0.91
1.00
   8 **           1        1        4        1     23.84     45.18    0.95
0.44
                                             3     46.00     62.95    0.00
0.00
                                             4     29.83     44.73    0.05
0.56
   9 **           1        4        4        1     7.631     9.622    0.27
0.12
                                             3    26.576    26.928    0.00
0.00
                                             4     5.591     5.599    0.73
0.88
  10 **           1        1        4        1     5.174     6.208    0.63
0.49
                                             3    24.754    24.666    0.00
0.00
                                             4     6.196     6.125    0.37
0.51
  12 **           1        4        4        1    10.001    13.259    0.31
0.10
                                             3    31.864    33.507    0.00
0.00
                                             4     8.431     8.790    0.69
0.90
  14 **           1        4        4        1    11.013    14.930    0.05
0.01
                                             3    30.185    31.761    0.00
0.00
                                             4     5.277     5.500    0.95
0.99
  16 **           1        1        4        1     10.90     14.74    0.59
0.24
                                             3     29.16     30.51    0.00
0.00
                                             4     11.61     12.45    0.41
0.76
  19 **           1        1        3        1     19.87     33.52    0.94
0.14
                                             3     26.35     30.01    0.04
0.83
                                             4     27.46     37.11    0.02
0.02
  24 **           3        4        4        1     8.920     8.822    0.06
0.06
                                             3    15.680    23.329    0.00
0.00
                                             4     3.444     3.469    0.94
0.94
  34 **           3        1        1        1     12.22     12.77    0.85
0.89
                                             3     19.98     33.24    0.02
0.00
                                             4     16.01     16.91    0.13
0.11
```

```
   56 **          4        1        1        1      6.681      6.679    0.51
0.71
                                            3     28.222     28.482    0.00
0.00
                                            4      6.776      8.446    0.49
0.29
```

Prediction for Test Observations

```
 Observation  Pred Group  From Group Sqrd Distnc Probability
           1           3
                                   1      68.554       0.000
                                   3      29.937       1.000
                                   4      58.479       0.000
           2           4
                                   1      25.307       0.032
                                   3      37.330       0.000
                                   4      18.517       0.967
           3           4
                                   1      78.880       0.000
                                   3      86.709       0.000
                                   4      63.197       1.000
           4           4
                                   1      92.554       0.001
                                   3     102.880       0.000
                                   4      79.229       0.999
           5           4
                                   1      80.461       0.001
                                   3      98.965       0.000
                                   4      65.380       0.999
           6           4
                                   1     138.275       0.007
                                   3     152.710       0.000
                                   4     128.415       0.993
           7           1
                                   1     202.608       0.600
                                   3     226.511       0.000
                                   4     203.419       0.400
```

Note that the cross-validated estimate of the classification rate is slightly lower and [of course] the classification of the new pots is unaltered by the cross-validation option.

1) *$x_1,\ldots,x_n$ are independent measurements of $N_p(\mu,\sigma^2 I_p)$*

i) *Shew that the maximum likelihood estimate of μ, subject to $\mu'\mu = r_0^2$ (a known constant) is the same whether σ is known or unknown.*

This example is very like example 5.5.3 in the lecture notes:

We have $\ell(\mu; X) = -\tfrac{1}{2}(n-1)\mathrm{trace}(S\sigma^{-2}) - \tfrac{1}{2}n(\bar{x}-\mu)'(\bar{x}-\mu)\sigma^{-2} -$

$$\tfrac{1}{2}np\log(2\pi) - \tfrac{1}{2}np\log(\sigma^2)$$

Let $\Omega = \ell(\mu) - \lambda(\mu'\mu - r_0^2)$ then $\frac{\partial\Omega}{\partial\mu} = n(\bar{x}-\mu)\sigma^{-2} - 2\lambda\mu$.

So we require $\hat{\mu} = \frac{n\bar{x}}{n+2\lambda\sigma^2}$ then $\mu'\mu = r_0^2$ implies $(n+2\lambda\sigma^2)^2 r_0^2 = n^2\bar{x}'\bar{x}$

and so $\hat{\mu} = \frac{\bar{x}\,r_0}{\sqrt{\bar{x}'\bar{x}}}$ which does not depend on σ^2.

ii) *Find the maximum likelihood estimate of σ when neither μ nor σ are known.*

$$\frac{\partial\Omega}{\partial\sigma} = (n-1)\mathrm{tr}(S)\sigma^{-3} + n(\bar{x}-\mu)'(\bar{x}-\mu)\sigma^{-3} - np\sigma^{-1}$$

$$\text{so}\quad \hat{\sigma} = \sqrt{\tfrac{1}{np}\left[(n-1)\mathrm{tr}(S) + n(\bar{x}-\hat{\mu})'(\bar{x}-\hat{\mu})\right]}$$

$$= \sqrt{\tfrac{1}{np}\left[(n-1)\mathrm{tr}(S) + n(\sqrt{\bar{x}'\bar{x}} - r_0)^2\right]}$$

iii) *Hence, in the case when $\sigma = \sigma_0$ (a known constant) construct the likelihood ratio test of $H_0 : \mu'\mu = r_0^2$ vs $H_A : \mu'\mu \neq r_0^2$ based on n independent observations of $N_p(\mu, \sigma_0^2 I_p)$.*

Under H_0

$$\ell_{\max} = K - \tfrac{1}{2}n(\sqrt{\bar{x}'\bar{x}} - r_0)^2\sigma_0^{-2}$$

Under H_A we have

$$\hat{\mu} = \bar{x} \quad \text{so} \quad \ell_{\max} = K$$

so LRT statistic is $n(\sqrt{\bar{x}'\bar{x}} - r_0)^2\sigma_0^{-2}$ and under H_0 this $\sim \chi_1^2$

[1 d.f. since p parameters in μ estimated under H_A and p with 1 constraint under H_0]

iv) *In an experiment to test the range of a new ground-to-air missile thirty-nine test firings at a tethered balloon were performed and the three dimensional coordinates of the point of ignition of the missile's warhead measured. These gave a mean result of (0.76, 0.69, 0.66)′ relative to the site expressed in terms of the target distance. Presuming that individual measurements are independently normally distributed with unit variance, are the data consistent with the theory that the range of the missile was set correctly?*

We have σ_0=1=r_0 and so

$$n(\sqrt{\bar{x}'\bar{x}} - r_0)^2\sigma_0^{-2} = 39(\sqrt{}(0.76, 0.69, 0.66)'\ (0.76, 0.69, 0.66) - 1)^2$$

= 1.894 (<<3.84=$\chi^2_{1;0.95}$) and so yes, the data are consistent with the theory that the range was set correctly

Background information for [partially seen] Quiz

Data are available on national track records for various distances held by women from 55 different countries (as they stood at the start of the 1984 Los Angeles Olympics). The distances are, in metres, 100, 200, 400, 800, 1500, 3000 and marathon. A small sample of the data is given below:

```
> womentrackrecords[1:10,]; womentrackrecords[46:55,]
         X100m X200m X400m X800m X1500m X3000m marathon
argentin 11.61 22.94 54.50  2.15   4.43   9.79   178.52
australi 11.20 22.35 51.08  1.98   4.13   9.08   152.37
austria  11.43 23.09 50.62  1.99   4.22   9.34   159.37
belgium  11.41 23.04 52.00  2.00   4.14   8.88   157.85
bermuda  11.46 23.05 53.30  2.16   4.58   9.81   169.98
brazil   11.31 23.17 52.80  2.10   4.49   9.77   168.75
burma    12.14 24.47 55.00  2.18   4.45   9.51   191.02
canada   11.00 22.25 50.06  2.00   4.06   8.81   149.45
chile    12.00 24.52 54.90  2.05   4.23   9.37   171.38
china    11.95 24.41 54.97  2.08   4.33   9.31   168.48
..................
               ..................
                              ..................
singapor 12.30 25.00 55.08  2.12   4.52   9.94   182.77
spain    11.80 23.98 53.59  2.05   4.14   9.02   162.60
sweden   11.16 22.82 51.79  2.02   4.12   8.84   154.48
switzerl 11.45 23.31 53.11  2.02   4.07   8.77   153.42
taipei   11.22 22.62 52.50  2.10   4.38   9.63   177.87
thailand 11.75 24.46 55.80  2.20   4.72  10.28   168.45
turkey   11.98 24.44 56.45  2.15   4.37   9.38   201.08
usa      10.79 21.83 50.62  1.96   3.95   8.50   142.72
ussr     11.06 22.19 49.19  1.89   3.87   8.45   151.22
wsamoa   12.74 25.85 58.73  2.33   5.81  13.04   306.00
```

The complete list of countries as held in the data file is

```
rownames(womentrackrecords)
 [1]  "argentin"  "australi"  "austria"   "belgium"   "bermuda"
"brazil"
 [7]  "burma"     "canada"    "chile"     "china"     "columbia"
"cookis"
[13]  "costa"     "czech"     "denmark"   "domrep"    "finland"
"france"
[19]  "gdr"       "frg"       "gbni"      "greece"    "guatemal"
"hungary"
[25]  "india"     "indonesi"  "ireland"   "israel"    "italy"
"japan"
[31]  "kenya"     "korea"     "dprkorea"  "luxembou"  "malaysia"
"mauritiu"
[37]  "mexico"    "netherla"  "nz"        "norway"    "png"
"philippi"
[43]  "poland"    "portugal"  "rumania"   "singapor"  "spain"
"sweden"
```

```
[49]  "switzerl"  "taipei"     "thailand"  "turkey"     "usa"
"ussr"
[55] "wsamoa"
```

(NB `gdr` the [former] East Germany. `frg` is the [former] West Germany, `gbni` is the UK and `png` is Papua New Guinea.

Below are some preliminary multivariate data analyses:

```
> print(wtrcov.pc$loadings,cutoff=0.001);
print(wtrcorr.pc$loadings)

Loadings:
         Comp.1 Comp.2 Comp.3 Comp.4 Comp.5 Comp.6 Comp.7
X100m    -0.010 -0.120  0.326 -0.150  0.925  0.002 -0.017
X200m    -0.025 -0.315  0.880 -0.014 -0.354 -0.025 -0.012
X400m    -0.062 -0.934 -0.328  0.122  0.013  0.022  0.025
X800m    -0.003 -0.026 -0.037 -0.049 -0.015 -0.262 -0.963
X1500m   -0.010 -0.039 -0.055 -0.340 -0.034 -0.899  0.265
X3000m   -0.024 -0.082 -0.088 -0.919 -0.130  0.349 -0.041
marathon -0.997  0.070 -0.002  0.020  0.002

Loadings:
         Comp.1 Comp.2 Comp.3 Comp.4 Comp.5 Comp.6 Comp.7
X100m    -0.368  0.490  0.286 -0.319 -0.231  0.620
X200m    -0.365  0.537  0.230               -0.711 -0.109
X400m    -0.382  0.247 -0.515  0.347  0.572  0.191  0.208
X800m    -0.385 -0.155 -0.585        -0.620        -0.315
X1500m   -0.389 -0.360        -0.430        -0.231  0.693
X3000m   -0.389 -0.348  0.153 -0.363  0.463        -0.598
marathon -0.367 -0.369  0.484  0.672 -0.131  0.142
```

Various questions will follow later.

Notes & Solutions to Exercises 1

4) Dataset `nfl2000.Rdata`★ gives performance statistics for 31 teams in the US National Football League for the year 2000. Twelve measures of performance were made, six relate to the home team performance and six to the opponent team (i.e. when the team was playing ‘at home’ and ‘away’). The measures of performance were

homedrives50	*drives begun in opponents' territory*
homedrives20	*drives begun within 20 yards of the goal*
oppdrives50	*opponents drives begun in team's territory*
oppdrives20	*opponents drives begun within 20 yards of goal*
hometouch	*touchdowns scored by team*
opptouch	*touchdowns scored against team*
homeyards	*total yardage gained by offence*
oppyards	*total yardage allowed by defence*
hometop	*time of possession by offence (in minutes)*
opptop	*time of possession by opponents' offence*
home1sts	*first downs obtained by offence*
opp1sts	*first downs allowed by defence*

The dataset contains a three letter abbreviation for the team as a row name. The coding is

initials	***team***	***initials***	***team***
ARI	*Arizona Cardinals*	***BAL***	*Baltimore Ravens*
ATL	*Atlanta Falcons*	***BUF***	*Buffalo Bills*
CAR	*Carolina Panthers*	***CIN***	*Cincinnati Bengals*
CHI	*Chicago Bears*	***CLE***	*Cleveland Browns*
DAL	*Dallas Cowboys*	***DEN***	*Denver Broncos*
DET	*Detroit Lions*	***IND***	*Indianapolis Colts*
GB	*Green Bay Packers*	***JAX***	*Jacksonville Jaguars*
MIN	*Minnesota Vikings*	***KC***	*Kansas City Chiefs*
NO	*New Orleans Saints*	***MIA***	*Miami Dolphins*
NYG	*New York Giants*	***NE***	*New England Patriots*
PHI	*Philadelphia Eagles*	***NYJ***	*New York Jets*
SF	*San Francisco 49ers*	***OAK***	*Oakland Raiders*
STL	*St. Louis Rams*	***PIT***	*Pittsburgh Steelers*
TB	*Tampa Bay Buccaneers*	***SD***	*San Diego Chargers*
WAS	*Washington Redskins*	***SEA***	*Seattle Seahawks*
		TEN	*Tennessee Titans*

i) Use principal component analysis to identify and describe the main sources of variation of the performances.

ii) Produce a scatter plot of the teams referred to their principal component scores and comment on any features you think worthy of mention.

*(**NB:** You are strongly advised to work through Task Sheet 2, Q3 if you have not already done so).*

★source: Journal of Statistics Education Data Archive

First, open the datafile `nfl2000` and then run the functions `screeplot()` and `identifyPCH()` (available from the folder R scriptfiles on the Semester 1 folder of MAS6011 module pages, as are the script file for the solution to this question). This is most easily done by downloading these files to a common folder and then double clicking on `nfl2000.Rdata` and then opening the two script files using the icon on the top left of the R session window.

```
> nfl2000[1:5,]                # gives first few lines of the
data file
    homedrives50    homedrives20    oppdrives50    oppdrives20
hometouch opptouch
ARI             17               2            27             6
24       52
ATL             27               4            26             3
25       46
CAR             26               2            29             3
30       35
CHI             15               1            22             5
22       43
DAL             22               3            30             4
31       41
    homeyards oppyards hometop opptop home1sts opp1sts
ARI      4756     5872    26.5   33.5      253     345
ATL      4380     5749    29.6   30.4      256     308
CAR      5036     5882    29.9   30.1      304     304
CHI      4741     5464    28.5   31.5      239     297
DAL      4724     5518    28.7   31.3      276     309
> options(digits=3)                 #    suppress    unnecessary
decimal places
>  sqrt(diag(var(nfl2000)))            #     gives     standard
deviations
homedrives50   homedrives20      oppdrives50      oppdrives20
hometouch     opptouch
        7.66            2.60             7.35             1.95
11.09        9.57
   homeyards        oppyards          hometop           opptop
home1sts      opp1sts
      757.83          531.59             1.94             1.94
42.35        35.45
>
> apply(nfl2000,2,sum)
homedrives50   homedrives20      oppdrives50      oppdrives20
hometouch     opptouch
         789             119              789              119
1146         1146
   homeyards        oppyards          hometop           opptop
home1sts      opp1sts
```

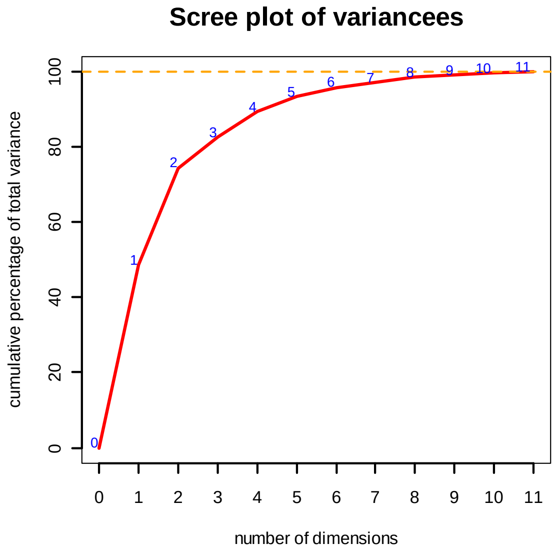
```
     165973.0       165973.0            929.6            930.4
9139.0       9139.0
```

Note wide the variation in standard deviations so perform PCA with correlations. Note also that `homeXXX` and `oppXXX` variables have matching totals so `homeXXX` counts events scored *by* the team and `oppXXX` the corresponding events *against* that team (rather than my initial guess of playing at *home* or *away, apologies*). Note that `sum` could equally well have been replaced by `mean` but since these are counts it seems natural to use totals. Also note that it would probably be sensible to look at some histograms to consider transforming the counts to achieve more symmetrical distributions (logarithmic or at least square roots are usually sensible for counts, but this is not critical for exploratory analysis such as PCA here).

```
>  nfl.pc<-princomp(nfl2000,cor=T)
>  summary(nfl.pc)
Importance of components:
                      Comp.1   Comp.2   Comp.3   Comp.4   Comp.5
Comp.6 Comp.7 Comp.8
Standard  deviation       2.415   1.764  0.9846  0.9128  0.6895
0.5374 0.4078 0.3835
Proportion  of  Variance   0.486   0.259  0.0808  0.0694  0.0396
0.0241 0.0139 0.0123
Cumulative  Proportion     0.486   0.745  0.8259  0.8953  0.9350
0.9590 0.9729 0.9851
                       Comp.9 Comp.10 Comp.11  Comp.12
Standard deviation     0.31153 0.23512 0.16141 5.40e-09
Proportion of Variance 0.00809 0.00461 0.00217 2.43e-18
Cumulative Proportion  0.99322 0.99783 1.00000 1.00e+00
```

Looking at cumulative proportion suggests a cuttoff around 4 or 5 components so look at a screeplot:

```
>  screeplot(nfl2000,T,maxcomp=11) # note use of maxcomp = 11
>                                   # since default is only 10
eigenvaues
```



This suggests a kink at 4 (although there is a sharper kink at 2 this is rather a low dimension and gives a total of rather less than 80%) so the majority of the 'information' is likely to be in the first four PCs and the remaining seven are regarded as 'noise' — though may be worth checking the fifth for any obvious interpretation.

```
>  print(nfl.pc$loadings, cutoff=0.01)

Loadings:
             Comp.1 Comp.2 Comp.3 Comp.4 Comp.5
homedrives50 -0.291 -0.189 -0.383  0.399  0.245
homedrives20 -0.250 -0.217 -0.579  0.305 -0.147
oppdrives50   0.253 -0.191 -0.434 -0.433 -0.535
oppdrives20   0.268 -0.101 -0.397 -0.410  0.723
hometouch    -0.208  0.445 -0.184        -0.041
opptouch      0.272  0.314 -0.310 -0.011 -0.255
homeyards    -0.222  0.445 -0.082 -0.253
oppyards      0.284  0.322 -0.150  0.345  0.099
hometop      -0.399  0.010 -0.031 -0.123 -0.069
opptop        0.399 -0.010  0.031  0.123  0.069
home1sts     -0.238  0.430 -0.094 -0.199  0.151
opp1sts       0.309  0.297 -0.054  0.374 -0.034
```

suppressing more decimal places may make this easier to assimilate:

```
> print(nfl.pc$loadings, cutoff=0.1,digits=1)

