OTHER WORKSHEETS FOR R/S-PLUS: A MISCELLANY

P.M.E.Altham, Statistical Laboratory, University of Cambridge.

October 1, 2008

These worksheets were originally constructed for my graduate teaching before I retired in September 2005. Since then, I have added some examples and graphs, and also made some minor editorial changes. For example, this new version of the worksheets includes a small index of the **commands** used, and also of the datasets.
If you have any comments or queries, please contact me at

p.m.e.altham@statslab.cam.ac.uk
http://www.statslab.cam.ac.uk/~pat

Special thanks must go to Dr R.J.Gibbens for his help in introducing me to S-Plus, and also to Professor B.D.Ripley for access to his S-Plus lecture notes. Several generations of keen and critical students for the Cambridge University Diploma in Mathematical Statistics, and since 1998 for the MPhil in Statistical Science, have made helpful suggestions which have improved these worksheets.

Most of the multivariate theory used is explained in my Lecture Notes at

http://www.statslab.cam.ac.uk/~pat/AppMultNotes.ps

These worksheets form a companion set to "Introduction to S-Plus for Generalized Linear Modelling", or (more recently) to my R worksheets for a similar course, which are available at

http://www.statslab.cam.ac.uk/~pat/redwsheets.ps

Nearly all of the examples given below will work in R, the free software (see link on my webpage).
Aristotle said
'For the things we have to learn before we can do them, we learn by doing them.'
This is a quotation I found at the start of the book by B.J.T.Morgan, 'Applied Stochastic Modelling', published by Arnold (2000).

Table of contents.

7. Hierarchical clustering.
8. Decision trees: the autolander data for the space shuttle.
9. Time series analysis.
10. Survival data analysis.
11. The British monarchy data: a question.
12. Classical multidimensional scaling and Chernoff's faces on student data.
Also, human rights abuses in 11 different countries.
13. A repeated measures design.
14. Fitting the beta-binomial distribution to Marshall and Spiegelhalter's data on *in vitro* fertilisation (52 British clinics).
15. Multinomial logistic regression and classification. New for August 2008: distribution of British Olympic medals for the last 10 Olympic games.
16. New for July 2003: Mohammad Raza's multivariate data on 50 famous films.
17. Eight men behaving badly (2004), and Hawks and Doves at the Monetary Policy Committee (2007).
18. New for December 2005: capture-recapture data. How many snowshoe hares are there in a given closed population? How many individuals with alcohol-related problems are there in a given closed population in a region of Northern Italy?

.....

References.

Venables, W.N. and Ripley, B.D. (2002) Modern Applied Statistics with S-Plus. New York: Springer-Verlag. Also, any of the 3 previous editions this book.
Webb, A. (1999) Statistical Pattern Recognition. London: Arnold (this shows the relevance of multivariate analysis to the topic of Statistical Pattern Recognition.)

Note added April 2008 A very interesting article by Michael Friendly, which has a good online dataset and some marvellous graphics, is

'A.-M.Guerry's *Moral Statistics of France*: Challenges for Multivariable Spatial Analysis', in *Statistical Science*, **22**, 368-399.

This is based on a nineteenth-century dataset.

Those of you with interests in financial mathematics, eg for your projects, should try

```
Splus6
module(finmetrics)
```

for example for fitting GARCH models, or copula models.

1. Classical statistical tests: t-tests and non-parametric.

The 2-sample t-test and the 2-sample Wilcoxon test

Notation: let $(x_1, x_2, \ldots, x_m)$ and $(y_1, y_2, \ldots, y_n)$ be independent random samples from the distribution functions $F(.), G(.)$ respctively.
If we know that F, G correspond respectively to $N(\mu_1, \sigma^2)$ and $N(\mu_2, \sigma^2)$ then the optimum test of
$H_0 : \mu_1 = \mu_2$ against the alternative $H_1 : \mu_1 < \mu_2$ is achieved by the '2-sample t-test', and here is an example, for a very small and obvious set of data.

```
  >x <- scan()
  3.7 2.1 4.5 7.1

  >y<- scan()
   6.1 7.9 10.3 11.4 13.7

  >summary(x)
    Min. 1st Qu. Median Mean 3rd Qu. Max.
      2.1     3.3    4.1 4.35    5.15  7.1

  >summary(y)
       Min. 1st Qu. Median Mean 3rd Qu. Max.
         6.1     7.9   10.3 9.88    11.4 13.7
 >t.test(x,y, alt ="less")

   Standard Two-Sample t-Test

  data:  x and y
  t  = -3.1364, df  = 7, p-value  = 0.0082
  alternative hypothesis: true difference in means is less than 0
  95 percent confidence interval:
  NA -2.189557

sample estimates:
 mean of x mean of y
   4.35      9.88
```

Thus here we reject H_0 in favour of H_1. Observe that here, $\mu_1 < \mu_2$ implies that $F(x) > G(x)$ for all x, ie the x's tend to be smaller than the y's.
But what if we want to test $HP_0 : F(x) = G(x)$ for all x against the alternative $HP_1 : F(x) > G(x)$ for all x, without making a specific assumption about the *shape* of F, G?
It turns out that we can get a long way (and in fact produce tests that are really rather efficient) simply by considering the *ranks* of the observations x_i, y_j.
This is what 'nonparametric' (or more accurately, 'distribution-free') statistical tests achieve, and as such they have a long history.
First, we find the ranks of $(x), (y)$ in the *combined sample*, which has $4 + 5 = 9$ elements.

```
  > rank(c(x,y))
   [1] 2 1 3 5 4 6 7 8 9
```

Then we find W, the sum of the ranks of $(x_1, \ldots, x_m)$ in the combined sample; here $W = 2 + 1 + 3 + 5 = 11$. We reject HP_0 in favour of HP_1 if W is sufficiently SMALL, say if $W \leq c$, where

$P(W \leq c|HP_0) = .05$, say.
The beauty of non-parametric statistics is that we can compute the 'null distribution' of W purely from m, n, the respective sample sizes.

```
>wilcox.test(x,y, alt ="less")

  Exact Wilcoxon rank-sum test

   data:  x and y
  rank-sum statistic W  = 11, n  = 4, m  = 5, p-value  = 0.0159
  alternative hypothesis: true mu is less than 0
```

How is the p-value computed? Note that under the null hypothesis, we can say by symmetry that all the orders of the $x_1, \ldots, x_m, y_1, \ldots, y_n$ are equally likely, and each such order must therefore have probability

$$1/\binom{m+n}{m} = q \text{ say.}$$

Here's how we build up the null distribution of W.
You can check that by definition, $q = .007936$. Further, by definition, $W \geq 10$, and $W = 10(= 1+2+3+4)$ with probability q.
And $W = 11 = (1+2+3+5)$ with probability q also, hence

$$P(W \leq 11|HP_0) = 2 \times q = .0159.$$

Note that in general (for reasonable sorts of distributions, in fact) the non-parametric test is **conservative** with respect to the corresponding t-test (we are throwing away some data by using only ranks) so that we should expect that the non-parametric test will have a larger p-value than the corresponding t-test.

Now we consider a new problem, **tests for paired samples**. Suppose we have data $(x_1, y_1), \ldots, (x_n, y_n)$, a random sample from the bivariate distribution function $F(x, y)$. We wish to test the hypothesis $HP_0 : F(x, y) = F(y, x)$ for all x, y against the alternative hypothesis HP_{alt} that the x's tend to be smaller than the corresponding y's. In the example given below, $n = 6$, and it is fairly obvious that the x's tend to be less than the y's, but the sample size is rather small. Now we know that if $F(.,.)$ is bivariate normal, then the optimum test of HP_0 against HP_{alt} is the paired sample t-test, carried out as follows:

```
> cbind(x,y,y-x)
        x     y
[1,] 12.3 12.43  0.13
[2,] 14.4 14.71  0.31
[3,]  2.3  2.97  0.67
[4,]  5.1  5.98  0.88
[5,]  6.7  6.12 -0.58
[6,]  9.1  9.99  0.89

>summary(y-x)
 Min. 1st Qu. Median   Mean 3rd Qu. Max.
-0.58   0.175   0.49 0.3833  0.8275 0.89


 > t.test(x,y,paired =T,alt = "less")
```

```
    Paired t-Test

    data:  x and y
     t  = -1.6687, df  = 5, p-value  = 0.078
     alternative hypothesis: true mean of differences is less than 0
     95 percent confidence interval:
             NA 0.07956285
     sample estimates:
     mean of x - y
          -0.3833333
```

Hence, the corresponding p-value is 0.078, so that at level 10% we reject HP_0 in favour of HP_{alt}.
How can we carry out the corresponding test if we make no assumption about the shape of $F(.,.)$?
Here's the way we do it.
Put $z_i = y_i - x_i$, then $z_1, \ldots, z_n$ is a random sample from the distribution function $G(.)$ say.
We test $H_0 : G(z) = 1 - G(-z)$, ie G corresponds to a pdf symmetric about 0,
against H_1, G corresponds to a pdf symmetric about a point > 0.
So, we compute $z_i = y_i - x_i$, find the ranks of $|z_i|, 1 \leq i \leq n$
and then compute as our test statistic V, defined as the sum of the ranks of the $|z_i|$ for which $z_i < 0$.

```
  > rank(abs(y-x))
 [1] 1 2 4 5 3 6
  > abs(y-x)
 [1] 0.13 0.31 0.67 0.88 0.58 0.89

  > wilcox.test(x,y,paired =T,alt ="less")

        Exact Wilcoxon signed-rank test

 data:  x and y
 signed-rank statistic V  = 3, n  = 6, p-value  = 0.0781
 alternative hypothesis: true mu is less than 0
```

How is the p-value computed?
Here it is $P(V \leq 3|H_0)$ and so we see that it is $P(V = 0, 1, 2 \text{ or } 3|H_0)$.
Let M = number out of $z_1, \ldots, z_n$ which are < 0. Then it can easily be seen that on H_0, M is distributed as $Bi(n, 1/2)$.
Hence ... it can be shown that, on H_0,

$$P(V = 0) = 1/2^6 = P(V = 1) = P(V = 2)$$

and

$$P(V = 3) = P(V = 1 + 2 \text{ or } V = 3) = 1/2^6 + 1/2^6$$

giving $P(V \leq 3|H_0) = 5/2^6 = .0781$ as given.
This way we can build up the null distribution of V, our test statistic, without even knowing the parent distribution $G()$.
For large n the asymptotic null distribution of V is normal, with mean and variance which are known functions of n, and a corresponding result holds for the 2-sample Wilcoxon test. You will find that R and SPlus use these asymptotic results to compute the p-values for large sample sizes.
Here is a very quick illustration, on the same very small sample, of **bootstrap methods**, here used to find 2 slightly different versions of a 95% confidence interval for the mean.

```
>z # this is our sample, of size 6.
[1]  0.13  0.31  0.67  0.88 -0.58  0.89
>t.test(z) # this will give a 95% confidence interval for mu,
# based on the assumption that the z's form a random sample
# from a Normal distribtion.
        One Sample t-test

data:  z
t = 1.6687, df = 5, p-value = 0.1560
alternative hypothesis: true mean is not equal to 0
95 percent confidence interval:
 -0.2071798  0.9738465
sample estimates:
mean of x
0.3833333
```

Now we will use the bootstrap library to find our confidence intervals for the mean. This construction does not depend on the assumption of normality. We generate 1000 bootstrap samples, and compute the mean for each such sample. Each sample is drawn *with* replacement from the original $z_1, \ldots, z_6$.

```
>library(boot)
>set.seed(1.7) # the arbitrary choice 1.7 ensures we get the same result each time
> z.boot = boot(data=z, statistic = function(x,i) mean(x[i]) , R =1000)
> boot.ci(z.boot, type=c("perc", "bca"))

BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
Based on 1000 bootstrap replicates

CALL :
boot.ci(boot.out = z.boot, type = c("perc", "bca"))

Intervals :
Level     Percentile            BCa
95%   (-0.0467,  0.7567 )   (-0.1350,  0.7183 )
Calculations and Intervals on Original Scale
```

Now a different example for you to try. The datafile
taxrevenue
contains, as rows, the taxrevenue for sales of
tobacco, spirits, beer, wine, cider and sherry, betting.
The columns are 1989-90,1990-91,1991-92.
These data are from " The Independent" Aug 18, 1993. (Note, data from a newspaper, while interesting and topical, does not usually constitute a "random sample". We press on regardless.)
Here is the dataset taxrevenue

```
  x       y       z
5035.3  5636.0  6289.5
1513.5  1703.0  1742.1
2074.2  2290.0  2324.9
 791.2   855.3   924.5
  58.8    68.6    73.8
 976.1  1006.4  1052.8
```

These data provide an opportunity for a tour through some S-Plus classical tests.

```
tax <-  read.table("taxrevenue", header  = T)
tax
attach(tax)
a <- (y-x)/x ; b <- (z-y)/y # we compare relative increases.
#first, one-sample tests on a.
a ; summary(a)
t.test(a,mu=.10)
names(t.test(a))
t.test(a)$conf.int
# Now the nonparametric version of this.
wilcox.test(a, mu =.10)
t.test(a,mu =.01) ; wilcox.test(a,mu =.01) #compare p-values.
```

Now compare a with b, but pretending a, b independent of each other.

```
t.test(a,b)             # This assumes the 2 variances are equal.
t.test(a,b,var.equal =F) # This doesn't assume the 2 variances equal.
```

But, the above were WRONGLY applied: they assumed independent a, b. So now we do it correctly, ie allowing for the PAIRING of a, b.

```
t.test(a,b,paired =T)
wilcox.test(a,b,paired =T)
```

Next we demonstrate 2 methods of testing the independence of a, b. The first, which uses the Pearson correlation coefficient, is effectively assuming that we have a random sample from a bivariate normal distribution. The second, constructed by Spearman in the context of intelligence-testing, tests for independence of a, b without making any assumption on the parent distribution $F(a, b)$, this is therefore a non-parametric test. It essentially replaces a_i, b_i by their ranks, eg $(1, 3), \ldots, (n, n)$ and works out the corresponding Pearson coefficient. On the null hypothesis of independence of a, b this has known distribution, depending only on n, the sample size.

```
cor.test(a,b)
cor.test(a,b,method ="spearman")
help(friedman.test) # experiment with this new non-parametric test.
# can you apply it to the data x, y, z ?
# Try some plots. Do they enlighten you ?
i <- 1:3 ; ttax <- t(tax)
matplot(i, ttax, type ="l") # might help
```

Here is another dataset, this time from The Independent, June 30, 1999, on the safety of multi-purpose vehicles (MPV's). The 8 types of vehicle were subjected to 'Front Impact' tests (in which the front impact takes place at 40mph (64kph)) and 'Side Impact' tests, in which the side impact takes place at 30mph (50kph)). The corresponding scores are given in the Table below: the higher the score, the better the vehicle.

	Frontal_score(%)	Side_Score(%)
RenaultEspace	67	100
ToyotaPicnic	61	93
Peugeot806	42	93
NissanSerena	34	100
VolkswagenSharan	36	96
MitsubishiSpWagon	24	96
Opel/VauxhallSintra	21	93
ChryslerVoyager	0	89

Questions for you:
i) Is the Frontal Score significantly less than the Side Score?

ii) Is there a positive association between these two scores?

'How the world is getting hungrier each year' is the headline in The Independent of 26 November, 2003, showing the following distressing data: for the following 40 countries, the percentage of the population that is undernourished, for the years 1999-2001, and for 1990-92.

		y99-01	y90-92
1	DRCongo	75	31
2	Somalia	71	68
3	Burundi	70	49
4	Afghanistan	70	58
5	Eritrea	61	63*
6	Mozambique	53	69
7	SierraLeone	50	46
8	Zambia	50	45
9	Haiti	49	65
10	Angola	49	61
11	CAR	44	50
12	Tanzania	43	35
13	Ethiopia	42	57*
14	Liberia	42	33
15	Rwanda	41	43
16	Zimbabwe	39	43
17	Mongolia	38	34
18	Cambodia	38	43
19	Kenya	37	44
20	Madagascar	36	35
21	Niger	34	42
22	Chad	34	58
23	NKorea	34	18
24	Yemen	33	35
25	Malawi	33	49
26	Bangladesh	32	35
27	Congo	30	37
28	Nicaragua	29	30
29	Guinea	28	40
30	PNewGuinea	27	25
31	Cameroon	27	33
32	Gambia	27	22
33	Iraq	27	7
34	Panama	26	20
35	Guatemala	25	16
36	Lesotho	25	27
37	Togo	25	33
38	DominicanR	25	27
39	Sudan	25	31
40	SriLanka	25	29

* corresponds to 1995-97, as the earlier figure was unavailable.

New for July 2008: The Tompkins Table for Cambridge Colleges Examinations results, 2000−2008. Each year The Independent publishes the examination rank order of the 29 Cambridge Colleges: Emmanuel has been at the top of the Table for each 2006 and 2007, but now (ie 2008) Selwyn is top.
Here is the Table of ranks for each of the last 9 year (note that certain colleges were only included in this Table from 2003 onwards). Suggestions for a non-parametric test, and a plot of the various college 'tracks' over the 8 years, are given below. First, here is the dataset.

```
College        y00  y01  y02  y03  y04  y05  y06  y07 y08
Christs         1   1   4   2   2   4   6   2 8
Churchill      15   9 10   9 19 18 13 15 6
Clare           9   6   3   6   4   9 12 17 13
CorpusC        10 20 18   7 10 16   8   8 9
Downing         8 10   8 12 17 15 11   3 12
Emmanuel        3   2   2   1   1   5   1   1 2
Fitzwilliam    21 13 20 20 15 13 19 14 21
Girton         18 17 16 17 25 24 22 21 22
G&Caius        12   8   7   4   5   2   2 10 4
Homerton       NA NA NA 25 24 26 25 26 25
HughesH        NA NA NA 27 27 29 29 29 26
Jesus          13 11   9 10   9   7 10   9 7
Kings          20 21 14 16 20 10 17 18 19
LucyC          NA NA NA 26 26 27 26 24 28
Magdalene      22 22 15 18 22 20 20 13 5
NewHall        16 23 24 24 23 25 24 23 23
Newnham        24 24 22 21 13 21 23 22 24
Pembroke        6   7   1   3   6   6   4   7 10
Peterhouse     14 19 23 22 21 22 21 25 17
Queens          5   5   5   5   8   8 14 11 16
Robinson       19 14 21 23 16 11 18 20 18
StCaths        11 18 12 11   7   1   3   5 11
StEdmunds      NA NA NA 29 29 28 28 28 29
StJohns         4   4 11 13 14 12 15 19 20
Selwyn          7 12 13 14 11 19   7   4 1
SidneyS        23 16 19 15 18 14   9 12 14
Trinity         2   3   6   8   3   3   5   6 3
TrinHall       17 15 17 19 12 17 16 16 15
Wolfson        NA NA NA 28 28 23 27 27 27
```

```
Tompkins <- read.table("Tompkins", header=T)
Tompkins <-Tompkins[-c(10,11,14,23,29),]#to remove the incomplete rows
matTomp <- as.matrix(Tompkins[,2:10])
friedman.test(t(matTomp)) #
```

Note that we transpose the matrix in order to test for the differences between the 24 colleges. The Friedman test results in a chi-squared statistic of 150.82 on 23 df, apparently showing that there are indeed systematic differences between these 24 colleges. However, this use of the Friedman test may not be strictly valid, since consecutive years ('blocks' in the parlance of the Friedman test) will not be **independent**. Each Tompkins score, for a given year and a given college, is obtained from the examination results of students from years 1, 2 and 3 of that college. Thus typically a particular student, arriving in say autumn 2001, will contribute to the scores of his/her college in 2002, 2003 and 2004.

```
> round(cor(matTomp),2)
     y00  y01  y02  y03  y04  y05  y06  y07  y08
y00 1.00 0.80 0.76 0.75 0.70 0.55 0.66 0.62 0.50
y01 0.80 1.00 0.83 0.75 0.69 0.65 0.62 0.54 0.47
y02 0.76 0.83 1.00 0.91 0.72 0.71 0.72 0.67 0.60
y03 0.75 0.75 0.91 1.00 0.78 0.71 0.83 0.73 0.65
y04 0.70 0.69 0.72 0.78 1.00 0.81 0.79 0.67 0.55
y05 0.55 0.65 0.71 0.71 0.81 1.00 0.78 0.62 0.45
y06 0.66 0.62 0.72 0.83 0.79 0.78 1.00 0.86 0.78
y07 0.62 0.54 0.67 0.73 0.67 0.62 0.86 1.00 0.77
y08 0.50 0.47 0.60 0.65 0.55 0.45 0.78 0.77 1.00
```

This does indeed fit in with the suggestion of positive correlation between successive years. In this case I suspect that the Friedman test statistic of 150.82 should actually be 'deflated' by a suitable factor (but what is this?) before referring it to the χ^2 distribution. Now we show a method of plotting the 'time tracks' of the 24 colleges.

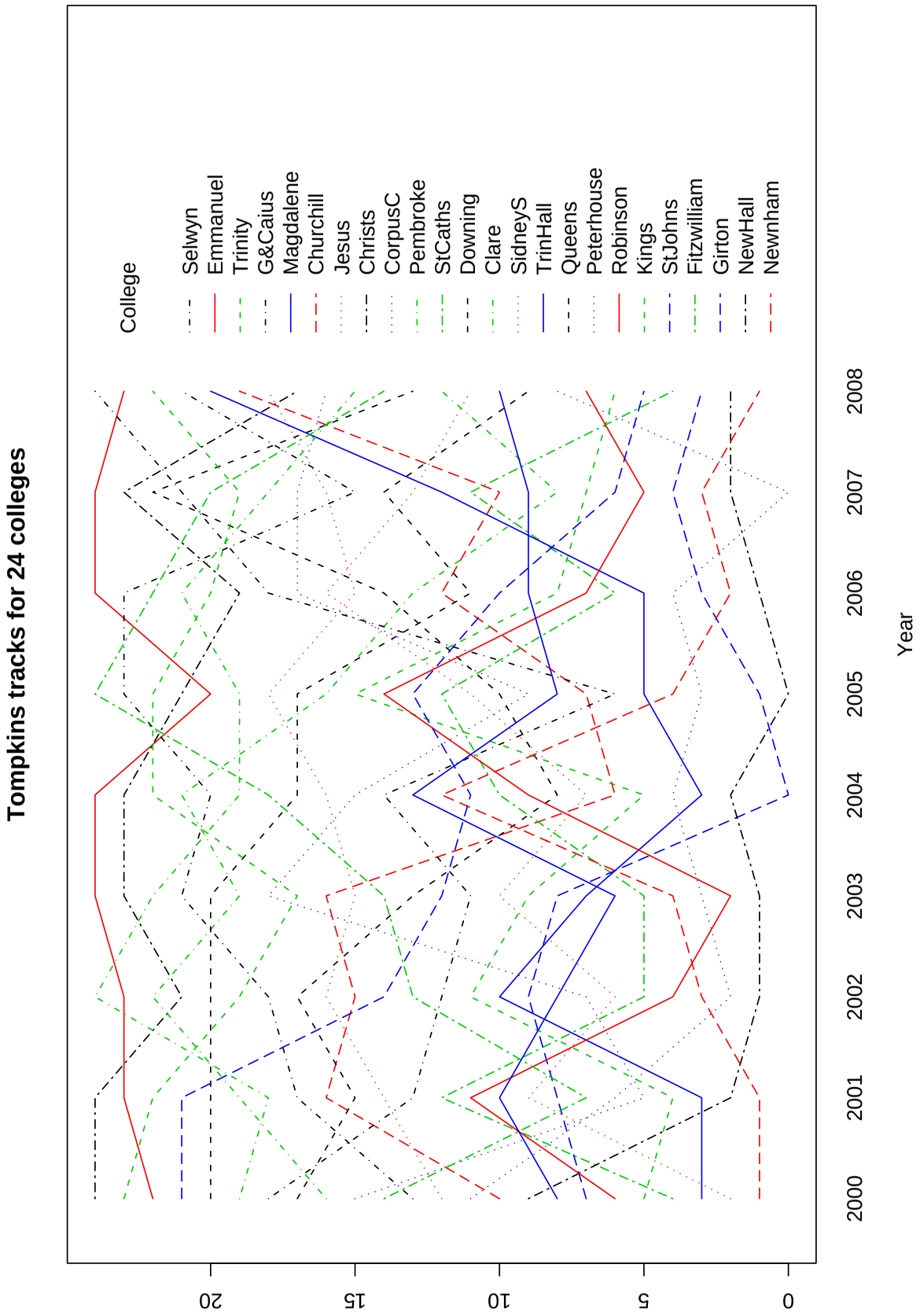
```
college <- Tompkins[,1] # to set up college names
# we could use 'matplot' to plot the tracks of the individual colleges,
# but 'interaction.plot' turns out to be quicker to use
v <- as.vector(matTomp) # this reads DOWN the rows
College <- gl(24,1, length=216, labels = college)
Year <- gl(9,24, length=216, labels=2000:2008)
y <- 25-v # to make graph give 'best' college at the TOP
interaction.plot(Year, College, y, col=c("black", "red", "green3", "blue"), ylab="")
```

