INTRODUCTION TO R (or S-PLUS) FOR GENERALIZED LINEAR MODELLING

P.M.E.Altham, Statistical Laboratory, University of Cambridge, CB3 0WB.

April 2, 2013

Contents

Preface

Special thanks must go to Dr R.J.Gibbens for his help in introducing me to S-Plus. Several generations of keen and critical students for the Cambridge University Diploma in Mathematical Statistics (pre 1998) and for the MPhil in Statistical Science have made helpful suggestions which have improved these worksheets. Readers are warmly encouraged to tell the author
P.M.E.Altham@statslab.cam.ac.uk
of any further corrections or suggestions for improvements.
Some draft solutions may be seen at `http://www.statslab.cam.ac.uk/~pat/bigsol.pdf` The S-Plus worksheets were prepared for the MPhil course. Since I retired in September 2005, I have gradually edited and updated all my worksheets. (This is an ongoing process)
I started to use R in about 1998, after the S-Plus worksheets were first constructed, and so many of the examples given here were used as the basis for my undergraduate worksheets in R: see for example `http://www.statslab.cam.ac.uk/~pat/redwsheets.pdf`
The R worksheets include many more graphs (with the corresponding R code) than are given in the S-Plus worksheets.
Additionally, I wrote S-Plus/R worksheets for multivariate analysis (and other examples) which may be seen at `http://www.statslab.cam.ac.uk/~pat/misc.pdf`
All worksheets may be used for any educational purpose provided their authorship (P.M.E.Altham) is acknowledged.

References

For R and S-plus material
Venables, W.N. and Ripley, B.D.(2002) Modern Applied Statistics with S-plus. New York: Springer-Verlag.
also the previous editions of the same.
For statistics text books
Agresti, A.(2002) Categorical Data Analysis. New York: Wiley.
Collett, D.(1991) Modelling Binary Data. London: Chapman and Hall. Also the 2nd edition (2003).
Crawley, M.(2002) Statistical Computing: an introduction to data analysis using S-Plus. Chichester: Wiley.
Dobson, A.J.(1990) An introduction to Generalized Linear Models. London: Chapman and Hall.
Lindsey, J.K. (1995) Modelling Frequency and Count Data. Oxford Science Publications.
Pawitan, Y. (2001) In All Likelihood: Statistical Modelling and Inference Using Likelihood. Oxford Science Publications.

The main purpose of the small index is to give a page reference for the $first$ occurrence of each of the S-plus commands used in the worksheets. Of course, this is only a small fraction of the total of S-plus commands available, but it is hoped that these worksheets can be used to get you started in S-plus.
Note that S-plus has an excellent help system : try, for example
?lm

You can always inspect the CONTENTS of a given function by, eg
lm
A problem with the help system for inexperienced users is the sheer volume of information it gives
you, in answer to what appears to be a quite modest request, eg
?scan
But you quickly get used to skimming through and/or ignoring much of what 'help' is telling you.
At the present time the help system still does not QUITE make the university teacher redundant,
largely because (apart from the obvious drawbacks such as lacking the personal touch of the univer-
sity teacher) the help system LOVES words, to the general exclusion of mathematical formulae and
diagrams. But the day will presumably come when the help system becomes a bit more friendly.
Thank goodness it does not yet replace a good statistical textbook, although it contains a wealth
of scholarly information (see the section on factor analysis as a good example of this).
Many many useful features of S-plus have NOT been illustrated in the worksheets that follow. The
keen user of lm() and glm() in particular should be aware of the following
i) use of 'subset' for regression on a subset of the data, possibly defined by a logical variable (eg
sex=="male")
ii) use of 'update' as a quick way of modifying a regression
iii) 'predict' for predicting for (new) data from a fitted model
iv) poly(x,2) (for example) for setting up orthogonal polynomials
v) summary(glm.example,correlation=F)
which is useful if we want to display the parameter estimates and se's, but not their correlation
matrix.
vi)summary(glm.example,dispersion=0)
which is useful for a glm model (eg Poisson or Binomial) where we want to ESTIMATE the scale
parameter ϕ, rather than force it to be 1. (Hence this is useful for data exhibiting overdispersion.)
Almost all of the analyses given below can be carried out, with minor changes of syntax, within
R, which is also available on the Statslab system. R is a free 'clone' of S-Plus: you can download
it for free if you wish. (See my web-page for further remarks on R.)
A free version of LaTeX may be obtained from
http://www.miktex.org/

Before you start using Splus or R on our system, you will need a basic knowledge of Linux (the
control language). I find that for everyday purposes, the only Linux commands that I use are as
given below, with small examples

```
passwd
ls (list what's in the current directory)
ls -ltr
ls -ltr *.tex  (to see all the files with a .tex postscript, in
reverse order of creation)
mkdir project (make the directory 'project')
rmdir project (remove the directory 'project')
cd oldproject (change to the directory 'oldproject')
cp data1 data2 (beware that this will over-write data2, if it already exists)
rm data1 (remove the file 'data1')
cat data1 data2>bigdata   (ie concatenate first 2 into the last-named file)
more useful (to see the file 'useful' on the screen)
lp  Tuesday.ps  (to print the document 'Tuesday.ps')
psnup -2 Tuesday.ps | lp (to print the same doc. with 2 pages per sheet)
gv useful.ps (to see a postscript file on the screen)
ps2pdf useful.ps (to convert a postscript file into a pdf file)
munpack funny (for files that reach me in funny format, eg a .doc)
ps x (to see my jobs)
kill -9 2073
 (kill job number 2073, useful if locked into say an S-Plus session)
chmod (for changing 'permissions', as in example below)
chmod go-r * (you may never need this)
```

You will also need to know how to mail (receiving, saving, sending etc) and how to edit files. Probably emacs is best for you.

Please READ your mail regularly, and delete unwanted messages, so that your filespace does not become cluttered.

You may get the odd VERY IMPORTANT email, eg about MPhil exam choices, but you'll also get a lot of rather unimportant messages (not quite junk, but almost so).

In this course I will ask you to write a brief report, each week, describing the results of your analysis. Please hand in these reports by the day I'll suggest, and then I will look at them before our next meeting. I hope that this will enable you to write better reports, which will help you with your Project Reports and with the Applied Statistics examination.

About 2 or 3 weeks into the course, I will introduce you to LaTeX for your report-writing, with examples of how to incorporate graphs into your reports.

Chapter 1

Reading in the data: simple linear regression

S-plus has very sophisticated reading-in methods and graphical output.

Here we simply read in some data, and follow this with linear regression and quadratic regression, demonstrating various special features of S-plus as we go.

Note: originally I wrote these notes using the symbol _ instead of $< -$ or $=$. This was a BAD habit: Iv should replace all occurrences of $= "by¡-"$ in the commands below. Note that $< -$ should be read as an arrow pointing from right to left; similarly $->$ is understood by S-plus as an arrow pointing from left to right.

R does not allow _, so if you want to start with good habits, you should use $< -$ or $=$ all the time.

S-plus differs from other statistical languages in being 'OBJECT-ORIENTED'. This takes a bit of getting used to, but there are advantages in being Object-Orientated.

First you should get the datafile 'weld' in place. I will send you all the data sets you need for this course: if you have a query about this, just email me.

Now start your S-plus session. Note that in the sequences of S-plus commands that follow, anything following a # is a comment only, so need not be typed by you.

```
# reading data from the keyboard
x =  c(7.82,8.00,7.95) #"c" means "combine"
x
# a slightly quicker way is to use scan( try help(scan))
x =  scan()
7.82 8.00 7.95
                        # NB blank line shows end of data
x
# To read a character vector
x =  scan(,"")
 A B C
 A C B

x
```

But mostly, for proper data analysis, we'll need to read data from a separate data file. Here are 3 methods, all doing things a bit differently from one another.

```
data=  scan("weld",list(x=0,y=0))
data
names(data)
```

```
x=  data$x ; y= data$y  # these data come from The Welding Institute,Abingdon
x;y # x=current(in amps) for weld,y= min.diam.of resulting weld
summary(x)
hist(x)
X =  matrix(scan("weld"),ncol=2,byrow=T) # T means "true"
X
weldtable =  read.table("weld",header=F) # F means "false"
weldtable
# for the present we make use only of x,y & do the linear regression of
# y on x, followed by that of y on x & x-squared.
plot(x,y)
teeny =  lm(y~x)
teeny      # This is an "object" in S-plus terminology.
summary(teeny)
anova(teeny)
names(teeny)
fv1 =  teeny$fitted.values
fv1
plot(x,fv1)
plot(x,y)
abline(teeny)
par(mfrow=c(3,2)) # to arrange that plots come in 3 rows, 2 columns
plot(teeny)
par(mfrow=c(1,1)) # to restore to 1 plot per screen
Y= cbind(y,fv1) ; matplot(x,Y,type="pl") #"cbind"is "columnbind"
res= teeny$residuals
plot(x,res)    # marked quadratic trend. See also
scatter.smooth(x,res)
xx=  x*x
teeny2 =  lm(y~x +xx ) # there's bound to be a slicker way to do this
teeny2
fv2 =  teeny2$fitted.values
plot(x,y)
lines(x,fv2,lty=2)
plot(teeny2, ask=T)
q()                      # to leave S-plus
```

Here is the data-set "weld", with x, y as first, second column respectively.

```
7.82 3.4
8.00 3.5
7.95 3.3
8.07 3.9
8.08 3.9
8.01 4.1
8.33 4.6
8.34 4.3
8.32 4.5
8.64 4.9
8.61 4.9
8.57 5.1
9.01 5.5
8.97 5.5
9.05 5.6
9.23 5.9
9.24 5.8
9.24 6.1
```

```
9.61 6.3
9.60 6.4
9.61 6.2
```

Chapter 2

A Fun example using data from Venables and Ripley

This worksheet is designed to show you some plotting and regression facilities.

```
library(MASS)
help(mammals)
mammals
attach(mammals) # to 'attach' the column headings, here 'body', 'brain' to the
# corresponding columns of the table. NB
# attach() WILL NOT OVER-WRITE EXISTING VARIABLES.
species =  row.names(mammals) ; species
x =  body ; y =  brain
postscript("simple.ps")  # to send the next graph to a named file
plot(x,y)                # this goes to "simple.ps")
dev.off()                # to turn off the current plotting device
plot(x,y)                # this will now come on your screen
identify(x,y,species) # find man, & the Asian elephant
                       # click middle button to quit
plot(log(x),log(y))
identify(log(x),log(y),species) # again, click middle button to quit
species.lm =  lm(y~x)  # linear regression, y "on" x
summary(species.lm)
par(mfrow=c(2,2))       # set up 2 columns & 2 rows for plots
plot(x,y) ; abline(species.lm) # plot line on scatter plot
r =  species.lm$residuals
f =  species.lm$fitted.values # to save typing
qqnorm(r) ; qqline(r)
 # this is a check on whether the residuals are NID(0,sigma^2)
 # they pretty obviously are NOT: can you see why ?
lx=  log(x) ; ly =  log(y)
species.llm =  lm(ly~lx) ; summary(species.llm)
plot(lx,ly) ; abline(species.llm)
rl =  species.llm$residuals ;fl =  species.llm$fitted.values
qqnorm(rl) ; qqline(rl)  # a better straight line plot
# click on "print graph" with left button to save graph to file
plot(f,r)  ; hist(r)
plot(fl,rl); hist(rl)    # further diagnostic checks
 # Which of the 2 regressions do you think is appropriate ?
mam.mat =  cbind(x,y,lx,ly)    # columnbind to form matrix
cor(mam.mat)                   # correlation matrix
```

8

```
round(cor(mam.mat),3)          # easier to read
par(mfrow=c(1,1))              # back to 1 graph per plot
pairs(mam.mat)

q()

ls
     now shows you where your .ps graph is.
lp ....
     enables you to print it out
rm ....
     enables you to remove it.(Saving later clutter)
```

Here is the data-set 'mammals', from Weisberg (1985, pp144-5). It is in the Venables and Ripley MASS library of data-sets.