Loadings:
             Comp.1 Comp.2 Comp.3 Comp.4 Comp.5
homedrives50 -0.3   -0.2   -0.4    0.4    0.2
homedrives20 -0.2   -0.2   -0.6    0.3   -0.1
oppdrives50   0.3   -0.2   -0.4   -0.4   -0.5
oppdrives20   0.3   -0.1   -0.4   -0.4    0.7
hometouch    -0.2    0.4   -0.2
opptouch      0.3    0.3   -0.3          -0.3
homeyards    -0.2    0.4          -0.3
oppyards      0.3    0.3   -0.2    0.3
hometop      -0.4                 -0.1
opptop        0.4                  0.1
home1sts     -0.2    0.4          -0.2    0.2
opp1sts       0.3    0.3           0.4
```

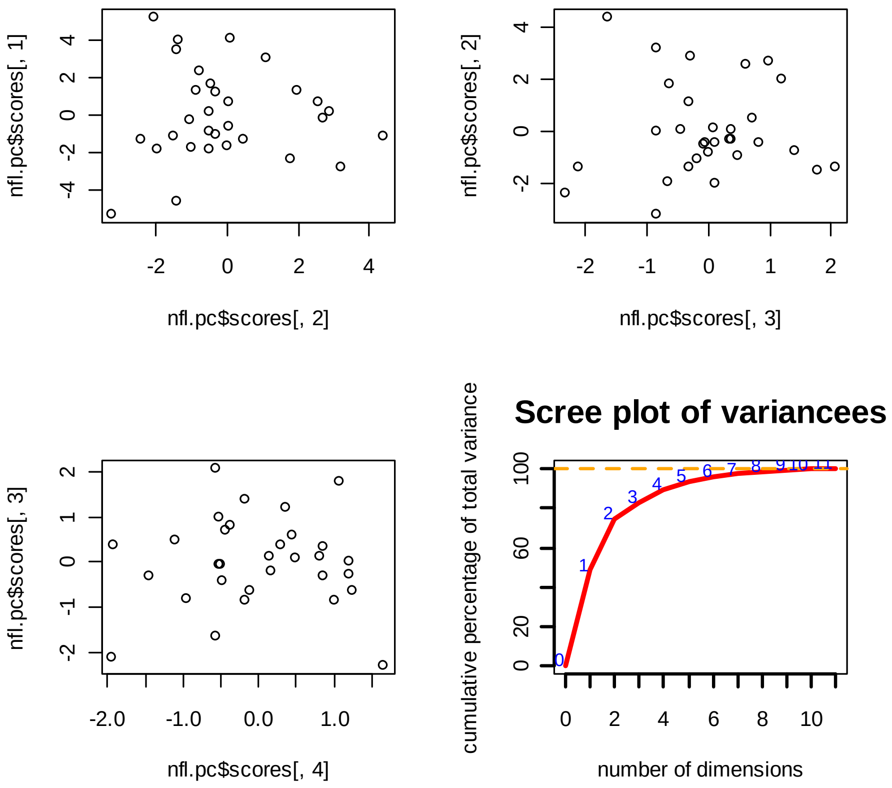
(note that PCs beyond the 5th have been edited away but they appear in the output).

i) Use principal component analysis to identify and describe the main sources of variation of the performances.

The first component has positive signs for all *opp* variables and negative for all *home* ones and so reflects variations in aggregates of *for – away* counts over the various categories. Component 2 reflects *drives* against all other variables. Component 3 has all signs the same for loadings of all variables (and checking without cutoff = 0.1 almost maintains this statement) and so reflects overall variations in total counts. Interpretation of component 4 would benefit from a greater knowledge of the intricacies of American football but it is notable that *home* and *opp* versions of pairs of variables have opposite signs with *drives* having positive signs and all others negative for the *home* versions*.* This suggests it is a contrast between the differences *for – against* between drives and the other variables. Note that this explanation is a little complicated and attempts to interpret the fifth component are even more convoluted, supporting the suggestion that this reflects noise rather than coherent information. So, in summary, the prime source of variation is that some teams have many more counts of events such as *drives*, *touches*, *and yards* than those scored against them whilst for others the converse is true. More simply the greatest source of variation is in the quality of the teams, some are very much better than others. Secondly, some teams score many more *drives* than other noted events while others many fewer; thirdly some teams are involved in matches where a large number of 'events' are noted whilst for other teams the matches might be considered less eventful. Finally, there is a small component of variation attributable arising from teams which exceed the number of drives accrued over those against by more than the corresponding counts of touches, yards etc compared with those for which the converse is true.

iii) Produce a scatter plot of the teams referred to their principal component scores and comment on any features you think worthy of mention.

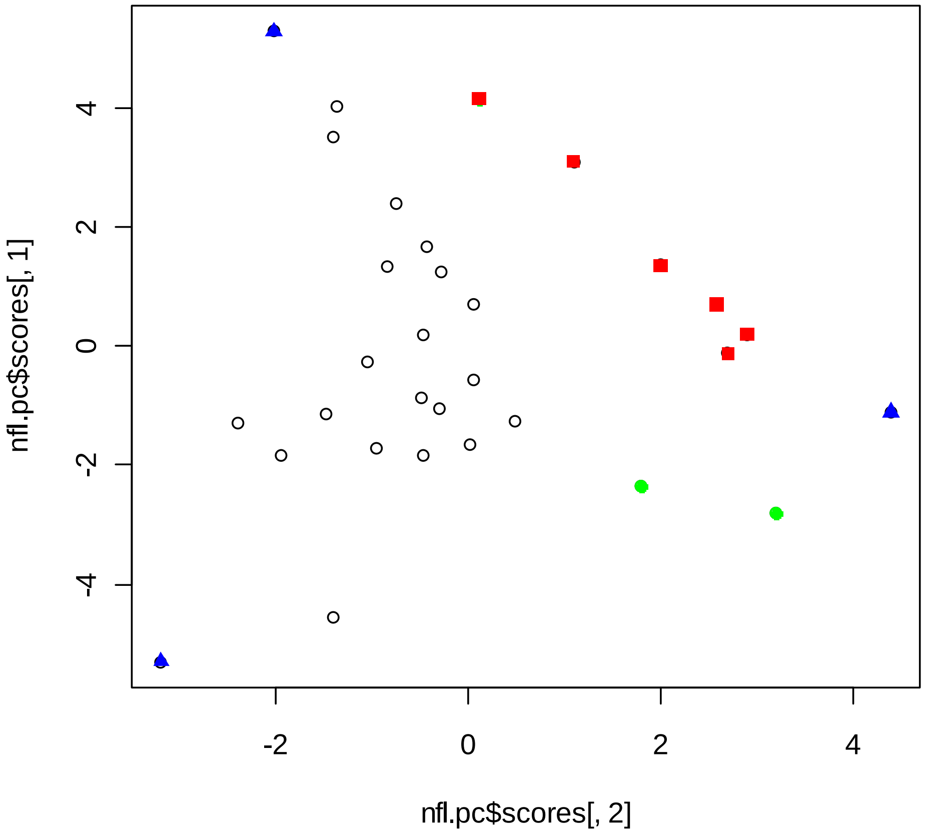
```
> par(mfrow=c(2,2))
> plot(nfl.pc$scores[,2],nfl.pc$scores[,1])
> plot(nfl.pc$scores[,3],nfl.pc$scores[,2])
> plot(nfl.pc$scores[,4],nfl.pc$scores[,3])
> screeplot(nfl2000,T,maxcomp=11)
```


nfl.pc$scores[, 1]
4
2
0
-2
-4
-2
0
2
4
nfl.pc$scores[, 2]
nfl.pc$scores[, 2]
4
2
0
-2
-2
-1
0
1
2
nfl.pc$scores[, 3]
nfl.pc$scores[, 3]
2
1
0
-1
-2
-2.0
-1.0
0.0
1.0
nfl.pc$scores[, 4]
Scree plot of variancees
cumulative percentage of total variance
100
60
20
0
0
1
2
3
4
5
6
7
8
9
10
11
0
2
4
6
8
10
number of dimensions

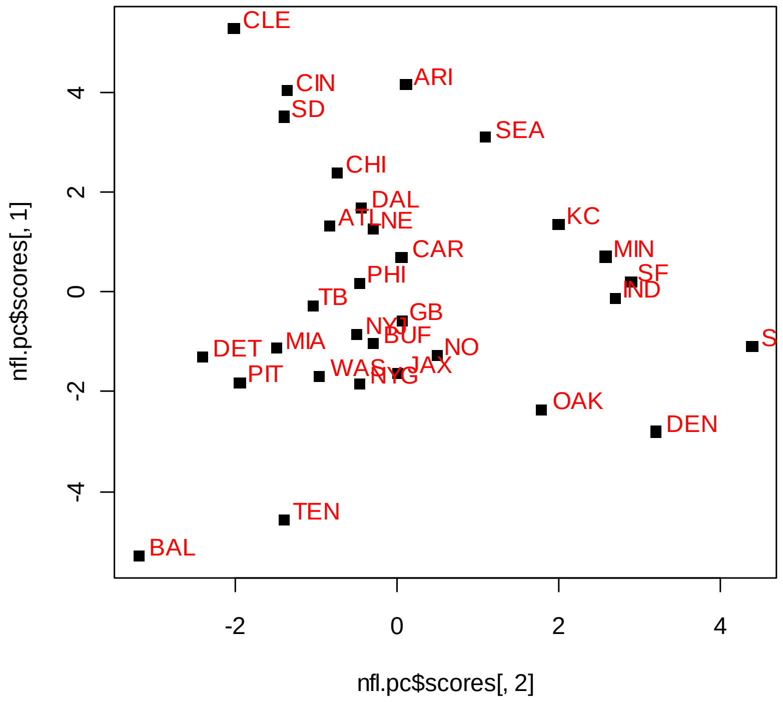
The plot on the first two components (but not on others) suggests it may be worth investigating a few of the outliers and there is maybe a group of half a dozen points towards the top right hand corner:

This could be done with `identifyPCH()` as below or else adding the row names as labels, though then there is some overlapping and illegibility but at least the outliers can be identified.

```
> par(mfrow=c(1,1))
> plot(nfl.pc$scores[,2],nfl.pc$scores[,1])
>
identifyPCH(nfl.pc$scores[,2],nfl.pc$scores[,1],col="red",pch=
15)
[1]  1 30 23  8 12 21
>
identifyPCH(nfl.pc$scores[,2],nfl.pc$scores[,1],col="blue",pch
=17)
[1] 16 19 13
>
identifyPCH(nfl.pc$scores[,2],nfl.pc$scores[,1],col="green",pc
h=19)
[1] 20 27
>
```


nfl.pc$scores[, 1]
4
2
0
-2
-4
-2
0
2
4
nfl.pc$scores[, 2]

```
> plot(nfl.pc$scores[,2],nfl.pc$scores[,1],pch=15)
> text(nfl.pc$scores[,2],nfl.pc$scores[,1],
row.names(nfl2000),
+ col="red",adj=c(-.2,-.1))
```



The next task is to say what it is that characterises the teams of interest. For example the team BAL appears on the bottom left corner so it has low scores on the first two PCs and so the Ravens have obtained exceptionally higher numbers of drives, touches etc than were obtained against them as well as unusually many more drives than touches and yards were counted in their games (need to look carefully at the signs of the coefficients of the various loadings in these two PCs to deduce these specific statements). Others can be described in a similar way, e.g. the Browns have obtained exceptionally lower numbers of drives, touches etc than were obtained against them as well as rather more drives than touches and yards were counted in their games. The Cardinals, Seahawks, City Chiefs, Vikings, 49ers and Colts have generally more counts of events *against* than *for* and comparably fewer *drives* than other events in matches involving them.

Comments: Note that this example does not have a first PC reflecting overall total level or amount (i.e. with identical signs for all loadings). Nevertheless the conclusion drawn from interpretation of the first PC is ‘obvious’ — of course some teams are much better than others and this accounts for the greater proportion of the variation.

2) Measurements of various chemical properties were made on 43 samples of soil taken from areas close to motorway bridges suffering from corrosion. The corrosion can be of either of two types and the ultimate aim of the investigation was to see whether these measurements could be used to discriminate between the two types. Before such a full-scale analysis was undertaken some preliminary analyses were performed, using MINITAB. The record of the session (edited in places) is given below.

(a) The principal component analysis has been performed on the correlation matrix rather than the covariance matrix. Why is this to be preferred for these data?

The measurements have very different standard deviations (a factor of five between smallest and largest, so a factor of >25 in variances). Additionally, the variables appear to be measuring different types of properties.

(b) By using some suitable informal graphical technique, how may components would you recommend using in subsequent analyses?

Draw a scree graph (not shewn here) — kink comes after 4 or 5 so recommend 4 or 5.

(c) What features of the samples do the first three components reflect?

PC1 = (approx)
.4(pH+Carbon)–.4(Water+pyrite+organic+masslosss) and so is a contrast between pH& Carbon and these other 4 variables
PC2: contrast between water, pyrite, carbon and the others (except pH)
PC3: primarily organic vs rest

(d) What, approximately, is the value of the sample correlation between the scores of PC-1 and PC-2?

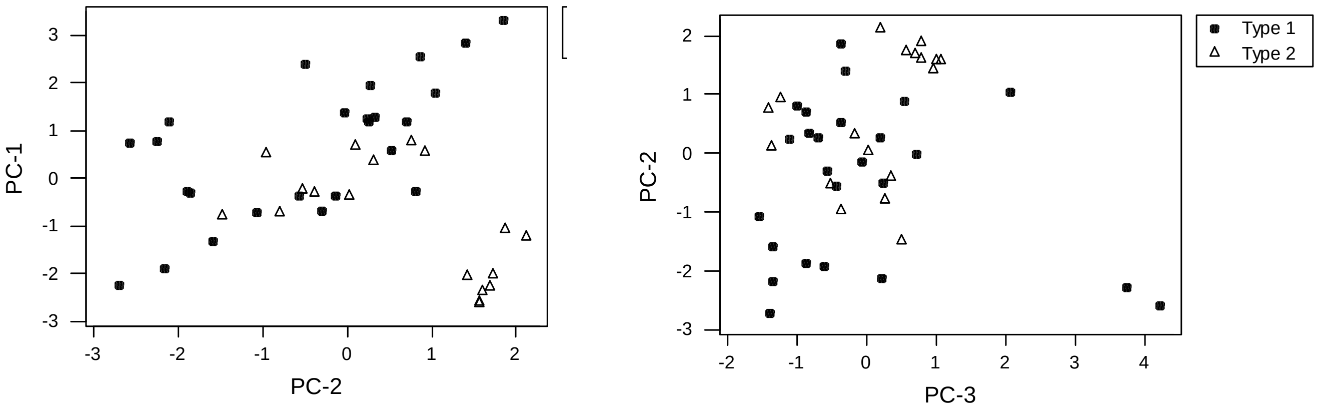
Zero! (up to rounding error), despite the appearance of the first diagram. (PCs are **always uncorrelated** by construction!)

(e) After looking at the various scatter plots of the principal component scores, what recommendation would you give to the investigator regarding the advisability of continuing with a discriminant analysis?

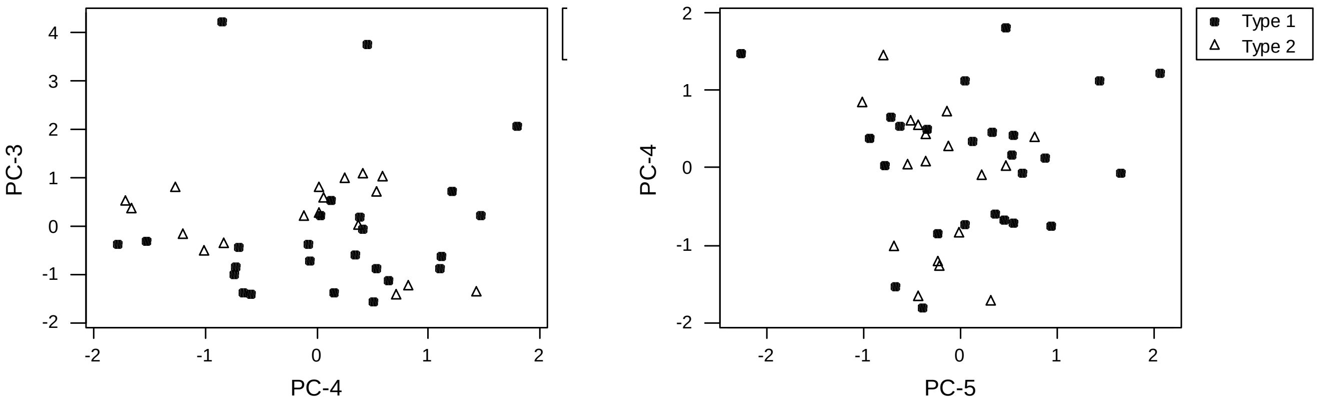
Discursive: key points are (i) that type 2 does not appear to be a homogeneous group — plots of PC1 vs PC2 and PC2 vs PC3 (71.5% of variation) reveal clear separation into distinct groups and so inadvisable to discriminate this from others, especially as one of the subgroups of type 2 appears to be very similar to type 1, (ii) No evidence of overall separation despite the first 5 PCs accounting for 90% of the variation. Could also mention (iii) outliers on PC 3.

```
Worksheet size: 100000 cells
MTB > Retrieve  "C:\soil.MTW".

MTB > desc c2-c9;
SUBC> by c1.

Descriptive Statistics
Variable   Type        N       Mean      StDev
pH         Type 1     25      8.416      0.962
           Type 2     18     8.0722     0.3102
Water      Type 1     25      1.693      0.716
           Type 2     18      2.831      1.812
Acid       Type 1     25     0.5672     0.3937
           Type 2     18     0.4322     0.2603
Pyrite     Type 1     25     0.4628     0.2563
           Type 2     18      1.019      0.500
Carbon     Type 1     25     11.251      4.230
           Type 2     18      9.783      1.862
Moisture   Type 1     25     23.712      4.975
           Type 2     18     21.922      2.647
Organic    Type 1     25      2.556      0.720
           Type 2     18      2.272      0.530
MassLos    Type 1     25      5.536      1.575
           Type 2     18      6.833      0.807


MTB > PCA  'pH'-'MassLos';
SUBC>   Coefficients c31-c38;
SUBC>   Scores'PC-1'-'PC-8'.
Principal Component Analysis
```

```
Eigenanalysis of the Correlation Matrix
Eigenvalue   2.351   1.862   1.504  0.827  0.612  0.412  0.230  0.197
Proportion   0.294   0.233   0.188  0.103  0.077  0.052  0.029  0.025
Cumulative   0.294   0.527   0.715  0.818  0.895  0.947  0.975  1.000

Variable       PC1     PC2     PC3     PC4     PC5     PC6     PC7     PC8
pH           0.348  -0.032   0.559   0.267  -0.126   0.599   0.334   0.095
Water       -0.455   0.270   0.339   0.219   0.042  -0.460   0.520   0.272
Acid        -0.002  -0.367   0.622  -0.053   0.347  -0.238  -0.520   0.168
Pyrite      -0.351   0.446   0.157   0.417  -0.344   0.157  -0.539  -0.214
Carbon       0.520   0.291  -0.077   0.022  -0.355  -0.285  -0.206   0.624
Moisture    -0.001  -0.582   0.068   0.148  -0.687  -0.318   0.090  -0.231
Organic     -0.204  -0.392  -0.387   0.616   0.181   0.188  -0.067   0.450
MassLos     -0.487  -0.118   0.048  -0.549  -0.336   0.363  -0.049   0.445

MTB > Plot 'PC-1'*'PC-2' 'PC-2'*'PC-3' 'PC-3'*'PC-4''PC-4'*'PC-5';
SUBC>   Symbol 'Type';
SUBC>     Type 6 19;
SUBC>     Size 1.0 1.5;
SUBC>   ScFrame;
SUBC>   ScAnnotation.
MTB > STOP
```



(This question is taken from the 1999/2000 examination)



3) ⋆⋆⋆ ***(Not for submission)*** *Suppose X={x_{ij} ; i=1,...,p, j=1,...,n} is a set of n observations in p dimensions with* $\sum_{j=1}^{n} x_{ij} = 0$ *all i=1,...,p (i.e. each of the p variables has zero mean, so* $\bar{x} = 0$ *) and S=XX′/(n-1) is the sample variance of the data. Let* $u_j=x_j{'}S^{-1}x_j$ *(j=1,...,n) (so* u_j *is the squared Mahalanobis distance of* x_j *from the sample mean 0). Suppose the data are projected into one dimension by Y=β′X (β a p×1 vector). Let* $y_j=\beta' x_j$ *and define* $U_j(\beta)=(n-1)y_j{'}(YY')^{-1}y_j$ *.*

   i) *Shew that* $U_j(\beta)$ *is maximized with respect to β by the (right) eigenvector of* $S^{-1}x_jx_j{'}$ *corresponding to its only non-zero eigenvalue.*

   $U_j(\beta)=(n-1)y_j{'}(YY')^{-1}y_j=(n-1)x_j{'}\beta(\beta'XX'\beta)^{-1}\beta'x_j$

   $=(n-1)x_j{'}\beta\beta'x_j/\beta'XX'\beta$ (noting that $\beta'XX'\beta$ is 1×p×p×1, a scalar)

   $=(n-1)\beta'x_j\ x_j{'}\beta/\beta'XX'\beta$ (noting that $\beta'x_j$ and $x_j{'}\beta$ are both scalars and so commute).

   $U_j(\beta)$ is invariant under scalar multiplication of β so we can impose the [non-restrictive] constraint that the denominator $\beta'XX'\beta=1$. (i.e. we only need to look for solutions amongst those βs for which $\beta'XX'\beta=1$).

   Let $\Omega=(n-1)\beta'x_j\ x_j{'}\beta-\lambda(\beta'XX'\beta-1)$:

   $\partial\Omega/\partial\lambda=2(n-1)x_j\ x_j{'}\beta-2\lambda XX'\beta$ so we require

   $(n-1)(XX')^{-1}x_j\ x_j{'}\beta-\lambda\beta=0$…………………..(★)

   and so we need β to be the [right] eigenvector of

   $(n-1)(XX')^{-1}x_j\ x_j{'}=S^{-1}x_j\ x_j$ which is of rank 1 (since $x_j\ x_j{'}$is of rank 1) and so has only one non-zero eigenvalue.

ii) *If this eigenvector is β_j, shew that this maximum value $U_j(\beta_j)$ is equal to this non-zero eigenvalue.*

multiplying (★) by $\beta'XX'$ gives $U_j(\beta)=(n-1)\beta'x_j\, x_j'\beta=\lambda\beta'XX'\beta=\lambda$

iii) *Shew that $u_j=U_j(\beta_j)$.*

iv) *Shew that the non-zero eigenvalue of $S^{-1}x_jx_j'$ is $x_j'S^{-1}x_j$ and the corresponding eigenvector is proportional to $S^{-1}x_j$*

$[S^{-1}x_j][x_j'(S^{-1}x_j)]=S^{-1}x_j(x_j'S^{-1}x_j)=[(x_j'S^{-1}x_j)][S^{-1}x_j]$ so $S^{-1}x_jx_j'$ has eigenvalue $\lambda=x_j'S^{-1}x_j$ and eigenvector proportional to $S^{-1}x_j$. Already shewn $U_j(\beta)=\lambda$.

Notes & Solutions to Exercises 2

1) *The data given in file dogmandibles.∗ (in various formats) are extracted, via Manly (1994), from Higham etc (1980), J.Arch.Sci, 149–165. The file contains 9 measurements of various dimensions of the mandibles of 5 canine species as well as records of the sex and the species, eleven variables in total. These are*

X_1: length of mandible

X_2: breadth of mandible

X_3: breadth of articular condyle

X_4: height of mandible below first molar

X_5: length of 1st molar

X_6 breadth of 1st molar

X_7: length between 1st to 3rd molar inclusive (1st to 2nd for Cuons)

X_8: length between 1st to 4th premolar inclusive

X_9: breadth of lower canine

X_{10}: gender (1 ≡ male, 2 ≡ female, 3 ≡ unknown)

X_{11}: species (1 ≡ modern dog from Thailand, 2 ≡ Golden Jackal,
3 ≡ Cuon, 4 ≡ Indian Wolf, 5 ≡ Prehistoric Thai dog)

All measurements are in mm; molars, premolars and canines are types of teeth; an articular condyle is the round knobbly bit in a joint; a Cuon, or Red Dog, is a wild dog indigenous to south east Asia and notable for lacking one pair of molars.

i) Ignoring the group structure, what interpretations can be given to the first two principal components?

Step 1 is to perform a PCA on the linear measurements for the complete data set (i.e. all 5 groups). Initial inspection shews (but not given here) that the standard deviations of the measurements vary widely — this is inevitable given that X_1 has values in the 100s and X_9 below 10 — so basing the PCA on the correlation matrix is preferable. (PCA on the covariance matrix gives the first eigenvalue as 0.956, with subsequent ones 0.027 and below, and first PC heavily dominated by X_1, however the overall conclusions on the PCs are much the same but less clear-cut.)

R Analysis:

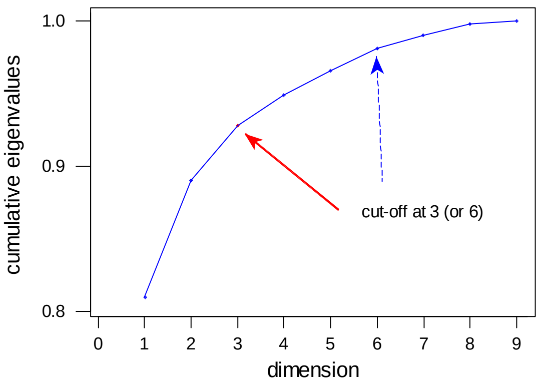
```
> attach(dogmandibles)
> dogmandibles[1:5,]
  length breadth condyle.breadth height molar.length molar.breadth
1    123    10.1              23     23           19           7.8
2    127     9.6              19     22           19           7.8
3    121    10.2              18     21           21           7.9
4    130    10.7              24     22           20           7.9
5    149    12.0              25     25           21           8.4
  first.to.3rd.length first.to.4th.length canine.breadth gender species
1                  32                  33            5.6      1       1
2                  32                  40            5.8      1       1
3                  35                  38            6.2      1       1
4                  32                  37            5.9      1       1
5                  35                  43            6.6      1       1
> dog.pc<-princomp(dogmandibles[,-c(10,11)],cor=T)
>
> summary(dog.pc)
Importance of components:
                          Comp.1     Comp.2     Comp.3     Comp.4     Comp.5
Standard deviation     2.6993793 0.85254056 0.58404915 0.43677899 0.38952230
Proportion of Variance 0.8096276 0.08075838 0.03790149 0.02119732 0.01685862
Cumulative Proportion  0.8096276 0.89038602 0.92828751 0.94948483 0.96634345
                           Comp.6      Comp.7      Comp.8      Comp.9
Standard deviation     0.35707481 0.296851411 0.262761145 0.135064109
Proportion of Variance 0.01416694 0.009791196 0.007671491 0.002026924
Cumulative Proportion  0.98051039 0.990301585 0.997973076 1.000000000
> print(dog.pc$loadings,digits=1)