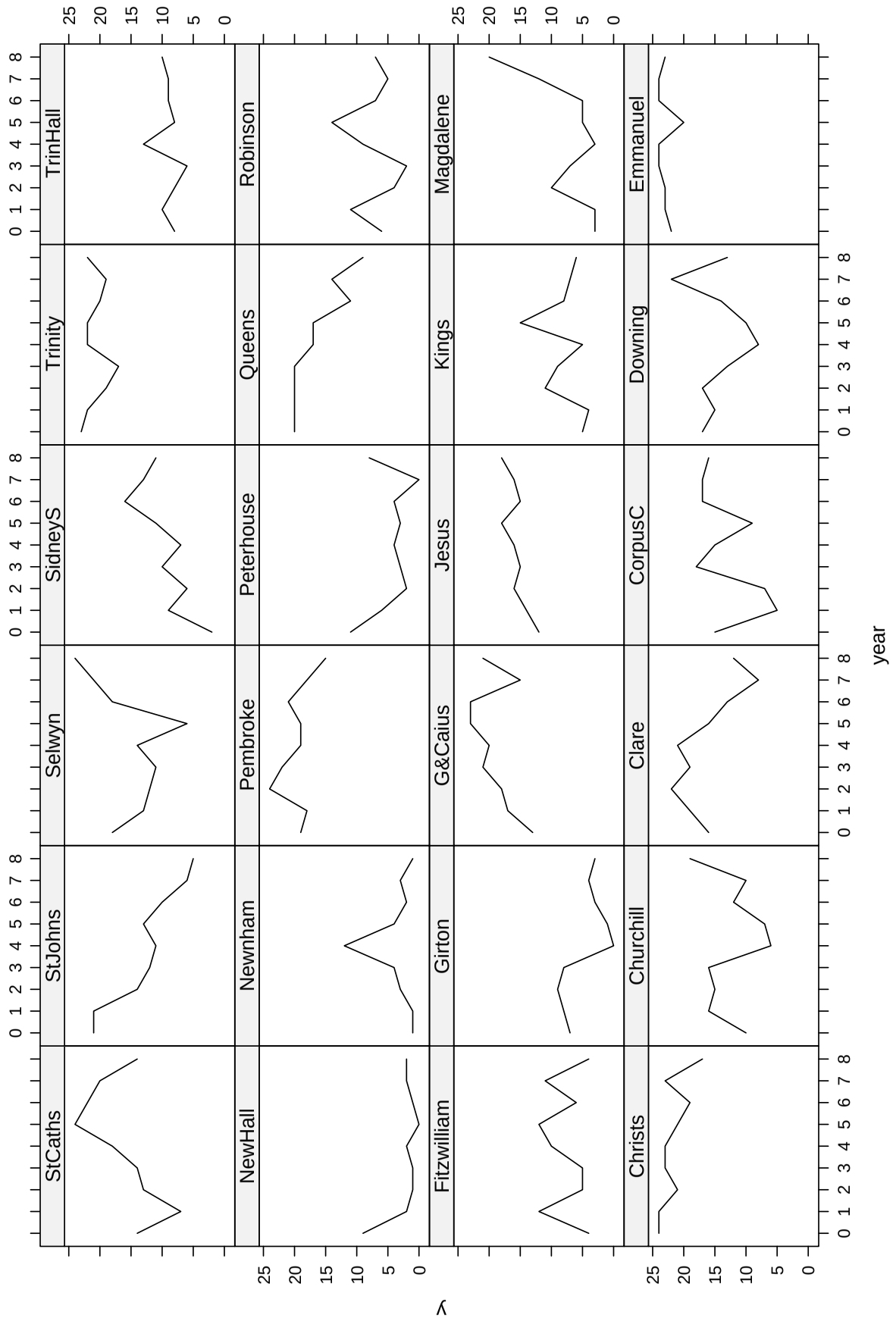
This results in Figure 1. Some of the middling 'tracks' do seem to go all over the place.
Following a suggestion by Dr Richard Gibbens, we could also also plot the 'tracks' another way, resulting in Figure 2.

```
library(lattice)
year <-  gl(9,24, length=216, labels=c(0:8)) # to reduce clutter on plot
xyplot(y~year|College, type="l")
```

Here is a fuller version of the Tompkins Table for 2008, in rank order.

```
 College score  %firsts
1 Selwyn   68.47   29.9
2 Emmanuel 68.30   30.6
3 Trinity  68.27   31.4
4 G&Caius  67.33   27.9
5 Magdalene 65.97  24.5
6 Churchill 65.72  27.1
7 Jesus     65.60  25.2
8 Christs   65.27  25.7
9 CorpusC   65.24  24.1
10 Pembroke 64.96  24.5
11 StCaths  64.63  23.5
12 Downing  64.48  22.8
13 Clare    64.44  22.5
14 SidneyS  64.22  20.9
15 TrinityH 63.76  19.3
16 Queens   63.58  22.3
17 Peterhouse 63.21 22.9
18 Robinson   63.20 20.6
```


Tompkins tracks for 24 colleges
College
Selwyn
Emmanuel
Trinity
G&Caius
Magdalene
Churchill
Jesus
Christs
CorpusC
Pembroke
StCaths
Downing
Clare
SidneyS
TrinHall
Queens
Peterhouse
Robinson
Kings
StJohns
Fitzwilliam
Girton
NewHall
Newnham
0
5
10
15
20
2000
2001
2002
2003
2004
2005
2006
2007
2008
Year


StCaths
StJohns
Selwyn
SidneyS
Trinity
TrinHall
NewHall
Newnham
Pembroke
Peterhouse
Queens
Robinson
Fitzwilliam
Girton
G&Caius
Jesus
Kings
Magdalene
Christs
Churchill
Clare
CorpusC
Downing
Emmanuel
y
year

```
19 Kings      63.07 22.5
20 StJohns    62.48 20.5
21 Fitzwilliam 61.08 18.2
22 Girton     60.84 15.3
23 NewHall    60.03 13.9
24 Newnham    59.96 13.3
25 Homerton   58.62 13.0
26 HughesHall 56.36 20.8
27 Wolfson    55.15  7.4
28 LucyC      52.61  8.7
29 StEdmunds  51.56 11.2
```

So you see that the two columns are correlated, but not perfectly correlated.
You may well want to know exactly how the Tompkins score is computed. The table allocates 5, 3, 2, 1, 0 points respectively for each of a First, a 2-1, a 2-2, a 3rd and 'granted an allowance'. (I think that complete failures are not counted at all.) Then to quote Wikipedia
"The scores in each subject are then weighted to a common average, to avoid the bias towards colleges with higher proportions of students entered for subjects which receive higher grades. The result is then expressed as a percentage of the total number of points available."
(Hence for construction of the Tompkins table, a First in Mathematics (for example) for college x 'counts' less than a First in English for college x.)

The Independent, 5 August 2008, presents the following cricketing data, under the headline 'For better or Worse: England's captains' performances since Tony Greig. (This item follows the resignation of Michael Vaughan as England's Captain.) We can compare the batting average of an individual player when he was Captain with his batting average when he was not Captain.
With the help of my colleague Dr Richard Samworth (using cricinfo) I have corrected the figures given in the Independent for **Stewart**.

```
          MC AvC   MnC AvnC
Greig     14 38.04 44 41.32
Brearley  31 22.48  8 24.28
Botham    12 13.14 90 36.74
Willis    18 21.59 72 26.31
Gower     32 43.59 85 45.50
Gatting   23 44.05 56 32.21
Gooch     34 58.72 84 35.93
Atherton  54 38.73 61 35.25
Stewart   15 39.22 118 39.59
Hussain   45 36.04 51 38.10
Vaughan   37 36.02 45 50.98
Flintoff  11 33.23 58 32.32

key: MC= number of matches as Captain,
   AvC = batting average  as Captain
   MnC = number of matches not as Captain,
  AvnC = batting average not as Captain.
```

Here is my suggestion for plotting the data. You may also like to think of some suitable non-parametric tests: eg is the batting average of a Captain smaller than his batting average when not a Captain? Can you do anything useful with the information on the numbers of matches played?

```
 Cricket <- read.table("Cricket.data", header=T)
attach(Cricket)
```

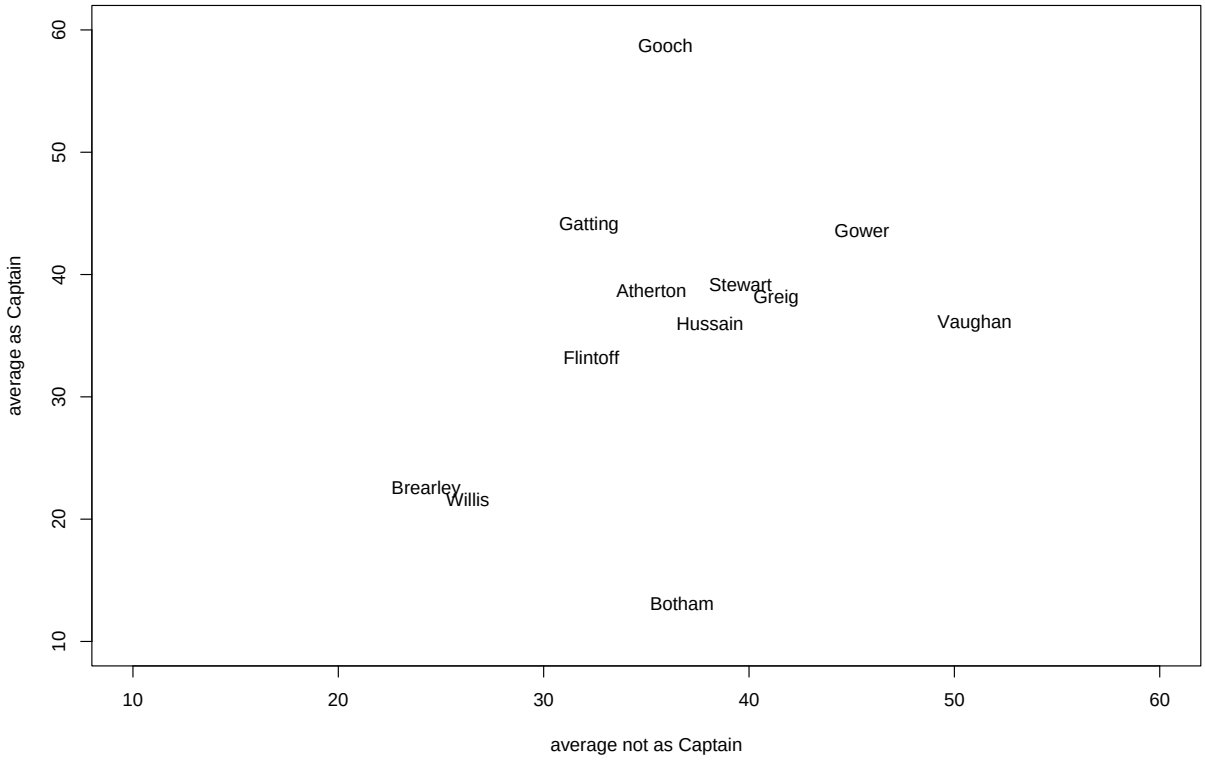


Figure 3: Comparing the batting averages of England Captains

```
captains <- row.names(Cricket)
plot(AvC ~ AvnC, type="n", xlab = "average not as Captain",
  ylab = "average as Captain", xlim=c(10,60), ylim=c(10,60))
text(AvC ~ AvnC, labels=captains)
```

This results in Figure 3 as shown.

2. Getting started in S-Plus for Multivariate Analysis

We start by simulating a sample of 200 observations from a given 3-dimensional normal. (You could do this via the function rmvnorm() if you prefer.)

```
i <- 1:200
x <- rnorm(i)   # to generate 200 NID(0,1) rvs
y <- rnorm(i)   # to generate a further set of 200 NID rvs
z <- rnorm(i)   # and again.
v1 <- 2*x + y + 29
summary(v1)
v2 <- x+z+10
v3 <- 3*x + 72  # now (v1,v2,v3) form a r.s. of 200 observations
 # from a specified 3-variate normal distribution.
a <- cbind(v1,v2,v3) # a is the corresponding data-matrix
plot(v1,v2)
pairs(a)
brush(a)       # can you see what's going on ?
options(digits =4)   # makes things easier on the eye
b.cov <- var(a) # the sample covariance matrix
b.cov          # how close is it to the true covariance matrix?
b.cor <- cor(a) # the sample correlation matrix
b.cor
b.lm <- lm(v1~ x+y)
summary(b.lm)
hist(v1)       # does this look as you would expect ?
e.cov <- eigen(b.cov) ; names(e.cov) # eigen-values etc
e.cor <- eigen(b.cor)
e.cov ; e.cor  # why are these 2 sets of e-vals different ?
   #(Now we do DIY calculation of sample covariance matrix)
col.means <- apply(a,2,mean)
col.means ; help(apply)
col.resid <- sweep(a,2,col.means) ; help(sweep)
# "apply" & "sweep" are not terms you'ld ever have thought of !
cov.diy <- t(col.resid) %*% col.resid  # t( ) is transpose
# You can probably find a more elegant way of computing cov.diy.
cov.diy <- cov.diy/199                 # %*% is matrix mult'n
cov.diy ;b.cov                        # for comparison
 # Here's another useful function.
b <- scale(a,center =T,scale =T)         # NB U.S. spelling
pairs(b)
var(a) ; var(b)
cor(a) ; cor(b) # Now try a Hotelling T-Test.
```

Now we set up a function to compute the bivariate normal density function, for correlation coefficient ρ, calculate this density at each point in a 20×20 grid, and demonstrate three ways of plotting this density.
We compute

$$f(x,y) = \exp -(x^2 - 2\rho xy + y^2)/2(1-\rho^2).$$

```
x <- seq(-2,2, length= 40); y <- x; rho <- .7
bivnd <- function(x,y){
exp(-(x^2 - 2*rho *x *y +y^2)/(2*(1- rho^2)))
}
z <- x %*% t(y)   # to set up z as a matrix of the right size
```

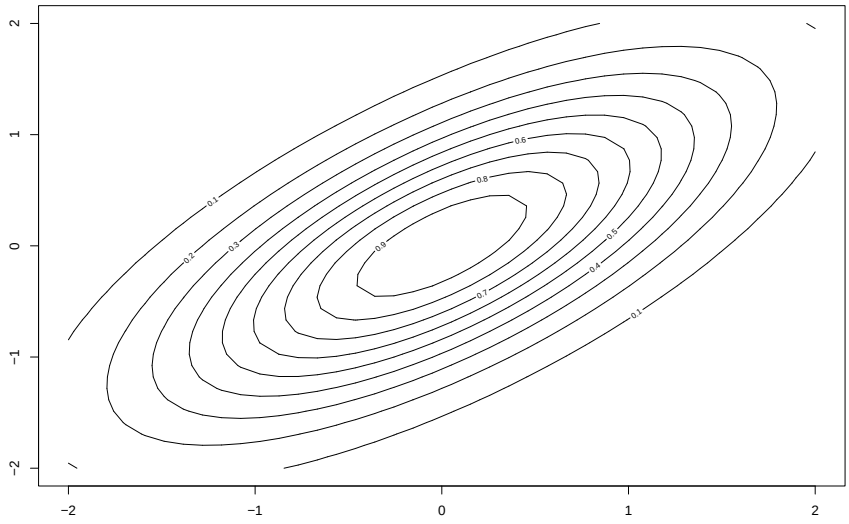


Figure 4: The bivariate normal density with $\rho = 0.7$, a contour plot

```
for (i in 1:40){
 for (j in 1:40){
  z[i,j] <- bivnd(x[i],y[j])
 }
}
contour(x,y,z)
image(x,y,z)
persp(x,y,z)
```

The resulting three plots are given respectively as Figures 4, 5 and 6.

Repeat, experimenting with different values of ρ. Think about the problem of simulating a sample of size n from this distribution, and then checking its empirical density.
In fact nested loops, while possible in SPlus, are to be avoided if possible (see Venables and Ripley's book). A little thought about matrix algebra shows us that they do not need to be used here. Try the following.

```
x2 <- x^2 ; y2 <- y^2
one <- rep(1, times= 40) # this is the unit vector, of length 40.
z0 <- (x2 %*% t(one) - 2 * rho * x %*% t(y) + one %*% t(y2))/(2*(1- rho^2))
z0 # to check that z0 is a matrix
z <- exp(-z0)
contour(x,y,z)  # and so on....
```

Finally, as an optional extra, we plot an ellipse, as shown in Figure 7, to show the shape of a contour of the bivariate normal density function.
The ellipse will be centred at (x0, y0). What are a, b and alpha?

```
a <- 3 ; b <- 4; alpha <- pi/3; x0 <- 1 ; y0 <- 2
theta <- seq(0, 2*pi, length=1000)
x <- x0 + a*cos(theta)* cos(alpha) - b*sin(theta)*sin(alpha)
y <- y0 + a*cos(theta)* sin(alpha) + b*sin(theta)*cos(alpha)
plot(x,y, type="l")
```

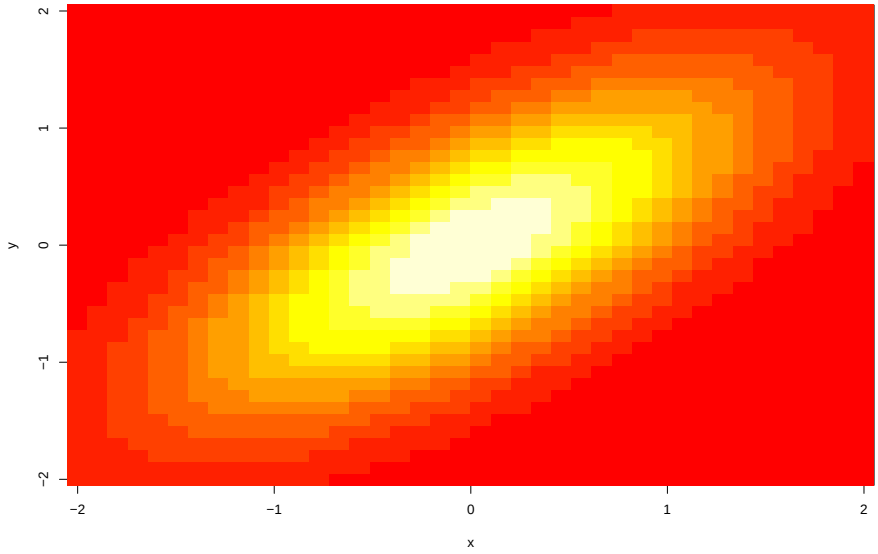


Figure 5: The bivariate normal density with $\rho = 0.7$, an image plot

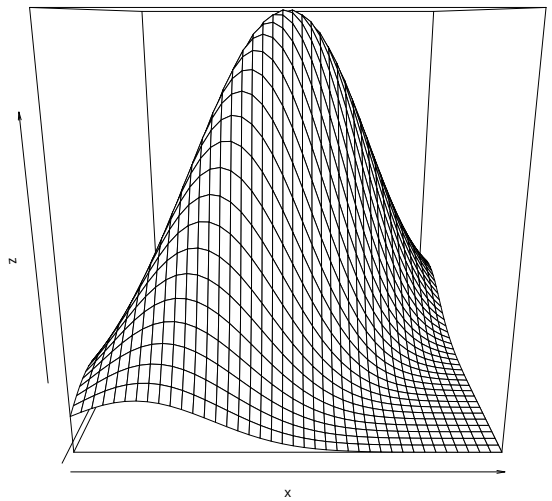


Figure 6: The bivariate normal density with $\rho = 0.7$, a perspective plot

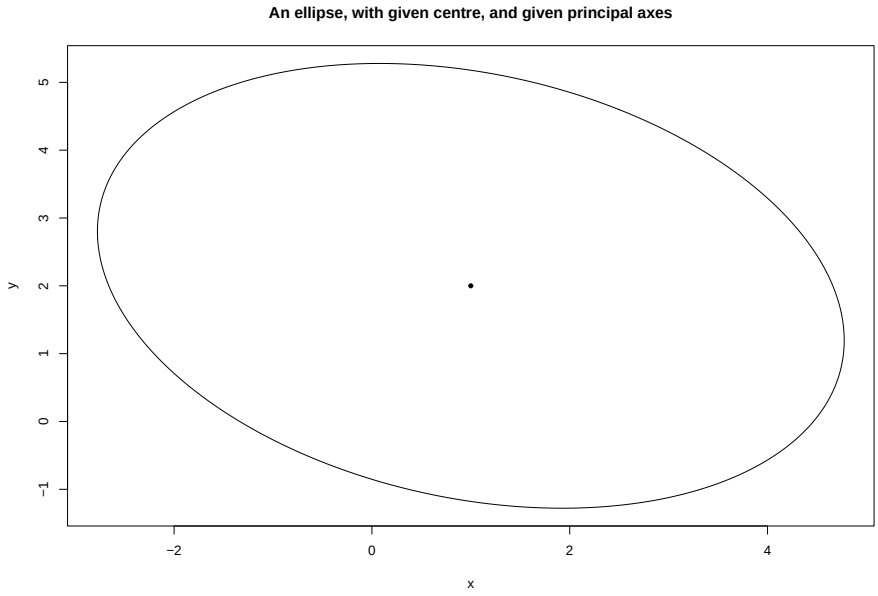


Figure 7: An ellipse

```
points(x0, y0, pch=20) # to show the centre of the ellipse
```

Why does this correspond to a pdf with NEGATIVE correlation?
I got the code from a reply to Rhelp in October 2006 by Alberto Monteiro. If you eliminate θ from the expressions for x, y you should be able to write the equation of the above ellipse as in the usual form, and hence find the correlation coefficient ρ in terms of a, b, α.

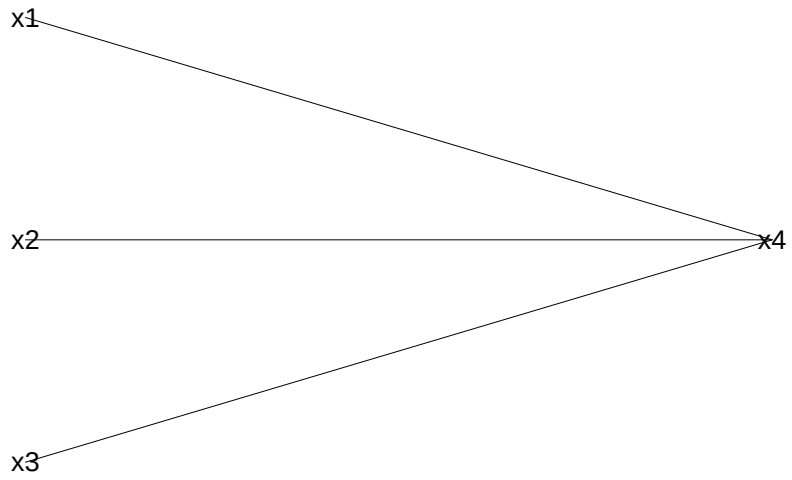


Figure 8: Illustrating conditional independence

3. Graphical models for normal distributions: a simulation, and **new for 2008** an analysis of the Times Online Good University Guide dataset.
First we generate data from a multivariate normal distribution, with x_1, x_2, x_3 mutually independent, conditional on x_4. So the corresponding graphical model for dependencies is as shown in Figure 8.