	body	brain
Arctic fox	3.385	44.50
Owl monkey	0.480	15.50
Mountain beaver	1.350	8.10
Cow	465.000	423.00
Grey wolf	36.330	119.50
Goat	27.660	115.00
Roe deer	14.830	98.20
Guinea pig	1.040	5.50
Verbet	4.190	58.00
Chinchilla	0.425	6.40
Ground squirrel	0.101	4.00
Arctic ground squirrel	0.920	5.70
African giant pouched rat	1.000	6.60
Lesser short-tailed shrew	0.005	0.14
Star-nosed mole	0.060	1.00
Nine-banded armadillo	3.500	10.80
Tree hyrax	2.000	12.30
N.A. opossum	1.700	6.30
Asian elephant	2547.000	4603.00
Big brown bat	0.023	0.30
Donkey	187.100	419.00
Horse	521.000	655.00
European hedgehog	0.785	3.50
Patas monkey	10.000	115.00
Cat	3.300	25.60
Galago	0.200	5.00
Genet	1.410	17.50
Giraffe	529.000	680.00
Gorilla	207.000	406.00
Grey seal	85.000	325.00
Rock hyrax-a	0.750	12.30
Human	62.000	1320.00
African elephant	6654.000	5712.00
Water opossum	3.500	3.90
Rhesus monkey	6.800	179.00
Kangaroo	35.000	56.00
Yellow-bellied marmot	4.050	17.00
Golden hamster	0.120	1.00
Mouse	0.023	0.40

```
      Little brown bat     0.010     0.25
            Slow loris     1.400    12.50
                 Okapi   250.000   490.00
                Rabbit     2.500    12.10
                 Sheep    55.500   175.00
                Jaguar   100.000   157.00
            Chimpanzee    52.160   440.00
                Baboon    10.550   179.50
Desert hedgehog     0.550     2.40
Giant armadillo    60.000    81.00
  Rock hyrax-b      3.600    21.00
        Raccoon      4.288    39.20
            Rat      0.280     1.90
E. American mole     0.075     1.20
       Mole rat      0.122     3.00
     Musk shrew      0.048     0.33
            Pig    192.000   180.00
        Echidna      3.000    25.00
Brazilian tapir  160.000   169.00
         Tenrec      0.900     2.60
      Phalanger      1.620    11.40
     Tree shrew      0.104     2.50
        Red fox      4.235    50.40
```

Chapter 3

One-way analysis of variance and a qqnorm plot

This worksheet shows you how to construct a one-way analysis of variance and how to do a qqnorm-plot to assess normality of the residuals.
Here is the data in the file 'potash'.

```
7.62 8.00 7.93
8.14 8.15 7.87
7.76 7.73 7.74
7.17 7.57 7.80
7.46 7.68 7.21
```

The origin of these data is lost in the mists of time; they show the strength of bundles of cotton, for cotton grown under 5 different 'treatments', the treatments in question being amounts of potash, a fertiliser. The design of this simple agricultural experiment gives 3 'replicates' for each treatment level, making a total of 15 observations in all. We model the dependence of the strength on the level of potash.
This is what you should do.

```
Splus6
y =  scan("potash") ; y
# Now we read in the experimental design.
x =  scan()      # a slicker way is to use the "rep" function.
36 36 36
54 54 54
72 72 72
108 108 108
144 144 144 # x is treatment(in lbs per acre) & y is strength
            # blank line to show the end of the data
tapply(y,x,mean) # gives mean(y) for each level of x
plot(x,y)
regr =  lm(y~x) ; summary(regr)
potash =  factor(x) ; potash
plot(potash,y)
teeny =  aov(y~potash)
names(teeny)
coefficients(teeny) # can you understand these ?
summary(teeny) # for conventional anova table
multicomp(teeny) # for multiple comparisons, between treatments
help(qqnorm)
qqnorm(resid(teeny))
qqline(resid(teeny))
plot(fitted(teeny),resid(teeny))
```

It's fun to generate a random sample of size 100 from the t-distribution with 5 df, and find its qqnorm, qqline plots, to assess the systematic departure from a normal distribution. To see how to do this, try

```
help(qqnorm)
```

Chapter 4

A two-way analysis of variance

This worksheet shows you a two-way analysis of variance, first illustrating the function expand.grid() to set up the factor levels in a balanced design. The data are given below, and are in the file 'IrishItalian'.
Under the headline
" Irish and Italians are the 'sexists of Europe'" The Independent, October 8,1992, gave the following table.

```
The percentage having equal confidence in both sexes for various occupations

86 85 82 86   Denmark
75 83 75 79   Netherlands
77 70 70 68   France
61 70 66 75   UK
67 66 64 67   Belgium
56 65 69 67   Spain
52 67 65 63   Portugal
57 55 59 64   W. Germany
47 58 60 62   Luxembourg
52 56 61 58   Greece
54 56 55 59   Italy
43 51 50 61   Ireland
```

Here the columns are the occupations bus/train driver, surgeon, barrister, MP. Can you see that the French are out of line in column 1 ?
You will need to delete the text (ie row labels) from the data for the reading-in given below to make sense.

```
Splus6
p =  scan("IrishItalian") ; p
occ =  scan(," ")                    # now read in row & column labels
bus/train surgeon barrister MP
                              # remember blank line
country =  scan(, " ")
Den Neth Fra UK Bel Spa
Port W.Ger Lux Gre It Irl
                              # remember blank line
x =  expand.grid(occ,country) ; x
OCC =  x[,1] ; COUNTRY =  x[,2]  # to pick out the columns of x
OCC =  factor(OCC) ; COUNTRY=  factor(COUNTRY) # factor declaration(redundant)
```

We wish to fit the model $y_{ij} = \mu + \alpha_i + \beta_j + \epsilon_{ij}$, and then explore the consequences of the orthogonality of the design.
The default parametrisation for factor effects in S-plus is both hard to understand, and different

from the common default parametrisation used in R.

The default setting for S-plus is the Helmert parametrisation. This compares the 2nd and subsequent levels to the average of lower levels. Such a parametrisation is very rarely required in practice.

To make interpretation easier, we will impose the 'corner-point' or GLIM parametrisation at the outset, which for the above model will mean that

$$\alpha_1 = 0, \beta_1 = 0.$$

Thus every treatment level is being compared with the FIRST treatment level. (This is an asymmetric constraint, but it is easy to understand, and is the convention.)

```
options(contrasts=c("contr.treatment", "contr.poly"))
```

```
ex2 =  aov(p~COUNTRY+OCC)
ex2 ; summary(ex2)
names(ex2)
ex2$coefficients
model.tables(ex2, type="means", se=T) # for useful summary
lex2 =  lm(p~COUNTRY+OCC) ; lex2 ; summary(lex2)# for comparison
summary(lm(p~COUNTRY),cor=F)
summary(lm(p~OCC), cor=F)  #what are the orthogonality consequences?
```

```
aov(p~ OCC + COUNTRY)     # What is this telling us?
tapply(p,COUNTRY,mean) ; tapply(p,OCC,mean) #compare these with coeff's.
#  Now we'll demonstrate some nice graphics.
boxplot(split(p,OCC))
boxplot(split(p,COUNTRY))
res =  ex2$residuals ; fv=  ex2$fitted.values
plot(fv,res)
design.first=  data.frame(OCC,COUNTRY,p) ;design.first
plot.design(design.first)
plot.design(design.first,fun=median)
hist(resid(ex2))
qqnorm(resid(ex2))   # should be approx straight line if errors normal
ls()      # opportunity to tidy up here,eg by
rm(ex2)
```

Note: it makes sense to stick with the GLIM parametrisation for all your analyses, so that you don't have to worry about it. You can arrange this by

```
.First =  function(){
options(contrasts=c("contr.treatment","contr.poly"))
}
```

so that from now on, every time you go into S-plus, you have the GLIM constraints for the parameter estimates for the factor levels.

Here is a 3-way design, taken from the MPhil Applied Statistics exam 2000, q3.

The Table below shows you the percentage of people with "excessive" alcohol consumption, classified by sex, age and year. Thus, for example, in 1996, 7% of women aged 65 and over had excessive alcohol consumption, that is, they consumed more than 14 units per week.

```
Health related behaviour: prevalence of alcohol consumption above 21/14 units
a week for men/women aged 18 and over, in England.
                  1986 1990 1992 1994 1996
men(above 21 units)
```

```
18-24                     39   37   38   36   42
25-44                     33   33   30   30   31
45-64                     24   26   24   27   27
65+                       13   14   15   17   18

women(above 14 units)
18-24                     19   18   19   20   22
25-44                     13   13   14   16   16
45-64                      8   10   12   13   14
65+                        4    5    5    8    7
```

Try the following analysis

```
p =  scan("drink.data") # p is thus a vector, with 40 elements
Sex =  scan(,"")
men women

Year =  scan(,"")
1986 1990 1992 1994 1996

Age =  scan(,"")
18-24 25-44 45-64 65+

x =  expand.grid(Year,Age,Sex) #order is crucial
cbind(x,p) # as a check
YEAR =  x[,1] ; AGE =  x[,2] ; SEX =  x[,3]
is.factor(YEAR) # as a check
design.first =  data.frame(YEAR,AGE,SEX,p)
plot.design(design.first) # for useful graphical summary
         # showing you what your prejudices suggest
first.lm =  lm(p~ (YEAR + AGE + SEX)^2)
summary(first.lm, cor=F) ; anova(first.lm)
# do we need all the pairwise interactions?
model.tables(aov(first.lm))
tapply(p,AGE,mean)
tapply(p,SEX,mean)
tapply(p,list(SEX,AGE),mean)
second.lm =  lm(p~YEAR + SEX*AGE)
interaction.plot(AGE,SEX,p) # what is this telling us?
plot(second.lm, ask=T) # for diagnostic plots
# Note, qqplot looks funny. This suggests that we may not be
# working in the correct scale. ?boxcox?
```

Chapter 5

Analysis of an unbalanced two-way design

These data are taken from The Independent on Sunday for October 6,1991. They show the prices of certain best-selling books in 5 countries in pounds sterling. The columns correspond to UK, Germany, France, US, Austria respectively. The new feature of this data-set is that there are some MISSING values (missing for reasons unknown). Thus in the 10 by 5 table below, we use NA to represent 'not available' for these missing entries.

We then use 'na.action...' to omit the missing data in our fitting, so that we will have an UNbalanced design. You will see that this fact has profound consequences: certain sets of parameters are NON-orthogonal as a result.

   Here is the data from

```
  bookprices


14.99  12.68   9.00  11.00  15.95 S.Hawking,"A brief history of time"
14.95  17.53  13.60  13.35  15.95 U.Eco,"Foucault's Pendulum"
12.95  14.01  11.60  11.60  13.60 J.Le Carre,"The Russia House"
14.95  12.00   8.45   NA     NA  J.Archer,"Kane & Abel"
12.95  15.90  15.10   NA    16.00 S.Rushdie,"The Satanic Verses"
12.95  13.40  12.10  11.00  13.60 J.Barnes"History of the world in ..."
17.95  30.01   NA    14.50  22.80 R.Ellman,"Oscar Wilde"
13.99   NA     NA    12.50  13.60 J.Updike,"Rabbit at Rest"
 9.95  10.50   NA     9.85   NA  P.Suskind,"Perfume"
 7.95   9.85   5.65   6.95   NA  M.Duras,"The Lover"
```

"Do books cost more abroad?" was the question raised by The Independent on Sunday.

```
Splus6
p =  scan("bookprices") ; p
au =  1:10 ; cou =  c("UK","Ger","Fra","US","Austria")
x =  expand.grid(cou,au) ; x # to check we've put au,cou the right way round
country =  x[,1] ;author =  x[,2]; author =  factor(author)
lmunb =  lm(p~ country + author,na.action=na.omit) ; summary(lmunb)
resid =  lmunb$residuals
resid                # note that this is a vector with less than 50 elements.
# Thus, plot(country,resid) would give us an error message
plot(country[!is.na(p)],resid)  # will deal with this particular difficulty
unbaov =  aov(p~ country + author,na.action=na.omit) ; summary(unbaov)
# Try lm(p~author +country,...)
# Try aov(p~author + country,...)
# Try lm(p~author,...)
# Discuss carefully the consequences of non-orthogonality of
```

```
# the parameter sets country, author for this problem.

# Was our model above on the correct scale? We try a log-transform.
lp =  log(p)
lmunblp =  lm(lp~ country+author,na.action=na.omit) ; summary(lmunblp)
qqnorm(resid(lmunb))
qqnorm(resid(lmunblp)) # which is best ?
```

Problems involving MONEY should be attacked with multiplicative rather than additive models : discuss this provocative remark.

```
library(MASS) ; boxcox(lmunb)
```

This confirms that $\lambda = 0$ is a good choice.
Ask your lecturer for an explanation of the Box-Cox transform.
Here is another dataset of similar format for you to analyse, taken from The Independent, November 21, 2001, with the headline
'Supermarkets to defy bar on cheap designer goods'.

```
How prices compare: prices given in UK pounds.
Item                      UK    Sweden  France  Germany  US
Levi 501 jeans            46.16  47.63  42.11   46.06    27.10
Dockers K1 khakis         58.00  54.08  47.22   46.20    32.22
Timberland women's boots  111.00 104.12 89.43   93.36    75.42
DieselKultar men's jeans  60.00  43.35  43.50   44.48    NA
Timberland cargo pants    53.33  48.58  43.54   58.66    31.70
Gap men's sweater         34.50  NA     26.93   27.26    28.76
Ralph Lauren polo shirt   49.99  42.04  36.41   40.26    32.48
H&M cardigan              19.99  17.31  18.17   15.28    NA
```

Chapter 6

Logistic regression for the binomial distribution

The data come from 'Modelling Binary Data', by D.Collett(1991). The compressive strength of an alloy fastener used in aircraft construction is studied. Ten pressure loads, increasing in units of 200psi from 2500 psi to 4300 psi, were used. Here

$n=$ number of fasteners tested at each load

$r=$ number of these which FAIL.