Loadings:
                    Comp.1 Comp.2 Comp.3 Comp.4 Comp.5 Comp.6 Comp.7 Comp.8
length              -0.4   -0.1   -0.3           0.2           0.1
breadth             -0.3    0.3    0.2    0.2   -0.3    0.4    0.2   -0.6
condyle.breadth     -0.3    0.3   -0.7          -0.3   -0.2    0.2
height              -0.3    0.4    0.2    0.6    0.3   -0.2   -0.3    0.3
molar.length        -0.3   -0.1          -0.4   -0.2    0.3   -0.7    0.1
molar.breadth       -0.3           0.4   -0.4          -0.7          -0.2
first.to.3rd.length -0.3   -0.7           0.4   -0.4   -0.1           0.1
first.to.4th.length -0.3   -0.3   -0.2           0.7    0.1          -0.4
canine.breadth      -0.3           0.3   -0.3           0.3    0.5    0.6
```

Note that in **R** the standard deviations on each component are the square roots of the eigenvalues. The rest of these solutions will concentrate on the interpretation of plots. These have been produced in a different package but equivalent one can of course be produced in **R.**

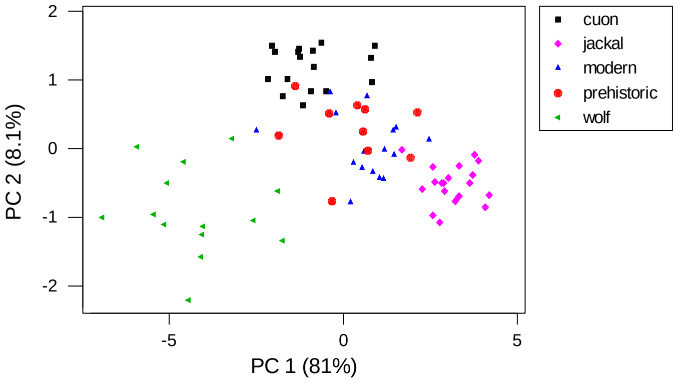
PC1 has coefficients all of the same sign and roughly the same magnitude. Thus low scores will be obtained (in this case, since the signs are all negative) by mandibles with all values of the variables which are large and there will be low values on PC1 when all variable are small. Thus PC1 reflects size and large mandibles will appear at the extreme negative end of the axis, small ones at the positive end.

PC2 has negative signs for X_1, X_5, X_7 & X_8 and positive signs for the other variables with X_6 & X_9 much smaller. These last two refer to the breadths of the 1st molar and the canine, so these are not important to PC2. Inspection of the variables reveals that those with negative signs are all lengths and those with positive signs are breadths and height so PC2 contrasts lengths with breadths and so can be interpreted as reflecting the ***shape*** of the mandible. [Aside: these interpretations of size and shape for linear measurements on physical objects are very common and are likely to be appropriate for high order PCs, though not necessarily the second one in the case of shape. This is not entirely for mathematical reasons, just the way the world is. One mathematical reason for size to be predominant is that linear measurements on objects are likely to be positively correlated – end of aside].

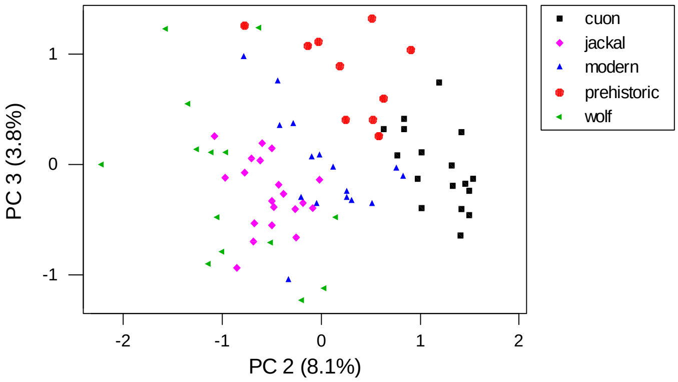
Although not asked for, the next steps given here for illustration were to produce a scree plot and plots on the PCs. This is provided here for comparison with the plots on crimcoords and is always a virtually vital step in any analysis for any purpose of multivariate data. The labels included the percentage of variation accounted for by that PC — a useful aid to the interpretation of the scatter plots which might alternatively be obtained by using equal scaling on the axes using **`eqscalplot()`** in the **`MASS`** library. The PC plots have the different groups distinguished, even though this information was ignored in the construction of the PCs.



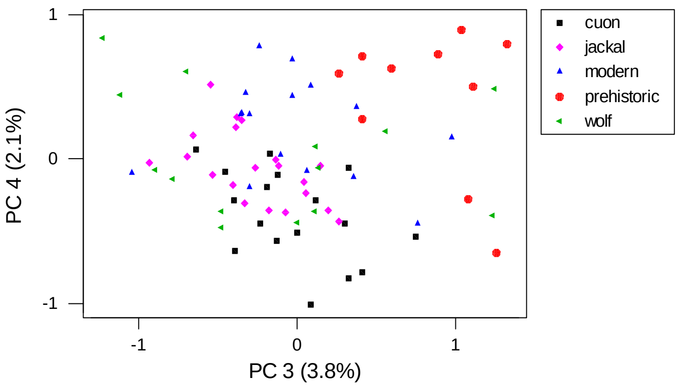
This plot shews that 3 PCs are adequate to capture most of the variation in the data.



The plot on the first two PCs displays 89% of the variation. It separates out the wolves to the left of the plot (i.e. they are ***bigger***) and the jackals to the right (i.e. they are ***smaller***) than the rest. Note that the prehistoric and modern dogs are near inseparable on this plot and that this plot displays most of the variation.



The plot on PCs 2 & 3 displays about 12% of the information. It separates the prehistoric (& the cuons) in the top right hand corner. To appear in the top r.h. corner cases have to have large values for those variables with positive coefficients on ***both*** PCs 2 & 3 and small values for those with negative coefficients on PCs 2 & 3, i.e. large values for X_2 and X_4 (ignoring any variable with a very small coefficient, even if positive) and small values for X_1 and X_8, i.e. prehistoric dogs and cuons have short ‘chunky’ mandibles.



The plot on PCs 3 & 4 shews that the prehistoric separates from most groups other than the modern dogs on the 4th PC, though this separation is very slight noting that PC4% contains only 2.1% of the variation. However, the fact that each of the groups is separated from the others on at least one of these plots suggests that it a discriminant analysis will be able to distinguish them.

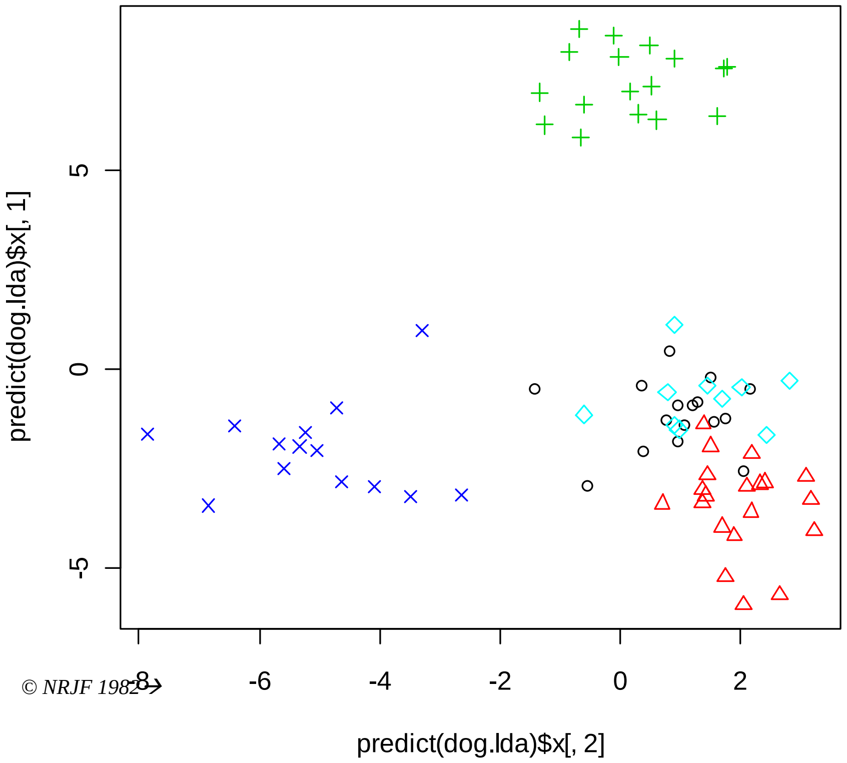
ii) Construct a display of the measurements on the first two crimcoords, using different symbols for the five different groups.

```
> library(MASS)
>
> dog.lda<-lda(species~length+breadth+condyle.breadth+height+
+ molar.length+molar.breadth+first.to.3rd.length+
+ first.to.4th.length+canine.breadth)
>
> print(dog.lda,digits=2)
Call:
lda(species ~ length + breadth + condyle.breadth + height +
molar.length +
    molar.breadth + first.to.3rd.length + first.to.4th.length
+
    canine.breadth)

Prior probabilities of groups:
   1    2    3    4    5
0.21 0.26 0.22 0.18 0.13

Group means:
  length   breadth   condyle.breadth   height   molar.length
molar.breadth
1      125       9.7                  21       21              19
7.7
2      111       8.2                  19       17              18
6.8
3      133      10.7                  24       24              21
8.5
4      157      11.6                  26       25              25
9.3
```

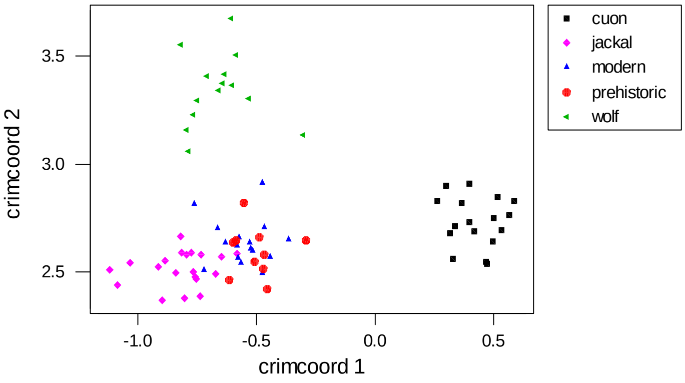
```
5      123      10.3                          20        23               19
8.2
  first.to.3rd.length first.to.4th.length canine.breadth
1                  32                  37            5.9
2                  30                  33            4.8
3                  29                  38            6.6
4                  40                  45            7.4
5                  33                  36            6.2

Coefficients of linear discriminants:
                       LD1    LD2    LD3    LD4
length               0.150 -0.027 -0.079 -0.015
breadth             -0.042  0.024  0.552  0.093
condyle.breadth     -0.347 -0.024 -0.087 -0.282
height               0.226  0.051  0.432  0.058
molar.length         0.885 -0.746 -1.131  0.680
molar.breadth        0.818  0.118  0.415  1.057
first.to.3rd.length -1.375 -0.181  0.338  0.018
first.to.4th.length -0.239 -0.090  0.014 -0.232
canine.breadth       1.512  0.487  1.279 -1.028

Proportion of trace:
   LD1    LD2    LD3    LD4
0.6539 0.2563 0.0859 0.0039
```

```
type<-unclass(species)
plot(predict(dog.lda)$x[,2],predict(dog.lda)$x[,1],
pch=type,col=type)
```

```
type<-unclass(species)
plot(predict(dog.lda)$x[,2],predict(dog.lda)$x[,1],
pch=type,col=type)
```



This basic plot needs to be enhanced with a legend and proper labelling of axes and this is done below (though produced in a different plotting package)

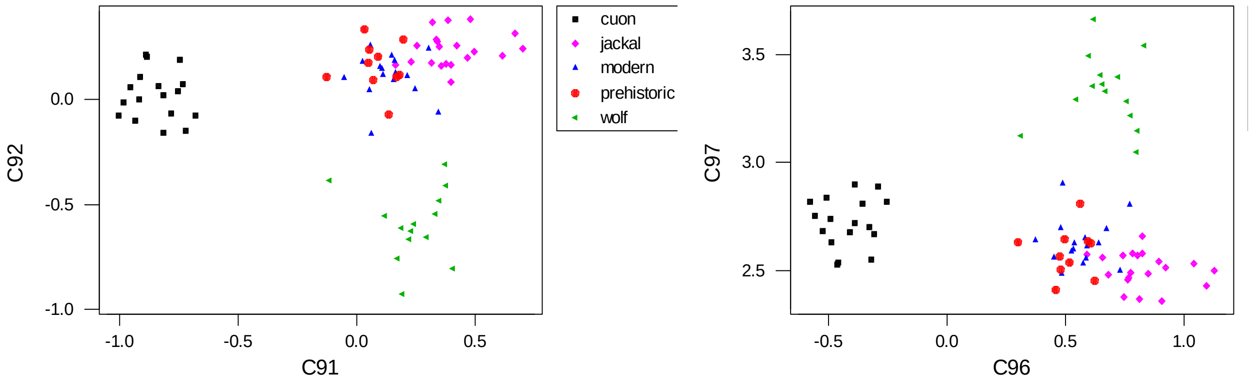


This plot shews clear separation of all the groups from each other with the exception of the modern and prehistoric dogs which are intermingled on this display.

iii) If the linear discriminant analysis were performed on the data after transformation to the full set of nine principal components what differences (if any) would there be in the plot on crimcoords and the eigenvalues and eigenvectors of the matrix $W^{-1}B$?

There are two ways of looking at this. One is to do it and see what happens, the other is to look mathematically, at least initially. Both are useful. However, on general principles, there should be no fundamental difference in the displays since a preliminary transformation to principal components is merely a rotation &/or a reflection of the data and no information is lost or gained. So plots on crimcoords after a PCA transformation should be expected to be essentially identical, up to perhaps a reflection. To get some idea mathematically, suppose the original data matrix is denoted by X′ and the matrix of eigenvectors (of either the covariance or the correlation matrix, whichever is used) is denoted by $A=(a_i)$. Then we know that since $a_i'a_i=1$ and $a_i'a_j=0$ for $i\neq j$ that $A'A = I_p$. The data referred to PCs is Y′ where $Y'=X'A$. If W and B are the within and between groups variances of the original data X′ then those of Y′ are $A'WA$ and $A'BA$ respectively. So the crimcoords of the data referred to PCs are the eigenvalues of $(A'WA)^{-1}$ $A'BA$, i.e. of $A^{-1}W^{-1}A'^{-1}A'BA = A'W^{-1}BA$. It is easy to see that the eigenvalues of this are identical to those of $W^{-1}B$. The original data referred to crimcoords are $X'U$ where U is the matrix of eigenvectors of $W^{-1}B$ and the [PCA transformed-]data referred to crimcoords after the PCA transformation are $Y'V=X'AV$ where V is the matrix of eigenvectors of $A'W^{-1}BA$. It can be shewn (but not here) that these differ only in scale and an arbitrary sign difference.

The try it and see approach is straightforward and is not given in detail here. The plots below are again in a different package and axes are not labelled (since this is just for a quick verification that nothing is essentially changed). It can be see that the three plots are essentially identical except for [arbitrary] reflections.



iv) Which group is separated from the other four by the first crimcoord?

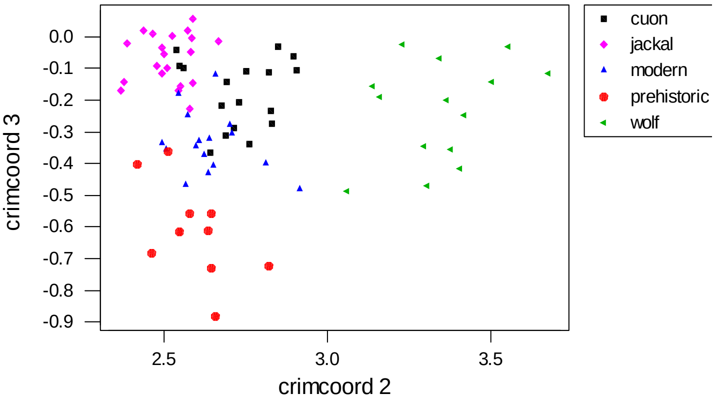
Cuons

v) Which group is separated from the other four by the second crimcoord?

Wolves

vi) Which group is separated from the other four by the third crimcoord?

Need a plot of the third crimcoord:



This shews that the third crimcoord separates the prehistoric dogs from the rest. Further, the third in conjunction with the second separates the jackals from the rest.

vii) What features of the mandibles provide discrimination between the various species?

The first three crimcoords (from the original data) are

```
                        LD1    LD2    LD3
length                0.150 -0.027 -0.079
breadth              -0.042  0.024  0.552
condyle.breadth      -0.347 -0.024 -0.087
height                0.226  0.051  0.432
molar.length          0.885 -0.746 -1.131
molar.breadth         0.818  0.118  0.415
first.to.3rd.length -1.375 -0.181  0.338
first.to.4th.length -0.239 -0.090  0.014
canine.breadth        1.512  0.487  1.279
```

High scores on crimcoord 1 are obtained by those cases which have big teeth, long mandibles and short distances between molars and premolars. These characteristics distinguish cuons from the others.

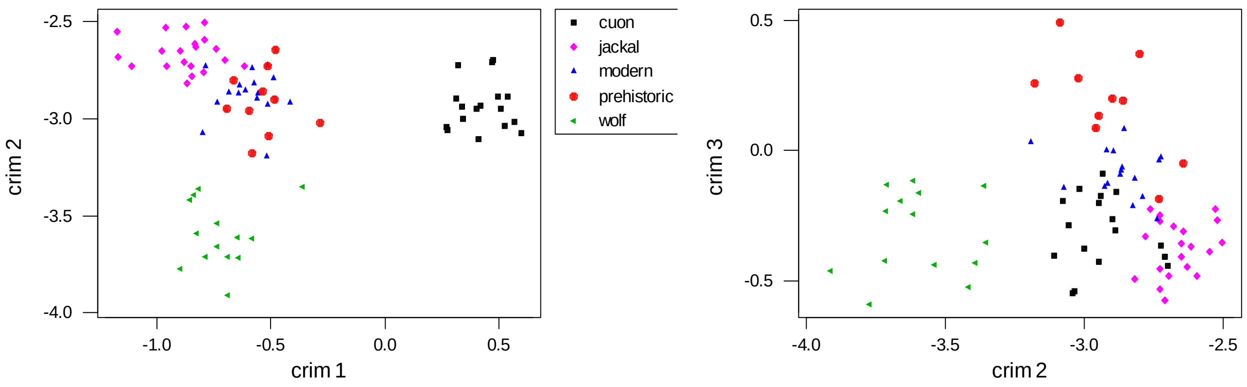
High scores on crimcoord 2 are obtained by long narrow mandibles with long narrow teeth, these characteristics distinguish wolves from the other species [rather more specifically than just overall size as was deduced from the PCA above].

Low scores on crimcoord 3 are obtained by short broad (i.e. ‘chunky’) molars, short broad (‘chunky’) mandibles, and longer distances between molars. These features distinguish the prehistoric dogs from the others. [note that this is again a little more specific than obtained from just the PCA].

2) *⋆ The question of prime interest in the study of canines was related to an investigation of the origin of the prehistoric dogs. Try calculating the discriminant analysis based on the four groups of modern canines and then plot the prehistoric cases on the same coordinate system a (c.f. informal data classification method (iii) on p140 of course notes) and seeing to which of the modern groups the majority of the prehistoric are closest.*

   *(The interpretation of the results of this exercise are **within** the scope of MAS465; the required computer skills to produce it are useful but a little beyond the scope of PA4370, i.e. if you do not attempt it ensure that you look carefully at the printed solution in due course.)*

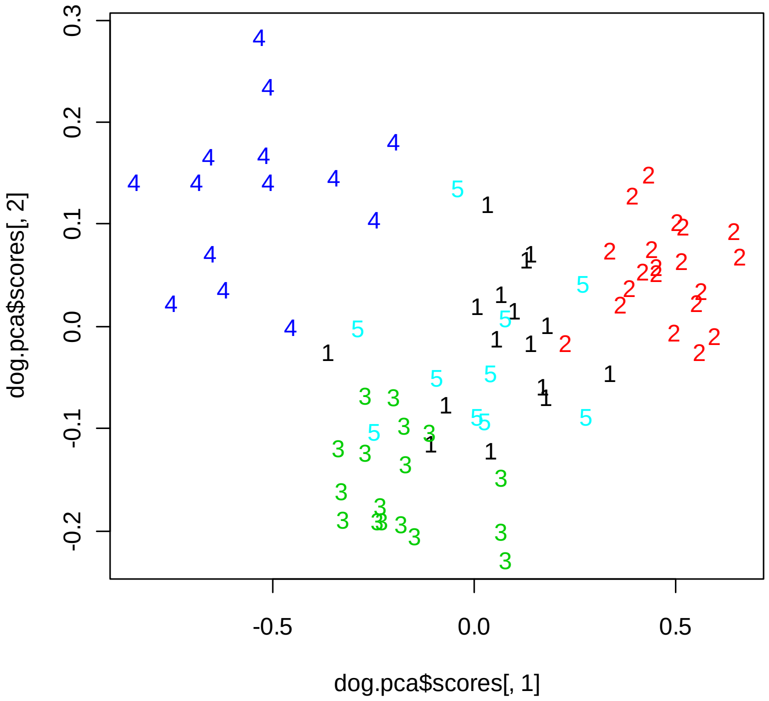
Below are plots produced in a different package but code to do the equivalent in R is given later



This shews that the prehistoric are superimposed on the modern on crimcoords 1 & 2 but there is a distinction on crimcoord 3.

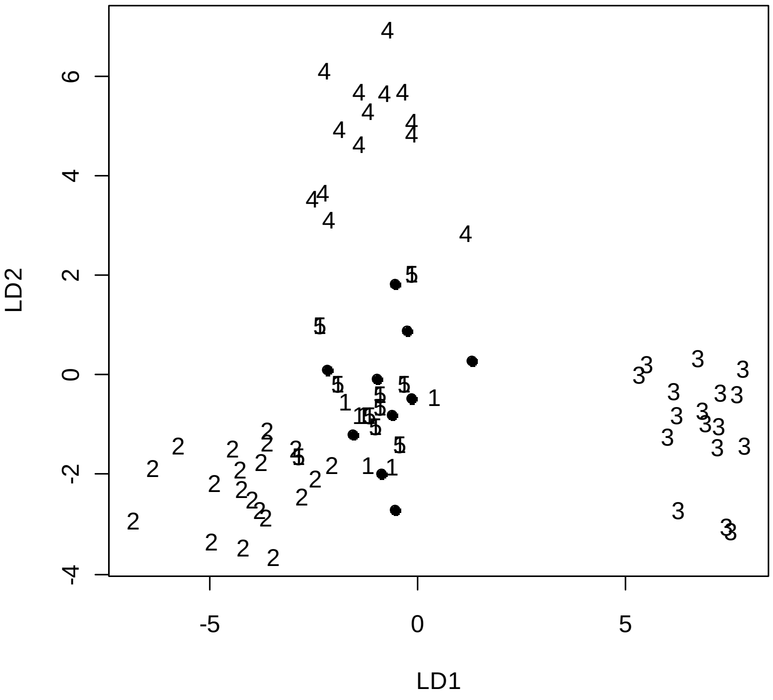
Below is guidance on producing the plots in **R**. One difficulty is that to add in points for the prehistoric samples onto existing plots on crimcoords for the modern dogs you may need to extend the plotting range o avoid trying to plot points outside the plotting area (hence use of the parameter **`ylim=c(.,.)`** below). Note also the removal from both the PCA and the LDA the columns 10 and 11 which are factors indicating gender and species. The code and examples below do not provide complete solutions to Q1 and Q2 but are intended as sufficient for you to adapt to your particular needs. Just for illustration and for comparison, the analyses below have been done after taking [natural] logs of all measurements.