We show that a more conventional multivariate analysis, ie principal components, will not pick up this structure.
(Would a factor analysis detect the structure?)

```
Splus6
i <- 1:100 ; x4 <- rnorm(i) # first generate x4
x1 <- 7*x4 + (.5)*rnorm(i);x2 <- 8*x4+(.7)*rnorm(i)
x3 <- -10*x4 + rnorm(i)
a <- cbind(x1,x2,x3,x4) # This is our data matrix
pairs(a)                # for pairwise associations
v <- var(a)             # sample covariance matrix
inv <- solve(v)         # inverse sample cov.matrix
round(v,2) ; round(inv,2) # to have a look at them.
```

Note:

$$var(x_4|\text{all remaining variables}) = 1/inv_{44}$$

where inv is the inverse of the covariance matrix of the $x's$.
Hence for example, we see that $var(x_4|x_1, x_2, x_3)$ is SMALL compared with $var(x_4)$, ie, x_4 is closely determined by x_1, x_2, x_3. Check this by

```
l.m <- lm(x4~x1+x2+x3) ; summary(l.m)
```

Note: standard theory also shows that, for example,

$$corr(x_1, x_2|\text{all remaining variables}) = -inv_{12}/\sqrt{inv_{11}inv_{22}}.$$

Inspection of *inv* also shows us for example, that $corr(x_1, x_2|\text{remaining variables})$ is LOW, and $corr(x_1, x_4|\text{remaining variables})$ is HIGH.
A more cunning way to find these conditional correlations is to use the linear model 'trick'.

```
y <- 1:100              #Invent a y-variable
trick.lm <- lm(y~x1+x2+x3+x4)
summary(trick.lm, cor= T)
```

This has given us MINUS the matrix of conditional correlations. (Ignore the column corresponding to 'intercept'.)
Now let's try principal components.

```
a.pr <- princomp(a)       #for principal components
first <- a.pr$x[,1];second <- a.pr$x[,2] # first 2 princ. comps
b <- cbind(a,first,second); pairs(b)
```

This plot will show us how x1,... are related to first and second principal components.

```
round(cor(b),2)        # x4 has no special role.
# We can use the D-matrix to compute appropriate test statistics:
d <- diag(inv) ; d<- 1/d ; d<- sqrt(d) #gives vector of 1/sqrt(d(i,i))
dd <- matrix(d)                      #turns it into 4X1 matrix
t.d <- dd %*% t(dd) #gives matrix of (1/sqrt(d(i,i)d(j,j)))
corr <- inv*t.d     # note, element by element multiplication
chi.sq <- 1 - corr^2
chi.sq <- -100*log(chi.sq)
chi.sq             # refer each term to chi-sq(1) to test sig.
                   # See Whittaker, p175
```

You could experiment with a 'heat map' picture of the correlation (or conditional correlation) matrix. You do have to remember that whereas a graph is indexed from the bottom left-hand corner, with a matrix we count from the top left-hand corner.

```
z = cor(a) # for the 4 by 4 correlation matrix
i = 1:4 ; j = 4:1
zz = z[i,j] # necessary 'flip'
ii= 1:4; jj = 1: 4 ; image(ii,jj,zz)
```

New for 2008: the Times Online Good University Guide data from April 28, 2008.
Firstly, here is the dataset for 2008, omitting the institution names. The column headings are R= Rank (there are some ties), StudSat = Student Satisfaction (with some NA's), ResQual= Research Quality,
ServSpend= Services and Facilities spend, Entry= Entry Standards, Compl= Completion rate, GoodH = percentage getting a 'Good Honours' degree
GradProsp = Graduate Prospects, Total= Total score.

	R	StudSat	ResQual	StudStaff	ServSpend	Entry	Compl	GoodH	GradProsp	Total
1	1	NA	6.2	13.0	2671	522	98.6	89.4	78.6	1000
2	2	NA	6.5	12.2	2097	530	97.9	84.5	87.9	995
3	3	3.9	5.8	9.7	2828	453	96.0	72.4	86.0	960
4	4	3.9	6.3	13.2	1416	471	96.9	74.6	83.0	915
5	5	4.1	5.3	15.4	1009	458	94.8	79.7	72.6	841
6	6	3.9	5.5	9.4	1623	429	94.3	72.5	79.8	832
7	7	NA	5.6	17.1	1724	453	96.7	79.3	73.0	813
8	8	3.9	5.2	14.9	1426	440	95.8	81.3	76.0	811
9	9	4.0	5.2	20.3	1250	452	96.4	75.4	75.8	810
10	10	4.0	4.7	12.4	1546	399	93.2	67.2	79.8	777
11	11	3.8	5.2	16.7	1235	443	95.3	76.8	78.9	764

12	12	4.1	4.3	17.6	1217	368	94.0	63.8	69.7	755
13	13	3.8	5.0	14.5	1254	458	92.2	77.8	76.4	742
14	14	4.0	5.4	16.7	1323	402	90.7	71.3	67.8	738
15	15	3.9	3.9	14.6	1701	346	88.4	64.3	76.9	734
16	16	3.9	5.5	15.0	1144	436	95.2	71.0	64.3	733
17	17	4.0	4.7	16.8	1016	387	94.8	74.6	65.6	726
18	18	3.8	5.3	9.5	1603	365	84.4	69.2	73.3	722
19	19	3.8	5.0	15.6	1368	431	96.2	74.5	70.7	721
20	20	4.1	5.0	17.0	1104	364	91.2	65.4	59.2	709
21	21	4.1	4.5	16.2	1149	371	92.9	62.5	64.6	705
22	22	3.9	4.5	15.1	1085	408	92.4	73.3	70.3	699
23	23	3.9	4.4	16.7	1402	390	92.3	67.6	72.9	694
24	24	3.9	5.2	13.9	1093	356	88.2	71.3	61.9	688
25	25	4.0	4.9	16.8	914	358	91.7	69.1	64.5	683
26	26	3.9	4.3	16.6	1200	389	92.4	69.6	70.2	678
27	27	3.9	5.4	15.8	1074	366	92.5	64.2	61.0	670
28	28	3.9	4.5	13.9	1136	377	90.1	66.3	70.1	658
29	29	3.8	5.1	14.6	1323	406	92.3	68.8	65.6	656
30	30	3.9	4.5	18.4	972	387	92.1	71.2	68.0	653
31	31	3.9	4.3	13.9	1130	425	85.5	64.8	68.0	650
32	32	NA	4.0	13.9	1174	447	78.8	64.3	75.8	648
33	33	3.9	4.3	17.2	1164	363	86.6	69.1	73.2	626
34	34	3.9	4.6	15.5	916	370	90.0	64.8	70.8	625
35	35	3.7	5.1	13.6	1136	382	88.3	70.3	61.6	621
36	36	3.9	4.8	14.7	1111	316	83.4	56.8	63.2	611
37	37	NA	3.9	14.5	1027	334	89.9	62.0	66.9	609
38	38	3.9	4.0	16.4	1093	324	87.0	60.5	65.1	608
39	39	4.1	4.0	19.2	976	299	89.8	61.1	56.1	607
40	40	3.7	4.7	15.9	1089	341	86.8	61.6	79.1	603
41	40	3.9	3.7	18.7	748	318	85.0	63.9	80.6	603
42	42	3.8	4.7	12.8	1088	342	88.5	61.6	72.6	599
43	42	4.0	3.2	19.0	761	294	83.9	57.3	71.1	599
44	44	NA	3.6	18.3	998	429	81.8	67.9	71.2	598
45	45	NA	4.2	16.8	1056	375	82.3	57.9	61.0	551
46	46	3.9	4.4	16.9	979	289	88.3	54.1	60.1	531
47	47	4.0	3.9	16.9	1042	290	77.0	50.0	65.8	530
48	48	3.9	3.4	18.6	1105	265	82.2	54.2	75.8	519
49	49	3.9	1.6	17.0	908	295	84.9	56.6	72.1	512
50	50	NA	4.2	17.6	939	382	68.7	58.6	74.6	502
51	51	3.7	3.1	16.7	1188	302	85.4	63.7	63.3	496
52	52	3.8	4.9	20.2	758	297	82.2	58.9	58.4	486
53	53	3.9	2.4	19.2	1345	279	77.0	61.8	59.2	485
54	54	3.8	4.0	18.1	693	314	84.3	56.0	64.1	484
55	55	NA	0.9	18.5	927	311	79.8	53.4	81.8	480
56	56	3.7	1.4	19.1	810	283	89.1	53.3	77.1	456
57	57	4.0	4.4	26.2	554	258	76.9	60.5	67.7	453
58	58	NA	1.5	21.3	710	333	77.7	60.3	75.2	447
59	59	3.5	4.4	18.7	873	391	87.1	57.9	51.3	442
60	60	3.9	1.6	17.5	1017	275	81.2	55.9	55.5	437
61	61	3.8	1.7	18.1	805	280	83.0	55.3	64.2	425
62	62	NA	1.2	18.1	767	354	74.8	60.0	61.6	419
63	62	3.7	0.7	18.5	851	278	85.2	59.8	63.7	419
64	64	3.9	1.1	18.1	943	241	79.8	52.2	53.3	399
65	65	3.9	1.3	16.3	1063	273	69.3	52.1	53.4	397

```
66   66    3.8     0.8     16.2     1024   252  77.3  57.7     62.5   396
67   67     NA     0.8     17.8      960   292  73.3  57.4     61.8   393
68   68    3.9     1.7     21.0      652   242  88.0  45.1     59.2   391
69   69    3.9     1.5     20.6      746   265  84.5  54.7     49.4   385
70   70    3.8     0.9     17.4      928   246  76.9  48.1     66.3   383
71   71    3.8     1.9     20.3     1257   237  79.5  52.7     55.6   382
72   72    3.8     1.6     19.8      867   266  80.6  56.7     59.8   380
73   73    3.8     1.1     21.9     1031   291  80.7  52.4     59.3   375
74   74    3.8     1.7     20.2     1000   242  80.5  51.3     53.7   371
75   75    3.8     0.8     20.7     1077   248  85.1  48.4     54.5   366
76   75    3.9     1.0     20.6      929   239  73.7  56.0     61.1   366
77   75    3.9     1.1     20.4      583   252  83.5  45.8     62.4   366
78   78    3.9     0.5     19.1      826   228  74.9  56.5     56.7   365
79   79    3.8     1.7     19.0      758   256  81.4  48.0     59.3   363
80   80    3.8     1.2     18.3      882   228  78.6  49.2     64.0   362
81   81    3.8     1.7     16.7      710   238  77.0  50.4     56.1   360
82   82    3.9     1.0     19.4      692   221  80.0  56.8     55.4   359
83   83    3.7     2.1     16.7      857   269  73.2  53.5     59.7   356
84   84     NA     0.6     23.4     1808   207  75.4  41.1     59.6   348
85   85    3.8     1.2     18.5      840   252  77.1  48.4     57.1   335
86   85    3.8     1.4     18.5      860   235  75.9  45.1     61.9   335
87   87    3.7     1.1     23.8      911   270  85.0  55.4     61.2   332
88   88    3.7     1.5     24.4      502   264  85.7  67.7     54.8   328
89   89    3.8     0.8     20.2      828   221  80.8  45.4     63.1   323
90   90    3.8     1.2     18.9      822   261  75.2  47.8     53.7   319
91   91    3.7     1.2     17.5      739   229  80.5  56.4     53.5   318
92   92    3.9     0.9     20.6      610   244  78.9  46.0     55.5   317
93   93    3.9     0.8     17.7      785   179  63.3  51.7     64.7   316
94   94     NA     0.7     20.0     1085   313  68.7  43.8     55.5   311
95   95    3.9     0.7     20.7      705   236  74.4  44.6     63.0   310
96   96    3.7     0.8     23.2      928   263  82.7  49.0     59.5   305
97   97    3.7     1.4     21.5      753   246  80.6  47.7     63.8   303
98   98    3.8     0.9     22.4      978   215  77.0  49.8     57.1   298
99   99     NA     1.4     22.3     1298   190  68.5  40.0     58.0   284
100 100    3.8     0.5     17.8      672   253  76.1  46.1     48.4   280
101 100     NA     0.7     19.8     1478   214  64.0  47.2     51.9   280
102 102    3.7     0.4     25.6      780   266  85.4  49.9     61.7   276
103 103     NA     1.3     23.1      800   191  72.0  51.7     61.9   270
104 104    3.8     0.5     19.9      615   242  64.8  52.7     61.5   262
105 105    3.7     0.5     21.6      940   224  78.2  43.9     48.8   247
106 106    3.9     0.4     22.8      248   243  78.3  38.1     55.7   242
107 107    3.6     0.6     22.8      626   257  76.6  46.5     70.7   229
108 108    3.7     1.3     29.6     1576   189  71.9  46.5     55.7   219
109 109    3.6     0.7     26.3      793   255  83.9  52.3     48.9   214
110 109    3.7     1.2     24.0      791   210  79.7  42.3     57.3   214
111 111    3.6     0.4     19.4      586   202  69.5  48.7     61.9   209
112 112    3.7     0.6     21.2      917   204  73.9  43.5     52.6   207
113 113    3.7     0.7     24.3      833   220  73.6  47.0     52.6   191
```

I used only the variables

```
ResQual, ..., GradProsp
```

for which the corresponding pairs plot is Figure 9. Finally, here is the matrix of conditional correlations, derived as above. You will see that the only two variables that have a strong correlation, conditional on the remaining five variables, are 'GoodHon' and 'Entry'.

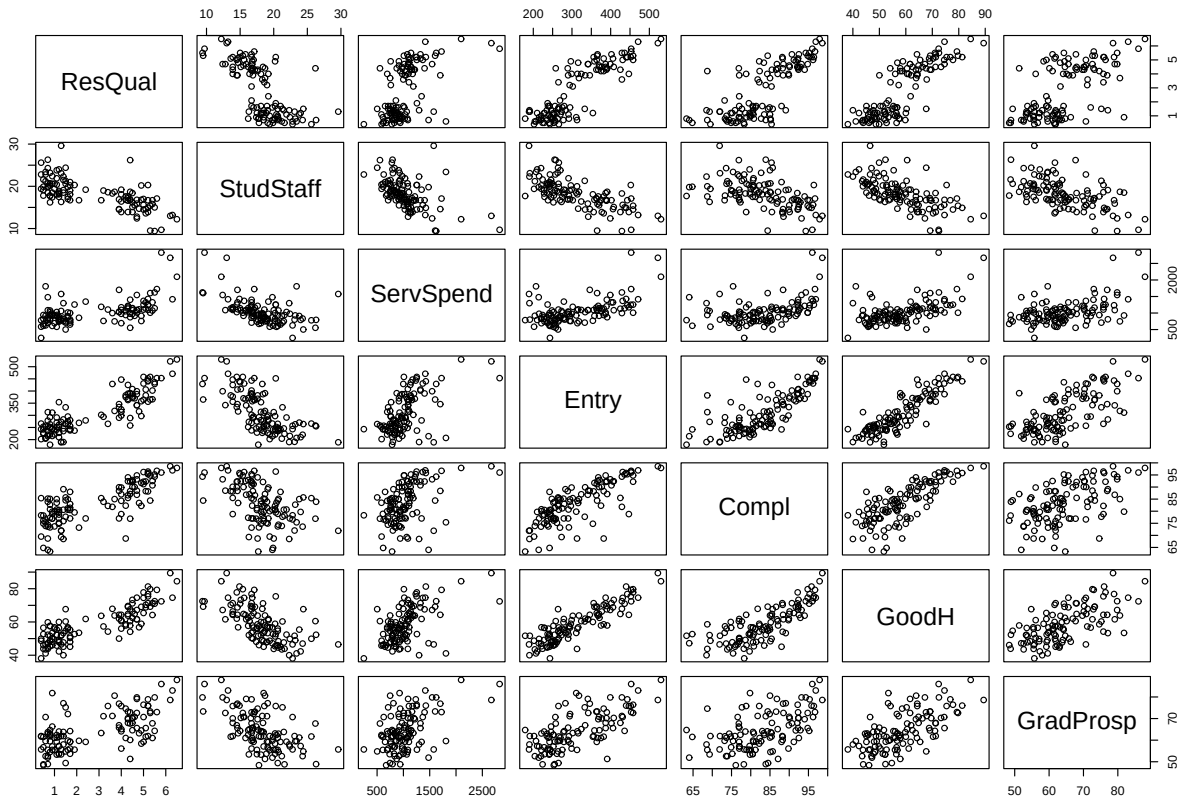


Figure 9: The pairs plot for the Times Online Good University Guide data, 2008

```
          ResQual StudStaff ServSpend Entry Compl GoodH GradProsp
ResQual      1.00     -0.24      0.06  0.31  0.28  0.21      0.00
StudStaff   -0.24      1.00     -0.15 -0.17  0.14 -0.02     -0.12
ServSpend    0.06     -0.15      1.00  0.08 -0.03  0.07      0.12
Entry        0.31     -0.17      0.08  1.00  0.18  0.48      0.24
Compl        0.28      0.14     -0.03  0.18  1.00  0.21     -0.02
GoodH        0.21     -0.02      0.07  0.48  0.21  1.00      0.07
GradProsp    0.00     -0.12      0.12  0.24 -0.02  0.07      1.00
```

Afterthought: the above analysis was perhaps a bit simple-minded, since

```
hist(ResQual)
```

shows that this variable has a clearly bi-modal distribution. You could try using only the first 60 rows of the data-matrix in your analysis.
In case you feel you really HAVE to know the rank order of the 113 Universities concerned, here it is (I abbreviated some of the names). I see that Cambridge is **second**. Huh!

```
  [1] Oxford                    Cambridge
  [3] Imperial_College          London_School_of_Economics
  [5] St_Andrews                University_College_London
  [7] Warwick                   Bristol
  [9] Durham                    Kings_College_London
 [11] Bath                      Loughborough
 [13] Edinburgh                 Southampton
 [15] Aston                     York
 [17] Exeter                    S_O_A_S
 [19] Nottingham                East_Anglia
 [21] Leicester                 Sheffield
 [23] Newcastle                 Royal_Holloway
 [25] Reading                   Birmingham
 [27] Lancaster                 Cardiff
 [29] Manchester                Leeds
 [31] Glasgow                   Aberdeen
 [33] Queens_Belfast            Liverpool
 [35] Sussex                    Essex
 [37] Stirling                  Kent
 [39] Aberystwyth               Surrey
 [41] City                      Queen_Mary_London
 [43] Hull                      Strathclyde
 [45] Heriot-Watt               Swansea
 [47] Bangor                    Bradford
 [49] Oxford_Brookes            Dundee
 [51] Brunel                    Goldsmiths_London
 [53] Ulster                    Keele
 [55] Robert_Gordon             Nottingham_Trent
 [57] Lampeter                  Queen_Margaret_Edinburgh
 [59] Univ_of_the_Arts,London   Plymouth
 [61] Brighton                  Glasgow_Caledonian
 [63] Bournemouth               Staffordshire
 [65] Glamorgan                 UCE_Birmingham
 [67] Napier                    Chichester
 [69] Winchester                Central_Lancashire
 [71] Roehampton                West_of_England
 [73] Northumbria               Gloucestershire
 [75] UWIC_Cardiff              Coventry
```

```
 [77] Canterbury_Christ_Church   Newport
 [79] Portsmouth                 Kingston
 [81] Sunderland                 Northampton
 [83] Salford                    Bedfordshire
 [85] LiverpoolJohnMoores        Hertfordshire
 [87] Sheffield_Hallam           Bath_Spa
 [89] Worcester                  Manchester_Metropolitan
 [91] Westminster                Huddersfield
 [93] Bolton                     Paisley
 [95] Teesside                   Leeds_Metropolitan
 [97] DeMontfort                 Derby
 [99] East_London                Chester
[101] Abertay                    York_St_John
[103] London_South_Bank          Anglia_Ruskin
[105] Southampton_Solent         Edge_Hill
[107] Cumbria                    Middlesex
[109] Lincoln                    Greenwich
[111] Thames_Valley              Wolverhampton
[113] Liverpool_Hope
```

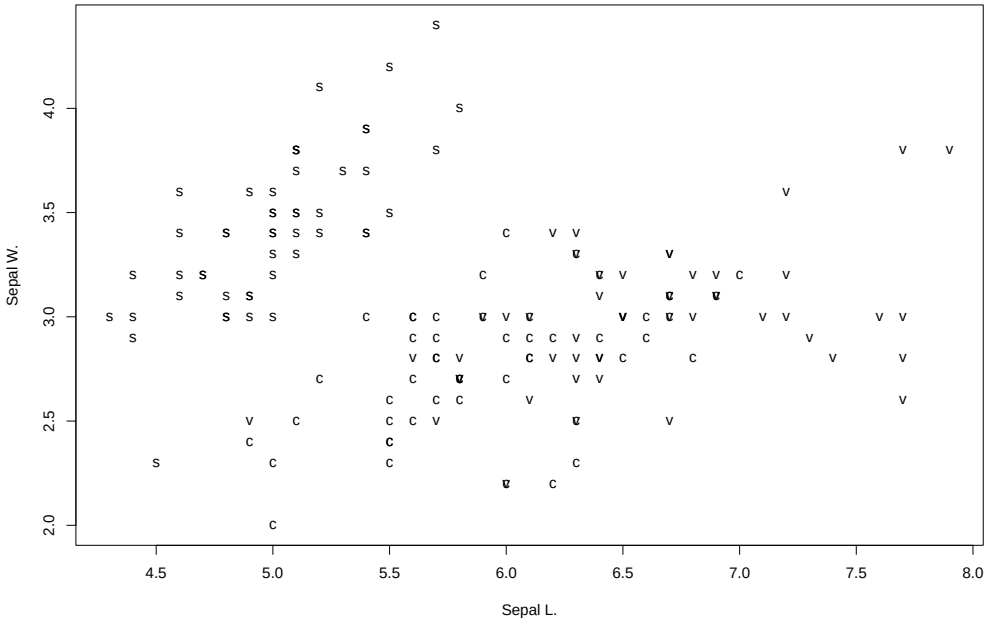


Figure 10: Fisher's Iris data: a simple pairs plot

4. Manova on 3 groups, using Fisher's classic Iris data

This dataset consists of 50 cases of each of 3 species, namely Iris setosa, Iris virginica, and Iris versicolor. Each case has 4 measurements on the length and width of its petals and sepals.

```
R
data(iris)
ir.species <- gl(3,50, length=150, labels=c("s", "c", "v"))
pairs(ir)  # not so revealing: we need to label the 3 species separately.
plot(ir[,1:2], type="n")
text(ir[,1:2], labels=as.character(ir.species)) # for a simple pairwise plot
# but, for a really good plot, we use the R example, thus
pairs(iris[1:4], main = "Anderson's Iris Data -- 3 species",
+  pch = 21, bg = c("red", "green3", "blue")[unclass(iris$Species)])
```

These pairwise plots result in Figures 10 and 11 respectively, and show some separation between the three groups.

```
summary(aov(ir[,1] ~ ir.species)) # for a 1-way anova on the 1st vector
iris.manova <- manova(ir~ ir.species) # for the manova
summary(iris.manova, univar=T) # compare with result of aov()
summary(iris.manova, test="wilk") # to look at the whole vector
liris.manova <- manova(log(ir)~ ir.species) #to try log-transform
summary(liris.manova, test="wilk")
```

The iris dataset works almost too well. For a fun dataset, where the separation between the groups is less clearcut, try the painters data (de Piles).

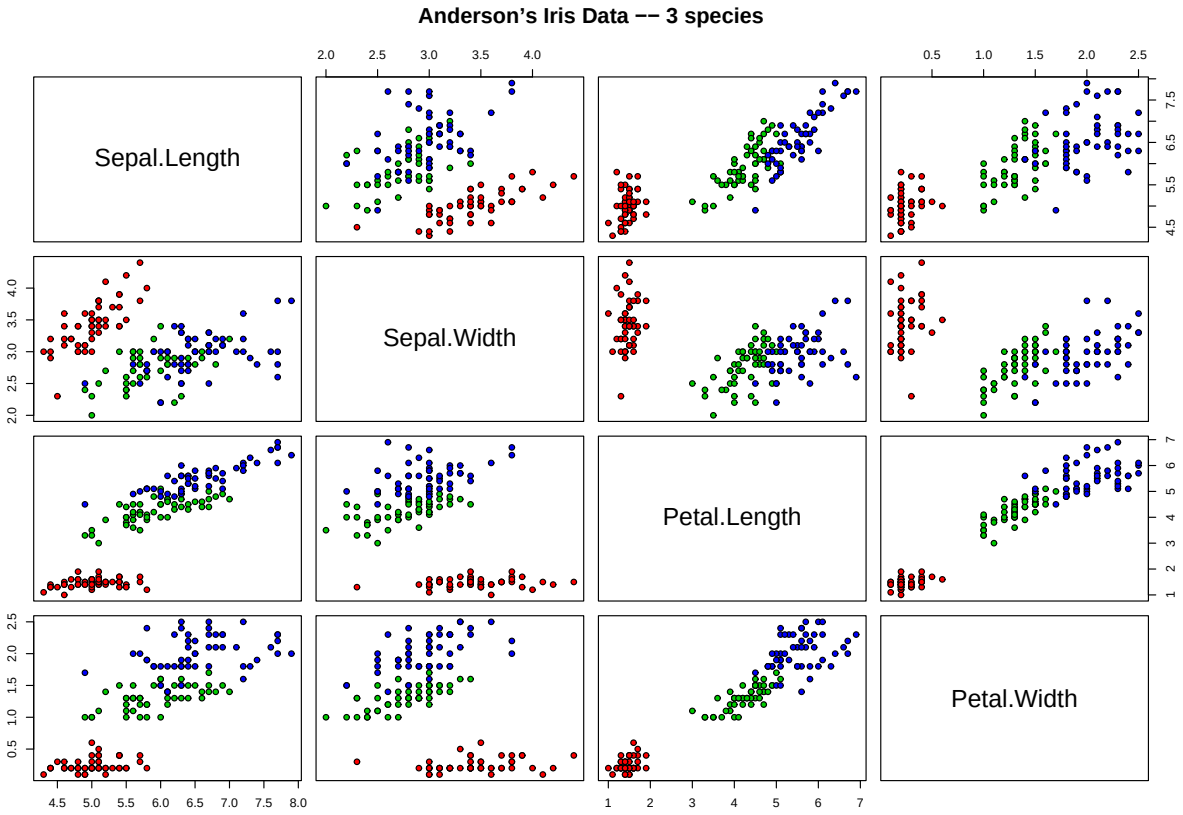


Figure 11: Fisher's Iris data: a full pairs plot

```
library(MASS)
?painters
x <- painters[,1:4] ; x <- as.matrix(x)
 school <- painters[,5]
 is.factor(school) ; table(school)
 painters.manova <- manova(x~ school)
 summary(painters.manova, univar =T)
 summary(painters.manova,test ="wilk")
 for (i in 1:4){
 + cat(round(tapply(x[,i],school,mean),3),"\n")
 + }
 # This shows us the differences between the Schools.
```

Exercise: do a 'pairs'plot of the 'painters' data, with a different plotting symbol for each of the 8 Schools.