We assume that r_i is Binomial(n_i, π_i) for $i = 1, \ldots, 10$ and that these 10 random variables are independent. We model the dependence of π_i on $load_i$, using graphs where appropriate.

The model assumed below is

$$log(\pi_i/(1 - \pi_i)) = a + b \times load_i.$$

[This is the LOGIT link in the glm]

Note that S-plus does the regression here with $p = r/n$ as the 'y-variable', and n as 'weights'. This is rather disagreeable: GLIM and Genstat allow you to take r as the y-variable and then declare n appropriately as the binomial 'denominator'. See

```
help(glm)
```

for more friendly syntax.

We take this exercise as an opportunity to show the use of 'source()': this allows us to access commands from separate file.

Here is

```
        littleprog3

data <-  read.table("alloyfastener",header=T)
attach(data)  # BEWARE, this will not over-write variables already present.
p <-  r/n
plot(load,p)
ex3 <- glm(p~load,weights=n,family=binomial) # 'weights' are the sample sizes
```

The corresponding data, given at the end of this sheet, is in the file called
    alloyfastener

So, first set up the files 'littleprog3' and 'alloyfastener'.

```
Splus6
source("littleprog3")
```

```
data; names(data); summary(data)
plot(load,p,type="l")
ex3 ; summary(ex3)
names(ex3)
plot(ex3,ask=T)     # for diagnostic plots
# Now we'll see how to vary the link function. Previously we were using the
#default link, ie the logit(this is canonical for the binomial distribution)
ex3.l =  glm(p~load,family=binomial(link=logit),weights=n)
ex3.p =  glm(p~load,family=binomial(link=probit),weights=n)
ex3.cll =  glm(p~load,binomial(link=cloglog),weights=n)
summary(ex3.l)
summary(ex3.p)    # the probit link
summary(ex3.cll) # the complementary loglog link
```

Observe that the ratio $\hat{a}/\hat{b}$ is about the same for the 3 link functions. This is a special case of a general phenomenon (proof unknown to me). Which link function gives the best fit, ie the smallest deviance ? In general the logit and probit will fit almost equally well. The logistic distribution with mean 0, scale parameter 1, has variance $(\pi^2/3)$. So we compare with the normal distribution, mean 0, variance $(\pi^2/3)$.

```
x =  1:200 ; x =  x/200 ;y =  log(x/(1-x));z =  qnorm(x);z =  z*(pi/(3^.5))
      # (3^.5) is perhaps better computed as sqrt(3)
plot(x,y)
plot(x,z)
Y =  cbind(y,z) ; matplot(x,Y,type="pl")
q()
```

Here is the dataset "alloyfastener".

```
load  n   r
25    50  10
27    70  17
29   100  30
31    60  21
33    40  18
35    85  43
37    90  54
39    50  33
41    80  60
43    65  51
```

Chapter 7

Logistic regression and safety in space

Swan and Rigby (GLIM Newsletter no 24,1995) discuss the data below, using binomial logistic regression. To quote from Swan and Rigby

'In 1986 the NASA space shuttle Challenger exploded shortly after it was launched. After an investigation it was concluded that this had occurred as a result of an 'O' ring failure. 'O' rings are toroidal seals and in the shuttles six are used to prevent hot gases escaping and coming into contact with fuel supply lines.

Data had been collected from 23 previous shuttle flights on the ambient temperature at the launch and the number of 'O' rings, out of the six, that were damaged during the launch. NASA staff analysed the data to assess whether the risk of 'O' ring failure damage was related to temperature, but it is reported that they excluded the zero responses (ie, none of the rings damaged) because they believed them to be uninformative. The resulting analysis led them to believe that the risk of damage was independent of the ambient temperature at the launch. The temperatures for the 23 previous launches ranged from 53 to 81 degreees Fahrenheit while the Challenger launch temperature was 31 degrees Fahrenheit (ie, -0.6 degrees Centigrade).'

Calculate $pfail = nfail/six$, where

```
six =  rep(6,times=23),
```

for the data below, so that $pfail$ is the proportion that fail at each of the 23 previous shuttle flights. Let $temp$ be the corresponding temperature. Comment on the results of

```
Splus6
glm(pfail~ temp,binomial,weights=six)
```

and plot suitable graphs to illustrate your results.
Are any points particularly 'influential' in the logistic regression?
How is your model affected if you omit all points for which $nfail = 0$?
Do you have any comments on the design of this experiment?

```
 The data (read this by read.table("...",header=T)) follow.
    nfail temp
       2     53
       1     57
       1     58
       1     63
       0     66
       0     67
       0     67
       0     67
       0     68
```

```
0      69
0      70
0      70
1      70
1      70
0      72
0      73
0      75
2      75
0      76
0      76
0      78
0      79
0      81
```

Chapter 8

Binomial and Poisson regression: the Missing Persons data

Some rather gruesome data published on March 8, 1994 in The Independent under the headline
" Thousands of people who disappear without trace "
are analysed below,

```
Splus
s=  scan()
33 63 157
38 108 159
                              # nb, blank line
r=  scan()
3271  7256    5065
2486  8877    3520
                              # nb, blank line
# r= number reported missing during the year ending March 1993
# s= number still missing by the end of that year
# figures from Metropolitan police.
sex = c(rep("males", times=3), rep("females", times=3))
 sex
  is.factor(sex)
  sex = factor(sex)
# Now we set up the factor age, in years
 age = rep(c("13&under", "14-18", "19&over"),times=2)
 age
 is.factor(age)
  age = factor(age)
  tapply(s/r, list(age, sex), sum) # for which the result is
              females       males
    13&under 0.01528560 0.01008866
        14-18 0.01216627 0.00868247
      19&over 0.04517045 0.03099704
# showing that those most likely to stay missing
# are females, in the 19&over age category
 interaction.plot(age, sex, s/r)
bin.add =  glm(s/r ~ sex+age,family=binomial,weights=r)
summary(bin.add)
```

What is this telling us?
Now the Binomial with large n, small p, is nearly the Poisson with mean (np). So we also try
Poisson regression, using the appropriate "offset".

```
 l =  log(r)
```

```
Poisson.add =  glm(s~sex + age+offset(l),family=poisson)
summary(Poisson.add)
```

Describe and interpret these results, explaining the similarities.

Chapter 9

Analysis of a 4×2 contingency table

In this analysis of a 4×2 contingency table we show three ways to get the same result
Check from the log-likelihood function WHY these give the same result.
The data were obtained by Professor I.M.Goodyer, as part of a study on 11-year-old children, to investigate the relationship between 'deviant behaviour' (no/yes) and several other variables. The results for the variable 'emotionality' are shown below (emo=1 meaning low emotionality,... emo=4 meaning high emotionality).

```
      behaviour = no     yes
emo=1             51       3
emo=2             69      11
emo=3             28      22
emo=4              7      13
```

```
a =  c(51,69,28,7)  ; b =  c(3,11,22,13)
indepB =  glm(cbind(a,b)~ 1 ,binomial) # nb a ONE
summary(indepB); x =  cbind(a,b); chisq.test(x)
y =  c(a,b)
RR =  c(1,2,3,4,1,2,3,4)
CC =  c(1,1,1,1,2,2,2,2)
RR =  factor(RR) ; CC =  factor(CC)
indepP =  glm(y~ RR + CC,poisson); summary(indepP)
fisher.test(x) # is an EXACT test for independence of rows and columns,
# based on the hypergeometric distribution. It will not work here, because
# the sum of frequencies is too large, but you might like to try it on a
# smaller table,eg
a =  c(28,3) ; b =  c(22,13)
x =  cbind(a,b); fisher.test(x); chisq.test(x)
```

Note added May 2006.
You can also make use of the

```
dhyper()
```

function to construct an exact test of the null hypothesis of no 3-way interaction, in a $2 \times 2 \times 2$ table.
For example, consider the following 3-way table.

```
First layer            Second layer               Total
  X1= 3   1  | k1=4       X2= 2    3  | k2 =5         u=5     4 | k1+k2=9
        2   4  |   6             7   1  |    8           9     5 |   14
```

```
------------------        -------------------       -----------------
  m1=5 n1=5|    10        m2 =9  n2=4|    13        m1+m2 n1+n2|   23
```

The first 2×2 table has cross-ratio 6; the second has cross-ratio $2/21$, and as these are quite different, it certainly *looks* as though there is a significant three-way interaction. How should we compute the relevant p-value?

Changing notation for convenience, the relevant null distribution will be

$$P(n_{111}|(n_{+jk}),(n_{i+k}),(n_{ij+})) \propto 1/\Pi_{ijk}n_{ijk}! \; .$$

(Looking at the relevant exponential family shows you that you seek the distribution of n_{111} conditional on all the pairwise marginal totals.)

The problem is, as Bartlett noted in 1935, there is no simple form for the constant of proportionality. Essentially, we seek the conditional distribution $P(X1 = x|X1 + X2 = u)$.

For the current example, X1 can have possible values $0, 1, 2, 3, 4$. Discuss the application of the following R commands.

```
x = 0:4
q = dhyper(x,m1,n1,k1)* dhyper(u-x,m2,n2,k2)
const = sum(q) ; q = q/const ; q
```

We seek the probability of getting an observation of $X1 = 3$ or larger, ie we seek

```
q[3] + q[4]   # which I reckon is .087, please check!
```

We end with a rather simple example.

Consider the following 6×2 table of 'Petrol Availability', at petrol filling stations, published in The Independent, September 18, 2000:

```
          total number, selling petrol
BP         1500       900
Shell      1100       651
Sainsbury  223        176
Tesco      310        309
Jet        547        316
Esso       1600       850
```

I would expect that the χ^2 statistic is very large: what do you find?

See http://bmj.bmjjournals.com for online version of the paper referred to below).

'Cannabis intoxication and fatal road crashes in France: population based case-control study' by Laumon et al, British Medical Journal, 2 December 2005, studied 10748 drivers, with known drug and alcohol concentrations, who were involved in fatal crashes in France from October 2001 to September 2003. The 'cases' were defined as the 6766 drivers considered at fault in their crash; the 'controls' were 3006 drivers selected from the 3982 other drivers. The authors studied many attributes of the drivers, but you will see below just one small summary table. Here 'High' refers to Blood concentration of Δ^9 tetrahydrocannabinol ≥ 1 ng/ml, ie testing positive for cannabis, and 'Low' refers to the rest, ie < 1 ng/ml. Show that the odds-ratio for the 2×2 table given below is 3.32, find the corresponding confidence interval, and interpret your findings.

```
         High Low
Cases    596  6170
Controls 85   2921
```

This is taken from their Table 2 'Drivers' responsibility associated with drugs and alcohol'.

For comparison, the corresponding odds-ratio when 'High' is taken as 'Blood concentration of alcohol ≥ 0.5 g/l, and 'Low' is low alcohol, has the value 15.5, with corresponding confidence interval $(12.4, 19.5)$.

Chapter 10

Loglinear modelling with the Poisson distribution

This exercise shows you use of the Poisson 'family' or distribution function for loglinear modelling. Also it shows you use of the 'sink()' directive in Splus.

As usual, typing the commands below is a trivial exercise: what YOU must do is to make sure you understand the purpose of the commands, and that you can interpret the output.