```
> attach(dogmandibles)
> library(MASS)
> □
> dog.pca<-princomp(log(as.matrix(dogmandibles[-c(10,11)])))
>
> plot(dog.pca$scores[,1],dog.pca$scores[,2],type="n")
>
text(dog.pca$scores[,1],dog.pca$scores[,2],labels=species,col=
type)
>
```

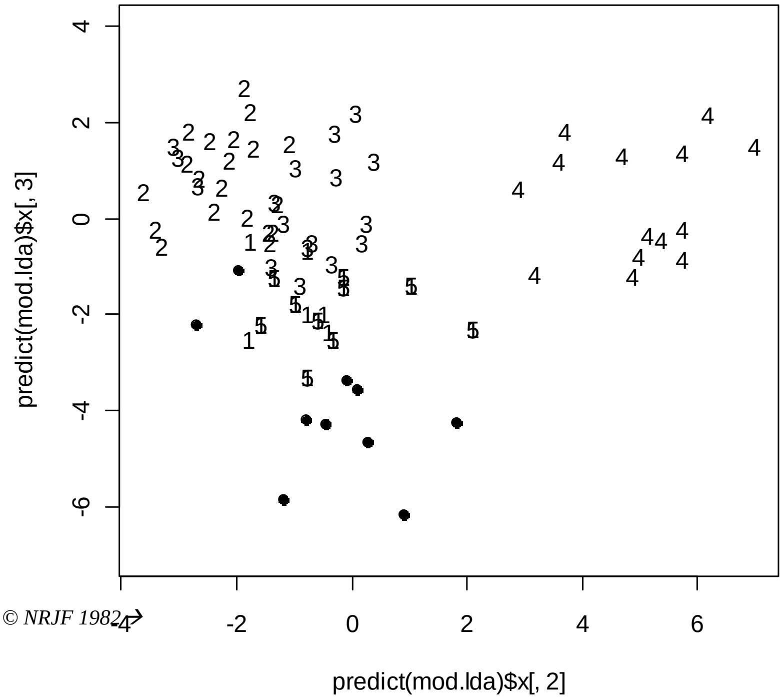


```
>
> mod <- log(as.matrix(dogmandibles[1:67, -c(10,11)]))
> pre <- log(as.matrix(dogmandibles[68:77, -c(10,11)]))
> spec<-species[1:67]
> mod.lda <- lda(mod, spec)
Warning message:
In lda.default(x, grouping, ...) : group 5 is empty
```

```
> plot(predict(mod.lda, dimen = 2)$x, type="n")
> text(predict(mod.lda)$x[,1],predict(mod.lda)$x[,2],
labels=species)
> points(predict(mod.lda,pre, dimen= 2)$x, pch=19)
>
```



```
> plot(predict(mod.lda)$x[,2],
predict(mod.lda)$x[,3],type="n",ylim=c(-7,4))
> text(predict(mod.lda)$x[,2],
predict(mod.lda)$x[,3],labels=tp)
> points(predict(mod.lda,pre)$x[,2],
predict(mod.lda,pre)$x[,3],pch=19)
>
```



Notes & Solutions to Exercises 3

1)

i) *Measurements of cranial length x_{11} and cranial breadth x_{12} on 35 female frogs gave* $\bar{x}'_1$*=(22.860, 24.397) and* $S_1 = \begin{pmatrix} 17.683 & 20.290 \\ * & 24.407 \end{pmatrix}$. T*est the hypothesis that* $\mu_{11}=\mu_{12}$.

Using the result from Task Sheet for week 8, Q2, illustrated Q3, we test $H_0 : C'\mu_1 = 0$ where $C' = (1,–1)'$ by comparing $35(22.860, 24.397)(1,–1)'[(1,–1)S_1(1,–1)']^{–1}(1,–1)(22.860, 24.397)'$ with $T^2(1,34)$, i.e. $35\times(–1.537)\times1.51^{–1}\times(–1.537) = 54.75$ and compare with $(34 – 1 + 1)/34\times1\times54.75$ with $F_{1,34–1+1}$ i.e. 7.4 with t_{34} and conclude that there is very strong evidence that the cranial lengths and breadths of female frogs are different.

ii) *Similar measurements on 14 male frogs gave* $\bar{x}'_2$*=(21.821, 22.843) and* $S_2 = \begin{pmatrix} 18.479 & 19.095 \\ * & 20.756 \end{pmatrix}$.

Calculate the pooled variance matrix for male & female frogs and test the hypothesis that female & male frogs come from populations with equal mean vectors.

Pooled variance matrix is

$(34\times S_1+13\times S_2)/47 = \begin{pmatrix} 17.903 & 19.959 \\ * & 23.397 \end{pmatrix}$ = S (say).

Now $S^{-1} = \begin{pmatrix} 1.140 & –0.973 \\ * & 0.873 \end{pmatrix}$

Hotelling's T^2 is $[35\times14/49]\times(1.039,\ 1.554)S^{-1}(1.039,\ 1.554)'$ = 1.968 and we compare this with $T^2(2,47) = 2.0434F_{2,46}$, i.e. compare 0.963 with $F_{2,46}$ and we conclude that the data give no evidence of a difference between the sizes of skulls of Male and Female frogs.

2) *Using you favourite computer package, access the British Museum Mummy Pots data (see task sheet for week 4) and calculate the two shape variables '`taper`' and '`point`'.*

Do the two batches of pots differ in overall shape as reflected by the calculated shape measures '`taper`' and '`point`'?

```
> attach(brmuseum)
> library(MASS)
> batch=factor(batch)
> taper=(rim.cir-base.circ)/length
> point=rim.cir/base.circ
> shape.manova=manova(cbind(taper,point)~batch)
> summary(shape.manova)
          Df  Pillai approx F num Df den Df Pr(>F)
batch      1 0.10362  1.27156      2     22 0.3002
Residuals 23
> summary(shape.manova,test="Hotelling-Lawley")
          Df  Hotelling-Lawley  approx  F  num  Df  den  Df
Pr(>F)
batch      1       0.1156   1.2716      2     22 0.3002
Residuals 23
```

You can see that the p-values for all of the tests are 0.3 (since only two groups all the tests are functionally related to each other and so equivalent. The value of Hotelling's T^2 is $23\times0.1156=2.6588$ and if you look at the corresponding p-value it is 0.300 (of course!). Note that there is one missing value and so this entire pot has been excluded from the analysis, leaving only 25 pots. The answer to the questions is no, there is no significant evidence that the pots differ in shape.

i) Do the two batches of pots differ in overall size?

```
>
size.manova=manova(cbind(length,rim.cir,base.circ)~batch)
> summary(size.manova)
          Df Pillai approx F num Df den Df   Pr(>F)
batch      1 0.4323   5.3301      3     21 0.006877 **
Residuals 23
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1
‘ ’ 1
> summary(size.manova,test="Hotelling-Lawley")
          Df  Hotelling-Lawley  approx  F  num  Df  den  Df
Pr(>F)
batch      1             0.7614   5.3301       3      21
0.006877 **
Residuals 23
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1
‘ ’ 1
```

Yes, all p-values are 0.007 and so conclude that there is very strong evidence of a difference in overall size between the batches.

In this two group case there is no actual advantage in looking at both the Pilai trace (the **R** default) and the Hotelling-Lawley statistics since they have precisely the same significance. In general you should decide which statistics you are going to base your inference on and you should definitely not choose the one with the most significant result. In most cases there should be general agreement between the available statistics (p-values differing only marginally); if there is a substantial difference then it indicates something most unusual and unexpected about the data which is worth investigating — it may mean that you have outliers or some other form of non-normality indicating our model is not appropriate.

ii) *Without doing any calculations,*

a) *would your answer to (ii) be different in any respect if you used the scores on the three PCs calculated from the size variables?*

b) *Would it make any difference were you to calculate the PCs using the correlation matrix instead of the covariance matrix?*

Since the PCs (provided you take all of them) are just a linear transformation of the data (whether the matrix of eigenvectors is calculated from the covariance or correlation matrix) there should be no difference in the results on using the PCs. If not convinced then look at the following:

```
> size.pc<-princomp(cbind(length,rim.cir,base.circ))
> sizepc.manova=manova(size.pc$scores~batch)
> summary(sizepc.manova,test="Hotelling-Lawley")
          Df  Hotelling-Lawley  approx  F  num  Df  den  Df
Pr(>F)
batch      1            0.7614   5.3301       3      21
0.006877 **
Residuals 23
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1
' ' 1
>
```

3) *$x_1,\ldots,x_n$ are independent measurements of $N_p(\mu,\sigma^2I_p)$*

i) *Shew that the maximum likelihood estimate of μ, subject to $\mu'\mu = r_0^2$ (a known constant) is the same whether σ is known or unknown.*

This example is very like example 5.5.3 in the lecture notes:

We have $\ell(\mu; X)= -½(n-1)\mathrm{trace}(S\sigma^{-2}) - ½n(\bar{x}-\mu)'(\bar{x}-\mu)\sigma^{-2} -$

$$½np\log(2\pi) -½np\log(\sigma^2)$$

Let $\Omega=\ell(\mu)-\lambda(\mu'\mu-r_0^2)$ then $\frac{\partial\Omega}{\partial\mu} = n(\bar{x}-\mu)\sigma^{-2} - 2\lambda\mu$.

So we require $\hat{\mu} = \frac{n\bar{x}}{n+2\lambda\sigma^2}$ then $\mu'\mu=r_0^2$ implies $(n+2\lambda\sigma^2)^2r_0^2=n^2\bar{x}'\bar{x}$

and so $\hat{\mu} = \frac{\bar{x}r_0}{\sqrt{\bar{x}'\bar{x}}}$ which does not depend on σ^2.

ii) *Find the maximum likelihood estimate of σ when neither μ nor σ are known.*

$$\tfrac{\partial\Omega}{\partial\sigma} = (n-1)\mathrm{tr}(S)\sigma^{-3} + n(\overline{x}-\mu)'(\overline{x}-\mu)\sigma^{-3} - np\sigma^{-1}$$

so $\hat{\sigma} = \sqrt{\tfrac{1}{np}\left[(n-1)\mathrm{tr}(S) + n(\overline{x}-\hat{\mu})'(\overline{x}-\hat{\mu})\right]}$

$$= \sqrt{\tfrac{1}{np}\left[(n-1)\mathrm{tr}(S) + n(\sqrt{\overline{x}'\overline{x}} - r_0)^2\right]} = \sqrt{\tfrac{1}{np}\left[\Sigma x_i'x_i - 2nr_0\sqrt{\overline{x}'\overline{x}} + nr_0^2\right]}$$

iii) *Hence, in the case when $\sigma = \sigma_0$ (a known constant) construct the likelihood ratio test of $H_0 : \mu'\mu = r_0^2$ vs $H_A : \mu'\mu \neq r_0^2$ based on n independent observations of $N_p(\mu, \sigma_0^2 I_p)$.*

Under H_0

$$\ell_{\max} = K - ½n(\sqrt{\bar{x}'\bar{x}} - r_0)^2\sigma_0^{-2}$$

Under H_A we have

$$\hat{\mu} = \bar{x} \quad \text{so} \quad \ell_{\max} = K$$

so LRT statistic is $n(\sqrt{\bar{x}'\bar{x}} - r_0)^2\sigma_0^{-2}$ and under H_0 this $\sim \chi_1^2$

[1 d.f. since p parameters in μ estimated under H_A and p with 1 constraint under H_0]

iv) *In an experiment to test the range of a new ground-to-air missile thirty-nine test firings at a tethered balloon were performed and the three dimensional coordinates of the point of ignition of the missile's warhead measured. These gave a mean result of (0.76, 0.69, 0.66)′ relative to the site expressed in terms of the target distance. Presuming that individual measurements are independently normally distributed with unit variance, are the data consistent with the theory that the range of the missile was set correctly?*

We have σ_0=1=r_0 and so

$$n(\sqrt{\bar{x}'\bar{x}} - r_0)^2\sigma_0^{-2} = 39(\sqrt{}(0.76, 0.69, 0.66)'\,(0.76, 0.69, 0.66) - 1)^2$$

= 1.894 (<<3.84=$\chi^2_{1;0.95}$) and so yes, the data are consistent with the theory that the range was set correctly

APPENDICES

The following pages contain various appendices.

Appendix 0 gives some background notes on some mathematical techniques and statistical methods which some people may not have seen before. **Everybody** should make sure that they are sufficiently acquainted with the material in this appendix so that they know where to look up references to properties of eigenvalues and eigenvectors, differentiation with respect to vectors, maximum likelihood estimation and likelihood ratio tests.

The other appendices are provided because they contain useful material for people involved in practical work both on courses in this University and in the future. Many of the techniques are relatively new and are still under development — they are tailored to people using **R** which is a **FREE** package downloadable from the web and is almost identical to S-PLUS. S-PLUS users will have little difficulty in converting the examples below to run in S-PLUS.

A full list of sites providing **R** can be found at

http://www.ci.tuwien.ac.at/R/mirrors.html

and the most local one is at http://cran.uk.r-project.org/

As well as the book by Venables & Ripley, a useful book recommended for additional reading about **R** is Nolan, D. & Speed, T. P. (2000), Stat Labs: Mathematical Statistics Through Applications. Springer. Support material is available at: http://www.stat.berkeley.edu/users/statlabs

It is emphasized that this further additional material that is **NOT** part of PAS470, nor part of the examined part of PAS6011. It has been provided because it is useful for practical studies and some people may find that they need to use these methods in the assessed projects in e.g. PAS354 and for the assessed project for PAS6011 and perhaps the MSc summer dissertation. Some of the topics are just mentioned on an 'awareness' level (e.g. Correspondence Analysis), for others (e.g. Neural Networks) there are detailed notes on how to run the analyses with commentaries on the examples.

APPENDIX 0: Background Results

A0.1 Basic Properties of Eigenvalues & Eigenvectors

Let A be a real p×p matrix.

The eigenvalues of A are the roots of the p-degree polynomial in λ:

$$q(\lambda)= \det(A - \lambda I_p)=0 \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots * $$

Let these be $\lambda_1,\lambda_2,\ldots\ldots\lambda_p$. Then, since the coefficient of λ^p in equation $*$ is $(-1)^p$ we have $q(\lambda)=\prod_{i=1}^{p}(\lambda_i - \lambda) \ldots\ldots\ldots\ldots\ldots\ldots **$

1. Comparing coefficients of λ^{p-1} in $*$ and $**$ gives

$$\sum_{i=1}^{p}\lambda_i = \text{trace}(A) = \sum_{i=1}^{p} a_{ii}$$

2. Putting λ=0 in $*$ and $**$ gives

$$\prod_{i=1}^{p}\lambda_i = \det(A) \ = |A|$$

Since the matrices $A - \lambda_i I_p$ are singular (i.e. have zero determinant) there exist vectors x_i called the **eigenvectors** of A such that

$$(A - \lambda_i I_p)x_i=0, \text{ i.e. } Ax_i - \lambda_i x_i=0.$$

[Strictly, if A is non-symmetric, the x_i are right-eigenvectors and we can define left-eigenvectors y_i such that $y_iA - \lambda_i y_i=0$]

3. Suppose C is any p×p non-singular square matrix, since $|A - \lambda I_p|=|C||A - \lambda I_p||C^{-1}|=|CAC^{-1} - \lambda I_p|$ we have:

   A and CAC^{-1} have the same eigenvalues.

4. If $Ax_i=\lambda_i x_i$ then $(CAC^{-1})(Cx_i)=\lambda_i(Cx_i)$ so the eigenvectors of CAC^{-1} are Cx_i

5. If A is n×p and B is p×n then

   $|AB - \lambda I_n|= (-\lambda)^{n-p}|BA - \lambda I_p|$ so the non-zero eigenvalues of AB and BA are identical.

6. Since, if $ABx_i=\lambda x_i$ then $(BA)(Bx_i)= \lambda(Bx_i)$, we have that the eigenvectors of BA are obtained by premultiplying those of AB by B.
7. Suppose now that A is symmetrical, i.e. $A=A'$, we can show that the eigenvalues of A are *real*, since suppose λ_i and x_i are the eigenvalues and vectors and that $\lambda_j=\mu_j+i\nu_j$, $x_j=y_j+iz_j$ then equating real and imaginary parts of $Ax_j=\lambda_j x_j$ gives

   $Ay_j=\mu_j y_j - \nu_j z_j$ ....∗∗∗ and $Az_j=\nu_j y_j+\mu_j z_j$ ....∗∗∗∗

   Premultiplying ∗∗∗ by z_j' and ∗∗∗∗ by y_j' and noting $z_j'Ay_j=(z_j'Ay_j)'$ (since it's a scalar)$=y_j'A'_j= y_j'Az_j$ (since A is symmetric by presumption) and subtracting the two equations gives the result.
8. Suppose again A is symmetric and that λ_j and λ_k are distinct eigenvalues with corresponding eigenvectors x_j and x_k. Then $Ax_j=\lambda_j x_j$ and $Ax_k=\lambda_k x_k$. Premultiplying these by x_k' and x_j' respectively and noting that $x_j'Ax_k=x_k'Ax_j$ since A is symmetric gives

   $(\lambda_j - \lambda_k)x_j'x_k=0$; since $\lambda_j\neq\lambda_k$ (by presumption) gives $x_j'x_k=0$,

   i.e. eigenvectors with distinct eigenvalues are orthogonal.

SUMMARY

$p\times p$ matrix A with eigenvalues $\lambda_1,...,\lambda_p$ and [right] eigenvectors $x_1,...,x_p$ then

1. $\sum_{i=1}^{p}\lambda_i = \text{trace}(A)$	5. AB and BA have identical non-zero eigenvalues.
2. $\prod_{i=1}^{p}\lambda_i = \det\|A\|$	6. Eigenvectors of BA = B ∗ those of AB
3. A and CAC^{-1} have identical eigenvectors for C non-singular	7. A symmetric ⇒ eigenvalues real
4. Eigenvalues of CAC^{-1} are Cx_i	8. A symmetric ⇒ eigenvectors corresponding to distinct eigenvalues are orthogonal.

A0.2 Differentiation With Respect to Vectors

A) if $x=(x_1,x_2,...,x_p)'$ is a p-vector and f=f(x) is a scalar function of x, we **define** $\partial f/\partial x$ to be the vector $(\partial f/\partial x_1, \partial f/\partial x_2, \ldots, \partial f/\partial x_p)'$.

B) Quadratic Forms

If $f(x)=x'Sx$, where S is a symmetric p×p matrix then $\partial f/\partial x=2Sx$.

Justification by example:

Case p=1: i.e. $x=(x_1)$, $S=(s_{11})$, $f(x)=x_1s_{11}x_1=x_1^2s_{11}$

$$\partial f/\partial x=2s_{11}x_1=2Sx$$

Case p=2: i.e. $x=(x_1,x_2)'$, $S=\begin{pmatrix} s_{11} & s_{12} \\ s_{12} & s_{22} \end{pmatrix}$,

$$\text{then } x'Sx=x_1^2s_{11}+2x_1x_2s_{12}+x_2^2s_{22}$$

$$\partial f/\partial x=(\partial f/\partial x_1, \partial f/\partial x_2)' = ((2x_1s_{11}+2x_2s_{12}), (2x_1s_{12}+2x_2s_{22}))'$$

$$= 2Sx.$$

General p: straightforward but tedious.

C) If S is not symmetric then $\partial(x'Sx)/\partial x = (S+S')x$

D) Special case: $S=I_p$, $\partial x'x/\partial x=2x$

E) Inner-products with scalars:

if $f(x)=a'x$ then $\partial f/\partial x = a$ (obvious)

A0.3 Lagrange Multipliers

Suppose $x=(x_1,...,x_n)'$. To maximize/minimize f(x) (a scalar function of x) subject to k scalar constraints $g_1(x)=0$, $g_2(x)=0,...,g_k(x)=0$ where k<n we define $\Omega = f(x) + \sum_{j=1}^{k}\lambda_j g_j(x)$ and max/minimize Ω with respect to the n+k variables $x_1,...,x_n$, $\lambda_1,...,\lambda_k$.

Proof: Omitted, other than 'by example'.

e.g. (1): $x=(x_1,x_2)'$, $f(x)=x'x=x_1^2+x_2^2$; 1 constraint $x_1+x_2=1$.

i.e. minimize $x_1^2+x_2^2$ subject to $x_1+x_2=1$.

Let $\Omega = x_1^2+x_2^2 +\lambda(x_1+x_2-1)$,

$\partial\Omega/\partial x_i = 2x_i+\lambda$ $(i=1,2)$, $\partial\Omega/\partial\lambda = x_1+x_2-1$. Setting these derivatives to zero yields $x_1=-\lambda/2$, $x_2=-\lambda/2$, $x_1+x_2=1$, so $\lambda=-1$ and solution is $x_{1opt}=+½$

CHECK: Substitute for x_2: $x_2=1-x_1$, $f(x)= x_1^2 + (1-x_1)^2$,

$\partial f/\partial x_1 = 2x_1 - 2(1-x_1)$ and so $x_{1opt} = +½$ ($= x_{2opt}$).

e.g. (2): Suppose $t_1,...,t_n$ are unbiased estimates of θ with variances $\sigma_1^2,...,\sigma_n^2$: to find the best linear unbiased estimate of θ. Let $\tau=\Sigma\alpha_i t_i$. We want to choose the α_i so that τ has minimum variance subject to the constraint of being unbiased. Now E[]=θ all i, so E[τ]=θ, so we have the constraint $\Sigma\alpha_i=1$. Also var(τ)= $\Sigma\alpha_i^2\sigma_i^2$. Let $\Omega=\Sigma\alpha_i^2\sigma_i^2 +\lambda(\Sigma\alpha_i -1)$:

$\partial\Omega/\partial\alpha_i = 2\alpha_i\sigma_i^2+\lambda$: $\partial\Omega/\partial\lambda = \Sigma\alpha_i - 1$.

So $\alpha_i=-½\lambda/\sigma_i^2$, so $\Sigma½\lambda/\sigma_I^2=-1$, so $\lambda = -\dfrac{1}{\sum 1/2\sigma_i^2}$ and so

$\alpha_i = \frac{1}{\sigma_i^2}\left(\sum\frac{1}{\sigma_i^2}\right)^{-1}$ and the BLUE estimate of θ is $\hat{\theta} = \left(\sum t_i/\sigma_i^2\right)\Big/\left(\sum 1/\sigma_i^2\right)$

A0.4 Maximum Likelihood Estimation

Suppose $x_1,\ldots,x_n$ are n independent observations of a random variable X which has density function $f(.;\theta)$ depending on an unknown parameter θ. There are various methods of estimating θ from the observations $x_1,\ldots,x_n$ — such as the method of least squares, the method of moments, the method of minimum chi-squared, ….. etc. The most central method in statistical work is ***the method of maximum likelihood***. The procedure is to calculate the ***the likelihood of θ for the data*** which is essentially 'the probability of observing the data $x_1,\ldots,x_n$' (this probability will be a function of the unknown parameter θ). Then we maximize this w.r.t. θ — the value of θ which maximizes the likelihood is the ***maximum likelihood estimate of θ***.

A0.4.1 Definition:

The likelihood of θ for data $x_1,\ldots,x_n$ is

$L(\theta;x_1,\ldots,x_n)=f(x_1;\theta)f(x_2;\theta)\ldots.f(x_n;\theta)$ if X is continuous

or $L(\theta;x_1,\ldots,x_n)=P[X=x_1;\theta]P[X=x_2;\theta]\ldots.P[X=x_n;\theta]$ if X is discrete

(i.e. it is the product of the values of the density function or probability function evaluated at each of the observations — it is the 'probability' of observing the data just obtained).

A0.4.2 Examples:

(all with data $x_1,\ldots,x_n$)

(i) $X\sim N(\mu,1)$: $f(x;\mu)=(2\pi)^{-½}\exp\{-½(x-\mu)^2\}$

$L(\mu;x_1,\ldots,x_n)=(2\pi)^{-½n}\exp\{-½\Sigma(x_i-\mu)^2\}$

(ii) $X\sim Ex(\lambda)$: $f(x;\lambda)=\lambda e^{-\lambda x}$

$L(\lambda;x_1,\ldots,x_n)=\lambda^n\exp\{-\lambda\Sigma x_i\}$

(iii) $X\sim Bin(m,p)$: $P[X=x]={}^mC_xp^x(1-p)^{m-x}$

$$L(p;\,x_1,\ldots,x_n)=\prod_{i=1}\binom{m}{x_i}p^{\Sigma x_i}(1-p)^{\Sigma(m-x_i)}$$

(iv) $X\sim N(\mu,\sigma^2)$: $f(x;\mu,\sigma)=(2\pi)^{-½}\sigma^{-1}\exp\{-½(x-\mu)^2/\sigma^2\}$

$L(\mu,\sigma;x_1,\ldots,x_n)=(2\pi)^{-½n}\sigma^{-n}\exp\{-½\Sigma(x_i-\mu)^2/\sigma^2\}$

(note that in this example the parameter $\theta=(\mu,\sigma)$ has two components)

(v) $X\sim Po(\lambda)$: $P[X=x]=\lambda^xe^{-\lambda}/x!$

$L(\lambda;x_1,\ldots,x_n)=\lambda^{\Sigma x}e^{-n\lambda}/\Pi x_i!$

To obtain the maximum likelihood estimates of the parameters in these cases we maximize the likelihoods w.r.t the unknown parameters. Generally it is often simpler to take the [natural] logarithms of the likelihood and maximize the log-likelihood (if the log is maximized then obviously the original likelihood will be maximized).

(i) $L(\mu;x_1,\ldots,x_n)=(2\pi)^{-\frac{1}{2}n}\exp\{-\tfrac{1}{2}\Sigma(x_i-\mu)^2\}$

$\log(L(\mu))=\ell(\mu)=-\tfrac{1}{2}n\log(2\pi)-\tfrac{1}{2}\Sigma(x_i-\mu)^2$

$\partial\ell/\partial\mu=\Sigma(x_i-\mu)$ and setting this to zero gives $\Sigma x_i=n\mu$ so $\hat{\mu}=\overline{x}$

(the hat ^ on the parameter indicates
that it is the estimate of the parameter)

(ii) $L(\lambda;x_1,\ldots,x_n)=\lambda^n\exp\{-\lambda\Sigma x_i\}\log(L(\lambda))=\ell(\lambda)=n\log(\lambda)-\lambda\Sigma x_i$

$\partial\ell/\partial\lambda=n/\lambda-\Sigma x_i$ and so $\hat{\lambda}=\frac{1}{\overline{x}}$

(iii) $L(p;\, x_1,\ldots,x_n)=\prod_{i=1}\binom{m}{x_i}p^{\Sigma x_i}(1-p)^{\Sigma(m-x_i)}$

$\log(L(p))=\ell(p)=\Sigma x_i\log(p)+\Sigma(m-x_i)\log(1-p)+K$

(K a constant not involving p)

$\partial\ell/\partial p=\Sigma x_i/p-\Sigma(m-x_i)/(1-p)$ and so $\hat{p}=\overline{x}/m$

(iv) $L(\mu,\sigma;x_1,\ldots,x_n)=(2\pi)^{-\frac{1}{2}n}\sigma^{-n}\exp\{-\tfrac{1}{2}\Sigma(x_i-\mu)^2/\sigma^2\}$

$\log(L(\mu,\sigma))=\ell(\mu,\sigma)=-\tfrac{1}{2}n\log(2\pi)-n\log(\sigma)-\tfrac{1}{2}\Sigma(x_i-\mu)^2/\sigma^2$

so $\partial\ell/\partial\mu=\Sigma(x_i-\mu)/\sigma^2$ and $\partial\ell/\partial\sigma=-n/\sigma+\Sigma(x_i-\mu)^2/\sigma^3$

giving $\hat{\mu}=\overline{x}$ and $\hat{\sigma}^2=\frac{1}{n}\sum(x_i-\overline{x})^2$

(v) $L(\lambda;x_1,\ldots,x_n)=\lambda^{\Sigma x}e^{-n\lambda}/\Pi x_i!$

$\log(L(\lambda))=\ell(\lambda)=\Sigma x_i\log(\lambda)-n\lambda+K$

$\partial\ell/\partial\lambda=\Sigma x_i/\lambda-n$ so $\hat{\lambda}=\overline{x}$.

A0.4.3 Further properties of MLEs:

Maximum likelihood estimates (mles) have many useful properties. In particular they are asymptotically unbiased and asymptotically normally distributed (subject to some technical conditions) — i.e. for large samples they are approximately normally distributed with mean equal to the [unknown] parameter and variance which can be calculated. This allows us to obtain standard errors of mles and so construct confidence intervals for them. In addition they can be used in the construction of [generalized] likelihood ratio tests.

To obtain the variance of the mle we need to calculate the expected value of the second derivative of the log-likelihood $E[(\partial^2\ell/\partial\theta^2)]$ and then the variance is the minus the reciprocal of this, i.e.

$$\mathrm{var}(\hat{\theta}) = -\{E[(\partial^2\ell/\partial\theta^2)]\}^{-1}$$

(note: if θ is a vector parameter of dimension p then we can interpret $\partial^2\ell/\partial\theta^2$ as a p×p matrix in which case we need to take the inverse of the matrix of expected values to get the variance-covariance matrix. To get just the variances of individual mles we can work with them singly and this is ok if we only want individual confidence intervals.]

A0.4.4 Examples: (continuing the examples above)

(i) $\partial\ell/\partial\mu=\Sigma(x_i-\mu)$, so $\partial^2\ell/\partial\mu^2=-n$ and thus $E[\partial^2\ell/\partial\mu^2]=-n$ and thus

$$\mathrm{var}\hat{\mu}=n^{-1}$$

(ii) $\partial\ell/\partial\lambda=n/\lambda-\Sigma x_i$, so $\partial^2\ell/\partial\lambda^2=-n/\lambda^2$, so $E[\partial^2\ell/\partial\lambda^2]=-n/\lambda^2$ and thus $\mathrm{var}(\hat{\lambda}) = \lambda^2/n$. In this case we would substitute the mle $\hat{\lambda}$ for λ to get the standard error of $\hat{\lambda}$ as $\hat{\lambda}/n^{½}$

(iii) $\partial\ell/\partial p=\Sigma x_i/p \;\; -\Sigma(m\text{-}x_i)/(1-p)$ so $\partial^2\ell/\partial p^2=-\Sigma x_i/p^2 \;\; +\Sigma(m\text{-}x_i)/(1-p)^2$ and thus

$$E[\partial^2\ell/\partial p^2]=-\Sigma mp/p^2 +\Sigma(m\text{-}mp)/(1-p)^2$$

(noting that $E[x_i]$=mp for each i

$$=-nm/p+nm/(1\text{-}p)=-nm/p(1-p)$$

and so $\mathrm{var}(\hat{p})=p(1-p)/nm$

(iv) $\partial\ell/\partial\mu=\Sigma(x_i-\mu)/\sigma^2$ and $\partial\ell/\partial\sigma=-n/\sigma+\Sigma(x_i-\mu)^2/\sigma^3$ so

$\partial^2\ell/\partial\mu^2=-n/\sigma^2$ and $\partial^2\ell/\partial\sigma^2=n/\sigma^2-3\Sigma(x_i-\mu)^2/\sigma^4$

and $\partial^2\ell/\partial\mu\partial\sigma=-\Sigma(x_i-\mu)/\sigma^3$

Now $E[x_i-\mu]=0$ and $E[(x_i-\mu)^2]=\sigma^2$ so $E[\partial^2\ell/\partial\mu^2]=-n/\sigma^2$,

$E[\partial^2\ell/\partial\sigma^2]=-2n/\sigma^2$ and $E[\partial^2\ell/\partial\mu\partial\sigma]=0$ and thus we have

$\mathrm{var}(\hat{\mu})=\sigma^2/n$, $\mathrm{var}(\hat{\sigma})=\sigma^2/2n$ and $\mathrm{cov}(\hat{\mu},\hat{\sigma})=0$.

(v) $\partial\ell/\partial\lambda=\Sigma x_i/\lambda$ so $\partial^2\ell/\partial\lambda^2=-\Sigma x_i/\lambda^2$ and we have that $E[x_i]=\lambda$ so $E[\partial^2\ell/\partial\lambda^2]=-n/\lambda$ and thus $\mathrm{var}(\hat{\lambda})=\lambda/n$

Again, in examples (iii)–(v) we would substitute the mles for the unknown parameters in the expressions for the variances to get standard errors (taking square roots) and thus obtain an approximate 95% confidence interval as **mle ± 2 × s.e.(mle)** ,

i.e. an approximate 95% confidence interval for θ is $\hat{\theta}\pm 2\times$s.e.($\hat{\theta}$)

A0.5 [Generalized] Likelihood Ratio Tests

A useful procedure for constructing hypothesis tests is an adaptation of the simple likelihood ratio test — recall that the Neyman-Pearson lemma shews that the most powerful test of a given size of one simple hypothesis against another is based on the likelihood ratio. (A simple hypothesis is one that involves no unknown parameters — the likelihood is fully specified under the hypothesis). The generalization is that [under suitable technical conditions] the [asymptotically] most powerful test of a composite hypothesis (i.e. one involving unknown parameters) against another can be based on the ratio of the maximized likelihoods, where any unknown parameters are replaced by their mles.

In fact, it is more usual to consider the [natural logarithm of this ratio (or equivalently the difference in maximized log-likelihoods since there are theoretical results that allow the significance level of this statistic to be calculated.

Specifically, if we have data $x_1,\ldots,x_n$ from a random variable X whose distribution depends on a parameter θ and if we are testing a hypothesis H_0 against and alternative H_A then ***the likelihood ratio statistic*** is $\lambda=2\{\ell(\hat{\theta}_A)-\ell(\hat{\theta}_o)\}$ where $\hat{\theta}_A$ and $\hat{\theta}_o$ are the estimates of θ under the hypotheses H_A and H_0 respectively. H_0 is rejected in favour of H_A if λ is sufficiently large. It can be shewn that for large sample sizes λ is approximately distributed as χ^2 on r degrees of freedom, where r is the difference in numbers of parameters estimated under H_A and H_0. Note that $\ell(\hat{\theta}_A)$ and $\ell(\hat{\theta}_o)$ are the actual maximum values of the log-likelihoods under H_A and H_0. Sometimes we cannot obtain mles explicitly (or algebraically) but we can obtain the maximum values of the log-likelihoods numerically using some general optimization program.

A0.5.1 Examples

(all with data $x_1,\ldots,x_n$)

(i) $X\sim N(\mu,1)$; to test H_0: $\mu=0$ *vs.* H_A: $\mu \neq 0$

Now $L(\mu)=(2\pi)^{-½n}\exp\{-½\Sigma(x_i-\mu)^2\}$

Under H_0 , $\mu=0$, so under H_0 the maximum (in fact the only) value of $L(\mu)$ is $(2\pi)^{-½n}\exp\{-½\Sigma x_i^2\}$,

i.e. $\hat{\mu}_0=0$ and $\ell(\hat{\mu}_0)=-½n\log(2\pi)-½\Sigma x_i^2$

Under H_A we just have the ordinary likelihood and the mle of μ is $\hat{\mu}_A=\overline{x}$ giving $\ell(\hat{\mu}_A)=-½n\log(2\pi)-½\Sigma(x_i-\overline{x})^{2,}$ this gives the likelihood ratio statistic as $\lambda=-2\{\ell(\hat{\mu}_A)-\ell(\hat{\mu}_0)\}=\Sigma x_i^2-\Sigma(x_i-\overline{x})^2=n\overline{x}^2$ and we reject H_0 if this is large when compared with χ_1^2.

(ii) $X\sim Ex(\lambda)$; to test H_0: $\lambda=\lambda_0$ *vs.* H_A: $\lambda \neq \lambda_0$

$L(\lambda)=\lambda^n\exp\{-\lambda\Sigma x_i\}$.

Under H_0 $\lambda=\lambda_0$ so $\hat{\lambda}_0=\lambda_0$ and $\ell(\hat{\lambda}_0)=n\log(\lambda_0)-\lambda_0\Sigma x_i\}$.

Under H_A we have $\hat{\lambda}_A = \frac{1}{\overline{x}}$ so $\ell(\hat{\lambda}_A)=n\log(\overline{x})-n$

and the lrt statistic is $2\{n\log(\overline{x})-n-n\log(\lambda_0)+\lambda_0\Sigma x_i\}$ which would be referred to a χ_1^2 distribution.

(iii) $X\sim N(\mu,\sigma^2)$; to test H_0: $\mu=0$ *vs.* $H_A\neq 0$ with σ^2 unknown.

Here we need to estimate σ under both H_0 (i.e. assuming $\mu=0$) and under H_A (not assuming $\mu=0$) and use these estimates in maximizing the likelihoods.

We have $\ell(\mu,\sigma)=-½n\log(2\pi)-n\log(\sigma)-½\Sigma(x_i-\mu)^2/\sigma^2$ so under H_0 we have

$\hat{\mu}_0=0$ and $\sigma_0^2=\frac{1}{n}\Sigma x_1{}^2$

and then $\ell(\hat{\mu}_0,\sigma_0^2)=-½n\log(2\pi)-½n\log(\frac{1}{n}\Sigma x_i{}^2)-½n$.

Under H_A we have $\hat{\mu}_A=\overline{x}$ and $\hat{\sigma}_A^2=\frac{1}{n}\Sigma(x_i-\overline{x})^2$ giving

$\ell(\hat{\mu}_A,\hat{\sigma}_A^2)=-½n\log(2\pi)-½n\log(\frac{1}{n}\Sigma(x_i-\overline{x})^2)-½n$ and thus the lrt statistic is

$\lambda=\{n\log(\Sigma x_1{}^{2)}-n\log(\Sigma(x_i-\overline{x})^2))$ which would be referred to a χ_1^2 distribution. (Note that the $\frac{1}{n}$ terms in the logs are $-n\log(n)$ and so cancel each other). It can be shewn that this statistic is a monotonic function of (and therefore equivalent to) the usual t-statistic for testing $\mu=0$ when σ is unknown.

Further examples to try are:

(i) $x_i\sim N(\mu,\sigma^2)$, H_0: $\sigma^2=\sigma_0{}^2$, μ unknown, H_A: $\sigma^2\neq\sigma_0{}^2$.

(ii) $x_i\sim N(\mu,\sigma^2)$, H_0: $\sigma^2=\sigma_0{}^2$, μ known, H_A: $\sigma^2\neq\sigma_0{}^2$.

(iii) $x_i\sim Bin(m,p)$, m known, H_A: $p\neq p_0$,

(iv) $x_i\sim Po(\lambda)$, H_0: $\lambda=\lambda_0$, H_A: $\lambda\neq\lambda_0$.

APPENDIX 1: Using discriminant analysis for Classification

A key objective of discriminant analysis is to classify further observations. In **R** or S-PLUS this can be done using the predict function `predict.lda(`*`lda-object,newdata`*`).` In the Cushings data (3 groups plus unknowns) we can perform the lda on the first 21 observations and then use the results to classify the final 7 observations of unknown categories. Note that we have to turn the final 7 observations into a data matrix `cushu` in the same way as we did with the *training data.*