5. Linear discrimination between groups.
Let x be the original data vector, and consider doing a 1-way anova on the scalar quantity $y = u^T x$.
We want to choose a u such that the 1-way anova on y gives maximal separation between the groups:
hence you can see that we aim to solve the problem:
choose u to maximise $u^T B u$ subject to $u^T W u = 1$, where B, W are the between-groups and within-groups sums of (squares and products) matrices, respectively.
This gives us that $Bu = \lambda W u$, and so taking λ as the largest such value gives the maximum value of $u^T B u / u^T W u$.
Hence, for the transformed variable $y = u^T x$, in the 1-way anova, we find that
'between groups ss/(within groups ss)' $= \lambda$, and hence
'between groups ss/ total ss' $= \lambda/(\lambda + 1) = R^2$,
where R^2 is the usual multiple regression R^2: in this case it measures how well the separation into groups explains the overall variation.
In the example below, each of $B,\ W$ is a 4×4 matrix, and since there are just 3 groups, it follows that B is of rank 2, hence the final λ, and hence the final R^2, is effectively zero.

We use the Iris dataset defined above, and compare 2 methods. (The function lda() is also used on this dataset in Venables and Ripley, 4th edition.)

```
a <- log(ir) ; grou <- ir.species # for convenience
teeny.dis  <- discr(a,3 )
teeny.dis
teeny.dv <- a %*% teeny.dis$vars #new coords
teeny.x <- teeny.dv[,1]
tapply(teeny.x,grou,mean)
```

Now relate this to teeny.dis output. I have always had great difficulty interpreting

```
teeny.dis$groups
```

but in fact the ?discr does tell you what to expect. Here goes.
Taking the first column of the 3×3 matrix

```
teeny.dis$groups,
```

we set

```
x1 <- c(rep(0.8074378, times =50), rep(-0.2986802,times =50), rep( -0.5087577,times =50))
cor.test(x1, teeny.x)
```

and sure enough, this reveals to us the correlation 0.9887738, as we get for the first component of

```
teeny.dis$cor
```

above.

```
par(mfrow =c(3,1))                  # 3 plots on 1 page
hist(teeny.x[grou=="s"]);hist(teeny.x[grou =="c"]);hist(teeny.x[grou =="v"])
par(pty ="s")                       # to make graph frames SQUARE
par(mfrow =c(1,1))
teeny.y <- teeny.dv[,2]
plot(teeny.x,teeny.y,type ="n",xlab ="first disc var",ylab ="second disc var")
text(teeny.x,teeny.y, labels = as.character(ir.species))
```

We now compare with data in original co-ordinates

```
v1 <- a[,1] ; v2 <- a[,2]
plot(v1,v2,type ="n") ; text(v1,v2, labels= as.character(ir.species))
library(MASS)
?lda
ir.lda <- lda(log(ir), ir.species)
ir.lda
plot(ir.lda) # we'll do this another way now
ir.ld <- predict(ir.lda, dimension =2)$x
plot(ir.ld, type ="n", xlab = "first lin discr", ylab = "second lin discr"))
text(ir.ld, labels = as.character(ir.species), cex =1.0)
```

Here's how to apply it for the painters' dataset.

```
summary(painters)
table(School)
k <- scan()
10 6 6 10 7 4 7 4

x <- painters[,1:4]
first.dis <- discr(x,k) ; first.dis
```

6. Principal Components Analysis

The data below are from Hartigan, 1975, "Clustering Algorithms" Our first object is to see whether the 9 points in 5 dimensions can be represented as 9 points in a plane. Here is Hartigan's data set.

```
       energy protein fat calcium iron
beef     180 22 10 17 3.7
chicken  170 25  7 12 1.5
clams     45  7  1 74 5.4
crabmeat  90 14  2 38 0.8
mackerel 155 16  9 157 1.8
salmon   120 17  5 159 0.7
sardines 180 22  9 367 2.5
tuna     170 25  7   7 1.2
shrimp   110 23  1  98 2.6
```

```
food <- read.table("food",header=T) ; food
attach(food)
a <- data.matrix(food) ; a
a.cov <- var(a) ; a.cov
a.corr <- cor(a) ; a.corr
pairs(a)
help(princomp)
a.pcp <- princomp(a) ; names(a.pcp)
a.pcp        # Can you understand what it's telling you ?
a.pcp$sdev   # What are these ?
help(eigen)  # We find out directly.
x <- eigen(a.cov) ; names(x)
x$values
z <- a.pcp$sdev  ; z <- z*z ; z
```

Do you see the connection? Let's get a plot of the 9 points using first 2 principal components.

```
a.pcp            # for a reminder
x1 <- a.pcp$scores[,1] # first column
x2 <- a.pcp$scores[,2] # second column
plot(x1,x2)      # but we really need to label the points
a.lab <-  row.names(food)
plot(x1,x2,type="n",xlab ="first principal component",ylab="second principal component")
text(x1,x2,a.lab)
```

This gives us Figure 12. Because of the high variability of calcium relative to the other 5 variables, this variable will dominate the dominate the first principal component, as is shown by Figure 13, which is obtained by

```
plot(x1,calcium,type="n") ; text(x1,calcium,a.lab)
```

We may prefer to standardise all the original variables to have mean 0, variance 1 before we do the principal components analysis. Thus, in effect, we find the eigen-values of the **correlation** matrix rather than those of the **covariance** matrix. Of course, this gives each of the 5 variables "equal weight" in the analysis. The final plots may look completely different from the plots which result from the unstandardised variables.
A problem for you: compute the standardised data matrix from a above, and do the principal components analysis on this.
Compare the results of the above with what you get from princomp() . You should also try

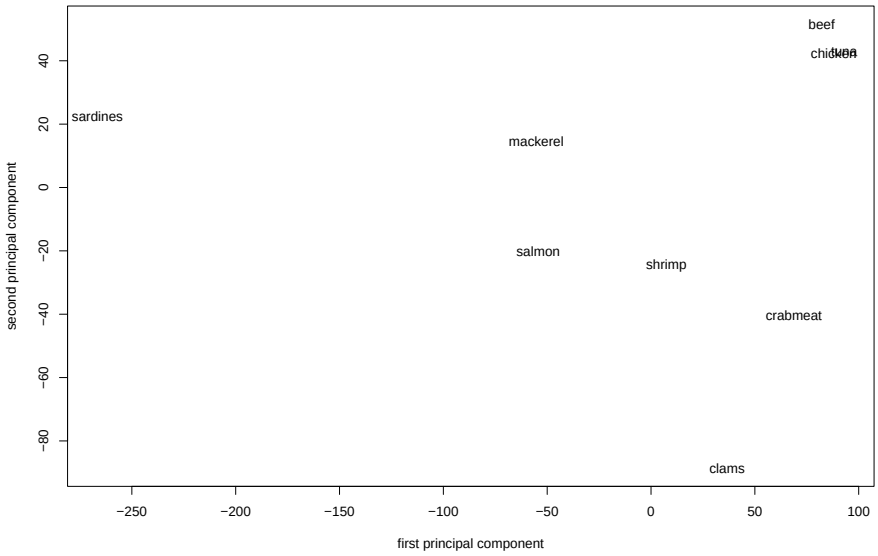


Figure 12: Principal components on the unstandardised food data

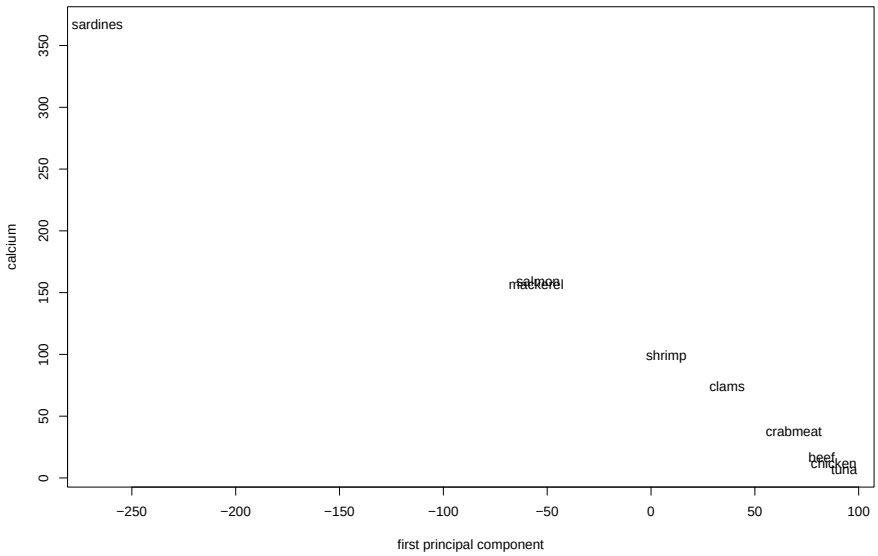


Figure 13: Showing that the first principal component is (almost) - calcium

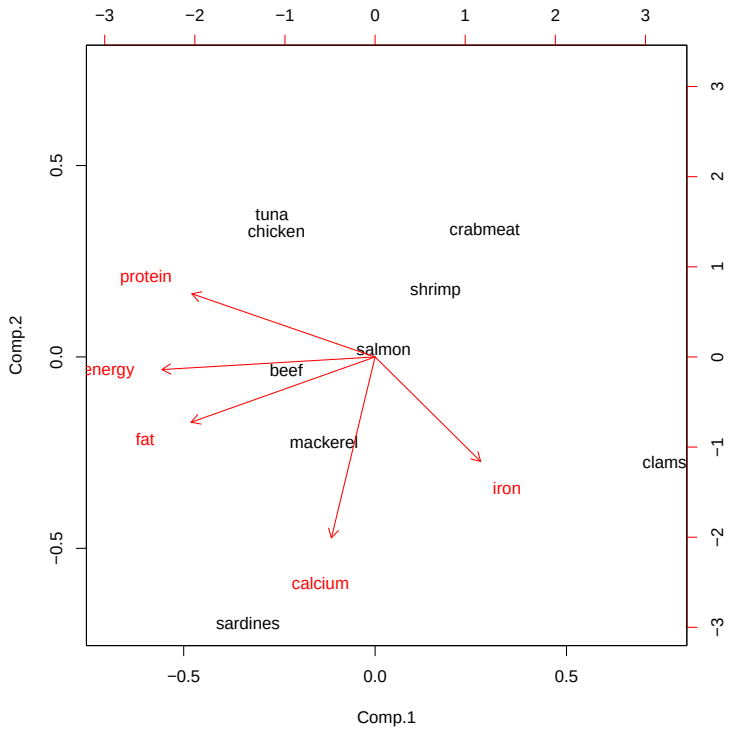


Figure 14: The biplot for the standardised food data

```
a.pcp <- princomp(a,cor =T)
a.pcp$loadings
Loadings:
        Comp.1 Comp.2 Comp.3 Comp.4 Comp.5
energy  -0.601        -0.173         0.778
protein -0.516  0.277         0.714 -0.374
fat     -0.519 -0.286 -0.286 -0.561 -0.503
calcium -0.123 -0.793  0.533  0.268
iron     0.297 -0.458 -0.773  0.321

               Comp.1 Comp.2 Comp.3 Comp.4 Comp.5
SS loadings       1.0    1.0    1.0    1.0    1.0
Proportion Var    0.2    0.2    0.2    0.2    0.2
Cumulative Var    0.2    0.4    0.6    0.8    1.0
```

Note: Figure 14 shows a **biplot** of the data. This is an ingenious 2-dimensional picture of the data (using the standardised variables) is produced by

```
biplot(a.pcp)
```

The directions of the arrows in the biplot correspond to the 'loadings' of components 1 and 2.

7. Hierarchical Cluster Analysis.

Here is a data set which represents 10 points in 3 dimensions, and I choose this very small data-set so that you can see what to expect from the result of the cluster analysis.

```
0 2 3
4 5 6
70 7 7
10 11 12
3   4  5
6   7 18
19 20 21
22 23 44
25 26 27
28 29 30
```

Here's how to analyse it in R (S-Plus will do the same, but with slightly different terminology). A fundamental problem with hierarchical cluster analysis is that there are several ways of choosing the **distance** function, and having made that particular choice, there are then several ways of choosing the particular method of clustering: this is because we can define the distance between two clusters in several different ways. You have to realise that cluster analysis is a 'data-analytic' method, ie a (sensible) way of reducing a complex dataset, but it does not depend on any fundamental statistical modelling ideas such as likelihood, parameters, goodness of fit etc.

```
a <- read.table("tinycluster") ; a
a <- data.matrix(a) ; a
```

Observe that R can cope with missing values in constructing a distance matrix.

```
d <- dist(a,method ="euclidean")
round(d,2) # which results in the interpoint distances below

       1     2     3     4     5     6     7     8     9
2   5.83
3  70.29 66.04
4  16.19 10.39 60.34
5   4.12  1.73 67.10 12.12
6  16.91 12.33 64.94  8.25 13.67
7  31.76 25.98 54.46 15.59 27.71 18.63
8  51.05 45.74 62.68 36.22 47.36 34.47 23.39
9  42.15 36.37 52.78 25.98 38.11 28.34 10.39 17.52
10 47.35 41.57 52.70 31.18 43.30 33.35 15.59 16.37  5.20
# You can see that points 2 and 5 are the closest of the 10.
par(mfrow=c(2,1))
h1 <- hclust(d,method ="complete") # this is the default method
names(h1)
plclust(h1)   # does this make sense to you ?
h2 <- hclust(d,method ="single")
plclust(h2)   # Observe differences from previous plot
# Now we'll put labels on the points
teeny.lab <- scan(,"")
```

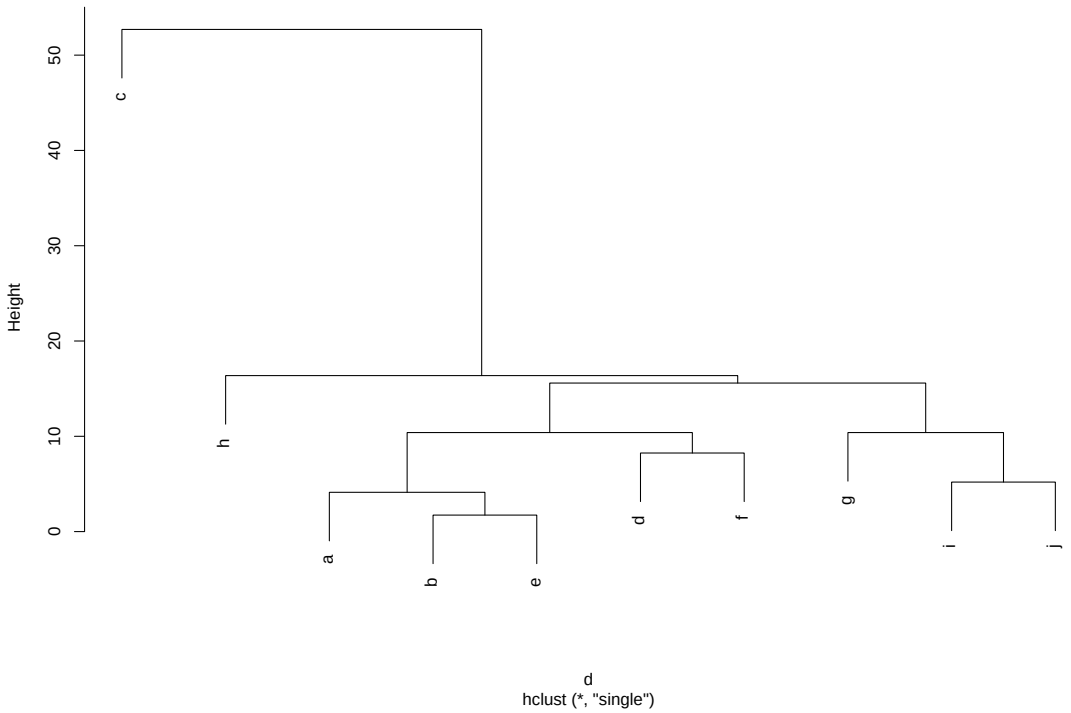


Figure 15: A dendrogram for the set of 10 points in 3 dimensions

```
a b c d e f g
h i j
             # NB,blank line
par(mfrow=c(1,1))
plclust(h2,labels =teeny.lab)
```

This results in the graph given in Figure 15.

Now we'll try an example of some BINARY data.
Here's my file for the Lent 2003 cohort of graduate students.
The questions are
1.do you eat eggs?
2.do you eat meat?
3.do you drink coffee?
4.do you like beer?
5. Are you a UK resident?
6. Are you a Cambridge graduate?
7. Are you female?
8. Do you play sports?
9. Do you have a full driving licence?
10.Are you left-handed?
The students gave the responses Yes or No, as 'y', 'n' respectively.
I admit these questions are BORING, but more interesting, personal questions might not be publicly usable, as these are.

```
.....................................................................
data for Lent 2003
```

```
        eggs  meat  coffee  beer  UKres  Cantab  Fem  sports driver Left-h
Vivienne  y    n      y      n      y      n      y     y      y      n
Taeko     y    y      y      n      y      y      y     n      n      n
Luitgard  y    n      y      n      n      n      y     y      y      n
Alet      y    y      y      y      n      n      y     n      y      n
Tom       y    y      y      y      y      y      n     y      y      n
LinYee    y    y      y      n      n      n      n     y      y      n
Pio       y    y      y      n      n      n      n     y      n      n
LingChen  y    y      n      n      n      n      y     y      n      n
HuiChin   y    y      y      n      n      n      y     y      y      n
Martin    y    y      y      y      y      n      n     y      y      n
Nicolas   y    y      y      y      n      n      n     y      y      y
Mohammad  y    y      y      n      n      n      n     n      y      n
Meg       y    y      y      n      n      n      y     y      n      n
Cindy     y    y      y      y      n      n      y     y      y      n
Peter     y    y      y      y      n      n      n     y      y      n
Paul      y    y      n      y      y      y      n     y      n      n
```

```
# What follows below was done in S-Plus
a <- read.table("students2003", header=T)
student.lab <- row.names(a)
a ; student.lab
a <- (a=="y")*1   # to convert to 0,1 data
a
d <- dist(a,"binary") ; d # can you understand it ?
s <- 1-d ; s              #s is the  SIMILARITY matrix
h <- hclust("compact",sim =s) # operating on the similarity matrix
plclust(h)               # does this make sense ?
h <- hclust(d,"compact") ; plclust(h) # now on the dissimilarity matrix
# essentially the same as the previous plot ?
# Now for fun with labels.
plclust(h,labels =student.lab)

ls()          # to show you all your S-Plus objects
# use rm() to remove unnecessary clutter
ls()          # shows you what you've done.
```

Exercise: do a cluster analysis on the 16 students using the FIRST 4 questions only.

8. Decision trees.

You are a trainee astronaut, learning how you should decide whether or not to use your autolander. You have a "training set" of 256 lines of data, telling you whether or not the autolander was used for all combinations of 6 factors(eg visibility yes/no) in the past. Here we show you how to use Splus to grow a "decision tree" to guide your actions in the future.

```
library(MASS)
help(shuttle)
shuttle
attach(shuttle)
summary(shuttle)
table(use,vis)
table(use,vis,error)   # and so on,for some useful summaries.
shuttle.tree <- tree(use~.,shuttle) # this grows a tree
 # making use of all 6 factors,if necessary.
summary(shuttle.tree)
shuttle.tree          # what is this telling you ?
 # Do we make use of "vis" in our decision ?
 # Do we make use of "wind" ?
plot(shuttle.tree)
text(shuttle.tree,srt =90)
```

Now try growing a tree using only the first 4 factors,and compare your results with the first tree obtained. For an interesting comparison with R, look at the function rpart() thus

```
library(rpart)  # rpart means 'recursive partitioning'
tree.rp <- rpart(use ~. , shuttle) ; tree.rp
plot(tree.rp,compress =T)
text(tree.rp,use.n =T)
post.rpart(shuttle.rp)  # for a  nice postscript graph
```

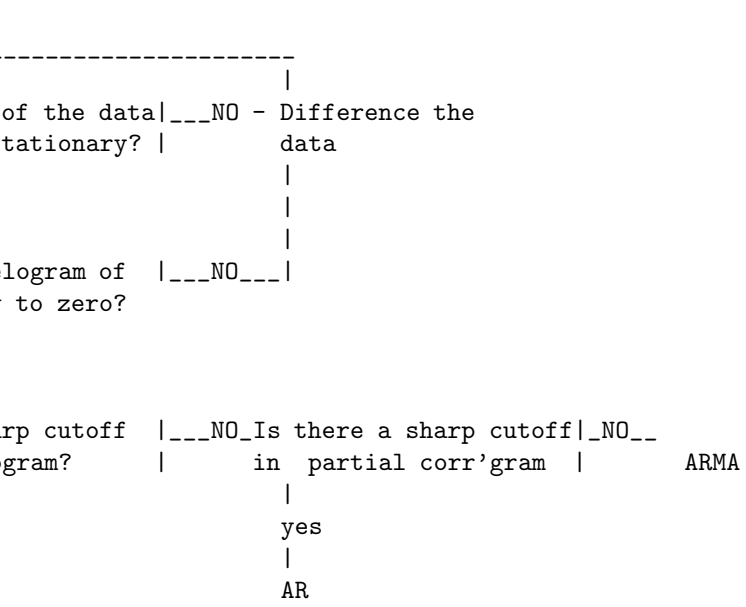
Note added June 2007. Now that I have discovered the new (ie 2007) book 'Data analysis and graphics using R: an example-based approach' by Maindonald and Braun, I realise that I should have also included the use of **the cross-validation error rate** to construct the best tree with rpart().
A very simple use of the tree() function is given in Worksheet 15, below, for the Cushing's dataset. Here are the first 50 rows of the data-set from the Venables and Ripley library(MASS), originally from D.Michie (1989).

```
   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
 4     xstab    LX   pp tail  Light  no auto
 5     xstab    LX   pp tail Medium  no auto
 6     xstab    LX   pp tail Strong  no auto
 7     xstab    LX   nn head  Light  no auto
 8     xstab    LX   nn head Medium  no auto
 9     xstab    LX   nn head Strong  no auto
10     xstab    LX   nn tail  Light  no auto
11     xstab    LX   nn tail Medium  no auto
12     xstab    LX   nn tail Strong  no auto
13     xstab    XL   pp head  Light  no auto
14     xstab    XL   pp head Medium  no auto
15     xstab    XL   pp head Strong  no auto
```

```
16     xstab    XL   pp tail  Light  no auto
17     xstab    XL   pp tail Medium  no auto
18     xstab    XL   pp tail Strong  no auto
19     xstab    XL   nn head  Light  no auto
20     xstab    XL   nn head Medium  no auto
21     xstab    XL   nn head Strong  no auto
22     xstab    XL   nn tail  Light  no auto
23     xstab    XL   nn tail Medium  no auto
24     xstab    XL   nn tail Strong  no auto
25     xstab    MM   pp head  Light  no auto
26     xstab    MM   pp head Medium  no auto
27     xstab    MM   pp head Strong  no auto
28     xstab    MM   pp tail  Light  no auto
29     xstab    MM   pp tail Medium  no auto
30     xstab    MM   pp tail Strong  no auto
31     xstab    MM   nn head  Light  no auto
32     xstab    MM   nn head Medium  no auto
33     xstab    MM   nn head Strong  no auto
34     xstab    MM   nn tail  Light  no auto
35     xstab    MM   nn tail Medium  no auto
36     xstab    MM   nn tail Strong  no auto
37     xstab    SS   pp head  Light  no auto
38     xstab    SS   pp head Medium  no auto
39     xstab    SS   pp head Strong  no auto
40     xstab    SS   pp tail  Light  no auto
41     xstab    SS   pp tail Medium  no auto
42     xstab    SS   pp tail Strong  no auto
43     xstab    SS   nn head  Light  no auto
44     xstab    SS   nn head Medium  no auto
45     xstab    SS   nn head Strong  no auto
46     xstab    SS   nn tail  Light  no auto
47     xstab    SS   nn tail Medium  no auto
   stability error sign wind   magn vis  use
48     xstab    SS   nn tail Strong  no auto
49      stab    LX   pp head  Light  no auto
50      stab    LX   pp head Medium  no auto
```