First. The total number of reported new cases per month of AIDS in the UK up to November 1985 are listed below(data from A.Sykes 1986). We model the way in which y, the number of cases depends on i, the month number.

```
Splus6
y =  scan()
0 0 3 0 1 1 1 2 2 4 2 8 0 3 4 5 2 2 2 5
4 3 15 12 7 14 6 10 14 8 19 10 7 20 10 19
                # nb, blank line
i=  1:36
plot(i,y)
aids.reg =  glm(y~i,family=poisson) # NB IT HAS TO BE lower case p,
# even though Poisson was a famous French mathematician.
aids.reg          # The default link is in use here, ie the log-link
summary(aids.reg) # thus model is log E(y(i))=a + b*i
sink("temp")      # to store all results from now on
# in the file called "temp". The use of
# sink(), will then switch the output back to the screen.
aids.reg          # no output to screen here
summary(aids.reg) # no output to screen here
sink()            # to return output to screen
names(aids.reg)
q()
more temp         # to read results of "sink"
```

Second. Accidents for traffic into Cambridge, 1978-1981
 How does the number of accidents depend on traffic volume, Road and Time of day?

```
Splus6
y =  scan()
11 9 4 4 20 4
```

```
v =  scan()
2206 3276 1999 1399 2276 1417
rd  = c(1,1,1,2,2,2) #rd=1 for Trumpington Rd,rd=2 for Mill Rd
ToD =  c(1,2,3,1,2,3) #ToD =1,2,3 for 7-9.30 am,9.30am-3pm,3-6.30 pm resp'ly
```

NB. We call time "ToD" rather than "t" because use of "t" will provoke S-plus into "warning messages": t() being an Splus function, (matrix transpose, in fact).
Now y=no. of accidents, v= est.of traffic volume

```
 lv=  log(v) ; RD= factor(rd) ;ToD= factor(ToD)
accidents=  glm(y~RD +ToD + lv,family=poisson)
accidents
summary(accidents)
anova(accidents,test="Chi")  # for analysis of deviance table
# NB The order in this table may not be sensible.
drop.road = update(accidents,.~. -RD) ; summary(drop.road)
drop.ToD  = update(accidents,.~.-ToD) ;summary(drop.ToD)
new= c(1,2,1, 1,2,1);NEW= factor(new); acc.new= glm(y~RD+NEW+lv,poisson)
# Can you see the point of the factor "NEW"?
summary(acc.new)
q()
```

Lastly, here are two other examples crying out for Poisson regression.
Under the heading "Great Britain's Medal Decline" on 14 August, 2001, The Independent gives the following table

```
1983 Helsinki  7 =(2,2,3)
1987 Rome      8 =(1,3,4)
1991 Tokyo     7 =(2,2,3)
1993 Stuttgart 10=(3,3,4)
1995 Gothenburg 5=(1,3,1)
1997 Athens    6=(0,5,1)
1999 Seville   7=(1,4,2)
2001 Edmonton  2=(1,0,1)
```

The figures in brackets refer, respectively, to the numbers of gold, silver and bronze medals. Thus at Edmonton the total of 2 medals came from 1 gold and 1 bronze.
I disagree with the Independent's headline, do you?
What is your predicted number of medals for Great Britain in 2003?
Under the heading, 'Police tell death-chase inquiry of 'red mist' risk, on 5 November, 2001, The Independent gives the following table of Car Chase Deaths, Fatalities from pursuits:

```
1997-98  9
1998-99 17
1999-00 22
2000-01 25
April 2001- 5 November,2001 (7 months) 26
```

Discuss these data, with the appropriate Poisson regression, using (12,12,12,12,7) as the corresponding 'time at risk'.

Chapter 11

Loglinear modelling and Simpson's 'paradox'

Here we use the Poisson distribution for log-linear modelling of a two-way contingency table, and compare the results with the corresponding binomial formulation.
We construct a fictitious 3-way table to illustrate Simpson's paradox.
The Daily Telegraph (28/10/88) under the headline 'Executives seen as DrinkDrive threat' presented the following data from breath-test operations at Royal Ascot and at Henley Regatta (these being UK sporting functions renowned for alcohol intake as well as racehorses and rowing respectively).
Are you more likely to be arrested, if tested, at R.A. than at H.R.?
You see below that a total of $(24 + 2210)$ persons were breathalysed at R.A., and similarly a total of $(5 + 680)$ were tested at H.R. Of these, 24 tested positive at R.A., and 5 at H.R.

```
Splus6
 r =  scan()
24 2210 # Royal Ascot
5  680  # Henley Regatta

Row =  c(1,1,2,2) ; Col =  c(1,2,1,2);ROW =  factor(Row);COL =  factor(Col)
# nb we do not use "row" because it is already an Splus function.
# Col= 1 for ARRESTED,Col= 2 for NOT arrested
   saturated =  glm(r~ ROW*COL,family=poisson)
independence =  glm(r~ ROW+COL,family=poisson)
summary(saturated)   # this shows us that the ROW.COL term can be dropped
summary(independence)
```

This looks like a very elaborate way of doing a chi-squared test for a 2×2 table!
Here is another way of answering the same question.
(Observe certain exact agreements and explain them.)

```
a =  c(24,2210) ; b=  c(5,680) ; tot =  a+b; p =  a/tot
Row =  c(1,2) ; ROW=  factor(Row)
 sat =  glm(p~ROW,family=binomial,weights=tot)
indep =  glm(p~  1 ,family=binomial,weights=tot)
```

Of course, 2 independent Poisson rv's conditioned on their sum gives a binomial.
Note: you don't have to specify "weights" for binomial regression. Here's a nicer way (it doesn't involve computation of p=a/tot).

```
satt =  glm(cbind(a,b)~ ROW,binomial)        # this is equivalent to "sat"
indepp =  glm(cbind(a,b)~ 1 ,binomial)    # this is equivalent to "indep"
summary(satt) ; summary(indepp) # this way we don't need to specify WEIGHTS
```

Now a little piece of fantasy,with a serious educational purpose (of course).
It can be very misleading to 'collapse' a 3-way table, say
Ascot/Henley × Arrest/NonArrest × Men/Women
over one of the dimensions, say Men/Women. For example (pure invention) suppose the above
2 × 2 table was in fact 'collapsed' from
24=23,1 2210=2,2208
5 =3,2 680=340,340
the first number of each pair being the number of men, the second being the number of women.
We analyse the resulting 2 × 2 × 2 table.

```
r =  scan()
23  2  1  2208
3  340 2  340

Row= c(1,1,1,1,2,2,2,2);Col= c(1,2,1,2,1,2,1,2);sex= c(1,1,2,2,1,1,2,2)
ROW=  factor(Row) ; COL= factor(Col) ;SEX=  factor(sex)
sat =  glm(r~ROW*COL*SEX ,poisson)  ; summary(sat)
q()
```

Of course we have invented an example with a strong 3-way interaction. You should consider the
following two questions. How does the arrest rate for men vary between Ascot and Henley? How
does the arrest rate for women vary between Ascot and Henley?
This is an example of 'Yule's paradox'.(An historical note: G.U.Yule was a Fellow of St John's
college, Cambridge at the start of the 20th century.)
It must be admitted that most people outside Cambridge call it Simpson's paradox (Simpson wrote
about 'his' paradox in 1951, whereas Yule had written about it about 50 years earlier.)

Chapter 12

How to plot contours of a log-likelihood function

Here we plot the contours for the Poisson regression log-likelihood surface corresponding to the 'Aids' dataset used above. We use nested loops to compute the log-likelihood surface. You will also see how to define and use a function in S-plus.

```
Splus6
y =  scan("aids")           # same data as before.
i =  1:36 ; ii=  i-mean(i) # to make the surface a better shape
aids.reg =  glm(y~ii,poisson)
summary(aids.reg)
```

shows us that $log(\mu_i) = a + b * ii$ with $\hat{a} = 1.51(se = .09)$ and $\hat{b} = .08(se = .008)$

```
loglik =  function(a,b){
  loglik =  - sum(exp(a+b*ii)) + a*t1 +b*t2
  loglik
}
```

Here t1 and t2 are the sufficient statistics, thus

```
t1 =  sum(y) ; t2 =  sum(ii*y)
```

We plot the loglikelihood surface for $a = 1.4, \ldots, 1.6$ and $b = .06, \ldots, .08$

```
a =  0:20 ; a =   a*(.01) + 1.4
b =  0:20 ; b =   b*(.001) + 0.06
zz =  1: (21*21) ; z =  matrix(zz,21,21) # to set up z as a matrix
for (x in 1:21){
 for (y in 1:21){
   z[x,y] =  loglik(a[x],b[y])
 }
}
round(z,1)
contour(a,b,z)
```

This first shot is disappointing, to say the least! But there is the suggestion of an elliptical contour (showing the negative correlation between the estimates of a and b). Use

```
  contour(a,b,z,v)
```

with the vector v suitably set, and/or rescale the a, b ranges, to get a more convincing demonstration that the loglikelihood function has approximately elliptical contours.

Chapter 13

A 4-way contingency table

Here we analyse the 4-way table given in 'Fitting Graphical Models to Multi-way Contingency Tables in GLIM' by P.M.E.Altham, GLIM Newsletter no.21, p4, 1992.

The S-plus (and R) versions uses glm( ), and generates a nested design for the 4 factors, each of which has two levels. We also show how to start with the 'saturated' model, and drop down, via stepAIC, by 'throwing away' unnecessary interactions between these 4 factors.

```
Splus
n =  scan()    # to read in the observed frequencies
89  2  4  1
 8  4  3  1
70  6  2  0
 1  0  1  1
           # blank line
```

The corresponding values of sex, dep, beh, anx could be read in from the table below,

```
sex dep beh anx
 1   1   1   1
 1   1   1   2
 1   1   2   1
 1   1   2   2
 1   2   1   1
 1   2   1   2
 1   2   2   1
 1   2   2   2
 2   1   1   1
 2   1   1   2
 2   1   2   1
 2   1   2   2
 2   2   1   1
 2   2   1   2
 2   2   2   1
 2   2   2   2
```

but this would clearly be a rather laboured way to set up the design.

Here the explanation of the 4 variables is:

sex=1,2 for girls, boys
dep=1,2 for depression=no, yes
beh=1,2 for behavioural symptoms absent, present
anx=1,2 for anxiety symptoms absent, present.

These data were collected by Prof I.M.Goodyer on 193 adolescents.

Since we have a NESTED design here, it is most efficient to use this fact in setting up the factor

levels. (Genstat has a similar command, "generate"). BE CAREFUL about the order of the arguments.

```
fnames =  list(anx=1:2,beh=1:2,dep=1:2,sex=1:2)
psych = expand.grid(fnames)
psych
attach(psych)          # warning: this will NOT over-write previous names
sex ; dep ; beh ; anx # to check that our labelling is correct.
sex = factor(sex); dep = factor(dep); anx= factor(anx); beh = factor(beh)
gl.sat =  glm(n~anx*beh*dep*sex,poisson)
   # This does not converge, because of zeros in cell-frequencies.
summary(gl.sat, cor=F)
gl.three =  glm(n~(anx+beh+dep+sex)^3,poisson)
   # this fits all 3-way interactions. Which can we drop?
gl.two =  glm(n~(anx+beh+dep+sex)^2,poisson)
   # this fits all 2-way interactions. Which can we drop?
library(MASS)
help(stepAIC)        # gives us a new idea
# ask your lecturer to explain the AIC (Akaike Information Criterion).
gl.try =  stepAIC(gl.three)
   # What does this end up with ?
gl.try =  stepAIC(gl.two)
   # What does this end up with ?
```

Of course, using stepAIC() may or may not end up with a GRAPHICAL model. We have to check for ourselves whether the final model is graphical. But we can tell at a glance from the correlation matrix of the parameter estimates whether the final model is DECOMPOSABLE. A decomposable model is necessarily graphical.
See Whittaker(1990) for explanation of 'graphical' and 'decomposable'.
The function stepAIC() tends to leave in some interactions that we can see are unnecessary. My final model for this dataset is

```
gl.last =  glm(n~(anx+beh+sex)*dep,poisson)
```

This final model, which has deviance 8.476 on 8 df, has a simple interpretation in terms of CONDITIONAL INDEPENDENCE. Here is a quick illustration of the corresponding conditional independence graph

```
                    anx
                   /
                  /
    sex-----------dep
                  \
                   \
                    beh
```
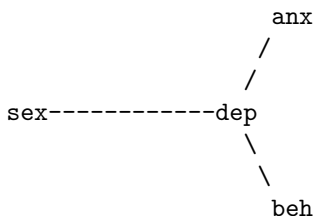
This shows that the 3 variables sex, anx, beh are only 'connected' through the variable dep: thus for example there is no direct link from anx to beh.
Perhaps it is more helpful to write this graphical model as
$P(anx, beh|dep, sex) = P(anx|dep, sex)P(beh|dep, sex) = P(anx|dep)P(beh|dep)$.
Observe that under this model, the 3 2-way tables (sex,dep), (dep,anx) and (dep,beh) form sufficient statistics, hence are a valid summary of the original data. Use

```
tapply(n,list(sex,dep),sum)
```

and so forth, to obtain these 3 2-way tables, and hence interpret the data.
Observe: Glim syntax would suggest you could do

```
 gl.another =  glm(n~(sex*dep*anx*beh)-*sex:anx,poisson)
```

to remove the (sex,anx) link. This doesn't work in S-plus, presumably because no-one's yet realised it could be useful. Note

```
glm(n~(sex*dep*anx*beh)-sex*anx,poisson)
```

means something different(and rather horrible).