```
> cush<-log(as.matrix(Cushings[1:21,-3]))
> cushu<-log(as.matrix(Cushings[22:27,-3]))
> tp<-factor(Cushings$Type[1:21])

> cush.lda<-lda(cush,tp)

> upredict<-predict.lda(cush.lda,cushu)
> upredict$class
[1] b c b a b b
```

These are the classifications for the seven new cases.

We can plot the data on the discriminant coordinates with

```
> plot(cush.lda)
```

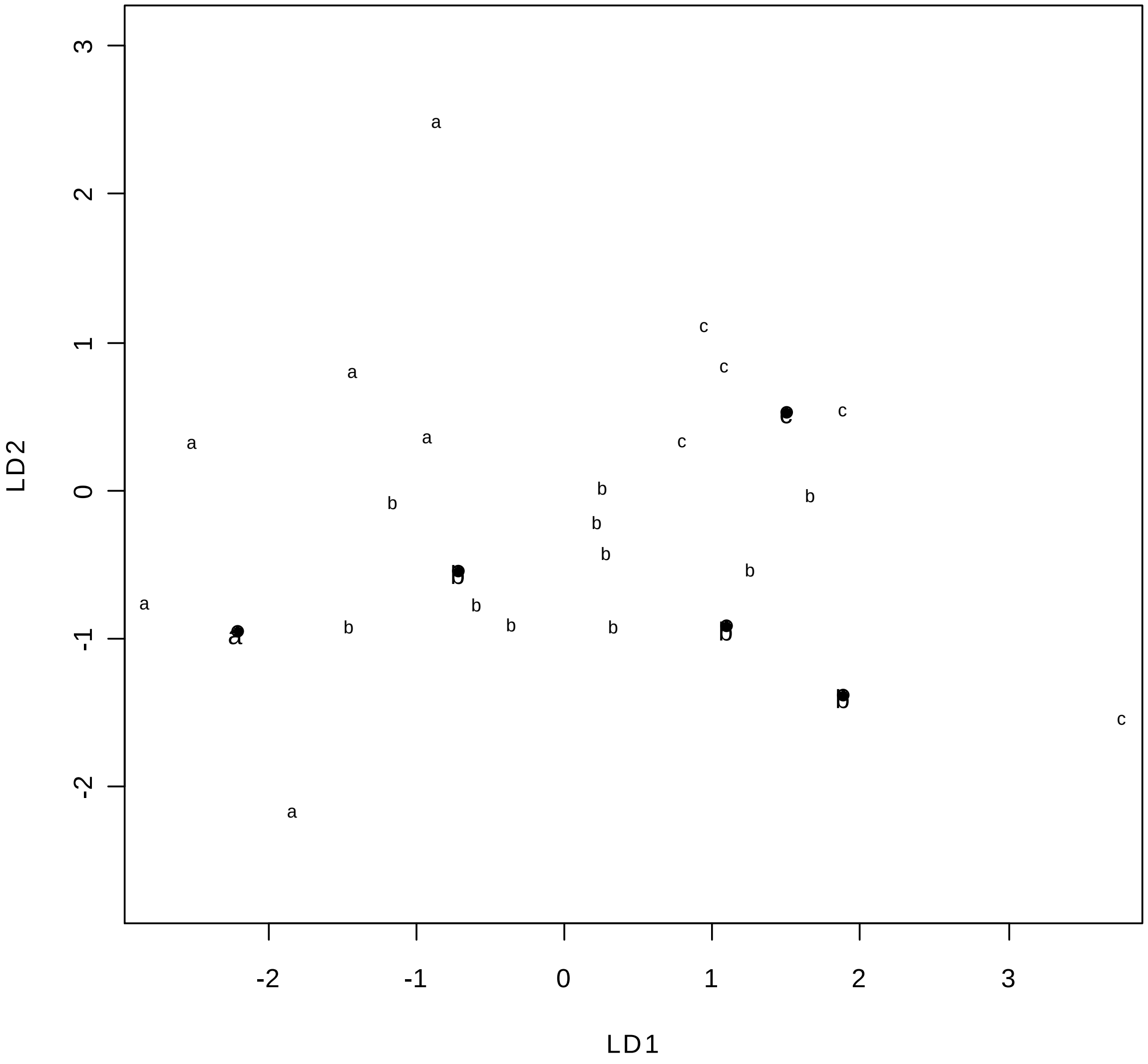
and then add in the unknown points with

```
> points(jitter(predict(cush.lda,cushu)$x),pch=19,)
```

and finally put labels giving the predicted classifications on the unknown points with

```
> text(predict(cush.lda,cushu)$x,pch=19,
+ labels=as.character(predict(cush.lda,cushu)$class))
```

(where the `+` is the continuation prompt from **R**) to give the plot below. The use of `jitter()` moves the points slightly so that the labels are not quite in the same place as the plotting symbol.



Example of discrimination and classification of three variants of Cushings syndrome (a, b and c, 21 cases in total) and classifying a further 6 unknown cases (•) (note one of these is outside the plotting range).

APPENDIX 2: Quadratic Discriminant Analysis

This generalizes lda to allow quadratic functions of the variables. Easily handled in **R** `qda()`.

```
> cush.qda<-qda(cush,tp)
> predict.qda(cush.qda,cushu)$class
[1] b c b a a b
```

It can be seen that the 5th unknown observation is classified differently by `qda()`. How can we see whether `lda()` or `qda()` is better? One way is to see how each performs on classifying the training data (i.e. the cases with known categories.

```
> predict.lda(cush.lda,cush)$class
 [1] a a a b b a b a a b b c b b b b c c b c c
```

and compare with the 'true' categories:

```
> tp
 [1] a a a a a a b b b b b b b b b b c c c c c
```

We see that 6 observations are misclassified, the 5th,6th,9th,10th,13th and 19th. To get a table of predicted and actual values:

```
> table(tp,predict.lda(cush.lda,cush)$class)

tp  a b c
  a 4 2 0
  b 2 7 1
  c 0 1 4
```

. Doing the same with `qda()` gives:

```
> table(tp,predict.qda(cush.qda,cush)$class)

tp  a b c
  a 6 0 0
  b 0 9 1
  c 0 1 4
```

so 19 out 21 were correctly classified, when only 15 using `lda()`.

If we want to see whether correctly classifying 15 out of 21 is better than chance we can permute the labels by sampling `tp` without replacement:

```
> randcush.lda<-lda(cush,sample(tp))
> table(tp,predict.lda(randcush.lda,cush)$class)

tp   a b c
  a 3 2 1
  b 1 9 0
  c 0 5 0
```

i.e. 12 were correctly classified even with completely random labels. Repeating this a few more times quickly shows that 15 is much higher than would be obtained by chance. It would be easy to write a function to do this 1000 times say by extracting the diagonal elements which are the 1st,5th and 9th elements of the object `table(.,.)`, i.e. `table(.,.)[1]`,`table(.,.)[5]` and `table(.,.)[9]`.