9. Introduction to Time-Series modelling with Splus

Diggle, 1990, p169, gives this excellent flowchart for guidance in arima modelling



We follow the approach in Venables and Ripley, and also use a PMEA data set.

```
library(MASS)
deaths    #  total UK monthly deaths from lung diseases for 1974-9
tsplot(deaths)
sablplot(sabl(deaths),title= "deaths") # seasonal components
acf(deaths)
acf(deaths,type= "partial")
spectrum(deaths)
spectrum(deaths,spans = 3)  # smoothed spectrum
spectrum(deaths,spans= c(3,3))
spectrum(deaths,"ar")
```

Now another dataset shown in Diggle, p42, on luteinising hormone.

```
lh ; tsplot(lh)
acf(lh)                  # looks like AR(1) or ARMA(1,1)
acf(lh,type= "partial")
spectrum(lh,"ar")
ar1 <- ar(lh,,1)
ar2 <- ar(lh)             #allowed free rein,chooses AR(3)
arima1 <- arima.mle(lh,model= list(order=c(1,0,0)))
                         # full MLE fit
2*arima1$loglik          # deviance - constant
arima.diag(arima1)       # diagnostics plot
arima3 <- arima.mle(lh,model=list(order= c(3,0,0)))
2*arima3$loglik          # not much better than AR(1)
arima.diag(arima3)
arima11 <- arima.mle(lh,model=list(order= c(1,0,1)))
2*arima11$loglik         #no better than AR(1)
arima.diag(arima11)
# Now use arima1 to forecast 12 steps
```

```
lh.fore <- arima.forecast(lh,n =12,model =arima1$model)
x <- lh.fore$mean ; sd <- lh.fore$std.err
tsplot(lh,x,x+2*sd,x-2*sd)
```

Now some popmusic data from 'The Independent', February 1994. First copy my files Splus/popmusic and Splus/popdata

```
source("popmusic")    # (this assumes you have BOTH files)
tsplot(ind)
acf(ind)
lind <- log(ind+1) ;tsplot(lind)  # and so on
```

Can you model the log-index? What is your prediction for 1994 ? Here is the popmusic file.
Data from 'The Independent', Wed Feb 23, 1994 "An Index of British penetration of the US singles market". The scoring system is :
give 30 points for the year's best-selling single, and go on down the scale to 1 point for the single that came 30th in that year's sale.
Thus the figure for 1993 is 28 pts for "UB40"(the 3rd best-seller)
+ 4 pts for "The Proclaimers"(27th)

```
pdata<- read.table("popdata", header=T); attach(pdata)
plot(year,ind)
# Here is the ``popdata" file.
year  ind
1960  0
1961  0
1962 14
1963  0
1964 179
1965 219
1966 131
1967 102
1968  48
1969  71
1970  61
1971  76
1972  38
1973  78
1974  36
1975 132
1976 105
1977 102
1978 166
1979  76
1980 142
1981  65
1982  36
1983 137
1984 111
1985 201
1986  70
1987  25
1988 170
1989  31
1990  38
1991  40
```

```
1992  48
1993  32
```

10. Survival Data Analysis.

We follow closely Venables and Ripley (1994) Chapter 11.
Two data-sets are used:
i)uncensored data on survival times for leukaemia (see Cox and Oakes, 1984, p9)
ii) The 2-sample Gehan data on remission times for leukaemia(Cox and Oakes, 1984, p7)

```
library(MASS)
attach(leuk) ; leuk
plot(log(time)~ag + log(wbc))       #log() is variance-stabilising here.
plot(survfit(Surv(time)~ag), lty= c(2,3))
```

These graphs suggest that
survival is BETTER with ag present than with ag absent, and
survival DECREASES as log(wbc) INCREASES.

```
legend(80,0.8,c("ag absent","ag present"),lty= c(2,3))
options(contrasts<-c("contr.treatment","contr.poly"))
leuk.glm <- glm(time ~ ag* log(wbc),Gamma(log))
```

Here we fit a gamma model, using the log-link. Check that you can write down the likelihood.

```
summary(leuk.glm,dispersion= 1)# sets df of gamma as 1. Thus, we have neg. exponential.
anova(leuk.glm)               # what is this telling us ?
# We drop the interaction term
leuk.glm <- update(leuk.glm, ~ . - ag:log(wbc))
summary(leuk.glm,dispersion= 1)
leuk.glmi <- glm(time ~ag*log(wbc),Gamma(inverse))
```

Does using the canonical link function improve the fit?

```
summary(leuk.glmi,dispersion= 1)
```

Again, we are forcing a neg exponential fit. Now we use survreg(), for exponential, Weibull and log-logistic regression analyses.

```
survreg(Surv(time) ~ag*log(wbc),dist= "exponential")
summary(survreg(Surv(time)~ag + log(wbc),dist= "exp"))
summary(survreg(Surv(time)~ag+log(wbc)))
summary(survreg(Surv(time)~ag+log(wbc),dist= "log"))
```

Now we will use a semi-parametric model, the Cox proportional hazards.

```
leuk.cox <- coxph(Surv(time)~ ag + log(wbc))
summary(leuk.cox)
detach("leuk")                  #to tidy our space.
```

Next we find the product-limit estimators of survival curves.

```
attach(gehan); gehan
plot.factor(gehan)
plot(log(time) ~ pair)          # variance- stabilising transformation again.
```

Now we will estimate the survivor function, using Greenwood's formula for standard errors. Some of what is written below is now out of date, since your version of Splus may have survfit() rather than surv.fit(). See Venables and Ripley, 1999, p371, for a method which replaces surv.fit() by survfit().

```
wt1 <- ifelse(treat=="control",1,NA) # to pick out control group
wt2 <- ifelse(treat=="6-MP",1,NA)    # to pick out treatment group
wt1 ; wt2                            # to check
fit1 <- surv.fit(time,cens,wt= wt1,type= "kaplan-meier",error= "greenwood")
fit1
fit2 <- surv.fit(time,cens,wt= wt2,type= "kaplan-meier",error= "greenwood")
fit2
surv.plot(time,cens,treat,lty= c(3,1),yscale= 100,
      xlab= "time of remission",ylab= "% survival")
legend(25,90,c("control","6-MP"),lty= c(1,3))

# or,a diy version,which has error-bars
plot(stepfun(fit1$time,fit1$surv),type= 'l',ylim= c(0,1),
      xlab= "time of remission",ylab= "survival")
t1 <- fit1$time ; s1 <- fit1$surv ; std1 <- fit1$std.err
t2 <- fit2$time ; s2 <- fit2$surv ; std2 <- fit2$std.err
lines(stepfun(t1,exp(log(s1) + 1.96*std1)),lty= 2)
lines(stepfun(t1,exp(log(s1) - 1.96*std1)),lty= 2)
lines(stepfun(t2,s2),lty= 3)
lines(stepfun(t2,exp(log(s2) + 1.96*std2)),lty= 2)
lines(stepfun(t2,exp(log(s2) - 1.96*std2)),lty= 2)
legend(1,0.2,c("control","6-MP","95% conf.int."),lty= c(1,3,2))
# or, use the packet-recipe
gehan.surv <- survfit(Surv(time, cens) ~ treat,conf.type= "log-log")
summary(gehan.surv)
plot(gehan.surv,conf.int= T,lty=c(3,2),log= T,
       xlab= "time of remission(weeks)",ylab=  "survival")

survreg(Surv(time,cens) ~ factor(pair)+treat,dist= "exp")
summary(survreg(Surv(time,cens)~treat,dist= "exp")
summary(survreg(Surv(time,cens)~treat))

help(surv.fit)
```

This enables us to find out about other options.
Now, to test for a difference between the 2 groups:

```
survdiff(Surv(time,cens) ~ treat, rho=0)    # This is the log-rank test
survdiff( Surv(time,cens) ~ treat,rho=1)    # almost Gehan-Wilcoxon test
                    # see Cox & Oakes p 124
```

11. Under the heading
"How long British monarchs have lived",
the Independent on Sunday (26/11/95) gave the Table below. This gives, for each of 40 monarchs, the date of death, the lifetime, and a 0 or 1 according to whether the death was natural or not. (The list omits Lady Jane Grey, who was executed aged 16 in 1553, after 2 weeks on the throne. Mary, wife of William of Orange, is listed separately as she was Queen in her own right.)

	death	length	natural
WilliamI	1087	60	0
WilliamII	1100	40	1
HenryI	1135	67	0
Stephen	1154	53	0
HenryII	1189	56	0
RichardI	1199	42	1
John	1216	48	0
HenryIII	1272	65	0
EdwardI	1307	68	0
EdwardII	1327	43	1
EdwardIII	1377	64	0
RichardII	1399	33	1
HenryIV	1413	47	0
HenryV	1422	34	0
HenryVI	1471	49	1
EdwardIV	1483	40	0
EdwardV	1483	12	1
RichardIII	1485	32	1
HenryVII	1509	52	0
HenryVIII	1547	55	0
EdwardVI	1553	15	0
Mary	1558	42	0
ElizabethI	1603	69	0
JamesI	1625	58	0
CharlesI	1649	48	1
CharlesII	1685	54	0
JamesII	1701	67	0
WilliamIII	1702	51	1
Mary(II)	1694	32	0
Anne	1714	49	0
GeorgeI	1727	67	0
GeorgeII	1760	76	0
GeorgeIII	1820	81	0
GeorgeIV	1830	67	0
WilliamIV	1837	71	0
Victoria	1901	81	0
EdwardVII	1910	68	0
GeorgeV	1936	70	0
EdwardVIII	1972	77	0
GeorgeVI	1952	56	0

We use R to plot the Survivor function for the **natural** lifetimes (so that, for example, William II counts as a CENSORED observation.)

```
library(survival}
monarchy.data = read.table("monarchy.data", header=T)
attach(monarchy.data) ; cens= 1-natural
```

```
Surv(length, cens)
fit = survfit(Surv(length, cens)~ 1) ; fit
summary(fit)
plot(fit) ;  abline(.5,0)
```

and here is the resultant survivor function. Counting forward from age 0 years, the first observed natural death was for Edward VI, who died aged 15 years: there are only 39 monarchs at risk at this age as (poor little) Edward V has been 'censored' at 12 years old.

time	n.risk	n.event	survival	std.err	lower 95% CI	upper 95% CI
15	39	1	0.9744	0.0253	0.9260	1.000
32	38	1	0.9487	0.0353	0.8820	1.000
34	35	1	0.9216	0.0435	0.8402	1.000
40	34	1	0.8945	0.0499	0.8018	0.998
42	32	1	0.8666	0.0557	0.7640	0.983
47	29	1	0.8367	0.0612	0.7249	0.966
48	28	1	0.8068	0.0659	0.6874	0.947
49	26	1	0.7758	0.0703	0.6495	0.927
52	23	1	0.7420	0.0749	0.6088	0.904
53	22	1	0.7083	0.0787	0.5696	0.881
54	21	1	0.6746	0.0819	0.5317	0.856
55	20	1	0.6408	0.0845	0.4950	0.830
56	19	2	0.5734	0.0880	0.4244	0.775
58	17	1	0.5397	0.0891	0.3905	0.746
60	16	1	0.5059	0.0897	0.3575	0.716
64	15	1	0.4722	0.0898	0.3253	0.686
65	14	1	0.4385	0.0895	0.2939	0.654
67	13	4	0.3036	0.0836	0.1769	0.521
68	9	2	0.2361	0.0774	0.1241	0.449
69	7	1	0.2024	0.0734	0.0994	0.412
70	6	1	0.1686	0.0684	0.0761	0.374
71	5	1	0.1349	0.0625	0.0544	0.335
76	4	1	0.1012	0.0552	0.0347	0.295
77	3	1	0.0675	0.0460	0.0177	0.257
81	2	2	0.0000	NA	NA	NA

The resulting plot is given in Figure 16.

Now compute the Kaplan-Meier estimates of the survivor function for the male monarchs and for the female monarchs, and try fitting parametric distributions to these. (Note, there are just 5 queens in the list.)

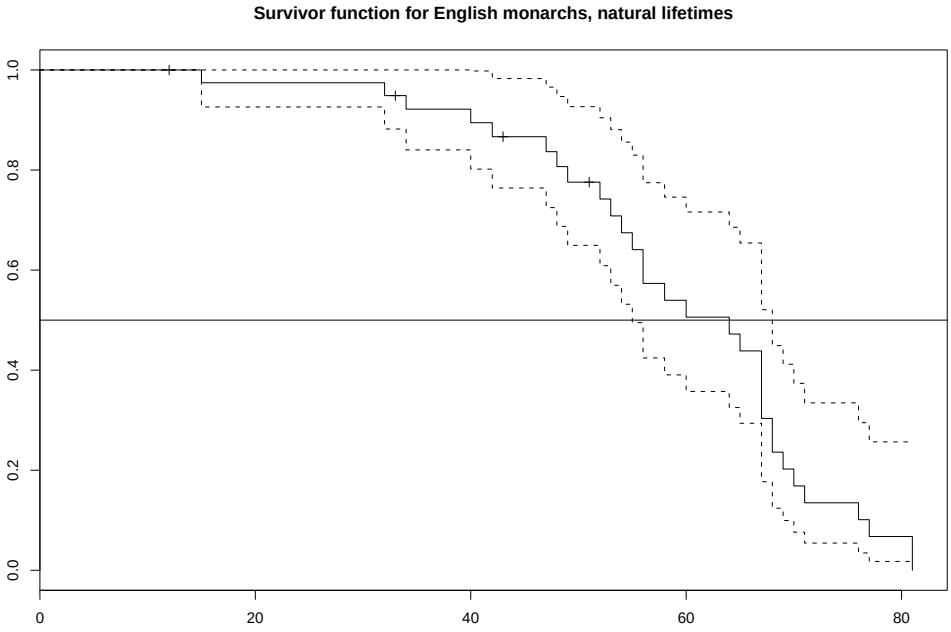


Figure 16: Survivor function for natural lifetime of English monarchs

12. Classical Metric Multidimensional Scaling, and Chernoff's faces.

```
a <- read.table("Dip97",header= T) # reads in the responses from 16 students
student.lab <- row.names(a)
a <- as.matrix(a)
d <- dist(a,metric= "binary")  # This  sets up the interstudent "distances"
new <- cmdscale(d,k= 2,eig= T) ; new
```

This finds the best 2-dimensional representation of the 16 points.

```
coord1 <- new$points[,1] # the first column
coord2 <- new$points[,2] # the second
par(pty<-"s")            # sets up a square plot
r <- range(new$points)
plot(coord1,coord2,type= "n")
text(coord1,coord2,seq(along= coord1)).
```

This labels the points by integers. Alternatively, we could use the default setting of a 2-dimensional representation, thus:

```
new <- cmdscale(d)
plot(new,type= "n")
text(new,labels= student.lab) # this time put the NAMES on the plot
faces(a, labels= student.lab)
```

How to insult your students!
Chernoff's faces (available in R via the package aplpack) represent up to 15 variables by features of cartoon faces as you will see in Figure 17.
The corresponding data set, 'Dip97', is given below.

```
eggs meat coffee beer UKres Cantab Female Sports Driver Left.h
```

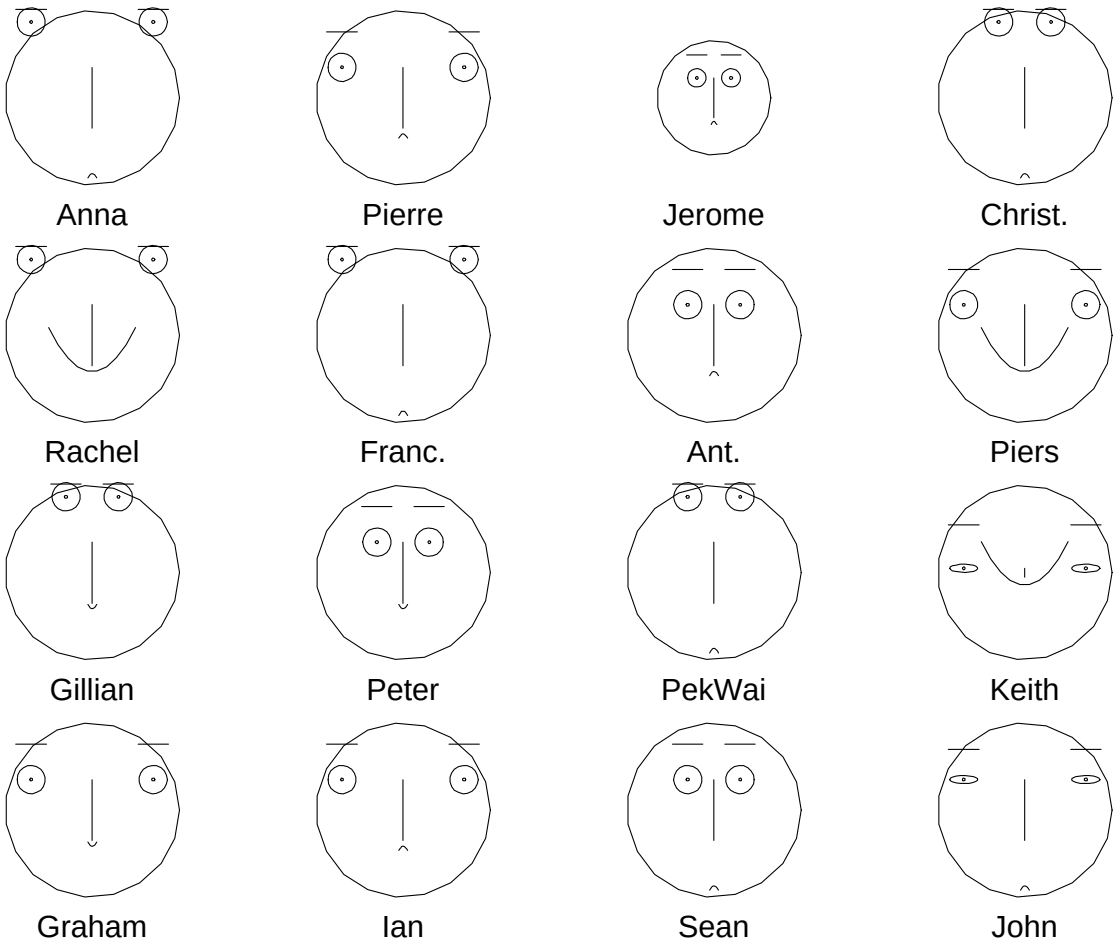


Figure 17: Chernoff's faces for the Diploma 1996-7 class

```
Anna    1    1     1    0    0      0     1       1      1      0
Rachel  1    1     1    1    1      1     1       1      1      0
Gillian 1    1     1    1    1      0     1       0      1      0
Graham  1    1     1    1    1      0     0       1      1      0
Pierre  1    1     1    1    0      0     0       1      1      0
Franc.  1    1     1    0    0      0     1       1      1      0
Peter   1    1     1    1    1      0     0       0      1      0
Ian     1    1     1    1    0      0     0       1      1      0
Jerome  0    1     1    1    0      0     0       0      1      0
Ant.    1    1     1    1    0      0     0       0      1      0
PekWai  1    1     1    0    0      0     1       0      1      0
Sean    1    1     1    0    0      0     0       0      1      0
Christ. 1    1     1    0    0      0     1       0      1      0
Piers   1    1     1    1    1      1     0       1      1      0
Keith   1    1     0    1    1      1     0       1      1      1
John    1    1     1    0    0      0     0       1      1      1
```

And finally, new for 2002, the following data MPhil/Part III, applied multivariate analysis, Feb 2002.

```
.....................................................................
        eggs  meat  coffee  beer  UKres  Cantab  Fem  sports driver Left-h specs
  Josh   y     y     y       y     y      n       n    y      n      n      y
  TjunKiat y   y     y       n     n      n       n    y      y      n      y
  Flora  y     y     y       y     y      n       y    y      y      n      y
  ChauLoong y  y     y       n     n      n       n    y      n      n      n
  Eleanor y    y     y       y     y      n       y    y      y      n      n
  Teresa  y    y     y       n     n      n       y    y      y      n      n
  Jim     y    y     y       y     y      y       n    y      y      y      n
  Mama    y    y     y       y     n      n       n    y      y      n      n
  Chao    y    y     y       y     n      n       n    y      y      n      y
  Qi      y    y     y       y     y      n       n    y      n      n      y
  LeeLee  y    y     n       y     n      n       y    y      y      n      y
  Karthi  y    y     y       n     n      n       n    n      n      n      y
  David   y    y     y       y     y      n       n    y      y      n      y
  Neeraj  y    y     n       n     y      n       n    n      n      n      y
  Cosme   y    y     n       n     n      n       n    y      y      n      y
  Arnaud  y    y     y       y     n      n       n    y      y      n      y
  Jochen  y    y     y       n     n      y       n    y      y      n      y
  Sophia  y    y     y       n     y      y       y    y      y      n      y
  Stephane y   y     n       n     n      n       n    y      y      n      n
  JimmyL  y    y     y       y     n      n       n    y      y      n      y
```

Note, the first 2 columns turn out to be unhelpful, so you may prefer to omit them before trying, eg

dist() for use with hclust() or cmdscale()

The above data set is of course based on rather trivial questions.

By way of complete contrast, here is a data set from The Independent, Feb 13, 2002. on 'Countries with poor human rights records where firms with British links do business'. It occurs under the headline

CORPORATE RISK: COUNTRIES WITH A BRITISH CONNECTION.

```
             1  2  3  4  5  6  7  8  9  10 11 12 13 14
SaudiArabia  1  0  0  0  0  1  0  1  0   0  1  1  0  1
```

```
Turkey       1  0  1  0  1  1  0  0  0   1  0  1  0  1
Russia       1  0  1  0  1  1  1  0  0   0  0  1  0  1
China        1  1  1  0  1  1  1  0  0   0  0  1  0  1
Philippines  1  1  1  0  0  0  0  0  1   0  0  1  1  0
Indonesia    1  1  1  0  0  1  1  1  0   0  0  1  0  0
India        1  0  1  0  1  0  0  1  1   0  1  1  0  0
Nigeria      0  0  1  0  0  0  1  0  0   0  0  1  1  0
Brazil       1  0  1  1  1  0  1  0  0   1  0  1  0  0
Colombia     1  1  1  1  1  0  0  0  0   1  0  1  0  0
Mexico       0  1  1  0  0  1  0  0  0   0  0  1  0  1
```

Key to the questions (1 for yes, 0 for no)
Violation types occurring in the countries listed

1 Torture
2 'Disappearance'
3 Extra-judicial killing
4 Hostage taking
5 Harassment of human rights defenders
6 Denial of freedom of assembly and association
7 Forced labour
8 Bonded labour
9 Bonded child labour
10 Forcible relocation
11 Systematic denial of women's rights
12 Arbitrary arrest and detention
13 Forced child labour
14 Denial of freedom of expression

Note that the *total* number of 1's in each row ranges from 4, for Nigeria, to 8, for China.
Figure 18 shows my 2-dimensional plot of the 11 countries, using the

```
method ="binary"
```

option in computing the between-countries distance matrix. (Of course, this treats the 14 different types of 'violation' as equally serious, which is not necessarily the correct thing to do.) In order to interpret the axes of this graph, I suggest the following:

```
a <- a[,-12] # to remove the 12th column from the matrix (it's all 1's)
b <- cbind(new, a) # new being the first 2 cmd co-ordinates
round(cor(b),2) # so that you can see, for example, which columns of a
are most closely correlated with new[,1]
```

You might like to compare the results of cmdscale with those of hierarchical clustering, as follows.

```
a = read.table("human.rights") # to read in the data
a = data.matrix(a)
d = dist(a, method="binary")
h = hclust(d, method="complete")
 # "complete" in R is same as "compact" in Splus
plclust(h)
```

The resulting graph is shown as Figure 19.