Experiment with these model formulae for yourselves. None of the above uses the superb S-plus graphics tools. Can you remedy this ?

Chapter 14

A balanced incomplete block design

Here we look at a very small-scale example of a balanced incomplete block design, using fictitious data, reminding you about factorial experiments.

We have say 4 cycle riders, to race on a track which can only take 3 riders at a time. We use just 4 races to compare the 4 riders, and we have invented data which shows a very strong following wind in the first race.

Check that with the design given below: $k = 3, t = 4, \lambda = 2, n = 12, b = 4$ (in standard notation for balanced incomplete blocks).

```
Splus
#first read in the treatment(rider) number, a.
 a =  scan()
2 3 4
1 3 4
1 2 4
1 2 3

# Now read in block(race) number bl, (or use rep( ) function)
bl =  scan()
1 1 1
2 2 2
3 3 3
4 4 4

a =  factor(a) ;bl =  factor(bl)
# Now read in the times of each rider.
y =  scan()
1.1 3.3 2.2
20 45 67
23 34 56
24 77 88

bib.y =  aov(y~ a+bl); summary(bib.y)
bib.y =  aov(y~bl+a) ; summary(bib.y)  # do you see the non-orthogonality?
lm.y  =  lm(y~a+bl)  ; summary(lm.y)
rlm.y =  lm(y~a)     ; summary(rlm.y)
# Check that you can understand all these 4 analyses.
design.bib =  data.frame(a,bl,y)

plot.design(design.bib)
q()
```

Are the 4 riders significantly different from one another? What is the standard error of the difference between any 2 riders?

Chapter 15

A fractional replication of a 5-factor experiment

This shows you how to analyse a fractional replication of a 5-factor experiment.
(See McCullagh and Nelder (1989) p366)
This is a (1/2) replicate of a 2^5 experiment, with each of the 16 treatment combinations replicated 3 times, to allow for a study of the variability at each factor combination.
The 5 factors are denoted by B, C, D, E, O (see below for an explanation) and the defining contrast for this design is BCDE.
Thus, for example, the interaction BD is aliased with the interaction CE.
You should be able to show that a good model for the response h is

```
 h ~ (B+C)*O + E
Splus
 h =  scan("Taguchidata")# Here are the readings of h, in inches
7.78 7.78 7.81
8.15 8.18 7.88
7.50 7.56 7.50
7.59 7.56 7.75
7.94 8.00 7.88
7.69 8.09 8.06
7.56 7.62 7.44
7.56 7.81 7.69
7.50 7.25 7.12
7.88 7.88 7.44
7.50 7.56 7.50
7.63 7.75 7.56
7.32 7.44 7.44
7.56 7.69 7.62
7.18 7.18 7.25
7.81 7.50 7.59
```

Now we must set up the design of the experiment. It is possible, but non-trivial(you try?) to coax fac.design() to set up the whole design for us. But here is a more straight-forward way (but long-winded) CAN YOU FIND A BETTER WAY?

```
  B=  rep(c(1,1,1,2,2,2),times=8) ; B
  C=  rep(c(rep(1,times=6),rep(2,times=6)),times=4) ; C
  D=  rep(c(rep(1,times=12),rep(2,times=12)),times=2) ; D
  O=  c(rep(1,times=24),rep(2,times=24));O
# the factor E we have to read in separately: thus
E =  scan()
```

```
1 1 1 2 2 2 2 2 2 1 1 1 2 2 2 1 1 1 1 1 1 2 2 2
1 1 1 2 2 2 2 2 2 1 1 1 2 2 2 1 1 1 1 1 1 2 2 2

# An alternative method is
x =  c(1,1,1) ; y =  c(2,2,2)
E =  c(x,y,y,x,y,x,x,y,x,y,y,x,y,x,x,y)
z =  B+C+D+E-4 ; z    # to check that BCDE is a defining contrast
cbind(B,C,D,E,O,h)   # to see the whole design & the data
B =  factor(B);C =  factor(C);D =  factor(D);E =  factor(E);O =  factor(O)
first =  lm(h~B+C+D+E+O) ; summary(first)
first =  aov(h~B+C+D+E+O) ; summary(first)
second =  lm(h~(B+C)*O+E) ; summary(second) #to check McC & N,p368
second =  aov(h~(B+C)*O+E) ; summary(second)
# Now try putting in more interactions
first =  lm(h~(B+C+D+O+E)^2)                    # "dumps":do you see why?
first =  lm(h~(B+C+D+O+E)^2,singular.ok=T)    # now you'll see why
# how can we find out from"first" which interaction is aliased with which?
lm.try =  step.lm(first)
design.fr =  data.frame(B,C,D,E,O,h)
plot.design(design.fr)
```

So far we have only modelled the MEAN response. McCullagh and Nelder, p368, show also how to analyse the DISPERSION. You could consider this for a future exercise.

Explanation of the data:
$h=$"free height of leaf springs", which may depend on any or all of
B=furnace temperature, C=heating time, D=transfer time, E=hold-down time, O=quench oil temperature.
The object of the experiment is to find the factor combination that gives h as close as possible to 8 inches, with as small a variability as possible.

Chapter 16

Analysis of matched case-control data

These data, and the Glim4 analysis, are presented by Tony Swan in the Glim4 Manual(1993), p520. Here the dataset has been slightly edited, so that missing values (coded as 9 or 99 in Swan's dataset) are given as NA: this is vastly more convenient for S-plus purposes.

The data come from the Los Angeles case-control study of endometrial cancer. (The endometrium is the lining of the uterus.) The purpose of the study was to assess the sizes and relative importance of a number of risk factors. Each case of endometrial cancer (necessarily female) was matched by age with each of 4 female controls.

The dataset is described and analysed in chapter 7 of Breslow and Day(1980).

The variables are:

   cact = case status (0 for control and 1 for case)
   set = identifier for each case-control set, set=1,...,63
   age = age in years at last birthday
   gall = gall bladder disease(0 for no, 1 for yes)
   hyper = high blood pressure(0 for no, 1 for yes)
   obese =obesity (0 for no, 1 for yes)
   est =history of any oestrogen use (0 for no, 1 for yes)
   edos =dosage of oestrogen, which has possible levels

0= none or single doses less than .1 mg/day,

1= .1-.299 mg/day,

2= .3-.625 mg/day,

3= .626+ mg/day, or

NA= unknown.

edur =duration of oestrogen use in months

(single doses are coded 0, and NA = not known)

   odrg =other non-oestrogen drug

   We assume that for set i, individual j, the contribution to the likelihood

i) from a case is $\mu_{ij} exp(-\mu_{ij})$

ii) from a control is $exp(-\mu_{ij})$,

where $log(\mu_{ij}) = \mu + \lambda_i + \beta^T x_{ij}$

and x_{ij} is the vector of covariate values for this individual. The parameter β is the parameter of interest, the parameter λ (corresponding to the 'sets') is a nuisance parameter. This is termed a 'conditional logistic regression model' .

The correct likelihood is achieved by declaring

cact

as the 'y-variate', with a Poisson distribution. (You see that this is a trick, because we KNOW that cact CAN only take the values 0 and 1.)

Observe that one consequence of the presence of the term λ in all the models given below is that we will always find (as we should certainly expect) that for each set, the sum of the observed values of cact (which of course is 1) agrees exactly with the sum of the corresponding fitted values.

Note added April 2007: in R, as an alternative to using the Poisson trick illustrated below, you can use

```
library(survival}
```

with the function clogit() (ie conditional logit) taking strata(set).

```
Splus
casecontrol =  read.table("casecontrol",header=T)
attach(casecontrol)
summary(casecontrol)
idur =  c(-.5,0.9,11.9,47.9,95.9,99.9)
dur =  cut(edur,idur)  #to give dur with same grouping as Swan's definition
gall =  factor(gall) ; hyper =  factor(hyper)
odrg =  factor(odrg) ; dose =  factor(edos)
set =  factor(set)
glm.first =  glm(cact~ set + (gall + hyper + dose + dur+odrg),poisson,
   na.action=na.omit)
```

You will see that this gives deviance$= 121.06, df = 223$.
Swan points out that the true df should be 200; "All the individuals in case-control sets where there are no cases"(ie for which one of the covariate values is NA) "must be weighted out to get the correct degrees of freedom although it is not essential because the estimates and model comparisons are unaffected."
Because he does this weighting out, his df here are 200.
In fact, because we have $0, 1$ data, we cannot use the deviance itself (compared with its df) as a measure of the fit of the model: we only use $changes$ in deviance (compared with the corresponding changes in df) to assess the significance of parameters.

```
glm.first  # gives us the parameter estimates, but of course all those
           # corresponding to "set" are of no interest
summary(glm.first,correlation=F) #lots of unwanted information: can
           # we improve the presentation here to eliminate "set" ?
# What about a model that includes 2nd-order interactions?
glm.first =  glm(cact~set+ (gall+hyper+dose+dur+odrg)^2,poisson,
                 na.action=na.omit)
```

You'll see this gives "lack of convergence" : we have too many parameters for the amount of data available. So, having seen what Swan does, I try a model with just one interaction, and then see if this model can be simplified.

```
glm.first =  glm(cact~set +(gall+hyper+dose+dur+odrg+dose:dur),poisson,
                 na.action=na.omit)
# We see if we can throw out unnecessary terms, but we remember that "set"
# must always be retained in the model.
glm.second =  step.glm(glm.first,list(upper=~.,lower=~set))
```

It's not clear to me that there is a dose-duration interaction.
A careful derivation of the likelihood for this problem is given by Collett, "Modelling Binary Data"(1991)pp 346-348.
Here is the dataset for this problem.

cact	set	age	gall	hyper	obese	est	edos	edur	odrg
1	1	74	0	0	1	1	3	96	1
0	1	75	0	0	9	0	0	0	0
0	1	74	0	0	9	0	0	0	0
0	1	74	0	0	9	0	0	0	0
0	1	75	0	0	1	1	1	48	1

```
1    2   67   0   0   0   1   3   96   1
0    2   67   0   0   0   1   3    5   0
0    2   67   0   1   1   0   0    0   1
0    2   67   0   0   0   1   2   53   0
0    2   68   0   0   0   1   2   45   1
1    3   76   0   1   1   1   1    9   1
0    3   76   0   1   1   1   2   96   1
0    3   76   0   1   0   1   1    3   1
0    3   76   0   1   1   1   2   15   1
0    3   77   0   0   0   1   1   36   1
1    4   71   0   0   9   1  NA   96   0
0    4   70   1   0   0   1   2    7   1
0    4   70   0   0   0   1   0    0   1
0    4   71   0   1   1   1   2    7   1
0    4   70   0   0   1   1   2   27   1
1    5   69   1   0   1   1   2   36   1
0    5   69   0   1   0   1   1   96   1
0    5   69   0   0   1   1   2    1   1
0    5   69   0   0   0   1   0    0   1
0    5   68   0   0   9   0   0    0   0
1    6   70   0   1   1   1   2   71   1
0    6   71   0   0   0   0   0    0   0
0    6   71   0   1   1   1   3    5   1
0    6   70   0   0   1   0   0    0   0
0    6   71   0   0   9   0   0    0   0
1    7   65   1   0   0   1   1   96   1
0    7   65   0   0   9   0   0    0   0
0    7   64   0   0   0   1   3   91   1
0    7   64   0   0   0   1   2   96   1
0    7   65   0   0   1   1   2   60   0
1    8   68   1   1   1   1   1   36   1
0    8   68   0   1   9   0   0    0   1
0    8   68   0   0   1   0   0    0   1
0    8   68   1   0   9   1   0    0   0
0    8   68   0   0   9   1   1    1   1
1    9   61   0   0   9   0   0    0   1
0    9   61   0   0   1   0   0    0   1
0    9   61   0   0   0   1   1   24   1
0    9   61   0   0   9   0   0    0   1
0    9   60   1   0   0   0   0    0   0
1   10   64   0   0   1   1   1   54   1
0   10   64   0   0   9   0   0    0   0
0   10   65   0   1   9   1   3    2   1
0   10   64   0   1   1   1   3   10   1
0   10   65   0   0   9   0   0    0   0
1   11   68   1   0   1   1   3   96   1
0   11   69   0   0   9   0   0    0   0
0   11   69   0   0   1   0   0    0   1
0   11   69   1   1   1   1   0    0   1
0   11   69   1   0   1   1   1   35   0
1   12   74   0   0   0   1   2   96   1
0   12   74   0   1   0   1   3    4   1
0   12   73   0   1   0   1   2   11   1
0   12   74   0   0   1   1   1    6   1
0   12   74   0   0   1   1   1   12   0
1   13   67   1   0   1   1   0    0   1
0   13   68   0   1   0   1   0    0   1
```