```
> randcush.lda<-lda(cush,sample(tp))
> table(tp,predict.lda(randcush.lda,cush)$class)

tp   a  b c
  a 1  5 0
  b 0 10 0
  c 0  5 0
> randcush.lda<-lda(cush,sample(tp))
> table(tp,predict.lda(randcush.lda,cush)$class)

tp   a b c
  a 1 5 0
  b 2 8 0
  c 1 4 0
> randcush.lda<-lda(cush,sample(tp))
> table(tp,predict.lda(randcush.lda,cush)$class)

tp   a  b c
  a 1  5 0
  b 0 10 0
  c 1  4 0
```

A2.2 Comments

- Generally discrimination and classification on the training data improves with the number of dimensions and with the complexity of the model — quadratic should generally be better than linear; if number of dimensions approaches number of observations then should be possible to get near perfect discrimination of known cases but this does not mean that the classification procedure will work so well on future data. The informal *principle of parsimony* suggests that one should look for procedures which are minimally data-dependent, i.e. which do not involve estimating large numbers of parameters from small amounts of data. However, neural-net classifiers (see later) seem to work extremely well in practice event though they are essentially highly complex non-linear discriminant functions involving very large numbers of estimated parameters.

APPENDIX 3: Outlier Displaying Components

Earlier it was shewn that PCA may or may not reveal outliers: as well as looking at both the high and the low order PCs and it is sensible to look at the 'cut-off' PCs as well. However, these are not *guaranteed* to reveal outliers. Instead, it is possible to display data on *outlier displaying components*.

The idea is to consider each observation in turn and consider which projection will highlight that observation most as an outlier — this will actually be the linear discriminant function for separating the groups consisting of that single observation alone as one 'group' and the remaining n–1 observations as the other. It can be proved that the standard test statistic for assessing that observation as an outlier is identical whether it is calculated from the original p dimensions or from the single 'outlier displaying dimension' for that observation (though the actual test of significance depends upon the number of original dimensions).

Further it can be shewn that the outlier test statistic for that observation calculated from ***any*** q-dimensional projection ($q \leq p$) is maximized when the projection includes the outlier displaying component (and it is then equal to the original p-dimensional calculation). Thus we can display the data on axes where the first is the outlier displaying dimension and the second is chosen orthogonally to that to maximize [e.g.] variance. Interpretation of loadings, display of supplementary points etc is useful.

In the case of two outliers we need to distinguish between two independent outliers (two outlier displaying components), or an outlying pair (one component).

APPENDIX 4: Correspondence Analysis

A technique for investigating/displaying the relationship between two categorical variables (i.e. frequency data in a contingency table) in a type of scatterplot. Broadly analogous to PCA except that instead of partitioning the total variance into successive components attributable to PCs it partitions the χ^2 statistic [known as the *total inertia*, $\Sigma(O–E)^2/E$] into components attributable to orthogonal underlying components. Interpretations of 'proportions of inertia explained' loadings of categories etc just as in PCA etc. Mathematically it relies on eigenanalyses of an appropriate matrix.

PCA of both covariances and correlations, CA, Biplots etc all have essentially the same 'mathematical engine' of eigenanalysis and differ only in the scaling of the central 'variance' matrix.

Canonical Correspondence Analysis (very bad and confusing name, invented by Cajo ter Braak) is CA incorporating continuous covariates, i.e. it analyses the relationship between two categorical variables after allowance has been made for the dependence of one them on a covariate. E.G.: Data on frequencies of occurrence of species on various sites arranged in a sites×species contingency table, suitable for CA, to see which species tend to group together etc. If also data on say soil pH and %LoI (percent loss on ignition — measures organic content) then can allow for these explanatory variables when investigating the dependence. Primarily used in ecology — maybe of use in other areas?? Main package is Canoco (from ter Braak).

Multiple Correspondence Analysis is a generalization of CA to 3 or more variables.

APPENDIX 5: Cluster Analysis

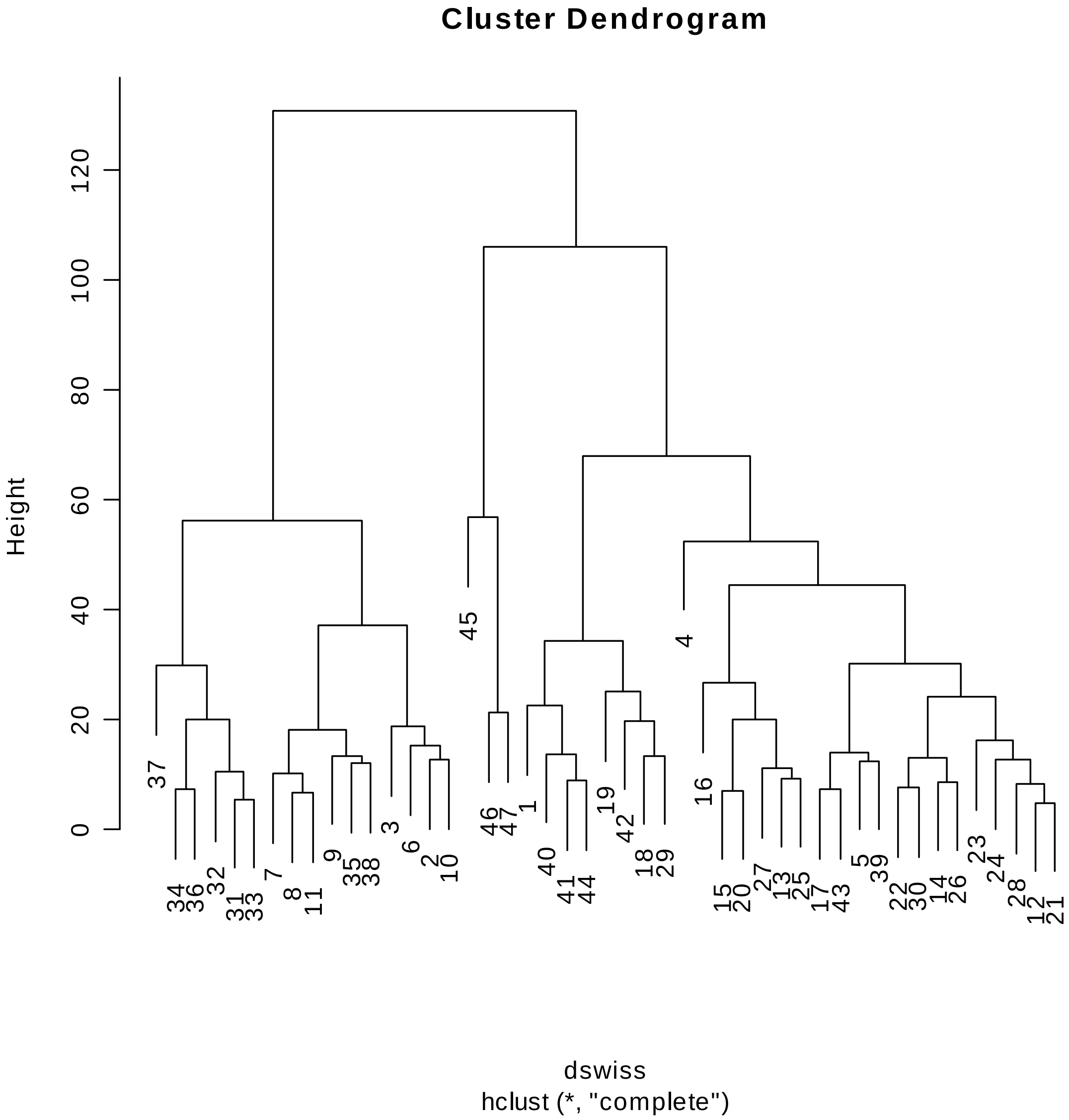
A5.1 Introduction

Cluster Analysis is a collection of techniques for *unsupervised* examination of multivariate data with an objective of discovering ‘natural’ groups in the data, often used together with scaling methods. Hierarchical methods start with each case in a separate cluster and proceed by agglomerating most similar cases into increasingly larger sized clusters. Thus a division into *r* clusters will be subdivison of one into *s* clusters when r>s. Non-hierarchical methods typically start by dividing all the cases into a *pre-specified* number of clusters. It is possible that a division in *r* clusters will bear little similarity with one into *s* clusters. For hierarchical methods the results can be displayed in a ***dendrogram*** rather like a family tree indicating which family objects belong to.

A5.2 Hierarchical Methods

The first example is on data `swiss` which gives demographic measurements of 47 provinces in Switzerland. The first step is to calculate a distance matrix, using `dist()` and then to perform hierarchical cluster analysis using `hclust()`, The result can then be plotted as a dendogram using the generic function `plot()`. This example has used the default clustering method of complete linkage, others you might try are average linkage, single linkage or Wards method

```
> data(swiss)
> summary(swiss)
   Fertility      Agriculture     Examination      Education
 Min.   :35.00   Min.   : 1.20   Min.   : 3.00   Min.   : 1.00
 1st Qu.:64.70   1st Qu.:35.90   1st Qu.:12.00   1st Qu.: 6.00
 Median :70.40   Median :54.10   Median :16.00   Median : 8.00
 Mean   :70.14   Mean   :50.66   Mean   :16.49   Mean   :10.98
 3rd Qu.:78.45   3rd Qu.:67.65   3rd Qu.:22.00   3rd Qu.:12.00
 Max.   :92.50   Max.   :89.70   Max.   :37.00   Max.   :53.00
    Catholic       Infant.Mortality
 Min.   :  2.150   Min.   :10.80
 1st Qu.:  5.195   1st Qu.:18.15
 Median : 15.140   Median :20.00
 Mean   : 41.144   Mean   :19.94
 3rd Qu.: 93.125   3rd Qu.:21.70
 Max.   :100.000   Max.   :26.60
> dswiss<-dist(swiss)
> h<- hclust(dswiss)
> plot(h)
```



This suggests three main groups, we can identify these with

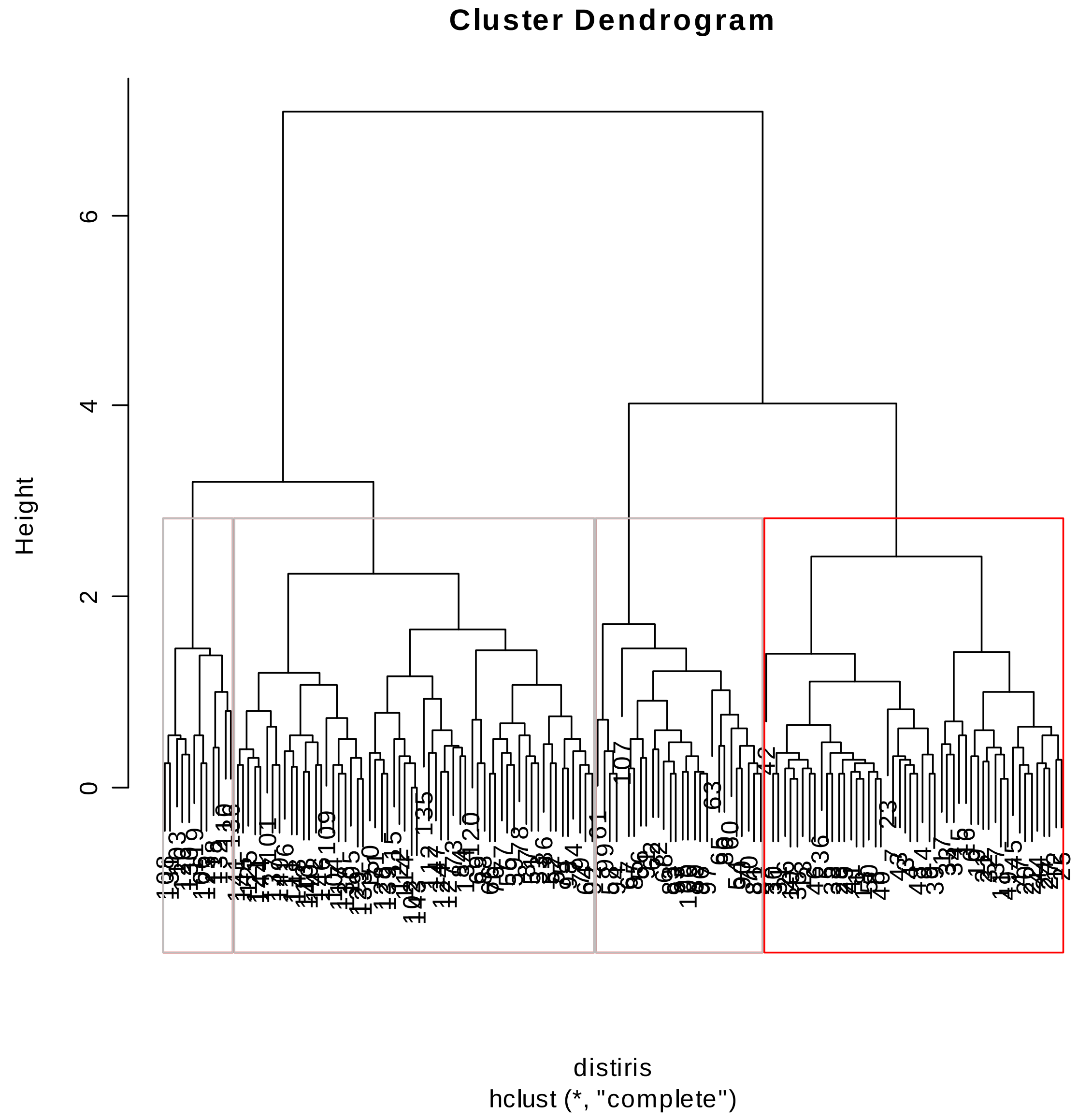
```
> cutree(h,3)
 1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21
22 23 24 25 26
 1  2  2  1  1  2  2  2  2  2  2  1  1  1  1  1  1  1  1  1  1
1  1  1  1  1
27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47
 1 1 1 1 2 2 2 2 2 2 2 2 1 1 1 1 1 1 3 3 3
```

which gives the group membership for each of the provinces.

Next, we look at the iris data (yet again) and use the interactive function `identify.hclust()` which allows you to point with the mouse and click on vertical bars to extract the elements in the family below. Click with the right button and choose stop to leave it.

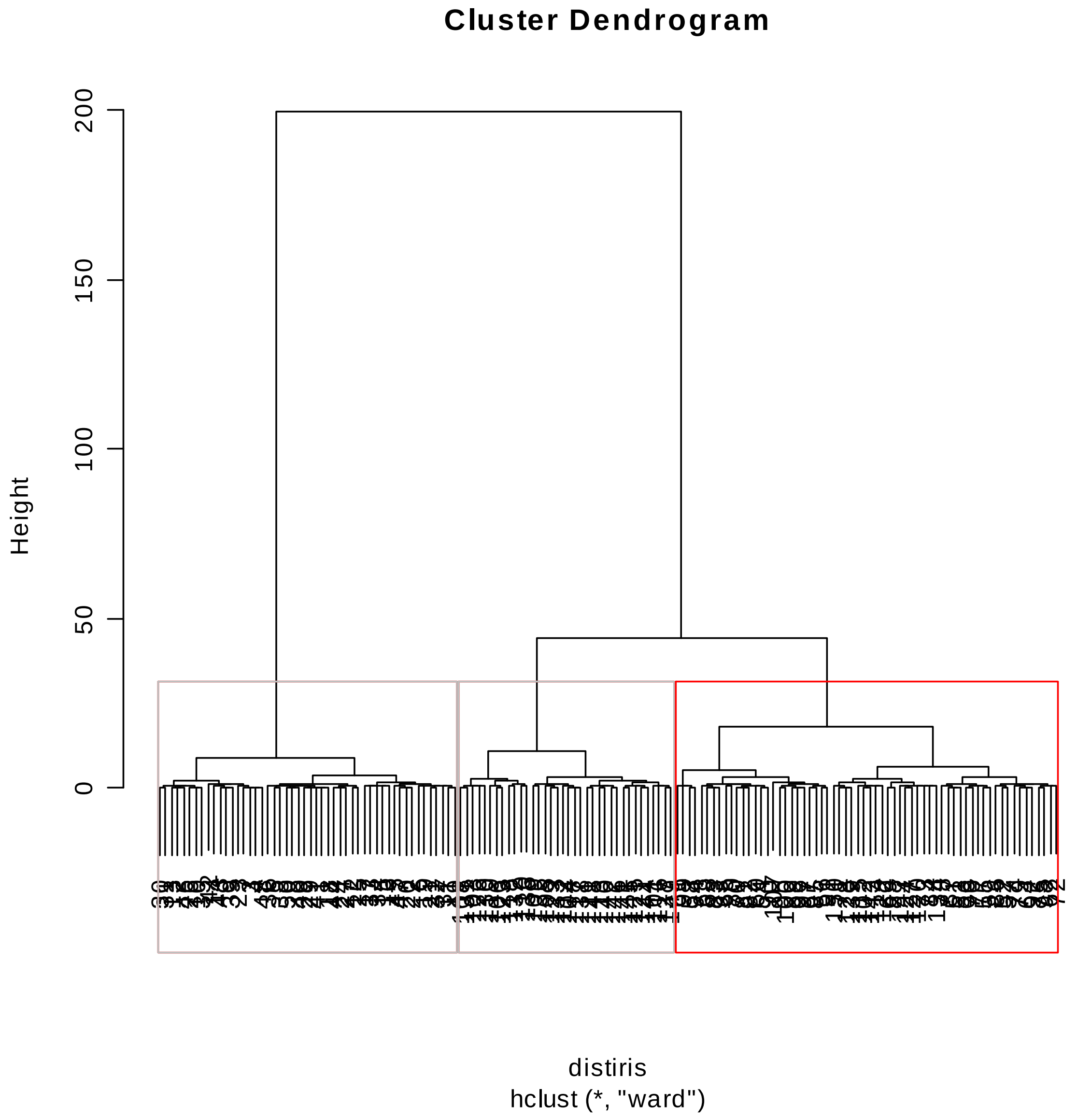
```
> distiris<-dist(ir)
> hiris<- hclust(distiris)
> plot(hiris)
>  identify.hclust(hiris, function(k) print(table(iris[k,5])))

    setosa versicolor  virginica
         0          0         12

    setosa versicolor  virginica
         0         23         37

    setosa versicolor  virginica
         0         27          1

    setosa versicolor  virginica
        50          0          0
>
```

The dendogram on the next pages shows four groups, and `identify.clust` was used to click on the four ancestor lines. Note that one of the groups is obviously the overlap group between versicolour and virginica.



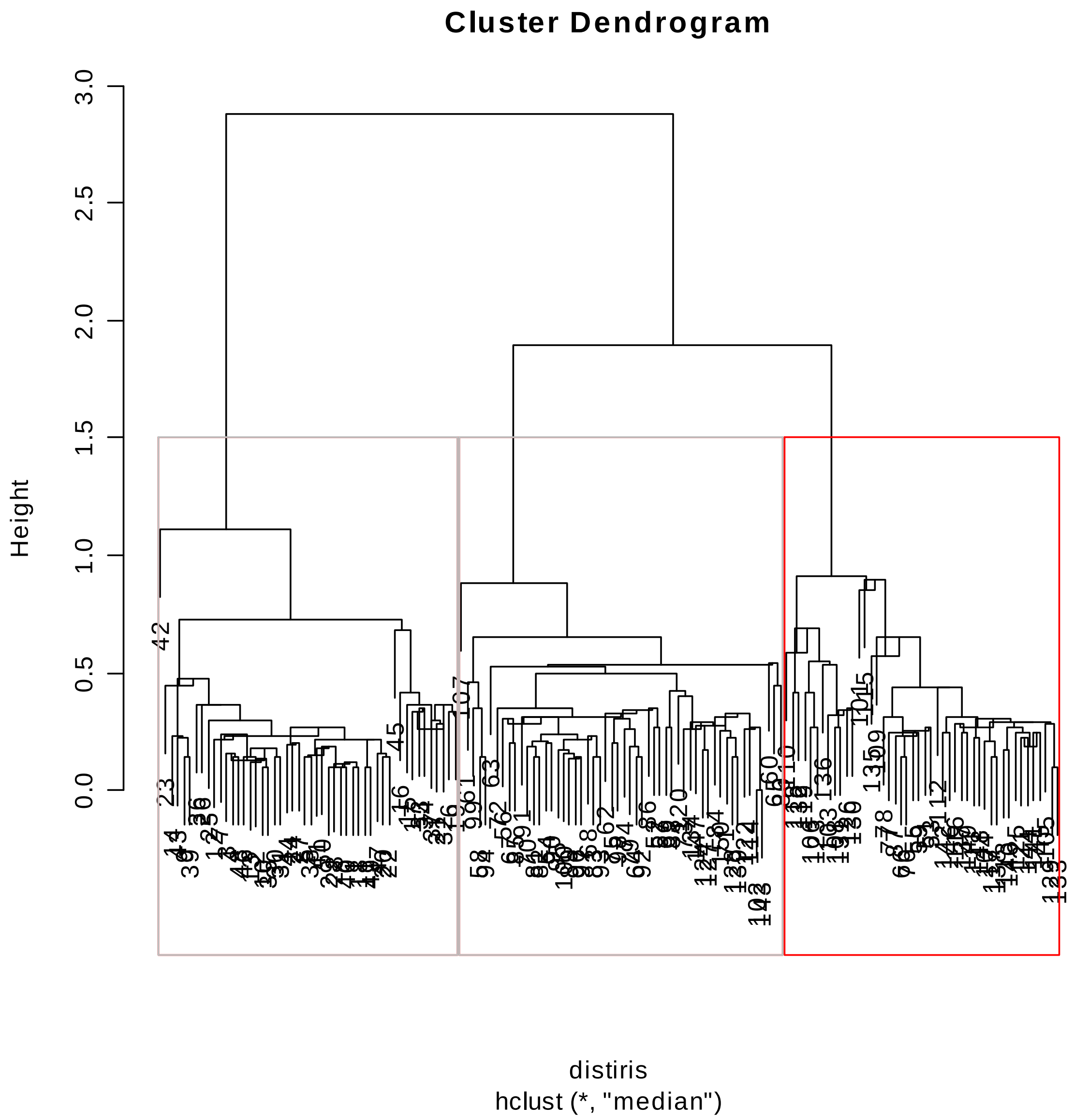
Using a different method (Ward's) gives:

```
> hirisw<- hclust(distiris,method="ward")
> plot(hirisw)
>  identify.hclust(hirisw,function(k) print(table(iris[k,5])))

    setosa versicolor  virginica
        50          0          0

    setosa versicolor  virginica
         0          0         36

    setosa versicolor  virginica
         0         50         14
```



And finally, using the 'median' method gives

```
> hirimed<- hclust(distiris,method="median")
> plot(hirimed)
>  identify.hclust(hirimed,function(k)print(table(iris[k,5])))
    setosa versicolor  virginica
        50          0          0
    setosa versicolor  virginica
         0         41         13
    setosa versicolor  virginica
         0          9         37
```


Cluster Dendrogram
Height
0.0
0.5
1.0
1.5
2.0
2.5
3.0
distiris
hclust (*, "median")

A5.3 Non-hierachical or Optimization Clustering

Optimization clustering techniques aim to find a **pre-specified** number k of clusters that satisfy some optimal condition. In principle, the optimisation is over all possible allocations of n objects into k clusters. However, the number of allocations is large for realistic problems — with 5 cases and 2 clusters it is 15, with 50 cases and 4 clusters it is 5.3×10^{28} and 100 cases and 5 clusters 6.6×10^{67}— is iterative techniques are used by considering moving that case which improves the optimisation criterion most, or (if large numbers of cases) by moving each observation in turn to that cluster which improves the criterion the most (rather than searching over all observations before deciding). The procedure then cycles through the complete data set until stability is achieved.

The measures of 'quality of clustering' can reflect separability between clusters or homogeneity within clusters or a mixture of both, e.g. any of the three standard statistics for MANOVA (multivariate analysis of variance) can be used, c.f. Ward's method for agglomerative hierarchical clustering.

A standard and very fast method is **K-means** clustering where each observation is [re]assigned to that cluster to whose centroid it is closest, clusters initially being defined with arbitrary centroids. Typically, the method is used on continuous data and the distance is Euclidean, but more general distance measures (e.g. single linkage) could be used.

In one study of competing clustering methods (Jonathan Myles, PhD thesis, OU ~1990, supervised by David Hand) th K-means method emerged as ‘overall best buy’ on medium (100 cases) to large (~10^6) data sets.

Since the criterion is optimised over discrete points, which are accessed sequentially, there can be apparent ‘local optima’, these may be dependent on the ordering of the cases and on the choice of starting clusters — JB reports this effect with SAS JMP, even with smaller data sets. This is inevitable with iterative techniques and a non-continuous ‘space’ of variables, i.e. although the space consists of discrete points, $\frac{1}{k!}\sum_{r=1}^{k}(-1)^{k-r}\binom{k}{r}r^n$ of them, the iterative technique only passes through a transect of them determined by the ordering of cases and starting points.

Clearly, stability to ordering and starting needs to be investigated, but it may not be a serious problem if the objective is (as it is likely too be with large numbers of cases) one of qualitative results rather than detailed classification of individual cases. However, maybe this is an opportunity for some more sophisticated optimisation technique.

The choice of the number of clusters, k, can be done by standard ‘scree-graph’ techniques of the optimal value of the criterion with each number of clusters.

A5.4 Further Directions:

The library `cluster` contains a variety of routines and data sets. The `mclust` library offers model-based clustering (i.e. a little more statistical).

Key reference: **Cluster Analysis, 4th Edition, (2001)**, by Brian Everitt, Sabine Landau and Morven Leese.

APPENDIX 6: Tree-Based Methods

Classification and regression trees are similar *supervised* techniques which are used to analyse problems in a step-by-step approach.

A6.1 Classification Trees

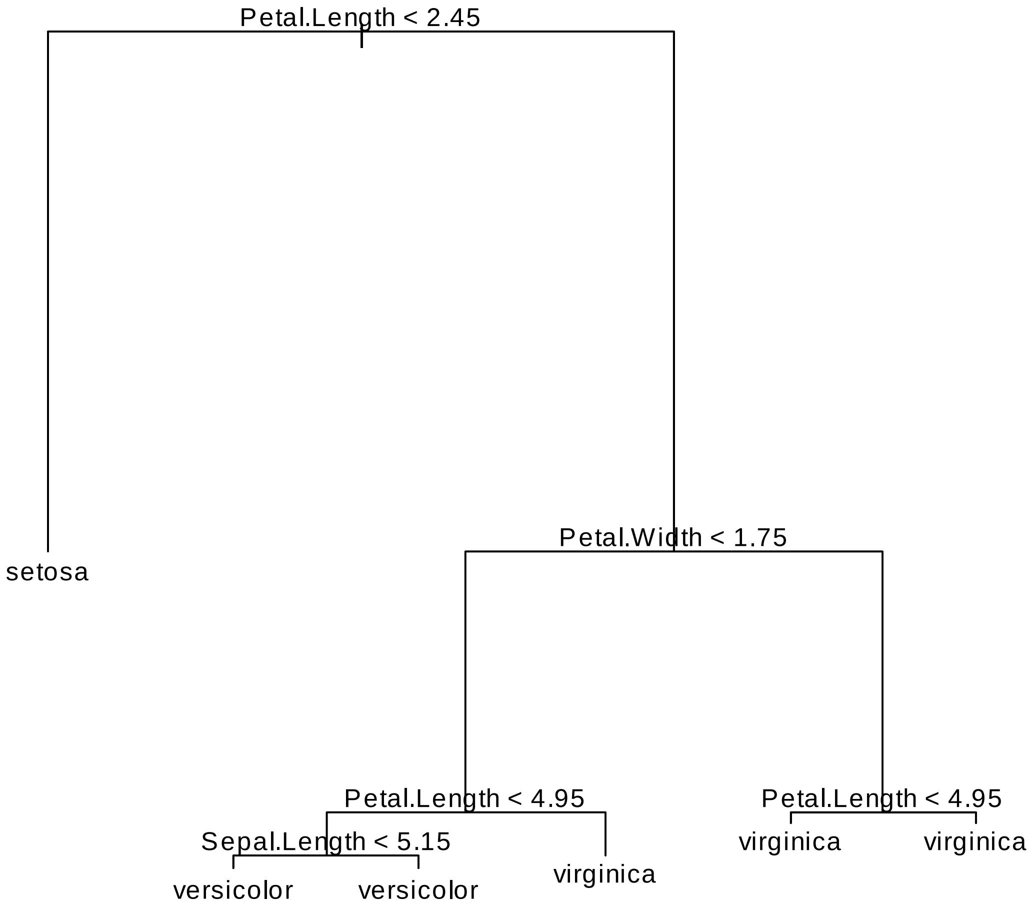
We start (even yet again) with the iris data where the objective is to find a set of rules, based on the four measurements we have, of classifying the flowers into on of the fours species. The rules will be of the form:

'if petal length>x then…. , but if petal length ≤x then something else'

i.e. the rules are based on the values of one variable at a time and gradually partition the data set into groups.