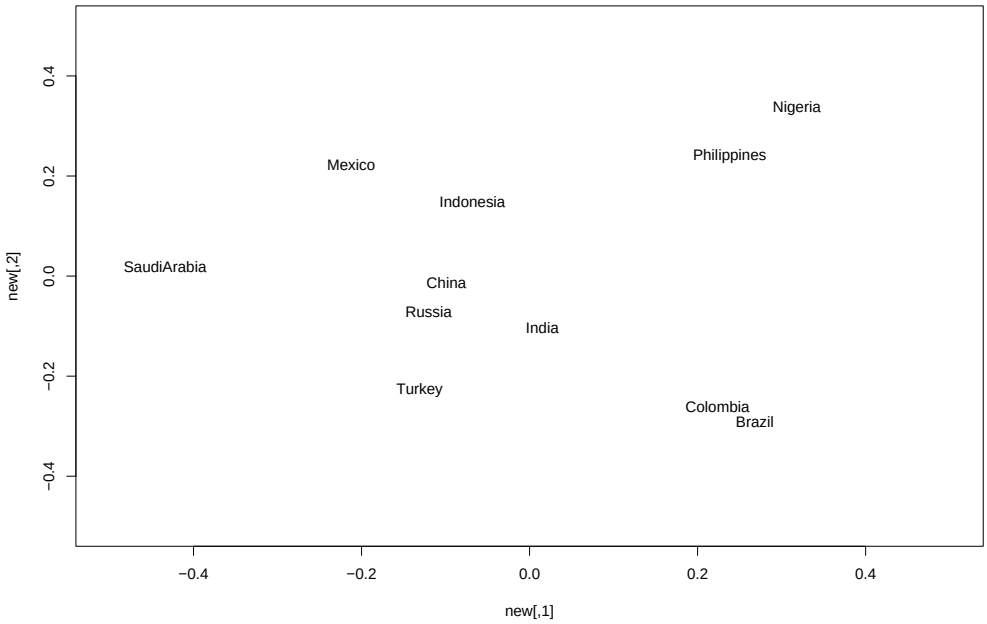


Figure 18: Countries with a British connection: human rights abuses

Figure 19: Hierarchical clustering for the countries data

And finally, a more cheerful dataset: 'Why workers in Britain are still chained to their desks', from The Independent, 30 April 2002. Here is the Table of 'How working conditions compare throughout Europe' (which I have edited slightly).

	Au	Be	De	Fi	Fr	Ge	Gr	Ir	It	Lu	Ne	Po	Sp	Sw	UK
statw	40	39	NA	40	35	48	48	48	40	40	45	40	40	40	48
Prod	90	128	99	99	113	102	74	94	113	199	119	63	81	95	92
AnnP	13	10	11	10	11	10	12	9	12	10	8	14	14	13	8
AnnL	25	20	25	25	25	20	22	20	NA	20	NA	22	20	25	20

Key: the countries are
Austria, Belgium, Denmark, Finland, France, Germany, Greece, Ireland, Italy, Netherlands, Portugal, Spain, Sweden and UK.
The rows are
statw= Statutory working hours per week
Prod = Productivity (GDP per hour) index: EU¡-100 (in 1999)
AnnP= Annual Public holidays
AnnL = Annual leave, days.

13. Analysis of a Repeated Measures design.

You see below the data from p28 of M.J.Crowder and D.J.Hand (1990) 'Analysis of Repeated Measures'.
To quote from Crowder and Hand, 'The effect of a vitamin E diet supplement on the growth of guinea pigs was investigated as follows. For each animal the body-weight was recorded at the ends of weeks 1,3,4,5,6 and 7. All animals are given a growth-inhibiting substance during week 1, and the vitamin E therapy was started at the beginning of week 5. Three groups of animals, numbering five in each, received respectively zero, low and high doses of vitamin E.'

The body weights (in grams) are given in the table below. The rows correspond to Animals $1, \ldots 15$, respectively, and the columns to the weeks $1, 3, 4, 5, 6, 7$. The first 5 rows are Group 1, the next 5 are Group 2, and the final 5 are Group 3. We reconstruct the analysis given by Crowder and Hand on p34, following Venables and Ripley (1997) Chapter 10.
This model allows for *three* sources of random variation: one is that between the 15 animals, one is the random interaction effect animals $\times$ occasions and and finally one is the 'error' variation.
The model to be fitted is

$$x_{ij} = \mu_{ij} + \alpha_{ij} + +\epsilon_{ij}$$

for $i = 1, \cdots, 15, j = 1, \cdots, 6$, where we assume that

$$\alpha_{ij} = \alpha_i^I + \alpha_{ij}^{IO}$$

where $\alpha_i^I, \alpha_{ij}^{IO}, \epsilon_{ij}$ are independent, with variances $\sigma_I^2, \sigma_{IO}^2, \sigma^2$ respectively. (The first 2 of these 3 terms are known as variance components.)
We assume that

$$\mu_{ij} = \mu_j^{(g)} \text{ for } \ i\epsilon \text{ Group } g.$$

```
x <- scan()
455 460 510 504 436 466
467 565 610 596 542 587
445 530 580 597 582 619
485 542 594 583 611 612
480 500 550 528 562 576
514 560 565 524 552 597
440 480 536 484 567 569
495 570 569 585 576 677
520 590 610 637 671 702
503 555 591 605 649 675
496 560 622 622 632 670
498 540 589 557 568 609
478 510 568 555 576 605
545 565 580 601 633 649
472 498 540 524 532 583

We <- c(1,3,4,5,6,7)
week <- We
```

First we plot the 15 'timetracks', on 3 separate plots, one for each of the 3 Groups. These are shown as Figures 20, 21 and 22 respectively.

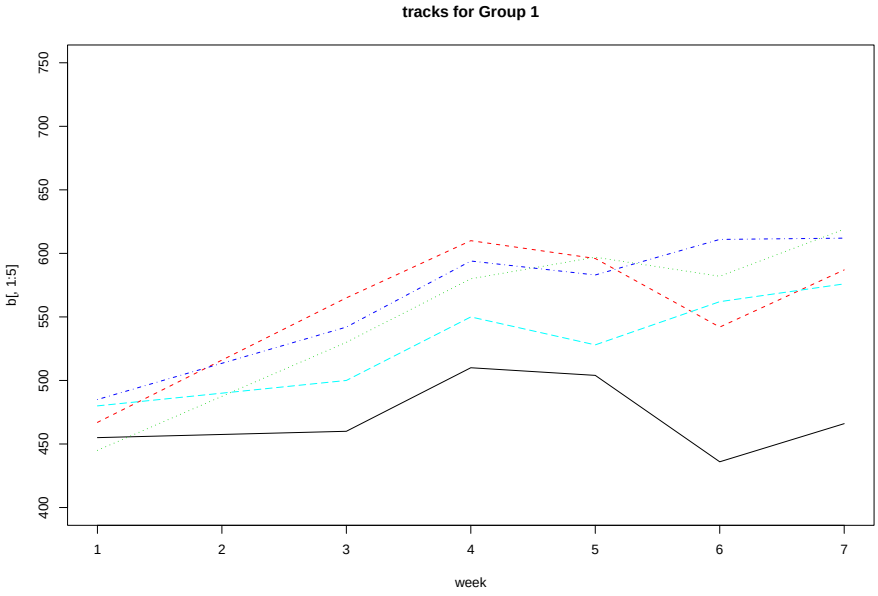


Figure 20: Time tracks showing growths for Group 1 guinea pigs

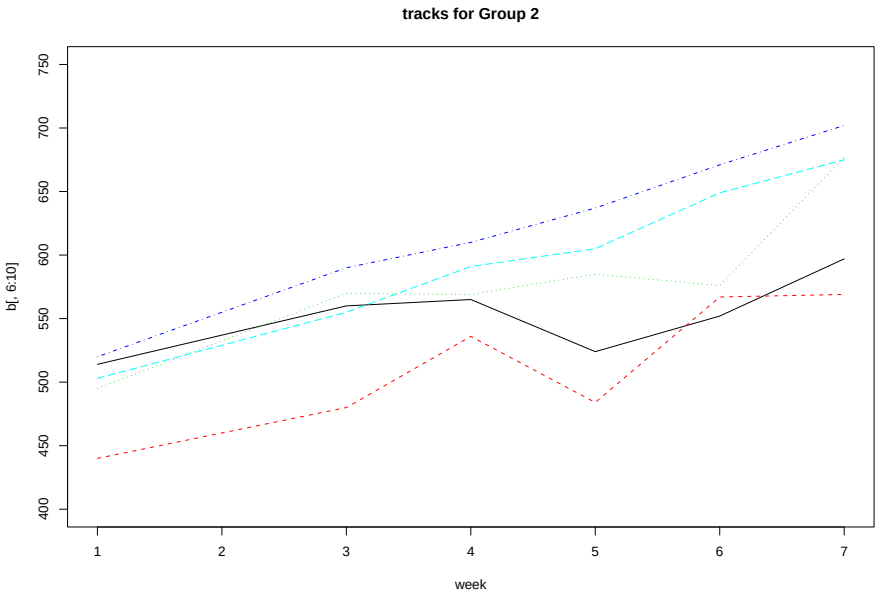


Figure 21: Time tracks showing growths for Group 2 guinea pigs

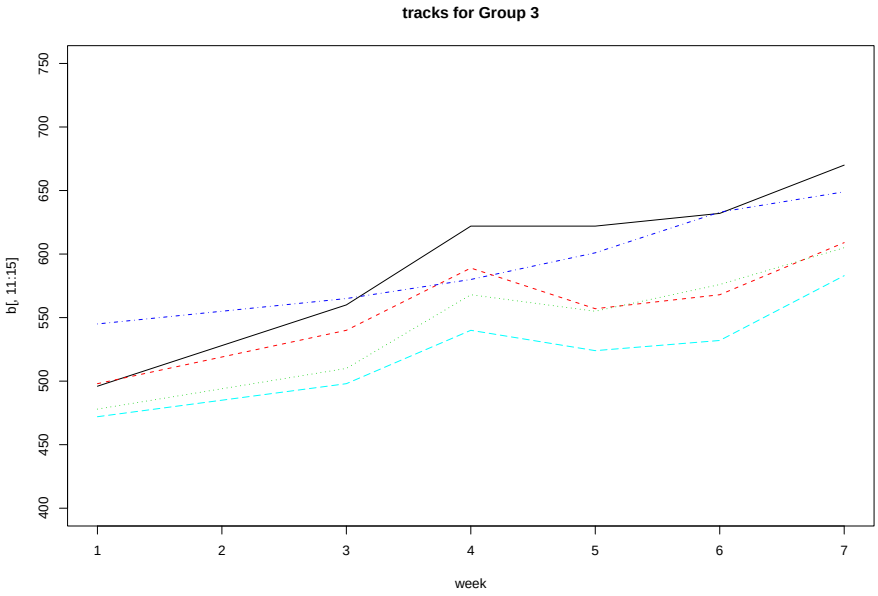


Figure 22: Time tracks showing growths for Group 3 guinea pigs

```
a <- matrix(x,nrow= 15,ncol= 6,byrow= T)
b<- t(a)
par(mfrow=c(3,1))
matplot(week,b[,1:5],type= "l", ylim= c(400,750))
matplot(week,b[,6:10],type= "l", ylim= c(400,750))
matplot(week,b[,11:15],type= "l", ylim= c(400,750))
library(nlme)  # for use in R
Gr <- 1:3 ; An <- 1:15
y <- expand.grid(We,An)
Week <- y[,1] ; Animal <- y[,2]
Group <- gl(3, 30, length=30, labels=c("zero", "low", "high")
Week <- factor(Week); Animal <- factor(Animal)
first.aov <- aov(x~Week*Group + Error(Animal))
summary(first.aov)
```

This shows that the Group*Week interaction is non-significant. So next we try

```
sec.aov <- aov(x~ Week + Group + Error(Animal))
summary(sec.aov)
```

This results in the following output, where you can see that the original 89 df have been partitioned into $89 = (2+12)+(5+70)$, giving us the 'between Animals' comparisons and the 'Within Animals' comparisons, respectively.

```
Error: Animal
          Df Sum Sq Mean Sq F value Pr(>F)
Group      2  18548    9274  1.0555 0.3782
Residuals 12 105434    8786

Error: Within
          Df Sum Sq Mean Sq F value    Pr(>F)
Week       5 142555   28511  47.164 < 2.2e-16 ***
```

```
Residuals 70  42315     605

sec.lme <- lme(x ~ Week + Group, random= ~1 | Animal)
summary(sec.lme) # for comparison
> summary(sec.lme)
Linear mixed-effects model fit by REML
 Data: NULL
       AIC      BIC    logLik
  831.9379 856.0051 -405.9689

Random effects:
 Formula: ~1 | Animal
        (Intercept) Residual
StdDev:    36.92713 24.58668

Fixed effects: x ~ Week + Group
               Value Std.Error DF   t-value p-value
(Intercept) 466.2333 18.068104 70 25.804220  0.0000
Week3        48.8000  8.977786 70  5.435639  0.0000
Week4        88.0667  8.977786 70  9.809397  0.0000
Week5        80.6000  8.977786 70  8.977715  0.0000
Week6        93.0667  8.977786 70 10.366328  0.0000
Week7       126.8667  8.977786 70 14.131176  0.0000
Grouplow     33.1333 24.202181 12  1.369023  0.1961
Grouphigh    26.7667 24.202181 12  1.105961  0.2904
 Correlation:
..................................................
Standardized Within-Group Residuals:
        Min          Q1         Med          Q3         Max
-2.62053931 -0.51705480  0.01798091  0.62523764  2.09203976

Number of Observations: 90
Number of Groups: 15
```

Note: I have used 'Groups' to mean treatments zero, low, high here. This was not such a good choice of name, as lme() uses 'Groups' to mean Animals in this context.
Compare this also with

```
summary(lm(x ~ Week + Group)) # which assumes that all observations are independent
```

Venables and Ripley show you how to look at residuals.
The current example follows the analysis given by Hand and Everitt. However 'Week' is definitely ordered in *time*, and so there may be a more suitable error structure than the symmetric one given here. See Venables and Ripley (1997) p312 for examples of other error structures.
Note that the function

```
glmmPQL()
```

available via library(MASS) provides a very general method of dealing with 'random effects' versions of generalized linear models. But beware: Hayley Jones, in her MPhil Applied Project, found that SPlus6 and R give different solutions when using this function on identical datasets.
(The problem seems to be connected with the fact that we are maximising a multi-modal log-likelihood function, and R may go off in the wrong direction.) For this reason we preferred to work with the SPlus version of the function. In either case, understanding all the output from glmmPQL() is tricky.

14. Fitting a beta-binomial distribution to the hospitals data

I checked the computations in Splus7 in July 2008.
E.C.Marshall and D.J.Spiegelhalter (1998) 'Reliability of league tables of *in vitro* fertilisation clinics: retrospective analysis of live birth rates', British Medical Journal, **316**, 1701-4 analyse the data from which the Table below has been constructed. To quote from E.C.Marshall's unpublished PhD thesis, which also includes these data, 'In July 1996 the Human Fertilisation and Embryology Authority reported on 25730 *in vitro* fertilisation treatments carried out in 52 clinics over the period from 1 April 1994 to 31 March 1995. An overall adjusted live birth rate of 14.5 % was found.'
In the Table below, r is the number of live births, and n the number of fertilisations. (r was computed from n and the observed percentage p, both of which were given in Marshall's PhD thesis.)

	r	n
Withington	7	147
ManchesterFS	41	506
Fazakerley	20	240
Ninewells	42	501
Hull	33	390
King'sColl	125	1453
BMIChiltern	13	149
Cromwell	39	427
Aberdeen	32	327
Walsgrave	45	458
Hartlepool	9	85
BUPALe'ster	12	110
UCH	41	366
WirralFC	17	141
GlasgowRI	105	876
SheffieldFC	80	661
Le'sterRI	14	114
LondonFC	100	786
StMary's	82	627
NewhamGH	9	68
EdinburghACU	59	447
BMIPortland	21	152
Washington	42	307
RoyalVicI	47	342
BourneHallC	185	1315
UHWales	24	168
BridgeFC	81	568
EsperanceH	31	212
WessexFS	60	404
ChurchillC	78	519
MidlandFS	120	787
UnivBristol	119	773
WolfsonFC	160	1004
RoyalMasonic	133	839
Northampton	36	223
NStaffs	19	116
LondonWomens	105	643
Guys&StThom	84	496
BMIPark	111	640
BUPARoding	38	211

```
    HollyHoFU  49  262
    BMIPriory  46  241
  S.Cleveland  20  104
    LeedsGenI 186  946
BMIChelsfield  42  208
    OxfordIVF 128  603
 SouthmeadGen  18   82
    Lister    244 1104
RMHBelfast    122  548
 StJames's    121  537
Birmingham     60  267
   NURTURE    204  861
```

First we will fit the binomial with constant probability p to these data, namely

$$r_i \sim \text{independent} \quad Bi(n_i, p), \ 1 \leq i \leq 52.$$

This is easily achieved by

```
hdata <- read.table("hospitals.data", header= T)
attach(hdata)
first.glm <- glm(r/n ~ 1, binomial, weights= n)
summary(first.glm)
```

which shows a deviance of 390.76, with $df = 51$. So we have substantial overdispersion with respect to the model of constant binomial parameter p. We will compute the binomial residuals, for comparison later with the betabinomial residuals.

```
p <- first.glm$fitted.values ; q <- 1-p
res <- (r-n*p)/sqrt(n*p*q)
sum(res^2) # as a check
chisq.test(cbind(r,n-r)) # as another check
# sqrt(n) * resid(first.glm) would give us the deviance residuals instead
```

Our next step is to allow one extra parameter: we assume that

$$r_i|p_i \sim Bi(n_i, p_i)$$

and assume further that p_i has the beta distribution, parameters θ, ϕ.
This has the consequence that each r_i then has a beta-binomial distribution, parameters n_i, θ, ϕ.
Again assume that all the r_i's are independent.
We pause to derive the frequency function for the beta-binomial. Now

$$f(r|p) = \binom{n}{r} p^r (1-p)^{n-r}, \ for \ r = 0, \cdots, n$$

where p has density $g(p)$ say, where

$$g(p) = \frac{\Gamma(\theta+\phi)}{\Gamma(\theta)\Gamma(\phi)} \ p^{\theta-1}(1-p)^{\phi-1}, \ \text{for } 0 \leq p \leq 1.$$

Thus, integrating with respect to p, we find that

$$\int f(r|p)g(p)dp = \binom{n}{r} \frac{\Gamma(\theta+\phi)}{\Gamma(\theta)\Gamma(\phi)} \frac{\Gamma(\theta+r)\Gamma(\phi+n-r)}{\Gamma(\theta+\phi+n)}.$$

In the S-Plus commands below, we compute

$$-\Sigma_i log f(r_i|\theta, \phi)$$

as MINUS the loglikelihood function, and then minimise it to find the maximum likelihood estimates of θ, ϕ. 'General optimization and maximum likelihood estimation' is given as Chapter 9 in Venables and Ripley (1997).

```
lbetabin <- function(p)
{
th <- p[1]
phi <- p[2]
sum( - lgamma(th + r) - lgamma(phi + n - r) + lgamma(th + phi + n) +
lgamma(th) + lgamma(phi) - lgamma(th + phi))
}
p <- c(.15,.85)
```

These are our initial estimates of θ, ϕ, taken from the binomial fit, and setting $\theta + \phi = 1$. One way to proceed is as follows

```
fit.first <- nlmin(lbetabin,p,print.level= 1) # this does not quite converge, but
fit.first$converged # shows that we have not yet reached convergence, but
fit.first$x         # shows that we have
#  estimates theta =10.73 , phi=63.07. So we use these as starting values, thus
p <-  fit.first$x
fit.next <-  nlmin(lbetabin,p,print.level= -1) # now quickly converges, giving
# the following estimates
fit.next$x
 10.89 63.04  # for theta, phi
```

Now we will try a different minimisation function.

```
p <- c(.15,.85) # the  same starting values as before
fit.betabin <- nlminb(start =  p, objective =  lbetabin, lower =  c(0, 0))
 # which gives convergence, and
fit.betabin$parameters
[1] 10.89 63.06            # and we need the corresponding se's, so
library(MASS)
vcov.nlminb(fit.betabin)   # gives us the approximate covariance matrix for these
                            parameter estimates
```

It is interesting that we find

$$\hat{\theta} = 10.89(se = 2.51), \hat{\phi} = 63.06(se = 14.85)$$

which corresponds to the beta-density for p, shown in Figure 23, which is quite sharply peaked. You can do this plot for yourself by

```
th <- 10.89; phi <- 63.06
p <- (1:100)/100
f <- dbeta(p,th,phi)
plot(p,f,type= "l")
```

We can use the parameter estimates to compute the correct estimated variance for r_i, and hence compute a χ^2 goodness of fit statistic for the model.

```
 th <- 10.89; phi <- 63.06; pi <- th/(th + phi)
betabin.resid <- (r - n*pi)/sqrt( n*pi *(1-pi)*(1+ (n-1)/(th + phi+1)))
plot(res,betabin.resid)
betabin.chi2 <- sum(betabin.resid^2)
```

This finds the χ^2 statistic as 50.35, with 50 df, showing that the inclusion of just 1 extra parameter gives a model that satisfactorily accounts for the 'over-dispersion' relative to the ordinary binomial. Here are the ordered binomial residuals.

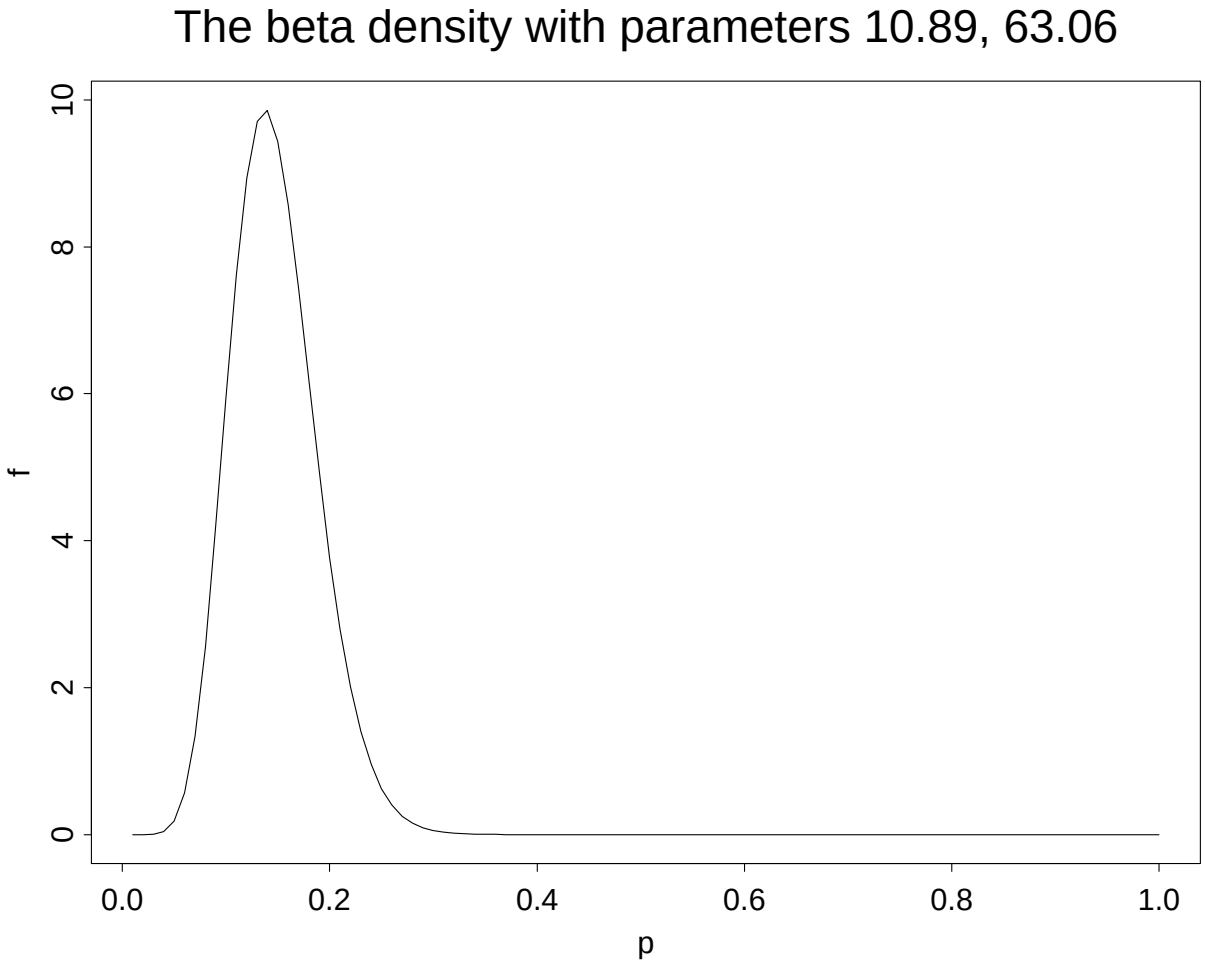


Figure 23: The beta density with parameters $10.89, 63.06$

```
round(sort(res),2)
# This shows us 'best' and 'worst' on crude 1-parameter binomial model
 King'sColl ManchesterFS Ninewells  Hull Withington Cromwell Walsgrave
      -6.85        -4.36     -4.16 -3.63      -3.48     -3.4     -3.11
 Fazakerley Aberdeen GlasgowRI BMIChiltern SheffieldFC   UCH LondonFC StMary's
       -2.9    -2.65     -2.51       -2.15        -2.1 -2.04     -1.8    -1.36
 BUPALe'ster Hartlepool EdinburghACU WirralFC BourneHallC Le'sterRI RoyalVicI
       -1.21      -1.14        -1.08    -0.98       -0.97     -0.82     -0.66
 Washington BridgeFC BMIPortland NewhamGH UHWales EsperanceH WessexFS
      -0.66    -0.51       -0.42    -0.41   -0.27      -0.16    -0.09
 ChurchillC MidlandFS UnivBristol NStaffs Northampton RoyalMasonic WolfsonFC
       0.01      0.18        0.29    0.41        0.47         0.67      0.81
 LondonWomens Guys&StThom S.Cleveland BUPARoding BMIPark HollyHoFU
         0.93        1.19         1.2       1.22    1.65      1.67
 SouthmeadGen BMIPriory BMIChelsfield Birmingham LeedsGenI OxfordIVF
         1.76      1.77          2.09       3.41         4      4.27
 RMHBelfast StJames's Lister NURTURE
       4.75      4.87   6.59    7.12
```

and here are the ordered beta-binomial residuals, which can also be compared to the standard

normal

```
round(sort(betabin.resid),2) # for betabinomial residuals
Withington ManchesterFS King'sColl Ninewells Hull Fazakerley Cromwell
      -1.99        -1.51      -1.46     -1.45 -1.4      -1.37    -1.26

 BMIChiltern Walsgrave Aberdeen   UCH Hartlepool BUPALe'ster GlasgowRI
       -1.2     -1.11    -1.09 -0.79      -0.74       -0.72     -0.64

 SheffieldFC WirralFC LondonFC Le'sterRI StMary's EdinburghACU NewhamGH
       -0.61    -0.53    -0.47     -0.47    -0.38        -0.35    -0.25

 Washington RoyalVicI BMIPortland BourneHallC BridgeFC UHWales EsperanceH
      -0.23     -0.22       -0.18       -0.16    -0.11   -0.09      -0.02

 WessexFS ChurchillC MidlandFS UnivBristol RoyalMasonic WolfsonFC Northampton
    0.03       0.07      0.12        0.16         0.26      0.29         0.3

 NStaffs LondonWomens Guys&StThom BMIPark BUPARoding S.Cleveland HollyHoFU
   0.32         0.37         0.5    0.61       0.69        0.84      0.86

 BMIPriory BMIChelsfield LeedsGenI SouthmeadGen OxfordIVF Birmingham
      0.93          1.15      1.16         1.28       1.5       1.67

 RMHBelfast Lister StJames's NURTURE
       1.73   1.74      1.79     2.1
```

We could compare the 2 sets of residuals graphically via

```
par(mfrow= c(2,1))
qqnorm(res) ; qqline(res)
qqnorm(betabin.resid); qqline(betabin.resid)
```

This gives the graphs (note that the y-axes have different scales) shown in Figure 24.
Exercise:
The sample correlation matrix for $\hat{\theta}, \hat{\phi}$ suggests that we could find a much 'better' parametrisation, in which the two parameters are closer to being orthogonal. Experiment with the parametrisation

$$\pi = \theta/(\theta + \phi),\ \psi = \theta + \phi.$$

Afterword.
One of the objectives of Marshall and Spiegelhalter in looking at this table was to produce a 'reliable' ranking of the hospitals, since a ranking based only on the crude success rate can be quite misleading. How do we address this question with the benefit of our beta-binomial model?