0	13	68	0	1	1	1	3	65	1
0	13	68	0	0	9	0	0	0	0
0	13	68	0	0	1	1	2	96	1
1	14	62	1	0	0	1	1	NA	1
0	14	62	1	0	0	0	0	0	0
0	14	63	0	0	1	0	0	0	0
0	14	63	0	0	9	0	0	0	0
0	14	63	0	0	1	1	2	NA	0
1	15	71	1	0	1	1	2	59	1
0	15	70	0	1	1	0	0	0	1
0	15	71	0	0	1	1	NA	NA	1
0	15	71	0	1	1	0	0	0	1
0	15	71	0	1	1	1	2	84	1
1	16	83	0	1	1	1	3	96	1
0	16	82	0	0	1	0	0	0	0
0	16	82	0	1	0	1	3	4	1
0	16	82	0	1	0	0	0	0	0
0	16	82	0	0	0	0	0	0	1
1	17	70	0	0	1	0	0	0	1
0	17	70	1	1	1	1	2	55	1
0	17	70	0	1	1	1	2	14	1
0	17	70	0	1	1	1	1	39	1
0	17	70	0	1	1	0	0	0	1
1	18	74	0	0	0	1	0	0	1
0	18	75	1	1	0	1	2	6	1
0	18	74	0	0	1	0	0	0	1
0	18	74	0	1	0	1	2	46	0
0	18	75	0	0	9	0	0	0	0
1	19	70	0	0	9	1	0	0	1
0	19	70	0	1	0	1	1	96	1
0	19	70	0	0	9	0	0	0	0
0	19	70	0	0	9	0	0	0	1
0	19	70	0	0	9	0	0	0	0
1	20	66	0	1	1	1	3	48	1
0	20	66	0	0	9	1	1	96	1
0	20	66	0	0	9	0	0	0	1
0	20	66	0	0	1	0	0	0	0
0	20	66	0	1	1	1	1	12	1
1	21	77	0	0	1	1	3	4	1
0	21	77	1	1	1	1	0	0	1
0	21	77	0	1	0	1	2	24	1
0	21	77	0	0	1	0	0	0	0
0	21	78	0	1	1	1	2	9	1
1	22	66	0	1	0	1	3	29	1
0	22	67	0	1	0	0	0	0	1
0	22	66	0	0	1	0	0	0	1
0	22	67	0	0	1	0	0	0	0
0	22	69	0	1	1	1	2	10	1
1	23	71	0	1	1	1	1	96	0
0	23	72	0	0	9	0	0	0	0
0	23	72	0	0	0	0	0	0	1
0	23	71	0	0	9	0	0	0	0
0	23	71	0	1	1	0	0	0	1
1	24	80	0	0	0	1	2	NA	1
0	24	79	0	0	9	0	0	0	0
0	24	79	0	0	0	0	0	0	0
0	24	79	0	0	1	0	0	0	0

```
0   24   80   0   0   0   0   0    0   0
1   25   64   0   0   1   1   2   NA   1
0   25   64   0   0   0   1   0    0   1
0   25   63   0   0   1   1   1   60   1
0   25   64   0   1   0   1   1    6   1
0   25   66   0   1   1   1   1   NA   1
1   26   63   0   0   0   1   1   60   1
0   26   63   0   1   0   1   1   96   1
0   26   65   0   0   0   1   1   25   0
0   26   65   0   0   0   0   0    0   1
0   26   64   0   0   0   1   1   96   1
1   27   72   1   0   1   0   0    0   1
0   27   72   0   1   9   0   0    0   0
0   27   72   0   1   1   0   0    0   1
0   27   72   0   1   0   1   1   48   1
0   27   72   0   1   1   1   0    0   1
1   28   57   0   0   0   1   3   12   0
0   28   57   0   1   1   1   0    0   1
0   28   58   0   0   1   1   1   36   1
0   28   57   0   0   0   1   1   36   0
0   28   57   0   0   0   1   0    0   0
1   29   74   1   0   1   0   0    0   1
0   29   74   0   0   1   0   0    0   1
0   29   73   0   0   1   1   2    2   1
0   29   75   0   0   1   0   0    0   1
0   29   75   0   0   9   0   0    0   0
1   30   62   0   1   1   1   2    6   1
0   30   62   0   0   1   1   2   37   1
0   30   62   0   0   1   1   2   63   1
0   30   63   0   0   9   0   0    0   0
0   30   61   1   1   1   1   3   96   1
1   31   73   0   1   1   1   1    4   1
0   31   72   0   0   0   1   2   90   1
0   31   73   0   0   0   1   3    5   1
0   31   73   0   1   0   1   1   15   1
0   31   73   0   1   0   0   0    0   0
1   32   71   0   1   1   1   1   NA   1
0   32   71   0   0   9   0   0    0   0
0   32   71   0   0   0   0   0    0   1
0   32   71   0   0   0   0   0    0   1
0   32   71   0   1   0   1  NA   NA   1
1   33   64   0   1   1   0   0    0   1
0   33   65   0   0   1   1   3   96   1
0   33   64   0   0   1   1   3   96   1
0   33   64   0   0   1   1   2   36   1
0   33   64   0   0   1   1   3   96   0
1   34   63   0   0   0   1  NA   96   1
0   34   64   0   0   1   0   0    0   1
0   34   62   0   0   1   0   0    0   1
0   34   64   1   0   0   1   1   18   0
0   34   64   0   1   1   1   3   NA   1
1   35   79   1   1   1   1   1   96   1
0   35   78   1   1   1   1   1   96   1
0   35   79   0   0   1   0   0    0   1
0   35   79   0   1   0   1   0    0   1
0   35   78   0   0   1   1   1   24   1
1   36   80   0   0   1   1   1   15   1
```

0	36	81	0	1	1	0	0	0	1
0	36	81	0	1	0	1	1	18	1
0	36	80	0	0	1	1	2	74	1
0	36	80	0	1	1	0	0	0	1
1	37	82	0	1	1	1	2	6	1
0	37	82	0	0	1	0	0	0	1
0	37	81	0	1	9	0	0	0	1
0	37	81	0	1	1	1	1	12	1
0	37	82	0	1	1	1	2	13	1
1	38	71	0	1	0	1	NA	84	1
0	38	71	0	1	1	0	0	0	1
0	38	71	1	0	1	0	0	0	1
0	38	71	1	0	1	1	1	96	1
0	38	71	0	0	0	1	1	30	1
1	39	83	0	1	1	1	3	14	1
0	39	83	0	1	1	0	0	0	1
0	39	83	0	0	0	0	0	0	1
0	39	83	0	1	1	1	2	16	1
0	39	83	0	0	0	0	0	0	1
1	40	61	0	1	0	1	3	96	1
0	40	60	0	0	0	0	0	0	1
0	40	61	0	0	0	1	1	24	1
0	40	62	0	0	1	0	0	0	1
0	40	61	0	0	0	1	0	0	1
1	41	71	0	0	0	1	1	96	1
0	41	71	0	0	1	0	0	0	0
0	41	71	0	1	1	0	0	0	0
0	41	70	0	0	0	0	0	0	0
0	41	71	0	1	1	1	1	3	1
1	42	69	0	1	1	1	2	40	1
0	42	69	1	0	1	0	0	0	1
0	42	70	0	1	0	1	0	0	0
0	42	70	0	1	0	1	1	32	1
0	42	70	0	0	1	1	NA	NA	1
1	43	77	0	0	1	1	3	73	1
0	43	76	0	1	0	1	0	0	1
0	43	76	0	1	1	1	0	0	1
0	43	77	1	1	1	1	0	0	1
0	43	77	0	1	0	0	0	0	1
1	44	64	0	0	1	1	1	37	0
0	44	64	0	0	1	1	3	6	0
0	44	63	1	0	1	0	0	0	0
0	44	63	0	1	0	1	NA	NA	1
0	44	63	0	1	1	0	0	0	1
1	45	79	1	0	0	0	0	0	0
0	45	82	0	0	1	1	1	NA	1
0	45	78	0	0	0	0	0	0	0
0	45	80	0	0	1	0	0	0	0
0	45	81	0	0	0	0	0	0	0
1	46	72	0	0	0	1	0	0	1
0	46	72	0	0	1	1	2	57	1
0	46	73	0	0	9	0	0	0	0
0	46	73	1	1	0	1	2	96	1
0	46	73	0	0	0	0	0	0	1
1	47	82	1	1	1	1	3	96	1
0	47	81	0	0	9	0	0	0	0
0	47	81	0	0	1	0	0	0	0

0	47	81	0	0	1	1	0	0	1
0	47	81	0	1	1	0	0	0	1
1	48	73	0	1	1	1	2	60	1
0	48	74	0	0	1	1	1	1	1
0	48	75	0	0	0	0	0	0	1
0	48	75	0	1	1	1	1	96	1
0	48	74	0	0	0	0	0	0	0
1	49	69	0	0	9	1	NA	NA	1
0	49	68	0	0	0	0	0	0	1
0	49	68	0	0	1	1	2	48	1
0	49	68	0	0	0	1	1	96	0
0	49	70	0	0	0	0	0	0	0
1	50	79	0	1	1	1	1	67	1
0	50	79	0	1	1	0	0	0	1
0	50	79	0	1	1	0	0	0	1
0	50	78	1	0	1	1	1	NA	1
0	50	79	0	0	1	0	0	0	1
1	51	72	0	0	1	1	3	60	0
0	51	71	0	0	0	1	0	0	1
0	51	72	0	0	0	0	0	0	1
0	51	72	0	1	1	1	3	96	1
0	51	71	0	1	1	1	3	12	1
1	52	72	0	1	1	1	1	27	1
0	52	72	0	1	1	1	1	3	1
0	52	71	0	0	9	0	0	0	0
0	52	72	0	1	1	0	0	0	1
0	52	72	0	1	1	0	0	0	1
1	53	65	0	1	1	1	2	16	1
0	53	67	0	0	0	0	0	0	0
0	53	67	0	0	9	0	0	0	0
0	53	66	0	0	1	0	0	0	1
0	53	66	0	0	0	1	2	3	0
1	54	67	0	1	1	1	2	96	1
0	54	66	0	0	1	1	2	56	1
0	54	66	0	0	1	0	0	0	0
0	54	67	0	0	1	1	1	NA	1
0	54	67	0	0	1	1	2	34	1
1	55	64	1	0	1	1	3	96	1
0	55	63	0	0	1	0	0	0	1
0	55	64	0	0	1	1	1	4	1
0	55	63	0	0	1	0	0	0	1
0	55	65	0	0	9	0	0	0	0
1	56	62	0	0	9	1	2	36	0
0	56	63	0	0	1	0	0	0	0
0	56	62	0	0	0	0	0	0	1
0	56	62	0	0	9	1	3	NA	0
0	56	62	0	0	9	0	0	0	0
1	57	83	1	1	9	0	0	0	1
0	57	83	1	0	9	0	0	0	0
0	57	82	0	0	0	1	2	6	1
0	57	83	0	0	9	0	0	0	1
0	57	83	1	0	9	0	0	0	1
1	58	81	0	0	1	1	0	0	1
0	58	79	0	0	9	0	0	0	0
0	58	80	0	0	1	0	0	0	1
0	58	82	0	0	1	0	0	0	1
0	58	80	0	0	0	0	0	0	0

```
1   59   67   0   0   1   1   2   96   1
0   59   66   0   0   1   1   2   40   1
0   59   68   0   0   9   0   0    0   1
0   59   65   0   0   0   0   0    0   1
0   59   66   0   1   1   1   1   96   1
1   60   73   1   1   1   1   1   NA   1
0   60   72   0   0   1   1   1   12   1
0   60   71   0   0   1   1   1   96   1
0   60   73   1   0   0   1   2   96   1
0   60   72   0   0   1   0   0    0   1
1   61   67   1   0   0   1   3   96   1
0   61   67   1   0   1   1   2   96   1
0   61   68   0   0   1   0   0    0   0
0   61   67   0   0   1   0   0    0   0
0   61   67   0   0   1   0   0    0   1
1   62   74   0   1   1   1   2    9   1
0   62   75   0   0   0   0   0    0   1
0   62   75   0   0   9   0   0    0   0
0   62   75   1   1   0   0   0    0   1
0   62   75   0   0   0   1   2   41   1
1   63   68   1   0   1   1   3   18   1
0   63   69   0   0   1   1   2   96   1
0   63   70   0   0   9   0   0    0   0
0   63   69   0   1   1   1   2   92   1
0   63   69   0   1   0   1   3   59   1
```

Chapter 17

The binomial distribution, and generalisations: sex-distribution data

You see below part of a very remarkable data-set collected by Geissler in nineteenth century Saxony. These data are also discussed by Lindsey (1995), chapter 6.