```
> data(iris)
> attach(iris)
> ir.tr<-tree(Species~.,iris)
> plot(ir.tr)
> summary(ir.tr)

Classification tree:
tree(formula = Species ~ ., data = iris)
Variables actually used in tree construction:
[1] "Petal.Length" "Petal.Width"  "Sepal.Length"
Number of terminal nodes:  6
Residual mean deviance:  0.1253 = 18.05 / 144
Misclassification error rate: 0.02667 = 4 / 150
> text(ir.tr,all=T,cex=0.5)
```

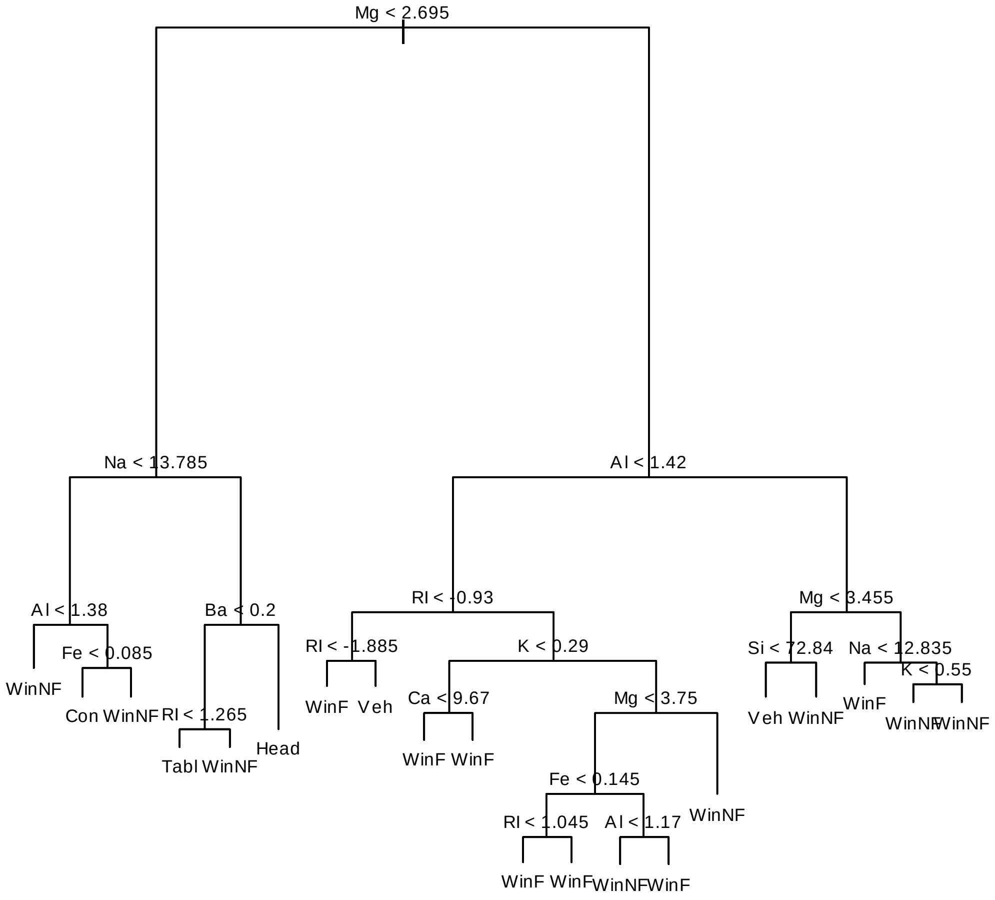
Now look at the graphical representation:



```
> ir.tr
node), split, n, deviance, yval, (yprob)
      * denotes terminal node

 1) root 150 329.600 setosa ( 0.33333 0.33333 0.33333 )
   2) Petal.Length < 2.45 50   0.000 setosa ( 1.00000 0.00000 0.00000 ) *
   3) Petal.Length > 2.45 100 138.600 versicolor ( 0.00000 0.50000 0.50000 )
     6) Petal.Width < 1.75 54  33.320 versicolor ( 0.00000 0.90741 0.09259 )
      12) Petal.Length < 4.95 48   9.721 versicolor ( 0.00000 0.97917 0.02083 )
        24) Sepal.Length < 5.15 5   5.004 versicolor ( 0.00000 0.80000 0.20000 ) *
        25) Sepal.Length > 5.15 43   0.000 versicolor ( 0.00000 1.00000 0.00000 ) *
      13) Petal.Length > 4.95 6   7.638 virginica ( 0.00000 0.33333 0.66667 ) *
     7) Petal.Width > 1.75 46   9.635 virginica ( 0.00000 0.02174 0.97826 )
      14) Petal.Length < 4.95 6   5.407 virginica ( 0.00000 0.16667 0.83333 ) *
      15) Petal.Length > 4.95 40   0.000 virginica ( 0.00000 0.00000 1.00000 ) *
>
```

Another example: the forensic glass data `fgl`. the data give the refractive index and oxide content of six types of glass.

```
> data(fgl)
> attach(fgl)
> summary(fgl)
       RI                Na              Mg              Al
 Min.   :-6.8500   Min.   :10.73   Min.   :0.000   Min.   
:0.290  
 1st Qu.:-1.4775   1st Qu.:12.91   1st Qu.:2.115   1st 
Qu.:1.190  
 Median :-0.3200   Median :13.30   Median :3.480   Median 
:1.360  
 Mean   : 0.3654   Mean   :13.41   Mean   :2.685   Mean   
:1.445  
 3rd Qu.: 1.1575   3rd Qu.:13.82   3rd Qu.:3.600   3rd 
Qu.:1.630  
 Max.   :15.9300   Max.   :17.38   Max.   :4.490   Max.   
:3.500  
       Si              K                Ca               Ba
 Min.   :69.81   Min.   :0.0000   Min.   : 5.430   Min.   
:0.0000  
 1st Qu.:72.28   1st Qu.:0.1225   1st Qu.: 8.240   1st 
Qu.:0.0000  
 Median :72.79   Median :0.5550   Median : 8.600   Median 
:0.0000  
 Mean   :72.65   Mean   :0.4971   Mean   : 8.957   Mean   
:0.1750  
 3rd Qu.:73.09   3rd Qu.:0.6100   3rd Qu.: 9.172   3rd 
Qu.:0.0000  
 Max.   :75.41   Max.   :6.2100   Max.   :16.190   Max.   
:3.1500  
       Fe             type   
 Min.   :0.00000   WinF :70  
 1st Qu.:0.00000   WinNF:76  
 Median :0.00000   Veh  :17  
 Mean   :0.05701   Con  :13  
 3rd Qu.:0.10000   Tabl : 9  
 Max.   :0.51000   Head :29  
> fgl.tr<-tree(type~.,fgl)
> summary(fgl.tr)

Classification tree:
tree(formula = type ~ ., data = fgl)
Number of terminal nodes:  20 
Residual mean deviance:  0.6853 = 133 / 194 
Misclassification error rate: 0.1542 = 33 / 214 
> plot(fgl.tr)
> text(fgl.tr,all=T,cex=0.5))
Error: syntax error
> text(fgl.tr,all=T,cex=0.5)
```


Mg < 2.695
Na < 13.785
Al < 1.42
Al < 1.38
Ba < 0.2
RI < -0.93
Mg < 3.455
Fe < 0.085
RI < -1.885
K < 0.29
Si < 72.84
Na < 12.835
K < 0.55
WinNF
Con WinNF
RI < 1.265
Tabl WinNF
Head
WinF Veh
Ca < 9.67
WinF WinF
Mg < 3.75
Fe < 0.145
WinNF
RI < 1.045
Al < 1.17
WinF WinF
WinNF WinF
Veh WinNF
WinF
WinNF WinNF

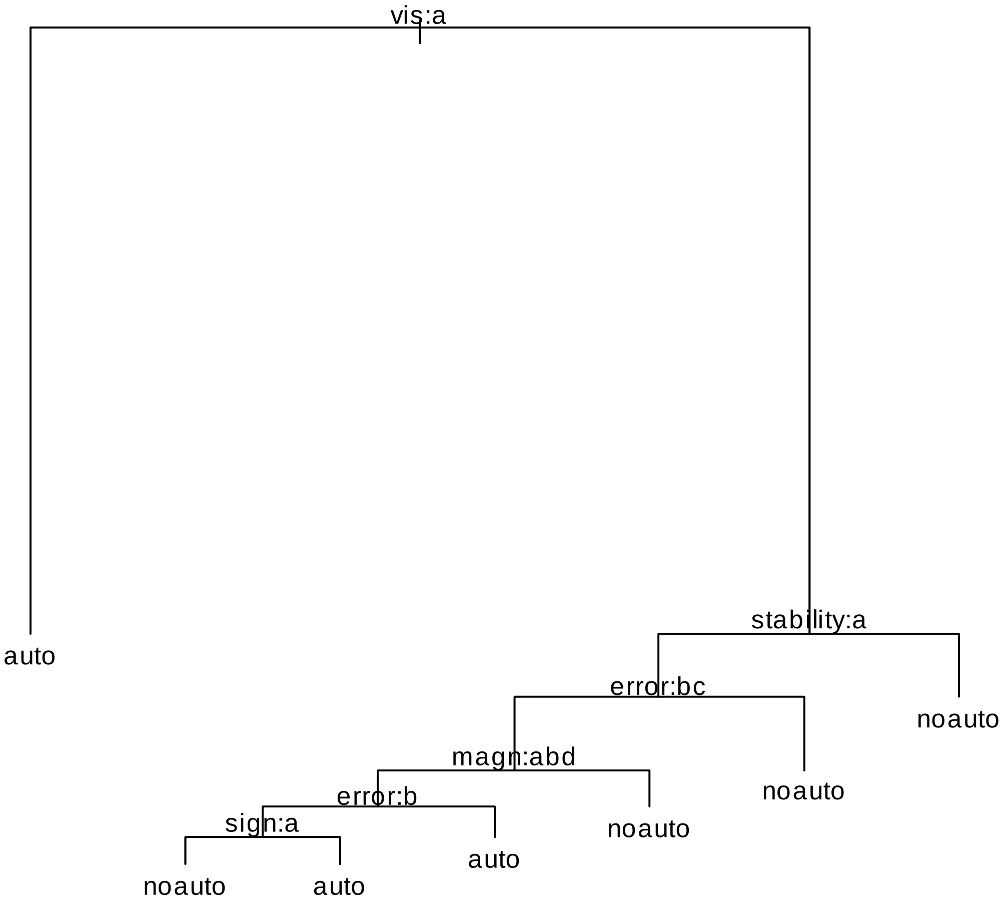
A6.2 Decision Trees

One common use of classification trees is as an aid to decision making — not really different from classification but sometimes distinguished.

Data shuttle gives guidance on whether to use autolander or manual control on landing the space shuttle under various conditions such as head or tail wind of various strengths, good or poor visibility (always use auto in poor visibility!) etc, 6 factors in all. There are potentially 256 combinations of conditions and these can be tabulated and completely enumerated but displaying the correct decision as a tree is convenient and attractive.

```
> data(shuttle)
> attach(shuttle)
> summary(shuttle)
 stability    error   sign         wind           magn      vis
use
 stab :128    LX:64   nn:128    head:128   Light :64    no :128
auto  :145
 xstab:128    MM:64   pp:128    tail:128   Medium:64    yes:128
noauto:111
             SS:64                       Out   :64
             XL:64                       Strong:64
> table(use,vis)
        vis
use       no yes
  auto   128  17
  noauto   0 111
> table(use,wind)
        wind
use      head tail
  auto     72   73
  noauto   56   55
```

```
> table(use,magn,wind)
, , wind = head

        magn
use      Light Medium Out Strong
  auto      19     19  16     18
  noauto    13     13  16     14

, , wind = tail

        magn
use      Light Medium Out Strong
  auto      19     19  16     19
  noauto    13     13  16     13

> shuttle
    stability error sign wind   magn vis    use
1       xstab    LX   pp head  Light  no   auto
2       xstab    LX   pp head Medium  no   auto
3       xstab    LX   pp head Strong  no   auto
…         …         …         …         …         …
…         …         …         …         …         …
…         …         …         …         …         …
…         …         …         …         …         …
255      stab    MM   nn head Medium yes noauto
256      stab    MM   nn head Strong yes noauto
>
> shuttle.tr<-tree(use~.,shuttle)
> plot(shuttle.tr)
> text(shuttle.tr)
```



In this default display, the levels of the factors are indicated by a,b,…. alphabetically and the tree is read so that levels indicated are to the left branch and others to the right, e.g. at the first branching `vis:a` indicates `no` for the left branch and `yes` for the right one. At the branch labelled `magn:abd` the right branch is for level c which is '`out of range'`; all other levels take the left branch. The plot can of course be enhanced with better labels.

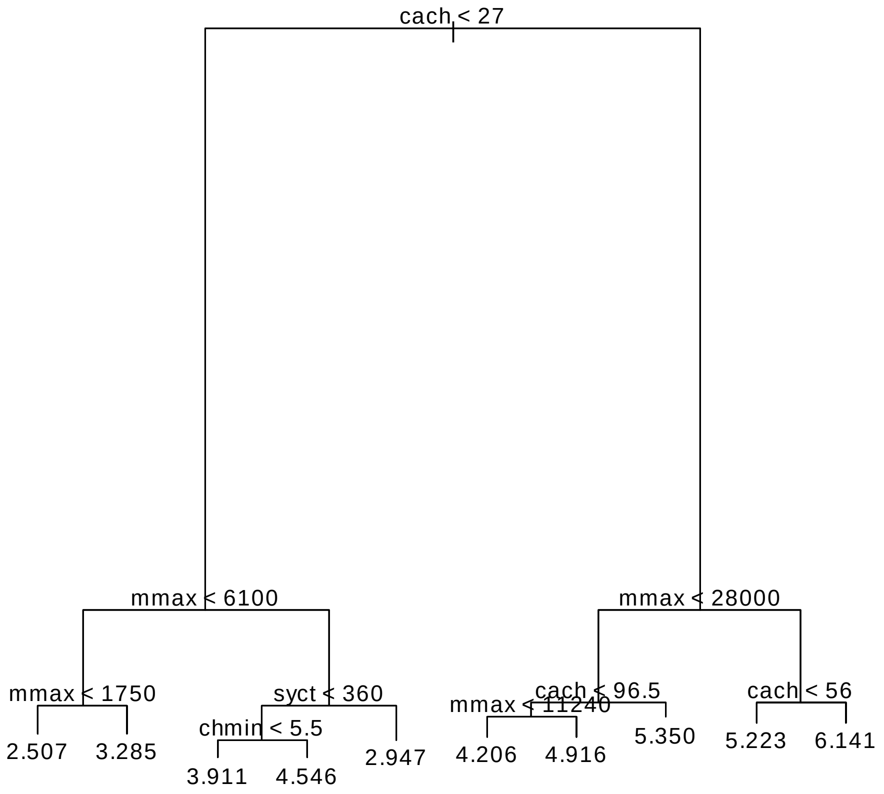
A6.3 Regression Trees:

We can think of classification trees as modelling a ***discrete*** factor or outcome as depending on various explanatory variables, either continuous or discrete**.** For example, the *iris species* depended upon values of the continuous variables giving the dimensions of the sepals and petals. In an analogous way we could model a ***continuous*** outcome on explanatory variables using tree-based methods, i.e. *regression trees*. The analysis can be thought of as categorizing the continuous outcome into discrete levels, i.e. turning the continuous outcome into a discrete factor. The number of distinct levels can be controlled by specifying the minimum number of observations (`minsize`) at a node that can be split and the reduction of variance produced by splitting a node (`mindev`). This is illustrated on data on c.p.u. performance of 209 different processors in data set `cpus` contained in the `MASS` library. The measure of performance is perf and we model the log of this variable.

```
> library(MASS)
> library(tree)
> data(cpus)
> attach(cpus)
> summary(cpus)
                   name                 syct                   mmin
mmax
 WANG VS10            :  1   Min.   :  17.0   Min.   :   64
Min.   :   64
 WANG VS 90           :  1   1st Qu.:  50.0   1st Qu.:  768
1st Qu.: 4000
 STRATUS 32           :  1   Median : 110.0   Median : 2000
Median : 8000
 SPERRY 90/80 MODEL 3:  1   Mean   : 203.8   Mean   : 2868
Mean   :11796
 SPERRY 80/8          :  1   3rd Qu.: 225.0   3rd Qu.: 4000
3rd Qu.:16000
 SPERRY 80/6          :  1   Max.   :1500.0   Max.   :32000
Max.   :64000
 (Other)              :203
      cach            chmin            chmax              perf
estperf
 Min.   :  0.00 Min.   : 0.000 Min.   :  0.00 Min.   :   6.0
Min.   :  15.0
```

```
 1st Qu.:  0.00 1st Qu.: 1.000 1st Qu.:  5.00 1st Qu.:  27.0
1st Qu.:  28.0
 Median :  8.00 Median : 2.000 Median :  8.00 Median :  50.0
Median :  45.0
 Mean   : 25.21 Mean   : 4.699 Mean   : 18.27 Mean   : 105.6
Mean   :  99.3
 3rd Qu.: 32.00 3rd Qu.: 6.000 3rd Qu.: 24.00 3rd Qu.: 113.0
3rd Qu.: 101.0
 Max.   :256.00 Max.   :52.000 Max.   :176.00 Max.   :1150.0
Max.   :1238.0
```

```
> cpus.tr<-tree(log(perf)~.,cpus[,2:8])
> plot(cpus.tr)
> text(cpus.tr)
```



The attraction of the display is that it gives a quick way of predicting cpu performance for a processor with specified characteristics. The accuracy of the predictions can be increased by increasing the number of terminal nodes (or leaves). However, this does not offer a substitute for more investigative modelling and outlier identification.

APPENDIX 7: Neural Networks

A7.1 Introduction

We have seen how we can consider classification and discrimination problems as a form of modelling the relationship between a categorical variable and various explanatory variables. We could make this more explicit and use, for example, logistic regression techniques. For example, suppose we have two categories, A and B, and explanatory variables $x_1,\ldots,x_k$ then we could model the probability that an object with values of $x_1,\ldots,x_k$ belongs to category A as a logistic function of the $x_1,\ldots,x_k$:

$$\text{P[belongs to A]} = \frac{\exp\{\alpha + \beta_1 x_1 + \ldots + \beta_k x_k\}}{1 + \exp\{\alpha + \beta_1 x_1 + \ldots + \beta_k x_k\}}$$

and then estimate the unknown parameters β_i from training data on objects with known classifications. New observations would be classified by classifying them as of type A if the estimated probability of belonging to A is > 0.5, otherwise classify them as of type B. The technique is widely used and is very effective, it is known as *logistic discrimination*. It can readily handle cases where the x_i are a mixture of continuous and binary variables. If there is an explanatory variable whish is categorical with k>2 levels then it needs to be replaced by k–1 dummy binary variables (though this step can be avoided with tree-based methods).

The idea could be extended to discrimination and classification with several categories, *multiple logistic discrimination.*

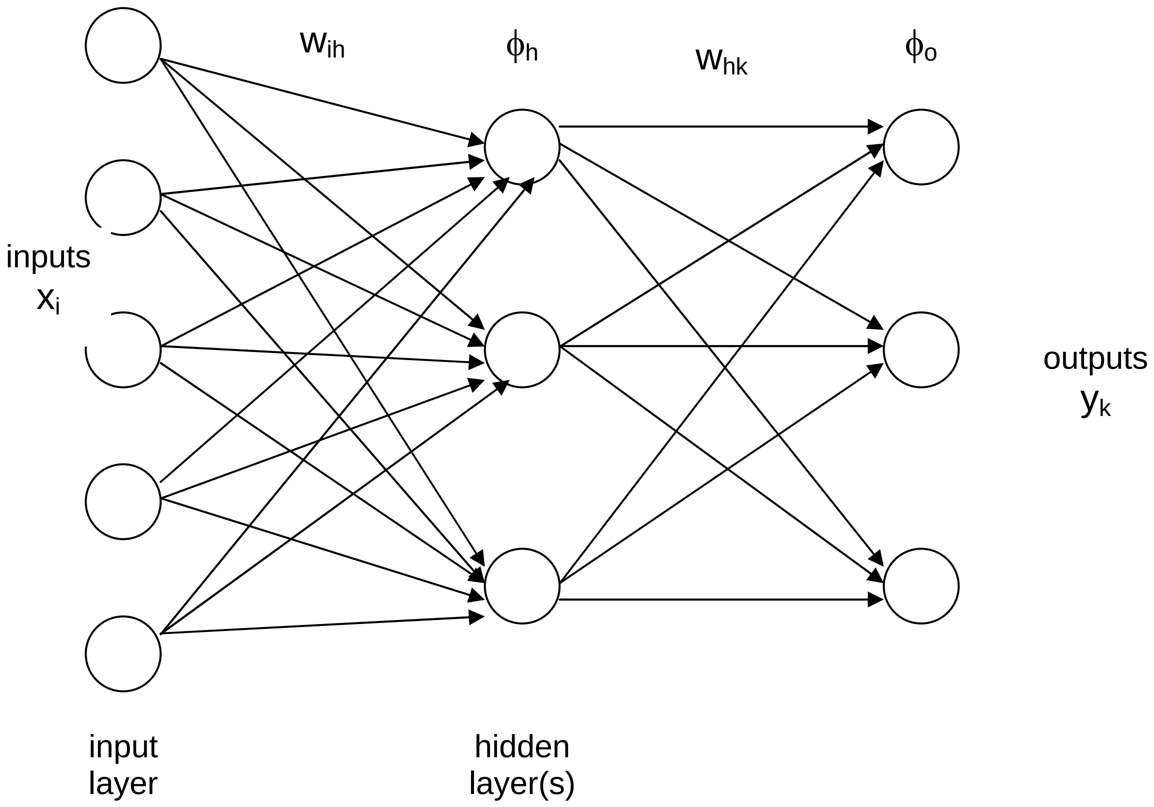
Neural networks, from a statistical point of view, can be thought of as a further extension of the idea and special cases of them are essentially non-linear logistic models. However, the technique is rather more general than just non-linear logistic modelling. It also has analogies with generalized additive modelling.

The full model for a feed-forward neural network with one hidden layer is

$$y_k = \phi_0\left(\alpha_k + \sum_h w_{hk}\phi_h(\alpha_h + \sum_i w_{ih}x_i)\right)$$

where the ‘inputs’ are x_i (i.e. values of explanatory variables), the ‘outputs’ are y_k (i.e. values of the dependent variable, and the α_j and w_{ij} are unknown parameters which have to be estimated (i.e. the network has to be ‘trained’) by minimising some fitting criterion, e.g. least squares or a measure of entropy. The functions ϕ_j are ‘activation functions’ and are often taken to be the logistic function $\phi(x)=\exp(x)/\{1+\exp(x)\}$. The w_{ij} are usually thought of as *weights* feeding forward input from the observations through a ‘hidden layer’ of units (ϕ_h) to output units which also consist of activation functions ϕ_o.

The model is often represented graphically as a set of inputs linked through a hidden layer to the outputs:



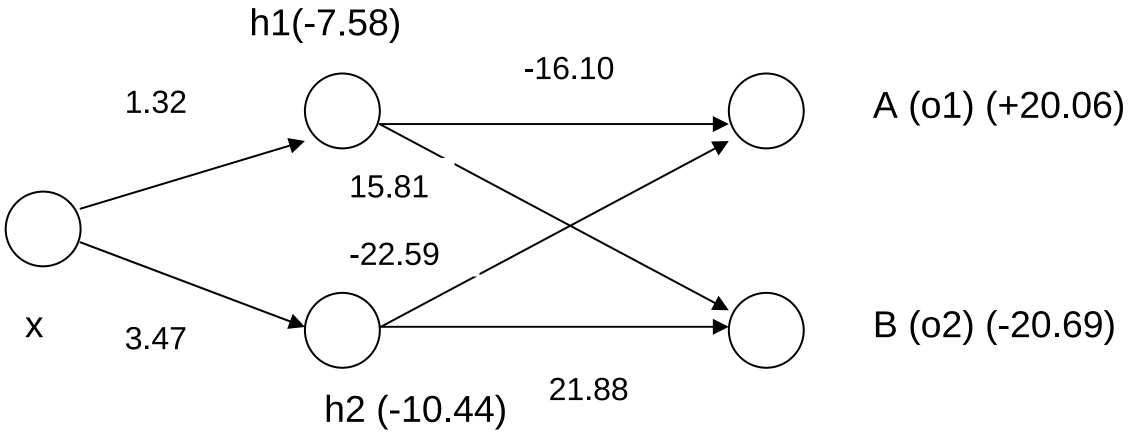
The number of inputs is the number of explanatory variables x_i (i.e. the dimension of the observations), the number of outputs is the number of levels of y_k (if y_k is categorical, though actually y_k would be regarded as a k-dimensional vector with components being the dummy variables indicating the k categories), or the dimension of y_k (if y_k is continuous) and the number of 'hidden units' is open to choice. The greater the number of hidden units the larger the number of parameters to be estimated and (generally) the better will be the fit of the predicted y_k with the observed y_k.

A7.2 A Simple Example:

This is an artificial example: the objective is to train a network with a hidden layer containing two units to return a value A for low numbers and a value B for high ones. The following code sets up a dataframe (nick) which has 8 rows and two columns. The first column has the values of x and the second the targets. The first five rows will be used for training the net and the last three will be fed into the trained net for classification, so the first 3 rows have low values of x and target value A, the next 2 rows have high values of x and the target value B and the final 3 rows have test values of x and unknown classifications.