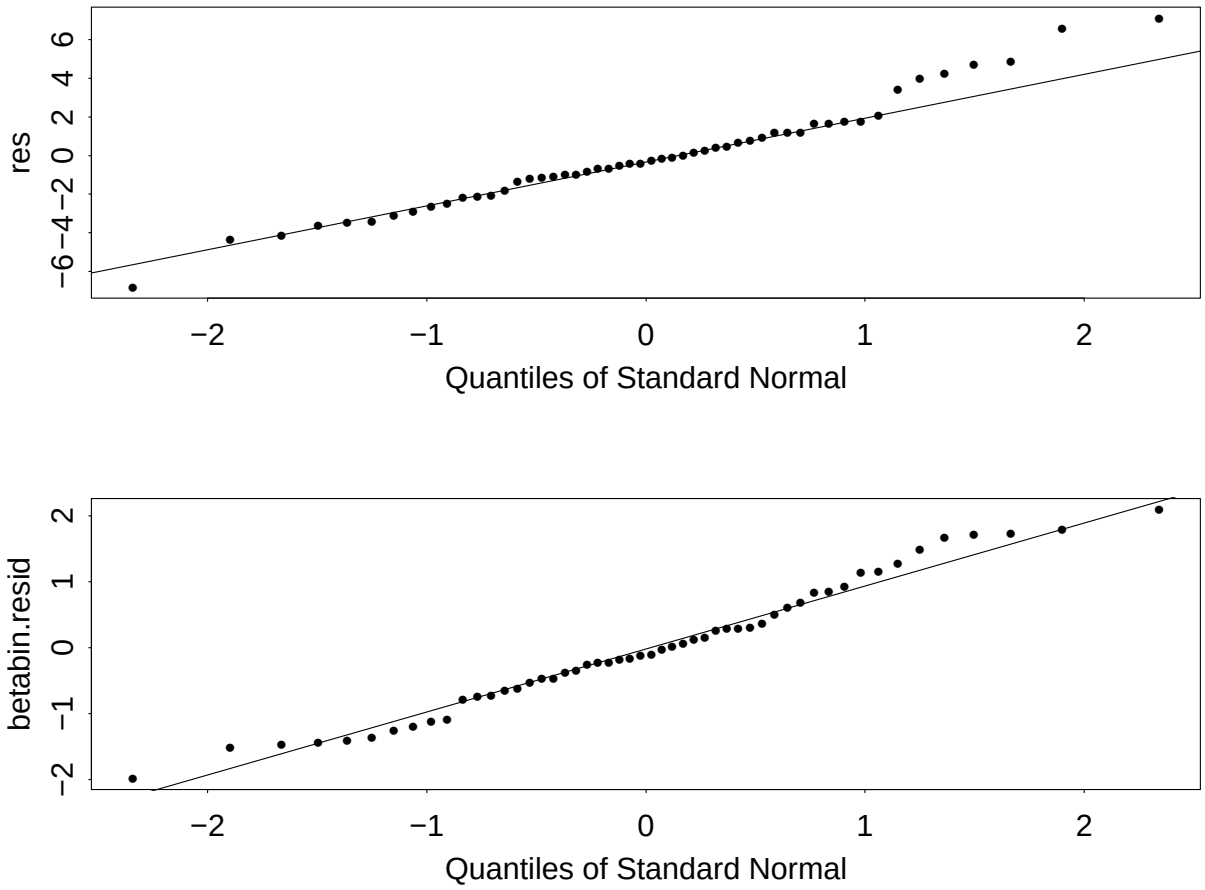


Figure 24: residuals for the binomial and the betabinomial distributions

15. Multinomial logistic regression and classification.
Here we follow the example on Cushing's syndrome, given in Venables and Ripley (1999) p350, and we give some supplementary explanation.
The dataset is given below.

	Tetrahydrocortisone	Pregnanetriol	Type
a1	3.1	11.70	a
a2	3.0	1.30	a
a3	1.9	0.10	a
a4	3.8	0.04	a
a5	4.1	1.10	a
a6	1.9	0.40	a
b1	8.3	1.00	b
b2	3.8	0.20	b
b3	3.9	0.60	b
b4	7.8	1.20	b
b5	9.1	0.60	b
b6	15.4	3.60	b
b7	7.7	1.60	b

```
   b8              6.5          0.40    b
   b9              5.7          0.40    b
  b10             13.6          1.60    b
   c1             10.2          6.40    c
   c2              9.2          7.90    c
   c3              9.6          3.10    c
   c4             53.8          2.50    c
   c5             15.8          7.60    c
   u1              5.1          0.40    u
   u2             12.9          5.00    u
   u3             13.0          0.80    u
   u4              2.6          0.10    u
   u5             30.0          0.10    u
   u6             20.5          0.80    u
```

The data 'are on diagnostic tests on patients with Cushing's syndrome, a hypersensitive disorder associated with over-secretion of cortisol by the adrenal gland. The dataset has three recognised types of the syndrome represented as

`a, b, c.`

(These encode 'adenoma', 'bilateral hyperplasia', and 'carcinoma', and represent the underlying cause of over-secretion. This can only be determined histopathologically.) The observations are urinary excretion rates (mg/24h) of the steroid metabolites tetrahydrocortisone and pregnanetriol, and are considered on a log scale.'
In the analysis given below, we do not use the last 6 rows of the data, for which the 'Type' was unknown. We fit the following logistic model

$$log(P(b|x)/P(a|x)) = \beta_2^T x, \quad log(P(c|x)/P(a|x)) = \beta_3^T x$$

with x as a 3-dimensional vector, having first element 1, and

$$P(a|x) + P(b|x) + P(c|x) = 1.$$

Thus, for example, if an object has covariate value x, we will predict it as b if $\beta_2^T x > 0$, & $\beta_2^T x > \beta_3^T x$. We use library(nnet) to maximise the resulting multinomial log-likelihood.

```
library(MASS)
library(nnet)
Cushings # to view the data
tp <- factor(Cushings$Type[1:21])
Cf <- data.frame(tp<-tp, Tetra <- log(Cushings[1:21,1]),
Pregna <- log(Cushings[1:21,2]))
attach(Cf)
Tetra <- Tetra- mean(Tetra) ; Pregna <- Pregna -mean(Pregna)
```

this improves the parametrisation, making convergence of maximisation algorithm faster.

```
cush.multinom <- multinom(tp ~ Tetra + Pregna, Hess =  T, maxit =  250)
cush.multinom
Call:
multinom(formula =  tp ~ Tetra + Pregna, Hess =  T, maxit =  250)

Coefficients:
  (Intercept)    Tetra    Pregna
b    7.288130 14.39930 -0.244936
```

```
c    2.385204 16.26469  3.358042

Residual Deviance: 12.30232
AIC: 24.30232
```

Note that the residual deviance is not an absolute measure of goodness of fit. In fact, the parameters are estimated rather imprecisely, as we see from

```
summary(cush.multinom)
Call:
multinom(formula =  tp ~ Tetra + Pregna, Hess = T, maxit = 250)

Coefficients:
  (Intercept)    Tetra    Pregna
b    7.288130 14.39930 -0.244936
c    2.385204 16.26469  3.358042

Std. Errors:
  (Intercept)    Tetra    Pregna
b    7.755119 13.73160 0.6692837
c    8.276217 13.38103 2.0996099

Residual Deviance: 12.30232
AIC: 24.30232
.........................................

round(predict(cush.multinom, type= "probs"),3)
      a    b    c
 1 0.89 0.01 0.10
 2 0.99 0.01 0.00
 3 1.00 0.00 0.00
 4 0.50 0.50 0.00
 5 0.43 0.56 0.00
 6 1.00 0.00 0.00
 7 0.00 0.99 0.01
 8 0.60 0.40 0.00
 9 0.58 0.42 0.00
10 0.00 0.99 0.01
11 0.00 1.00 0.00
12 0.00 0.29 0.71
13 0.00 0.97 0.03
14 0.00 1.00 0.00
15 0.01 0.99 0.00
16 0.00 0.91 0.09
17 0.00 0.10 0.90
18 0.00 0.06 0.94
19 0.00 0.63 0.37
20 0.00 0.13 0.87
21 0.00 0.03 0.97
```

The above shows that there is considerable uncertainty about the predicted class for some of the observations, eg numbers 8, 9.

```
predict(cush.multinom)
 [1] a a a a b a b a a b b c b b b b c c b c c
table(predict(cush.multinom),tp)
```

```
  a b c
a 5 2 0
b 1 7 1
c 0 1 4
```

which shows that the 'confusion matrix' is not so bad as we might have expected: the total of the offdiagonal terms is 5, so that the misclassification error rate with this method is 5/21, ie .24.
For this dataset, the logistic multinomial regression is actually less successful in prediction than the simple classification tree, which we can easily obtain as follows.

```
> first.tree <- tree(tp ~ Tetra + Pregna) # use rpart() if in R
> first.tree
node), split, n, deviance, yval, (yprob)
      * denotes terminal node

1) root 21 44.220 b ( 0.2857 0.4762 0.2381 )
  2) Tetra<-0.323364 8  8.997 a ( 0.7500 0.2500 0.0000 ) *
  3) Tetra>-0.323364 13 17.320 b ( 0.0000 0.6154 0.3846 )
    6) Pregna<0.582761 7  0.000 b ( 0.0000 1.0000 0.0000 ) *
    7) Pregna>0.582761 6  5.407 c ( 0.0000 0.1667 0.8333 ) *
```

```
> summary(first.tree)
Classification tree:
tree(formula = tp ~ Tetra + Pregna)
Number of terminal nodes:  3
Residual mean deviance:  0.8002 = 14.4 / 18
Misclassification error rate: 0.1429 = 3 / 21
```

What this is telling us is the following.
If you know neither Tetra nor Pregna, then you should predict all 21 cases to be 'b'.
But, this is not a 'terminal node' (in fact it is the root node), and we can improve our prediction. Our next step is
now look at Tetra, there are 8 cases for which Tetra < -0.323364, and all these cases should be predicted as 'a',
The remaining 13 cases have Tetra > -0.323364, and if you are allowed no further information, then predict all these cases as 'b'.
But this also is not a 'terminal' node: you can improve things further by
looking at Pregna for these 13 cases.
The 7 cases for whom Pregna < 0.582761 should be predicted as 'b' (this will be perfectly correct, and so must be a terminal node).
The remaining 6 cases for whom Pregna > 0.582761 should be predicted as 'c' (this will be not quite correct, but is a terminal node nonetheless).
You can check that this classification tree is then incorrect in exactly 3 out of the 21 cases, so the overall error rate is 0.1429.
I haven't given you the story here about the deviance, but that's something you can work out for yourself. The root deviance is easily seen to be

$$44.220 = -2n \sum p_i \log(p_i)$$

where $n = 21$, and $p_1 = 6/21, p_2 = 10/21, p_3 = 5/21$.

```
> post.tree(first.tree, file="tree.ps", pointsize=6) # for a 'pretty' plot
```

We show how the sample space is divided up by the following plot, given as Figure 25.

```
> plot(Tetra, Pregna, type="n") # blank plot so far
```

```
> text(Tetra, Pregna,c("a","b","c")[tp]) # putting the points on with their labels
> abline(v= - 0.323364) # for the vertical dividing line
> abline(h=   0.582761) # for the horizontal dividing line
```

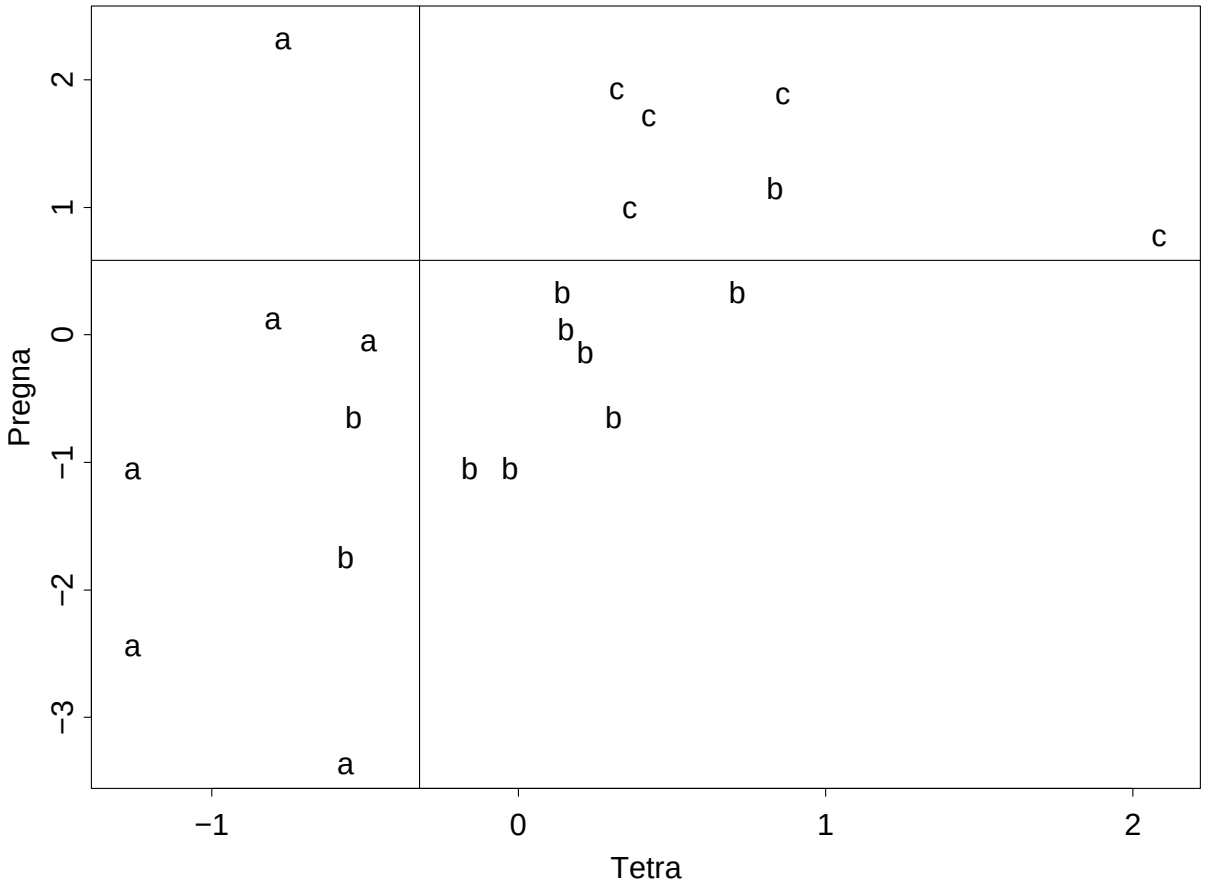


Figure 25: How the classification tree for the Cushings data is constructed

New for August 2008: Olympic medals data
The Independent, 6 August 2008, presents the dataset below on 'British medal hauls at the past 10 Olympics'.

```
           Gold Silver Bronze
Athens2004  9    9     12
Sydney2000 11   10      7
Atlanta96   1    8      6
Barcelona92 5    3     12
Seoul88     5   10      9
LosAngeles84 5  11     21
Moscow80    5    7      9
Montreal76  3    5      5
Munich72    4    5      9
MexicoCity68 5   5      3
```

Suppose we wish to find whether the distribution of medals into Gold, Silver, Bronze has changed

over these 10 Games. Specifically we will fit

$$\log(p_{2i}/p_{1i}) = \alpha_2 + \beta_2 i, \text{ and } \log(p_{3i}/p_{1i}) = \alpha_3 + \beta_3 i,$$

for $i = 10, 9, \ldots, 1$ (the first row corresponding to 2004) with p_{1i}, p_{2i}, p_{3i} being the respective probabilities that in a given year, a medal is Gold, Silver, Bronze. ($p_{1i} + p_{2i} + p_{3i} = 1$, of course.) If you apply the analysis below, you will see that $\beta_2 = 0, \beta_3 = 0$ and that for the British teams, the probabilities of Gold, Silver, Bronze, respectively in any given year have remained more or less constant at $.24, .33, .42$.
Suggestion for analysis:

```
library(MASS) ; library(nnet)
Olympics <- read.table("Olympics.data", header=T)
Year <- 10:1 ; attach(Olympics) ; Olmat <- cbind(Gold, Silver, Bronze)
chisq.test(Olmat)
```

Strangely, the chisq statistic is 19.8 on 18 df, so really there's not a lot more to be said, but we will press on with the more complex multinomial logistic model as an exercise.

```
par(mfrow=c(1,2))
plot(Silver/Gold ~ Year) ; plot(Bronze/Gold ~ Year)
# these plots show no obvious trends
Total <- Gold + Silver + Bronze
first.multinom <- multinom(Olmat~ Year, Hess=T)
summary(first.multinom)
eigen(first.multinom$Hess) # to check Hessian is positive-definite
Olp <- predict(first.multinom, type="probs"); round(Olp,2) # for fitted probabilities
Olp <-  Olp*Total ; Olp<- round(Olp,2) # for fitted frequencies
cbind(Olp, Olmat) # for comparison
base.multinom <- multinom(Olmat ~1, Hess = T) # baseline model
#in which probabilities do not change with year
round(predict(base.multinom, type="probs"),2)
```

The resulting graphs are shown as Figure 26.
Note that there is a perceptible increase in the **Total** number of medals gained by Great Britain since 1968. This must be due in part to the increase in the number of Olympic events over the years; there were 172 events in 1968, and in 2008 there will be a total of 302 events. Try

```
plot(Total ~ Year)
```

But we see that Los Angeles 1984, in which there was a Total of 37 medals, was a 'strange' year, and in fact that was the Olympic Games which was boycotted by nearly all the Eastern Bloc countries. For this reason we now try

```
first.glm <- glm(Total[-6] ~ Year[-6], poisson) # to omit Los Angeles 1984
summary(first.glm) # shows a residual deviance of 6.83 on 7 df, hence a good fit
YYear <- 11:1
fv = exp(2.58660 +(0.07147*YYear))
plot(fv ~ YYEAR, type="l") # for fitted values, including for 2008
```

This gives
a predicted Total of 29.2 medals in 2008 (can you work out a confidence interval?), of which we expect
$7.1, 9.7, 12.4$ as Gold, Silver, Bronze respectively.
Contrast this with the rather upbeat prediction given before the start of the 2008 games by Nick Harris in The Independent. He predicted
$16, 17, 26$ as Gold, Silver, Bronze respectively.
As of August 19, 2008, it looks as though the actual outcome will be much better than my predictions of $7.1, 9.7, 12.4$**!**

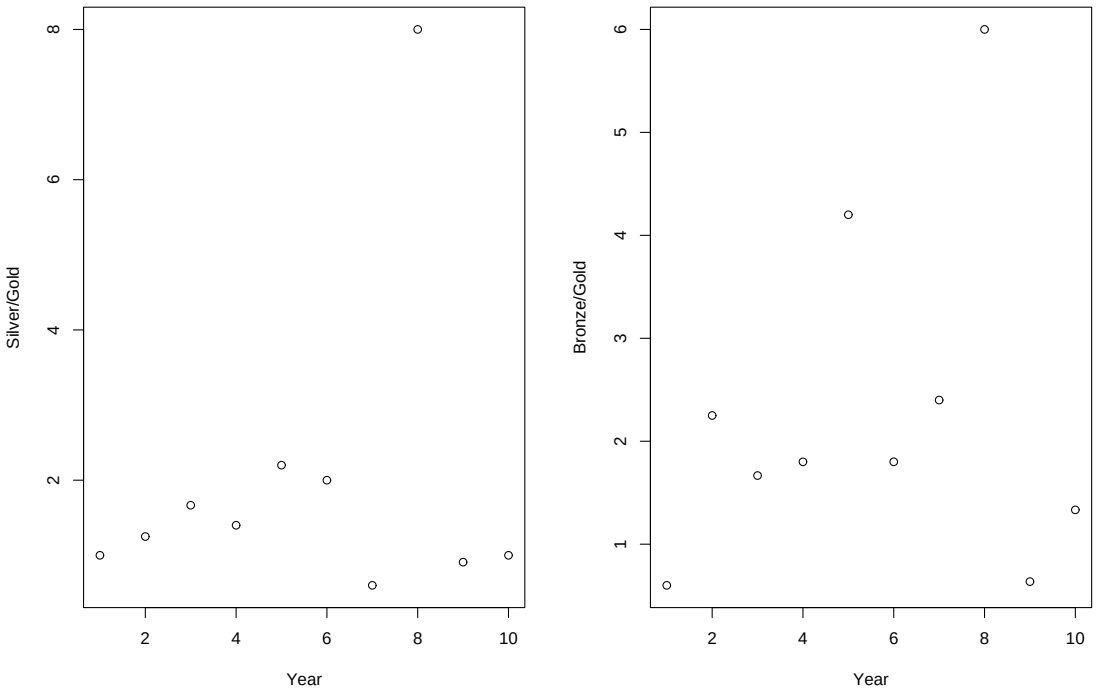


Figure 26: Have the British medals distributions changed from 1968 to 2004?