There were 6115 families with exactly 12 children, and the 2 columns below give i, the number of males in the family, and n, the number of families with the corresponding number of males. Thus for example, there were 24 families consisting of 1 male and 11 females. If the number of males in the family followed a binomial distribution, parameters $12, p$ say, then we would find that

$$E(n_i) = n \binom{12}{i} p^i (1-p)^{(12-i)}$$

for $i = 0, \ldots, 12$, where $n = 6115$.

Hence

$$log(E(n_i)) = constant - log(i!(12-i)!) + i \times log(p/(1-p)).$$

This is a model that we can easily fit by declaring n as a Poisson variable, with the log-link, and then doing regression on i, with $-log(i!(12-i)!)$ as our offset. (Once again we are using the relation between the multinomial and independent Poisson variables.)

Try doing this. You will see that

i) because we have included the 'constant' term, (ie the intercept), we will automatically find that sum of observed = sum of 'fitted' frequencies = 6115.

ii) the binomial is in fact a poor fit (the deviance is 97.01 with 11 df) because we have clear over-dispersion relative to the binomial: apparently the sexes of the individual children in a family are not independent random variables, but are (slightly) positively related. This may in fact just be a consequence of the random variation of the parameter p over the population.

Here is the dataset 'family'.

```
i    n
0    3
1   24
2  104
3  286
4  670
5 1033
6 1343
7 1112
8  829
9  478
```

```
10 181
11  45
12   7

data =  read.table("family",header=T) ; attach(data)
lg =  -lgamma(i+1) - lgamma(12-i+1) # useful to have the log-gamma function
first.glm =  glm(n ~ i + offset(lg),poisson)
summary(first.glm)
```

The following simple generalisation of the binomial, suggested by Altham in 1978, was found computationally almost intractable at that time.

$$E(n_i) = n \ c\binom{12}{i}p^i(1-p)^{(12-i)}\theta^{i(12-i)}.$$

This is because there is no closed-form analytic expression for the normalising constant c. But following the work of Lindsey(1995) we see that the distribution can easily be fitted by Poisson regression, with a log-link function. Once again, including the intercept term in the fit will take care of the problem of the normalising constant c.

```
sec.glm =  glm(n ~ i + i^2 + offset(lg),poisson)
summary(sec.glm)
```

This 'multiplicative' binomial fits very well: it has deviance 14.469 on 10 df.
Since $log(\theta)$ is estimated as $0.02615(se = .00275)$, we see that θ is estimated as 1.0265. (It may seem a little odd to find $\theta > 1$ when we clearly have OVER-dispersion relative to the binomial, but this is not impossible, as Gianfranco Lovison pointed out. See
Lovison, G. (1998)' An alternative representation of Altham's multiplicative-binomial distribution.'
Statistics and Probability Letters 36:415-420.
In a sense the 'multiplicative' binomial is a discrete version of the normal distribution, since
i) it is of 2-parameter exponential family form, and
ii) fitting it by maximum likelihood ensures that both the sample mean and variance exactly match the fitted mean and variance.
This distribution is used in 'Numbers of CNV's and false negative rates will be underestimated if we do not account for the dependence between repeated experiments' by Lynch, Marioni and Tavare, in The American Journal of Human Genetics, 2007, 81:418-420. (CNV's are 'genomic copy-number variations')
Returning to the Geisser data from families of size 12, here are the observed frequencies n, and then the fitted frequencies, first for the binomial model (fv1) and then for the 'multiplicative' binomial (fv2).

```
 i     n     fv1     fv2
 0     3     0.9     2.3
 1    24    12.1    22.6
 2   104    71.8   104.8
 3   286   258.5   310.9
 4   670   628.1   655.7
 5  1033  1085.2  1036.1
 6  1343  1367.3  1257.9
 7  1112  1265.6  1182.3
 8   829   854.2   853.8
 9   478   410.0   462.0
10   181   132.8   177.8
11    45    26.1    43.7
12     7     2.3     5.2
```

We can plot the curves of the corresponding fitted frequencies as follows, with the observed frequencies n superimposed as points on this graph: see Figure 17.1.

```
matplot(i, cbind(fv1,fv2), type="l", col=c(1,2), ylab="fitted frequencies")
legend("topleft",
 legend=c("binomial", "multiplicative binomial"), lty=c(1,2), col=c(1,2))
points(i,n)
```
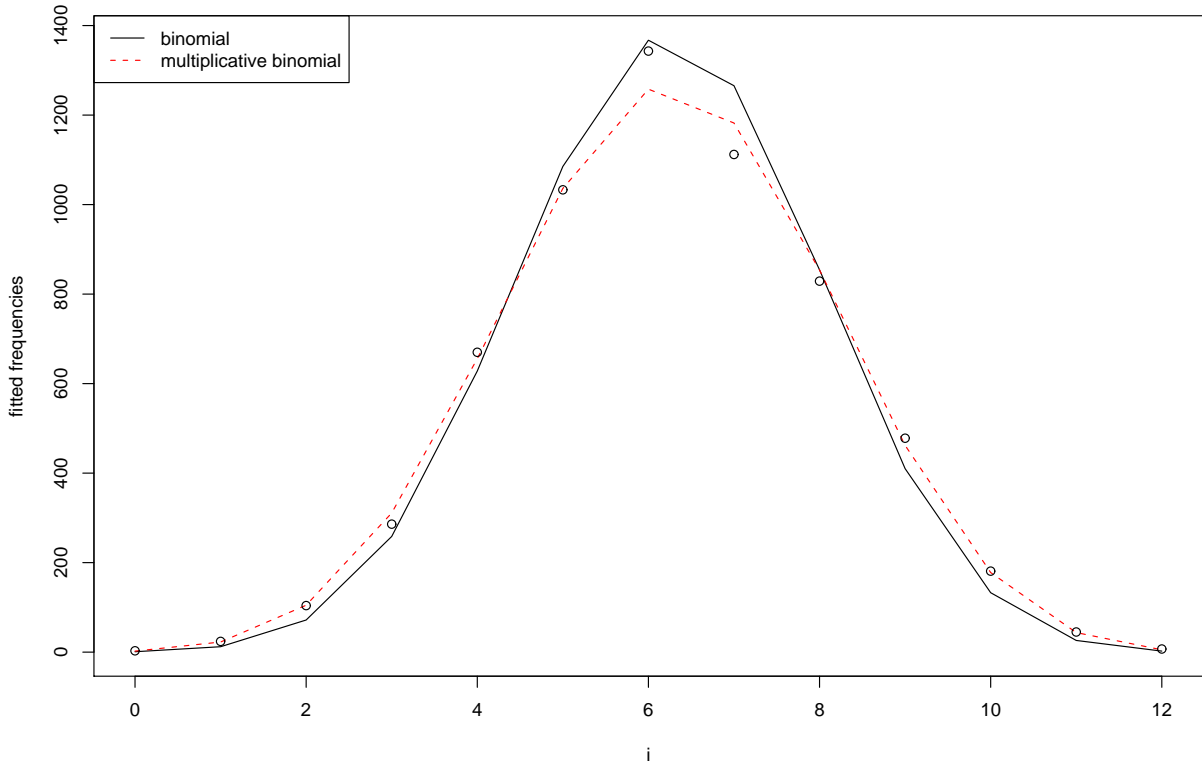

Figure 17.1: The binomial and multiplicative binomial fitted frequencies for size 12 families

Note that for many years geneticists have used the 'coefficient of in-breeding F' as a measure of departure from the binomial distribution: in the context of genetics F measures the departure from Hardy-Weinberg mating. F is defined by

$$F = 1 - N_m/E_m$$

where N_m is the observed number of mixed sex families, and
E_m is the expected number of mixed sex families, under the null hypothesis of a binomial distribution. Hence assuming that we are dealing with families of size k (here $k = 12$) we see that

$$F = \frac{(n_0 - e_0) + (n_k - e_k)}{(n - e_0 - e_k)};$$

here $n_0 = 3, e_0 = 0.9, n_k = 7, e_k = 2.3$ and $n = 6115$. This gives $F = 0.00111$. You can check that by definition, $-1 \le F \le 1$, with $F = 0$ only for $N_m = E_m$, ie an exact fit to the binomial.

The beta-binomial can also be shown to fit these data very well, but we cannot readily use glm() for to find the parameter estimates in this case. However, it is very quick to fit the beta-binomial by simply matching the first and second moments, thus.
Suppose Y, the number of males in a family of size 12, has the following distribution. Conditional on p, Y is Binomial, parameters $(12, p)$ and p is Beta, parameters (α, β). Then we say that Y is

beta-binomial, parameters $(12, \alpha, \beta)$. You can then show that $E(Y) = 12\alpha/(\alpha+\beta) = 12p'$ say, and $var(Y) = 12p'(1-p') + 12(12-1)\rho p'(1-p')$ where $\rho = 1/(1+\alpha+\beta)$, and ρ is thus the correlation between the sexes of any two siblings of a family.

I estimate that $E(Y) = 6.23058$ and $var(Y) = 3.48973$. This corresponds to a beta-binomial with parameters $(12, 34.13, 31.61)$, so the corresponding beta density is quite peaked. (ρ comes out to be 0.0150: with the beta-binomial ρ has to be positive.)

Note added June 2006. Shmueli et al, Applied Statistics 2005, pp 127-142 presented 'A useful distribution for fitting discrete data: revival of the Conway-Maxwell-Poisson distribution'. This included a generalization of the binomial distribution, allowing for under-dispersion or over-dispersion. Here we can very simply fit this particular generalization (the CMP-binomial distribution, using the definition of Shmueli et al) by the command

```
next.glm = glm(n ~ i + lg, poisson)
```

This has deviance 13.365 on 10 df, and the estimated coefficient of lg is .8433, (se= .01625). We thus show that with a_i as $\binom{12}{i}$ the model

$$E(n_i) = n \ k(a_i)^\nu p^i (1-p)^{(12-i)}$$

is a good fit, with ν estimated as $0.8433(se = .01625)$.

That's all very well, but it isn't easy to see a natural interpretation of the parameter ν, apart from noting that $\nu = 1$ corresponds to the ordinary binomial distribution.

Postscript

1. You may think that *under*-dispersion is very unlikely to occur in nature. But curiously the 1991 paper 'Modelling sub-binomial variation in the frequency of sex combinations in litters of pigs' by R.H.Brooks, W.H.James and E.Gray, Biometrics **47**, 403–417 contains several datasets exhibiting sub-binomial variation, with possible explanations and possible models.

2. Altham and Hankin (2010) have generalized the multiplicative binomial to the multiplicative multinomial distribution, with a corresponding R package and several datasets. This paper was published in 2012.

Chapter 18

Overdispersion and the Poisson, fitting the negative binomial

J.Hinde(1996) in the *GLIM Newsletter* no. 26 , "Macros for Fitting Overdispersion Models", discusses the data-set given below. This data-set was originally published by D.P.Gaver and I.G.O'Muircheartaigh (1987) "Robust empirical Bayes analysis of event rates", Technometrics 29,1-15. We quote from J.Hinde. "Gaver and O'Muircheartaigh present data on the number of failures s_i and the period of operation t_i (measured in 1,000's of hours) from 10 pumps from a nuclear plant. The pumps were operated in two different modes; four being run continuously (C) and the others kept on standby (S) and only run intermittently."
In the analysis that follows, we illustrate the use of the MASS library negative binomial fitting.
First we read in $(s_i), (t_i)$. The latter we refer to in S-Plus as time to avoid confusion with the function $t()$. Similarly we read in the mode as *Mode* in order not to confuse S-Plus.

```
s =  scan()
5  1  5 14  3 19  1  1  4 22
             # blank line
time =  scan()
94.320  15.720  62.880 125.760   5.240  31.440   1.048   1.048   2.096 10.480

Mode =  scan(," ")
C S C C S C S S S S

Mode =  factor(Mode)
plot(time,s,xlab="time",ylab="s",type= "n")
text(time,s,c("C","S")[Mode])
# Now try fitting a Poisson, using log(time) as an offset.
  glm.P =  glm(s~ Mode + offset(log(time)),poisson)
  summary(glm.P,cor=F)
Deviance Residuals:
      Min        1Q     Median        3Q       Max
 -4.576166 -1.210474 -0.3688074 1.019757 5.202038
Coefficients:
               Value Std. Error     t value
(Intercept) -1.989462  0.1522853 -13.064043
       Mode  1.881986  0.2327862    8.084611
(Dispersion Parameter for Poisson family taken to be 1 )
    Null Deviance: 124.5384 on 9 degrees of freedom
Residual Deviance: 71.43254 on 8 degrees of freedom
Number of Fisher Scoring Iterations: 4
Correlation of Coefficients:
     (Intercept)
Mode -0.6541852
```