```
> library(nnet) # open nnet library
> nick<-
+ data.frame(x=c(1.1,1.7,1.3,5.6,7.2,8.1,1.8,3.0),
+ targets=c(rep("A",3),rep("B",2),rep("U",3)))
> attach(nick)
# check dataframe is ok
> nick
     x   targets
1   1.1       A
2   1.7       A
3   1.3       A
4   5.6       B
5   7.2       B
6   8.1       U
7   1.8       U
8   3.0       U
> nick.net<-nnet(targets~.,data=nick[1:5,],size=2)
# weights:  10
initial  value 3.844981
final  value 0.039811
converged
Warning message:
group(s) U are empty in: nnet.formula(targets ~ .,
data = nick[1:5, ], size = 2)
```

```
# check predictions on training data
> predict(nick.net,nick[1:5,],type="class")
[1] "A" "A" "A" "B" "B"
# now classify new data
> predict(nick.net,nick[6:8,],type="class")
[1] "B" "A" "A"

# see what the predictions look like numerically
> predict(nick.net,nick[6:8,])
             A            B
6 1.364219e-15 1.000000e+00
7 1.000000e+00 4.659797e-18
8 1.000000e+00 1.769726e-08
> predict(nick.net,nick[1:5,])
             A            B
1 1.000000e+00 2.286416e-18
2 1.000000e+00 3.757951e-18
3 1.000000e+00 2.477393e-18
4 1.523690e-08 1.000000e+00
5 2.161339e-14 1.000000e+00
>
# look at estimates of weights.
> summary(nick.net)
a 1-2-2 network with 10 weights
options were - softmax modelling
 b->h1 i1->h1
 -7.58   1.32
 b->h2 i1->h2
-10.44   3.47
 b->o1 h1->o1 h2->o1
 20.06 -16.10 -22.59
 b->o2 h1->o2 h2->o2
-20.69  15.81  21.88
```



To check how the calculations proceed inside the net we have an input x=1.1 which is to be classified as A and looking at the values given in the middle of the previous page this means that we want the net to produce the output vector (1,0)′ (actually it has produced (1, 2.286416e–18).

To see how this calculation goes, we see that the output at A is made of the sum of two elements obtained by the two possible routes from the input to A. The first of these is

$$\phi\{20.06 - 16.10\phi() - 22.59\phi(-10.44+3.47\times\mathbf{1.1})\}$$

and the second is

$$\phi\{-20.69 + 21.88\phi(-10.44+3.47\times\mathbf{1.1})+15.81\phi(-7.58+1.32\times\mathbf{1.1})\}$$

where $\phi(t)=\exp(t)/\{1+\exp(t)\}$.

A7.3 Examples on Iris Data

These next two examples are taken from the `help(nnet)` output and are illustrations on the iris data yet again, this time the classification is based on (i.e. the neural network is trained on) a random 50% sample of the data and evaluated on the other 50%. In the first example the target values are taken to be the vectors (1,0,0), (0,1,0) and (0,0,1) for the three species (i.e. indicator variables) and we classify new data (i.e. with new values of the sepal and petal measurements) by which column has the maximum estimated value.

```
> library(nnet)
> data(iris3)
># use half the iris data
> ir <- rbind(iris3[,,1],iris3[,,2],iris3[,,3])
> targets <-class.ind(c(rep("s",50),rep("c",50),rep("v",50)))
> samp<-c(sample(1:50,25),sample(51:100,25),
+ sample(101:150,25))
>ir1 <- nnet(ir[samp,], targets[samp,], size=2, rang=0.1,
+                   decay=5e-4, maxit=200)
# weights:  19
initial  value 54.827508
iter  10 value 30.105123
iter  20 value 18.718125
… … … … … … … … … … … …
… … … … … … … … … … … …
iter 190 value 0.532753
iter 200 value 0.532392
final  value 0.532392
stopped after 200 iterations
>      test.cl <- function(true, pred){
+              true <- max.col(true)
+              cres <- max.col(pred)
+              table(true, cres)
+      }
>      test.cl(targets[-samp,], predict(ir1, ir[-samp,]))
    cres
true  1  2  3
   1 24  0  1
   2  0 25  0
   3  2  0 23
```

Thus, the classification rule only misclassifies 3 out of the 75 flowers which were not used in the analysis. If we used a net with only 1 unit in the hidden layer:

```
> ir1 <- nnet(ir[samp,], targets[samp,], size=1, rang=0.1,
+              decay=5e-4, maxit=200)
# weights:  11
initial  value 57.220735
iter  10 value 35.168339
…    …    …    …    …    …
iter  60 value 17.184611
final  value 17.167133
converged
>      test.cl <- function(true, pred){
+              true <- max.col(true)
+              cres <- max.col(pred)
+              table(true, cres)
+      }
>      test.cl(targets[-samp,], predict(ir1, ir[-samp,]))
    cres
true  1  2  3
   1 22  0  3
   2  0 25  0
   3  0  0 25
>
```

then it is still only 3, though a different 3 clearly. To see what the actual values of the predictions are we can print the first five rows of the estimated target values:

```
> predict(ir1, ir[-samp,])[1:5,]
             c         s v
[1,] 0.1795149 0.9778684 0
[2,] 0.1822938 0.9747983 0
[3,] 0.1785939 0.9788104 0
[4,] 0.1758644 0.9813966 0
[5,] 0.1850007 0.9714523 0
```

and we see that although it does not estimate the values as precisely (0,1,0) (or (1,0,0) or (0,0,1)) they are close. Hence the use of the `mac.col` function above.

We can find out more about the actual fitted (or trained) network, including the estimated weights with `summary()` etc:

```
> ir1
a 4-1-3 network with 11 weights
options were - decay=5e-04
> summary(ir1)
a 4-1-3 network with 11 weights
options were - decay=5e-04
 b->h1 i1->h1 i2->h1 i3->h1 i4->h1
 -0.15   0.41   0.74  -1.01  -1.18
 b->o1 h1->o1
 -0.06  -1.59
 b->o2 h1->o2
 -6.59  11.28
 b->o3 h1->o3
  3.75 -39.97
```

and we could draw a graphical representation putting in values of the weights along the arrows.

Another way of tackling the same problem is given by the following:

```
> ird <- data.frame(rbind(iris3[,,1], iris3[,,2], iris3[,,3]),
+          species=c(rep("s",50), rep("c", 50), rep("v", 50)))
>      ir.nn2 <- nnet(species ~ ., data=ird, subset=samp,
+ size=2, rang=0.1,  decay=5e-4, maxit=200)
# weights:  19
initial  value 82.614238
iter  10 value 27.381769
…    …    …    …    …
iter 200 value 0.481454
final  value 0.481454
stopped after 200 iterations
>      table(ird$species[-samp], predict(ir.nn2, ird[-samp,],
type="class"))

     c  s  v
  c 24  0  1
  s  0 25  0
  v  2  0 23
```

again, 3 of the new data are misclassified. However, if try a net with only one hidden unit we actually succeed slightly better:

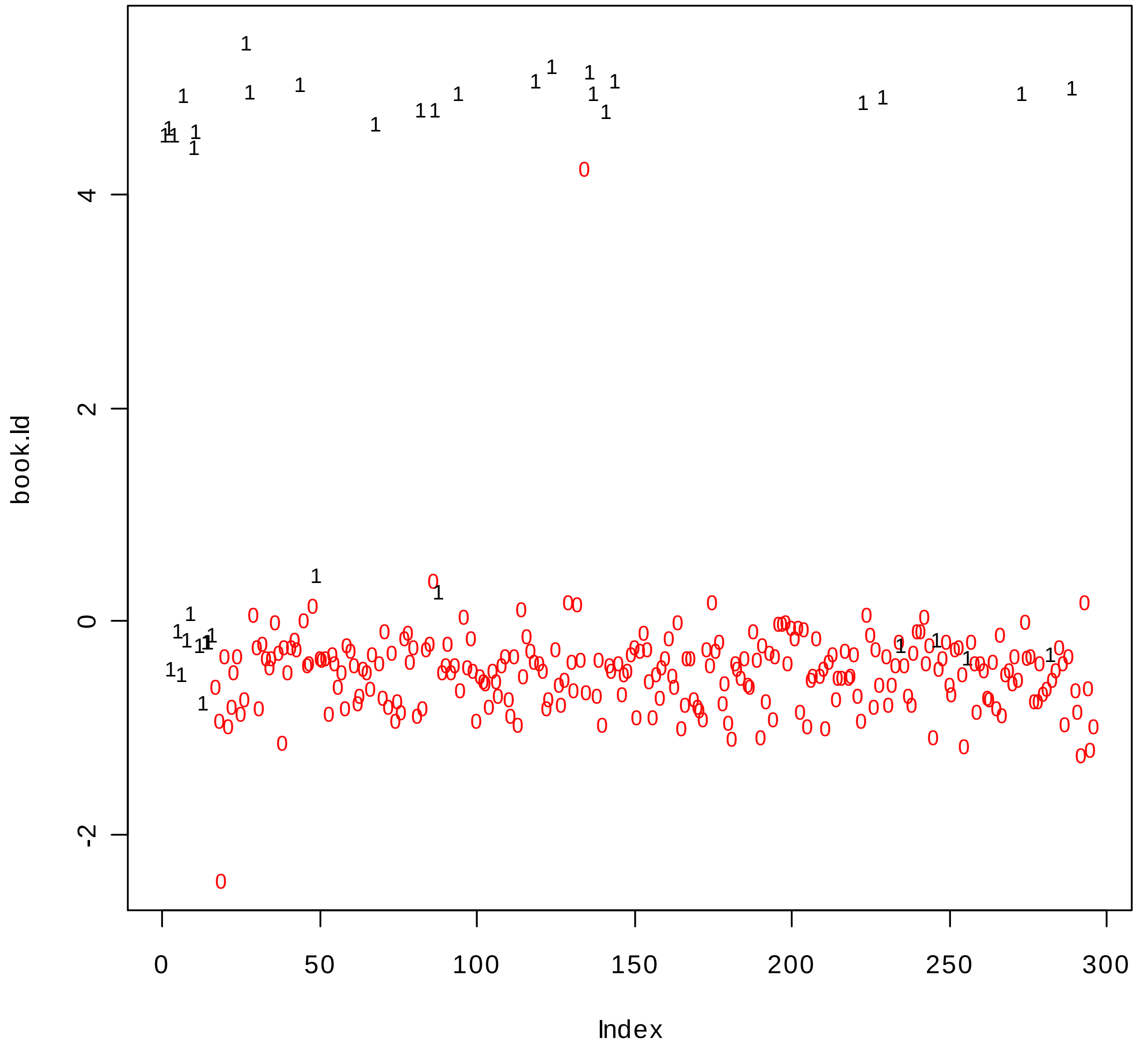
```
>  ir.nn2 <- nnet(species ~ ., data=ird, subset=samp, size=1,
+ rang=0.1, decay=5e-4, maxit=200)
# weights:  11
initial  value 82.400908
final  value 3.270152
converged
>    table(ird$species[-samp], predict(ir.nn2, ird[-samp,],
+ type="class"))

     c  s  v
  c 24  0  1
  s  0 25  0
  v  1  0 24


> summary(ir.nn2)
a 4-1-3 network with 11 weights
options were - softmax modelling  decay=5e-04
 b->h1 i1->h1 i2->h1 i3->h1 i4->h1
 -1.79  -0.44  -0.91   1.05   1.65
 b->o1 h1->o1
  7.11  -0.99
 b->o2 h1->o2
 12.30 -36.31
 b->o3 h1->o3
-19.45  37.43
```

Further experiments: Taking a random sample of 10 from each group I found that the misclassification rate on the new data was 6 out of 120 and even with training samples of 5 from each species it was 8 out of 135).

A7.4 Extended example: Data set Book

This data set has 16 variables, plus a binary classification (QT). The variables are a mixture of continuous (5 variables), binary (8 vars) and ordinal (3). An exploratory PCA [in MINITAB] on the correlation matrix (not shewn here) on 'raw' variables (i.e. ordinal not transformed to dummy variables) indicates very high dimensionality, typical of such sets with a mixture of types. The first 6 PCs account for 75% of variability, the first 9 for 90%. Plots on PCs indicate that there is some well-defined structure revealed on the mid-order PCs but strikingly the cases with QT=1 are clearly divided into two groups, one of which separates fairly well from cases with QT=0 but the other is interior to those with QT=0 from all perspectives. The background to this example reveals that it is particularly important to classify the QT=1 cases correctly and so rather than the overall or raw misclassification rate the more relevant measure is the misclassification rate of the QT=1 cases.

A Linear Discriminant Analysis emphasizes that these latter points are consistently misclassified. The plot below (from R) shews the data on the first (and only) crimcoord against sequence number. There then follows various analyses using a random subset to classify the remainder using both LDA and various simple neural nets. In this exercise LDA appears to win. Again the 'raw' variables are used for illustration but recoding would be better.



Plot of Book data on first discriminant (*vs* index in data file)

```
     Predicted
         0  1
true 0 256  1
     1   16 23
     17
```

Misclassification rates: raw=17/296(6%), QT=1 cases: 16/39(41%)

Next take random samples of 200 and then use the LDF obtained to classify the remaining 96 (details of code and some output suppressed):

```
> samp<- sample(1:296,200)
> books.lda<-lda(book[samp,],qt[samp])
> table(qt[-samp],predict(book.lda,book[-samp,])$class)
                    predicted
             0 1      0 1       0 1      0 1      0 1
   true   0 82 0   0 87 0    0 84 1   0 88 0   0 84 0
          1  7 7   1  2 7    1  3 8   1  4 4   1  6 6
misclassif
rates (raw)7        2         4        4        6   /96
(QT=1)     7/14     2/9       3/11     4/8      6/12

            0  1      0  1       0 1      0 1      0 1
   true  0 81  1  0 79  0    0 81 0   0 89 0   0 83 1
         1  4 10  1  6 11    1  7 8   1  2 5   1  4 8
misclassify
rates (raw)  5        6          7        2        5
(QT=1)       4/14     6/17       7/15     2/7      4/12
```

i.e. raw about 5% but QT=1 cases=37%

Now try a neural net with 8 hidden units:

```
> book.net<-nnet(book,qt,size=8,rang=0.1,decay=5e-4,maxit=200)
> q<-class.ind(qt)
> book.net<-nnet(book,q,size=8,rang=0.1,decay=5e-4,maxit=200)
> book.net
a 16-8-2 network with 154 weights
options were - decay=5e-04
> test.cl(q,predict(book.net,book))
    pred
true   0  1
   0 257  0
   1  11 28
    11
```

Raw misclassification rate 11/296 (3.7%), QT=1cases is 11/39 (39%).

Now again with 15 hidden units:

```
> book.net<-nnet(book,q,size=15,rang=0.1,decay=5e-4,maxit=200)
# weights:  287
> test.cl(q,predict(book.net,book))
    pred
true   0  1
   0 257  0
   1   9 30
    9
```

Raw misclassification rate 9/296 (3%) & QT=1 cases 9/39 (23%)

Now again with 20 hidden units:

```
> book.net<-nnet(book,q,size=20,rang=0.1,decay=5e-4,maxit=200)
# weights:  382
> test.cl(q,predict(book.net,book))
    pred
true   0  1
   0 257  0
   1   4 35
     4
```

Raw misclassification rate 4/296 (1.4%), & QT=1 cases 4/39 (10%)

Now try training the net on 200 randomly selected cases and classify the remaining 96.

```
> book.net<-
nnet(book[samp,],q[samp,],size=20,rang=0.1,decay=5e-
4,maxit=200)
# weights:  382
> test.cl(q[-samp,],predict(book.net,book[-samp,]))
    pred        pred       pred          pred        pred
true  0 1   true  0 1  true  0  1    true  0 1   true  0 1
   0 82 2      0 77 9     0 73 10       0 79 9      0 77 5
   1  6 6      1  5 5     1  6  7       1  3 5      1  8 6
misclassif
 rates
raw  8         14         16            12          13   /96
(QT=1) 6/12     5/10       6/13          3/8          8/14


    pred        pred        pred        pred        pred
true  0 1   true  0 1  true  0  1    true  0 1   true  0 1
   0 81 4      0 81 1      0 88 2      0 81 9      0 85 5
   1  6 5      1  7 7      1  2 4      1  2 4      1  2 4

misclassif
 rates
raw 10          8           4           11          7   /96
(QT=1) 6/11     7/14        2/6          2/6        2/6

   pred        pred
true  0 1   true  0  1
   0 86 4      0 80 10
   1  2 4      1  2  4
misclassif
 rates
raw    6           12
(QT=1) 2/6      2/6
```

i.e. overall about 10% and QT=1 cases 37%

Next, try this again with only 5 hidden units:

```
> book.net<-
nnet(book[samp,],q[samp,],size=5,rang=0.1,decay=5e-
4,maxit=300)
# weights:  97
    pred        pred        pred        pred        pred
true  0 1   true  0 1  true  0  1    true  0 1   true  0 1
   0 85 5      0 82 4      0 77 7      0 82 2      0 77 8
   1  2 4      1  3 7      1  6 6      1  7 5      1  4 7
misclassif
 rates
raw 7           7           13          9           12
(QT=1) 2/6      3/10        6/12        7/12        4/11

    pred        pred        pred        pred         pred
true  0 1   true  0 1  true  0  1    true  0 1   true  0 1
   0 76 5      0 78 5      0 79 5      0 75 10     0 78 6
   1  8 7      1  6 7      1  3 9      1  6  5      1  6 6
 misclassif
 rates
raw 13          11          8           16           12
(QT=1) 8/15     6/13        3/12          6/11         6/12

>
```

i.e. overall about 11% & QT=1 cases 44%

A7.4.1 Comment

The misclassification rate of the LDA on the QT=1 cases is about 40% on both the complete data set and on random samples of 96 when the rule is based on the remaining 200. Simple neural nets with large numbers of hidden units appear to do much better on the overall data,i.e about 110% of QT=1 cases are misclassified when the rulle calculated from the complete data is used but when tested on random samples of 96 not used in the construction of the rule (a more reliable assessment) the misclassification rate is about 40% and is comparable with that of the LDA. Thus this simple form of neural net offers no appreciable advantage over the LDA.

A7.5 SVMs

Simple feed-forward single hidden layer neural nets such as that used above typically fail in discrimination & classification problems because the cases are not *linearly separable*, as is revealed by the PCA. Support Vector Machines try to overcome this by initially mapping (via an *inner product Kernel function*) the input vectors into a higher dimensional space (perhaps very high) termed the ***feature space*** where they are linearly separable so that standard LDA can be applied. Different Kernel functions produce different learning machines such as polynomial, radial-basis or two-layer peceptron. For example, the first with Kernel function $(\mathbf{x}'\mathbf{x}_i+1)^p$ (p chosen *a priori* but **not** the dimension of the data) is equivalent to expanding the original variables to p-degree polynomials in them, e.g. in the case of original variable (x_1,x_2) and taking p=2 as well we obtain the variables in 5-dimensional feature space of $(x_1,x_2,x_1^2,x_2^2,x_1x_2)$. If p=3 then the feature space has dimension 9.

A7.5 Summary

The above account is only a very brief introduction to simple neural networks, with particular reference to their use for classification. As with tree-based methods they can also be used for regression problems.

Some comparisons win performance in classification with LDA methods has been given where it the importance of assessing performance on new data (rather than the training set) is highlighted.

Little has been said about the use and choice of activation functions, fitting criteria etc and examples have been given entirely in the context of the simple and basic facilities offered in the `nnet` library of **R**. To find out more then look at the reference Ripley (1996) given on p1, this is written with statistical terminology and largely from a statistical point of view. Another definitive reference is Chris Bishop (1995), *Neural Networks for Pattern Recognition,* Oxford, Clarendon Press. A readable but ‘black box’ account is given in Simon Haykin (1999), *Neural Networks*, (2nd Ed.) Macmillan, 1998.

APPENDIX 8: Kohonen Self-Organising Maps

Kohonen mapping is an *unsupervised* technique (i.e. essentially a form of non-hierarchical cluster analysis) but which has similarities with feed-forward neural nets. The objective is to produce a classification of objects into 'clusters' of similar cases and simultaneously construct a map of these clusters with neighbouring clusters as similar as possible. Kohonen says (quoted in Ripley, 1996): "I just wanted an algorithm that mapped similar patterns (pattern vectors close to each other in input space) into contiguous locations in output space". A good account is given in Everitt *et al,* (2001).

Viewed as a network, it consists of two layers —

- an input layer of p-dimensional observations
- an output layer (represented by a grid) consisting of k nodes of the k clusters

Initially each node has a p-vector of weights associated with it, initially arbitrarily chosen (e.g. as random weights between 0 and 1). The iteration proceeds by associating each observation to that node to which it is closest (inner product of observation and node weight). The winning node (and to a lesser extent, the neighbouring nodes) are 'rewarded' by moving the weights of that node[s] towards the observation. The process cycles round the complete set of observations until stability is reached, perhaps varying the degree of movement of nodes towards observations to a smaller and smaller amount.