18. **New for 2003.** Here are two interesting new data sets with which you can experiment. The first data set was collected by Mohammad Raza of Wolfson College, Cambridge, for his May 2003 Mathematical Tripos Part III essay 'Analysis of a large and complex data set'. I am very grateful to Mohammad for his permission to use the data.
We have data for 50 famous recent movies, compiled via the Internet Movie Database (IMDB).
KEY to the variables
year= year when the film was released
gross= amount of money in millions of US dollars, the film made, in the US
budget= amount in millions of $ US spent in making the film,
(note that the 2 monetary figures given above are not 'adjusted' in any way, eg for inflation.)
rating (also male rating & female rating)= 'opinion of registered users of the IMDB website'. Each individual gives an integer score between 1 ('awful') and 10 ('excellent') for a given film. The rating for the film is then calculated as the average score. The 'male rating & female ratings' correspond to scores given by men, women, respectively.
AWARDS:
These are AA, GG, BAFTA for Academy Award, Golden Globe Award and British Academy of Film & Television Arts Awards, with a 1 indicating that the film won the 'Best Picture' award, and a 0 indicating that it did not.
GENRE:
Each movie is given a 1 or a 0 to indicate whether it was in a particular 'genre', eg comedy, scifi .... A film can be in more than one such genre.

	year	gross	budget	rating	malerating	femalerating
Titanic	1997	600.743	200.000	7.0	6.8	7.2
StarWars	1977	460.936	11.000	8.8	8.8	8.6
ET	1982	434.949	10.500	7.8	7.8	8.1

```
                        SWPhantomMenace 1999 431.065 115.000    6.7        6.7          6.5
                             SpiderMan 2002 403.706 139.000    7.7        7.7          7.8
                           JurassicPark 1993 356.763  63.000    7.3        7.3          7.1
                           ForrestGump 1994 329.452  55.000    8.0        8.0          8.0
                           HarryPotterI 2001 317.558 130.000    7.3        7.2          8.0
                         LOTRFellowship 2001 313.364 109.000    8.9        8.9          9.0
                           TheLionKing 1994 312.775  79.300    7.6        7.5          7.9
                          TheGodfather 1972 134.821   6.000    9.0        9.1          8.6
                TheShawshankRedemption 1994  28.341  25.000    9.0        9.0          9.0
                         TheGodfatherII 1974  57.300  13.000    8.9        8.9          8.4
                         SchindlersList 1993  96.067  25.000    8.8        8.8          8.9
                     ShichininNoSamurai 1954   0.187   0.500    8.9        9.0          7.7
                              Memento 2000  25.530   5.000    8.8        8.8          8.6
                         DrStrangelove 1964   9.164   1.800    8.7        8.8          8.1
                       SWReturnOfTheJedi 1983 309.064  32.500    8.1        8.1          8.0
                         IndependenceDay 1996 306.200  75.000    6.0        5.9          6.5
                          TheSixthSense 1999 293.502  55.000    8.3        8.2          8.4
                  SWEmpireStrikesBack 1980 290.159  18.000    8.7        8.7          8.4
                            HomeAlone 1990 285.761  15.000    6.2        6.2          6.7
                                Shrek 2001 267.652  60.000    8.1        8.1          8.3
            HowTheGrinchStoleChristmas 2000 260.031 123.000    6.0        5.9          6.4
                                 Jaws 1975 260.000  12.000    8.2        8.3          7.9
             OneFlewOverTheCuckoosNest 1975 112.000   3.000    8.7        8.7          8.5
                          RearWindow 1954   1.559   1.000    8.7        8.7          8.6
                  RaidersOfTheLostArk 1981 242.374  20.000    8.6        8.7          8.4
                     TheUsualSuspects 1995  23.272   6.000    8.7        8.7          8.7
                     NorthByNorthwest 1959  13.275   4.000    8.6        8.7          8.5
                          PulpFiction 1994 107.930   8.000    8.6        8.7          7.9
                               Psycho 1960  32.000   0.800    8.6        8.6          8.3
                 TheSilenceOfTheLambs 1991 130.727  22.000    8.5        8.6          8.5
                     LawrenceOfArabia 1962   0.342  12.000    8.6        8.6          8.4
                         Monsters,Inc 2001 255.870 115.000    8.1        8.0          8.4
                               Batman 1989 251.189  35.000    7.3        7.3          7.1
                           MenInBlack 1997 250.148  90.000    6.8        6.8          7.0
                           ToyStory2 1999 245.823  90.000    8.2        8.2          8.3
                             Twister 1996 241.700  92.000    5.9        5.8          6.3
                          GhostBusters 1984 238.600  30.000    7.4        7.4          7.5
                        BeverlyHillsCop 1984 234.760  15.000    7.1        7.1          7.1
                           CastAway 2000 233.630  90.000    7.3        7.3          7.4
           TheLostWorldJurassicPark 1997 229.074  73.000    5.4        5.4          5.2
                     AmericanBeauty 1999 130.058  15.000    8.5        8.5          8.3
                         Goodfellas 1990  46.836  25.000    8.5        8.6          8.0
                            Vertigo 1958   3.200   2.479    8.5        8.6          8.3
                       ApocalypseNow 1979  78.800  31.500    8.5        8.5          8.0
                           TheMatrix 1999 171.383  63.000    8.5        8.5          8.4
                          TaxiDriver 1976  21.100   1.300    8.4        8.5          7.9
                        SomeLikeItHot 1959  25.000   3.500    8.5        8.5          8.6

                                AA GG BAFTA comedy drama action horror fantasy
                            Titanic  1  1     1      0     1      0      0       0
                           StarWars  1  1     1      0     0      1      0       1
                                 ET  1  1     1      0     0      0      0       1
                        SWPhantomMenace  0  0     0      0     0      1      0       0
                             SpiderMan  0  0     0      0     0      1      0       1
```

```
                          JurassicPark  0  0     0      0     0      1      1       0
                           ForrestGump  1  1     1      1     1      0      0       0
                           HarryPotterI  0  0     0      0     0      0      0       1
                         LOTRFellowship  1  1     1      0     0      0      0       1
                           TheLionKing  0  0     1      0     0      0      0       0
                           TheGodfather  1  0     1      0     1      0      0       0
                    TheShawshankRedemption  1  0     0      0     1      0      0       0
                           TheGodfatherII  1  0     1      0     1      0      0       0
                           SchindlersList  1  1     1      0     1      0      0       0
                        ShichininNoSamurai  0  1     0      0     1      1      0       0
                                  Memento  0  0     0      0     1      0      0       0
                             DrStrangelove  1  1     0      1     0      0      0       0
                          SWReturnOfTheJedi  0  0     0      0     0      1      0       1
                            IndependenceDay  0  0     0      0     0      1      0       0
                             TheSixthSense  1  1     0      0     1      0      1       0
                       SWEmpireStrikesBack  0  0     0      0     0      1      0       1
                                 HomeAlone  0  0     1      1     0      0      0       0
                                     Shrek  0  1     1      1     0      0      0       1
HowTheGrinchStoleChristmas  0  0     0      1     0      0      0       1
                                      Jaws  1  1     1      0     0      1      1       0
 OneFlewOverTheCuckoosNest  1  1     1      0     1      0      0       0
                               RearWindow  0  1     0      0     0      0      0       0
                       RaidersOfTheLostArk  1  1     0      0     0      1      0       0
                          TheUsualSuspects  0  1     0      0     0      0      0       0
                          NorthByNorthwest  0  0     0      0     0      0      0       0
                                PulpFiction  1  1     1      0     1      1      0       0
                                     Psycho  0  0     0      0     0      0      1       0
                      TheSilenceOfTheLambs  1  1     1      0     0      0      1       0
                          LawrenceOfArabia  1  1     1      0     1      0      0       0
                              Monsters,Inc  0  0     0      1     0      0      0       1
                                    Batman  0  0     0      0     0      1      0       1
                                 MenInBlack  0  0     1      1     0      1      0       0
                                  ToyStory2  0  0     1      1     0      0      0       1
                                    Twister  0  0     0      0     0      1      0       0
                                GhostBusters  0  0     1      1     0      0      0       1
                            BeverlyHillsCop  0  0     1      1     0      1      0       0
                                   CastAway  0  0     0      0     1      0      0       0
                  TheLostWorldJurassicPark  0  0     0      0     0      1      1       0
                             AmericanBeauty  1  1     1      0     1      0      0       0
                                 Goodfellas  1  1     1      0     1      0      0       0
                                    Vertigo  0  0     0      0     1      0      0       0
                              ApocalypseNow  1  1     1      0     1      1      0       0
                                 TheMatrix  0  0     0      0     0      1      0       0
                                TaxiDriver  1  1     0      0     1      0      0       0
                              SomeLikeItHot  0  1     1      1     0      0      0       0

                                    scifi romance thriller animation
                                  Titanic     0       1        0         0
                                 StarWars     1       0        0         0
                                       ET     1       0        0         0
                           SWPhantomMenace     1       0        0         0
                                 SpiderMan     1       0        0         0
                              JurassicPark     1       0        1         0
                               ForrestGump     0       0        0         0
```

HarryPotterI	0	0	0	0
LOTRFellowship	0	0	0	0
TheLionKing	0	0	0	1
TheGodfather	0	0	0	0
TheShawshankRedemption	0	0	0	0
TheGodfatherII	0	0	0	0
SchindlersList	0	0	0	0
ShichininNoSamurai	0	0	0	0
Memento	0	0	1	0
DrStrangelove	1	0	0	0
SWReturnOfTheJedi	1	0	0	0
IndependenceDay	1	0	0	0
TheSixthSense	0	0	1	0
SWEmpireStrikesBack	1	0	0	0
HomeAlone	0	0	0	0
Shrek	0	1	0	1
HowTheGrinchStoleChristmas	0	0	0	0
Jaws	0	0	1	0
OneFlewOverTheCuckoosNest	0	0	0	0
RearWindow	0	0	1	0
RaidersOfTheLostArk	0	0	0	0
TheUsualSuspects	0	0	1	0
NorthByNorthwest	0	1	1	0
PulpFiction	0	0	1	0
Psycho	0	0	1	0
TheSilenceOfTheLambs	0	0	1	0
LawrenceOfArabia	0	0	0	0
Monsters,Inc	0	0	0	1
Batman	0	0	1	0
MenInBlack	1	0	0	0
ToyStory2	0	0	0	1
Twister	0	0	1	0
GhostBusters	1	0	0	0
BeverlyHillsCop	0	0	0	0
CastAway	0	0	0	0
TheLostWorldJurassicPark	1	0	1	0
AmericanBeauty	0	0	0	0
Goodfellas	0	0	0	0
Vertigo	0	0	1	0
ApocalypseNow	0	0	0	0
TheMatrix	1	0	1	0
TaxiDriver	0	0	1	0
SomeLikeItHot	0	1	0	0

Worksheet 17. Fun and Games for British Union leaders (2004), and Hawks and doves at the Monetary Policy Committee (2007)

The dataset below was given in The Independent on May 29, 2004, under the headline
'One out of order, all out of order'.
'This week's unseemly brawl (at a barbecue in Hampstead) by Aslef officials continued a rich tradition of union leaders' excesses.'
The columns in the Table below give the 'out of order rating' for the categories
1. Fisticuffs, 2. Big Dinners, 3. Champagne socialist, 4. Luxury Travel,
5. Beer (no sandwiches), 6. Colourful Language, 7. Expenses Enthusiast, 8. Gender issues.
(By the way, if English is not your first language, you may need to get someone to explain some of the above (euphemisms) to you.)
The ratings are given for the 8 union leaders listed below (I omit their surnames).

```
                    1 2 3 4 5 6 7 8
Joe 'The Cherub'    0 0 3 2 0 0 0 0
John 'Big Boss'     0 0 0 0 0 1 1 2
Andy 'Chasse-Spleen'0 3 1 0 0 0 1 0
Derek               2 0 0 0 0 1 0 0
Shaun               2 0 0 0 2 1 0 0
Roger 'The Dodger'  0 1 1 0 0 0 2 0
Bollinger Bob       0 1 3 0 0 1 1 1
Raucous railwayman  0 1 2 0 2 2 0 0
```

You could, for example, construct a matrix to show the dissimilarities between all pairs of the 8 union bosses listed above.
Moving to a much more respectable scenario, but with data of the same structure, The Independent on July 2, 2007, gave the following data set on the voting of the 9 members of the Monetary Policy Committee with respect to the UK interest rates, under the heading 'Hawks, doves and pigeons: who influences UK interest rates?'. There were 9 successive monthly meetings, the first being 4/5 October, 2006.
Here we denote 1 to mean 'votes for an increase (of 0.25%), 0 to mean 'votes for no change in interest rate' and -1 to mean 'votes for a decrease (of 0.25%)'.
(For the record, interest rate was initially 4.75%, and at the end of the 6/7June meeting was 5.5%.)
We illustrate the positions of the 9 members of the MPC firstly with hierarchical clustering, and then by classical scaling. You may disagree with my (default) choice of metric. The individuals are plotted via hierarchical clustering in Figure 27, and via classical scaling in Figure 28. This latter is less successful as there are 3 pairs of coincident points, reflecting the fact that there are 3 pairs of individuals who vote identically. You could experiment with 'jitter' to improve the look of this plot.

```
             4/5Oct 8/9Nov 6/7Dec 10/11Jan 7/8Feb 7/8Mar 4/5Apr 9/10May 6/7Jun
Blanchflower      0      0      0       0       0      -1      0       1       0
Besley            1      1      0       1       1       0      1       1       1
Sentance          1      1      0       1       1       0      1       1       1
King              0      1      0       1       0       0      0       1       1
Gieve             0      1      0       1       0       0      0       1       1
Tucker            0      1      0       0       0       0      0       1       0
Bean              0      1      0       0       0       0      0       1       0
Barker            0      1      0       1       0       0      0       1       0
Lomax             0      0      0       0       0       0      0       1       0
```

```
a <- read.table("MPCdata.July2", header=T)
MPCnames <- row.names(a)
a <- as.matrix(a)
```

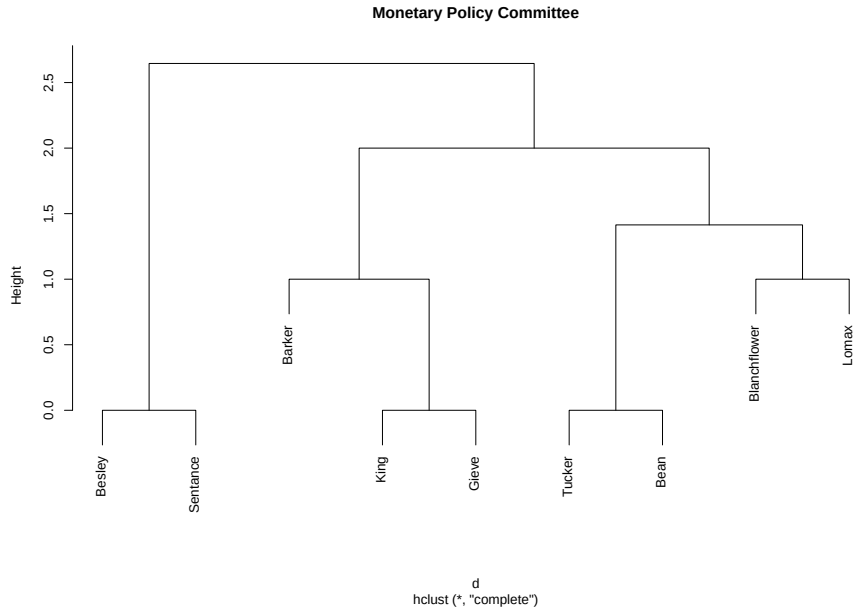


Figure 27: The monetary policy committee, hierarchical clustering

```
d <- dist(a)
clust.MPC <- hclust(d)
postscript("MPC.ps")
plclust(clust.MPC, hang=0.1,labels=MPCnames,main="Monetary Policy Committee")
dev.off()
loc <- cmdscale(d) ; x <- loc[,1] ; y <- loc[,2]
plot(x,y, type="n", main = "cmdscale for Monetary Policy Committee")
text(x,y, MPCnames, cex=1 )
```

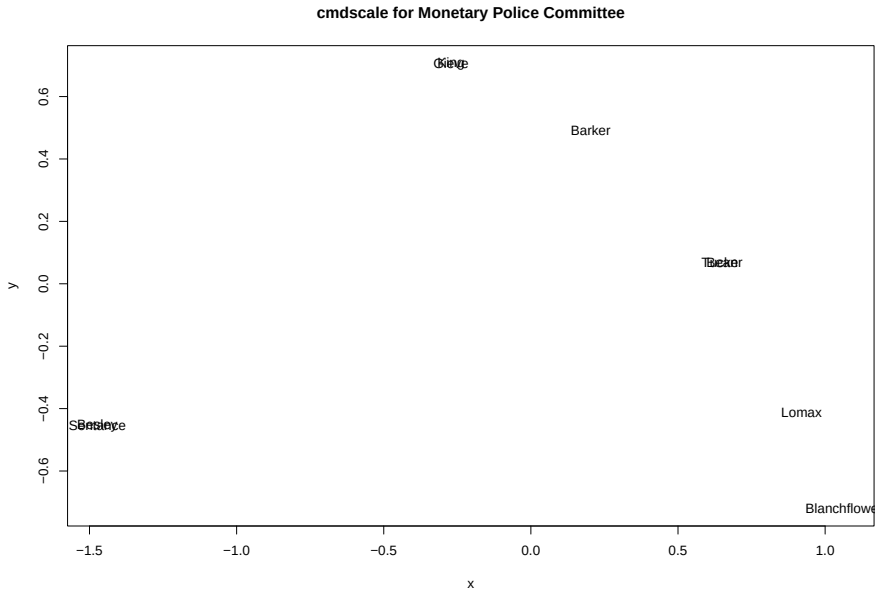


Figure 28: The monetary policy committee, classical multidimensional scaling

Worksheet 18: using capture-recapture data to estimate a total population size.
Agresti, in his 1994 *Biometrics* article
'Simple capture-recapture models permitting unequal catchability and variable sampling effort', Vol 50, pp 494-500, (and also in his 2002 book) gives a table of counts, first discussed by Cormack in 1985. This dataset was obtained from the results of a Capture-Recapture study of Snowshoe Hares, and consists of a 2^6 contingency table, with one missing entry, on the numbers of hares in a closed population which were trapped on each of 6 successive trapping days.
We need to set up suitable notation in order to describe the data precisely, thus
let $a = 0$ if an animal is NOT captured on the first day, and let $a = 1$ if it was captured on the first day.
Define $b = 0, 1, ..., f = 0, 1$ for the remaining sequence of 5 days.
The sequence of 64 entries in the variate 'count' follows the pattern

```
a= 0 1 0 1 0 1.................    0 1
b= 0 0 1 1 0 0 1 1 ............    1 1
c= 0 0 0 0 1 1 1 1 0 ...
d= 0 0 0 0 0 0 0 0 1 1  .....
e= 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 ..
f=  32 0's followed by 32 1's
```

We can set up this nested pattern of 0's and 1's using expand.grid() as shown below.

```
count <- scan()
NA 3 6 0 5 1 0 0
3  2 3 0 0 1 0 0
4  2 3 1 0 1 0 0
1  0 0 0 0 0 0 0
4  1 1 1 2 0 2 0
4  0 3 0 1 0 2 0
2  0 1 0 1 0 1 0
1  1 1 0 0 0 1 2
# data from Agresti (2002) p512
```

Thus you see that we don't know the number NOT caught on any of the 6 days, and for example, 6 animals were caught on day 2, but not on any of the other 5 days.
(There were 2 wretched creatures who were caught on every one of the 6 days.)

```
x <- expand.grid(a=0:1, b=0:1, c=0:1,d=0:1, e=0:1,f=0:1)
x[1:10,] # as a check
attach(x)
sum(count[2:64])
```

This shows that a total of 68 animals were seen at least once each.
Our aim is to fit a model to this table of $2^6 - 1$ counts, in order to estimate the number of hares in this (closed) population which were never seen at all: this enable us to estimate the total population size.

```
A <- factor(a) ; B <- factor(b); C <- factor(c) ; D <- factor(d); E<- factor(e)
F<- factor(f)
```

First we fit a model which assumes mutual independence of the 6 catching occasions, but which does not assume equal catchability. (The output has been slightly reduced.)

```
> first.glm <- glm(count~ A+B+C+D+E+F, poisson)
> summary(first.glm)
 ......................

    Coefficients:
                Estimate Std. Error z value Pr(>|z|)
(Intercept)   1.9553     0.2891   6.765 1.34e-11 ***
A1           -1.3061     0.2875  -4.543 5.55e-06 ***
B1           -0.5194     0.2491  -2.085 0.037052 *
C1           -1.0128     0.2681  -3.778 0.000158 ***
D1           -0.6351     0.2520  -2.520 0.011735 *
E1           -0.8170     0.2585  -3.160 0.001575 **
F1           -0.2970     0.2460  -1.207 0.227357

    Null deviance: 112.846  on 62  degrees of freedom
    Residual deviance:  58.314  on 56  degrees of freedom
    AIC: 154.50

> exp(1.9553)
[1] 7.066039
```

(You can see that a total of 16 animals were caught on day 1, compared with a total of 32 animals caught on day 6: perhaps the animals were getting tired, and/or the trappers were getting better at their task.)
The Residual deviance of 58.314 on 56 degrees of freedom shows us that the model does not fit very well, but we will still use the estimate of the intercept to provide us with an estimate of the count for which $a = 0,\ b = 0, \ldots f = 0$, giving us a value of $exp(1.9553) = 7.066039$, and hence an estimate of $68 + 7.1 = 75.1$ as our estimate of N, the total population size. (We could use $68 + exp(1.9553 + / - 2 * 0.2891)$ to give us our confidence interval for N.)
Agresti (1994) discusses various models which might fit better than the simple model of mutual independence, and therefore which might provide more accurate estimates of N. For simplicity here, we discuss only one generalization of the independence model: namely the model which allows all 2-factor interactions between $A, B, \ldots, F$. There are 15 such interactions, each with 1 df. We again edit the resulting output somewhat.

```
next.glm <- glm(count~ (A+B+C+D+E+F)^2 , poisson)
summary(next.glm)
```

```
    Coefficients:
                Estimate Std. Error z value Pr(>|z|)
(Intercept)  3.60498    0.51850   6.953 3.58e-12 ***
A1          -2.28082    0.57134  -3.992 6.55e-05 ***
B1          -1.82090    0.51966  -3.504 0.000458 ***
C1          -2.31716    0.55609  -4.167 3.09e-05 ***
D1          -2.14780    0.54364  -3.951 7.79e-05 ***
E1          -2.06819    0.53846  -3.841 0.000123 ***
F1          -2.09074    0.54177  -3.859 0.000114 ***
.....................
A1:E1        1.46053    0.66617   2.192 0.028348 *
.....................
C1:F1        1.59899    0.64082   2.495 0.012588 *
D1:F1        1.79685    0.60052   2.992 0.002770 **

    Null deviance: 112.846  on 62  degrees of freedom
    Residual deviance:  32.424  on 41  degrees of freedom
    AIC: 158.61

> exp(3.60498)
[1] 36.78095
```

Hence this more complex model (for which in fact only 3 of the 15 interactions are significant) gives us a point estimate of N as $36.8 + 68 = 104.8$.
The dataset discussed above is by now rather old (though a classic, no doubt). Agresti's 1994 paper gave a GLIM program for the log-linear analysis, discussed a variety of possible models (including latent class models) for this dataset. If you want a new dataset as a challenge, try the following, taken from 'Capture-recapture methods to size alcohol-related problems in a population' by Corrao, Bagnardi, Vittadini and Favilli, *J.Epidemiol. Community Health* 2000;54;603-610.
Our object is to estimate the total number of individuals with alcohol-related problems (ARP) in the target population, by combining data from 4 different (and clearly non-independent) 'flagging' sources.
Here is the table of data, as published on p 606 by Corrao et al. (You may need to think a bit in order to put it into R/S-Plus.)

```
Gender   M   M  F   F
Age      Y   O  Y   O  Total
Patients flagged by exactly 1 source
F1      30  40  6  12  88
F2      31  12  7   7  57
F3      12   5  3   2  22
F4      46  81 12  16  155

Patients flagged by exactly 2 sources
F1&F2    2   1  0   0   3
F1&F3    1   0  0   0   1
F1&F4    1   2  0   0   3
F2&F3    2   2  0   1   5
F2&F4    3   3  1   0   7
F3&F4    3   1  0   0   4

Patients flagged by exactly 3 sources
```

```
F1&2&3   0   0  0   0   0
F1&2&4   0   0  0   0   0
F1&3&4   1   0  0   0   1
F2&3&4   1   0  1   0   2

Patients flagged by all 4 sources
F1&2&3&4 1   0  0   0   1
----------------------------
Total  134 147 30  38   349
----------------------------
```

Key to above table
Gender, M = male, F =female
Age, Y = under 50 yrs, O = 50 yrs or older
The sources for 'flagging' the patients were
F1 = self-help volunteering groups (similar to Alcoholics Anonymous)
F2 = psychiatric ambulatory
F3 = Public Alcohology Service
F4 = hospital discharges.
The catchment area was 'all residents in the area of Voghera, a Northern Italy rural area with an economy based on vinegrowing and wine production... with a resident population of 132618 over 15 years in age.'
Corrao et al discuss various log-linear models, and conclude that the target population contained approximately 2500 individuals with ARP.
How does this compare with your estimate? What is your confidence interval? Do you have to treat the table for men differently from that for women?
(The authors conclude that the answer to this question is No, but that Young and Old should be analysed separately.)

Index