This simple model shows a significant effect of 'Mode', but has a huge overdispersion relative to the Poisson. Thus the standard errors for the parameter estimates which are given above are unrealistically low, and hence the corresponding t-values are unrealistically high.
A 'quick fix' for this over-dispersion problem is to use

```
summary(glm.P, dispersion=0,cor=F)
```

This has the effect of assuming that $E(s_i) = \mu_i$, $var(s_i) = \phi\mu_i$ where now we estimate the scale parameter ϕ from $deviance/df$. This gives the revised summary below:

```
Call: glm(formula = s ~ Mode + offset(log(time)), family = poisson)
Deviance Residuals:
      Min        1Q     Median       3Q       Max
 -4.576166 -1.210474 -0.3688074 1.019757 5.202038
Coefficients:
               Value Std. Error   t value
(Intercept) -1.989462  0.5073913 -3.920962
      Mode   1.881986  0.7756080  2.426466
(Dispersion Parameter for Poisson family taken to be 11.1012 )
    Null Deviance: 124.5384 on 9 degrees of freedom
Residual Deviance: 71.43254 on 8 degrees of freedom
Number of Fisher Scoring Iterations: 4
```

But a better approach is to use the function provided in the Venables-Ripley library to model the over-dispersion relative to the Poisson by the negative binomial distribution: observe that this is a generalization of the Poisson.

```
library(MASS)
 negbin.1 =  glm.nb(s~ Mode + offset(log(time)))
summary(negbin.1)
Call: glm.nb(formula =s~Mode+offset(log(time)),init.theta = 1.29788210331571,
link = log)
Deviance Residuals:
      Min         1Q      Median        3Q        Max
 -1.961858 -0.7909716 -0.3287076 0.4257677 1.427814
Coefficients:
               Value Std. Error   t value
(Intercept) -1.603410  0.4610432 -3.477786
        Mode  1.672888  0.6292875  2.658384
(Dispersion Parameter for Negative Binomial family taken to be * )
    Null Deviance: 15.80313 on 9 degrees of freedom
Residual Deviance: 9.738841 on 8 degrees of freedom
Number of Fisher Scoring Iterations: 1
Correlation of Coefficients:
     (Intercept)
Mode -0.7326432
             Theta:  1.29788
         Std. Err.:  0.62693

 2 x log-likelihood:   195.44261


# Now let's compare the observed values with the fitted values.
fv =  negbin.1$fitted.values
round(cbind(s,fv,time),3)

    s      fv     time
 1  5 18.978  94.320
 2  1 16.851  15.720
 3  5 12.652  62.880
```

```
 4 14 25.304 125.760
 5  3  5.617   5.240
 6 19  6.326  31.440
 7  1  1.123   1.048
 8  1  1.123   1.048
 9  4  2.247   2.096
10 22 11.234  10.480
```

You will observe from the above table of s and the corresponding fitted values fv that at first sight the fit looks terrible, although the deviance of 9.7388 with 8 df is actually telling us that the negative binomial model fits very well. This apparent contradiction is due to the fact that we are used to thinking of the 'fit' as being determined by

$$\Sigma(o-e)^2/e$$

(using an obvious notation for o, e). Of course it is well known that if $\Sigma o = \Sigma e$, then this is approximately

$$2\Sigma o \ log(o/e)$$

but of course this latter quantity is only the deviance appropriate for testing the fit in the case of a Poisson (or multinomial) model.

We now derive an approximation for the deviance for testing the fit of a negative binomial model, valid for any link function, in the special case of known θ. We use the notation of Venables and Ripley (1997, p242) to explore these properties.

Suppose that the observations are (y_i), and that these are independent negative binomial random variables, and that y_i has frequency function $f(y_i|\theta, \mu_i) =$

$$\frac{\Gamma(\theta+y_i)\mu_i^{y_i}\theta^\theta}{\Gamma(\theta)y_i!(\mu_i+\theta)^{\theta+y_i}},$$

for $y_i = 0, 1, 2, ...$

Then $E(y_i) = \mu_i$, and $var(y_i) = \mu_i + \mu_i^2/\theta$, and $\theta \uparrow \infty$ will give a Poisson distribution.

Assume for simplicity that θ is known, and that we wish to fit the model

$$H_0 : g(\mu_i) = \beta^T x_i,$$

for $i = 1, \ldots, n$ where $g(.)$ is a given link function and (x_i) are known covariates.

We will derive an approximation for the deviance for testing H_0 against the more general hypothesis $H_1 : \mu_i$ any positive numbers, $i = 1, \ldots, n$.

The loglikelihood is a constant $+L$, where

$$L = \Sigma y_i log(\mu_i/(\mu_i+\theta)) - \theta\Sigma log(\mu_i+\theta).$$

It is easily seen that this is maximized under H_1 by $\mu_i = y_i$ for all i. Suppose L is maximized under H_0 by $\beta = \hat{\beta}$, and let (e_i) be the corresponding 'fitted values' under H_0, so that

$$g(e_i) = \hat{\beta}^T x_i$$

for all i.

Then it is easily checked that the deviance for testing H_0 against H_1 is say D, where

$$D/2 = \Sigma y_i log(y_i/e_i) - \Sigma(y_i+\theta)log((y_i+\theta)/(e_i+\theta)).$$

Now put $y_i = e_i + \Delta_i$, for $i = 1, ..., n$ and assume that (Δ_i) is 'small'. Then as usual,

$$2\Sigma y_i log(y_i/e_i) = 2\Sigma(e_i+\Delta_i)log(1+\Delta_i/e_i)$$

which may be shown to be approximately

$$2\Sigma\Delta_i + \Sigma\Delta_i^2/e_i.$$

Similarly, since $y_i + \theta = e_i + \theta + \Delta_i$ for $i = 1, ..., n$, we can apply the same argument to

$$2\Sigma(y_i + \theta)log((y_i + \theta)/(e_i + \theta))$$

to show that this is approximately

$$2\Sigma\Delta_i + \Sigma\Delta_i^2/(e_i + \theta).$$

Hence the deviance D may be written as

$$D \approx \Sigma\Delta_i^2/e_i - \Sigma\Delta_i^2/(e_i + \theta).$$

Note that in general, Σy_i is different from Σe_i.

For this proof we do not need to make the assumption $\Sigma\Delta_i = 0$.

Hence we see that the role of the parameter θ in assessing the 'fit' of the negative binomial model is to 'moderate' the usual Pearson χ^2 (or to make us 'feel better' about the terrible fit as apparently shown by the Pearson χ^2). We may rewrite this approximation as

$$D \approx \Sigma(y_i - e_i)^2/(e_i + e_i^2/\theta)$$

which of course is just what we should expect, given our assumption about $var(y_i)$.

For the dataset of the example, where the deviance is 9.738841, the approximation for the deviance (with θ set to 1.29788) is 8.537757. Observe that we are effectively assuming that for each observation y, $Y|E$ is Poisson, mean μE, and θE is gamma(θ). This latter distribution is very nearly negative exponential (hence with a heavy right 'tail') for the given θ.

Observe that the approximation for D given above reveals another interesting feature: the approximation for D (which of course is $-2\times$ the loglikelihood function), for *fixed* values of (y_i, e_i) is clearly a monotone increasing function of θ, and so will be MINIMISED by taking θ as *zero*. This is somewhat paradoxical, and may be related to the difficulty of maximizing the true log-likelihood with respect to θ in a slightly different setup: see

'Log-Linear Modeling with the Negative Multinomial Distribution' (1997) by L.A.Waller and D.Zelterman, Biometrics 53,971-982 for a more general discussion of this difficulty.

McCullagh and Nelder(1989, p374) discuss the estimation of θ (which in their notation is k).

Here's how we could plot the profile loglikelihood function for θ.

For each $\theta = 2, 2.1, 2.2, \ldots, 3$ say, use

```
first.glm =  glm(s~Mode + offset(log(time)),family =neg.bin(\theta))
fv =  first.glm$fitted.values
```

and now use these fitted values (which are themselves functions of θ) to compute in turn the values of

$$\Sigma[log\Gamma(\theta + y_i) + y_i log(\mu_i) + \theta log(\theta) - log(\Gamma(\theta)) - (\theta + y_i)log(\mu_i + \theta)]$$

where we replace μ_i by the fitted values fv in the sum above.

Assessing the fit of the negative binomial with unknown θ is analogous to assessing the fit of the normal distribution with unknown scale parameter. Venables and Ripley (1996, p245) suggest a qq-plot of the *deviance residuals* as a way of assessing the fit of the negative binomial. In the S-Plus commands below, we compare the deviance residuals for the Poisson models with those of the negative binomial, and give the corresponding qqplots in Figure 18.1..

```
resP=  residuals(glm.P,type="deviance")     # Poisson residuals
resnb =  residuals(negbin.1,type="deviance")# negative binomial residuals
round(cbind(s,resP,resnb),3)
    s   resP  resnb
 1  5 -2.514 -1.178
 2  1 -4.576 -1.962
 3  5 -1.333 -0.856
 4 14 -0.798 -0.595
 5  3 -0.843 -0.570
 6 19  5.202  1.428
```

```
 7  1  0.060 -0.088
 8  1  0.060 -0.088
 9  4  1.340  0.597
10 22  3.490  0.824
par(mfrow=c(2,1))
qqnorm(resP) ; qqline(resP)
qqnorm(resnb); qqline(resnb)
```
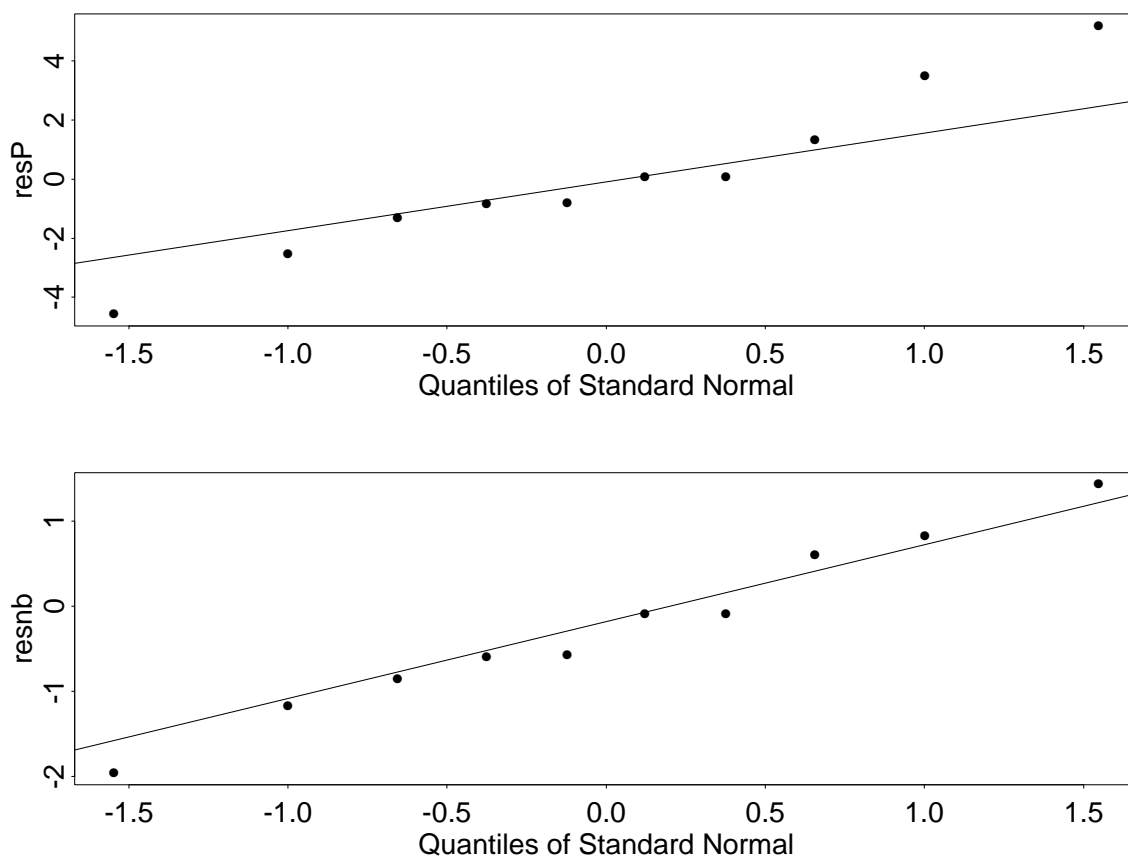
Figure 18.1: Checking the fit of the negative binomial

Chapter 19

Diagnostics for binary regression: the Hosmer Lemeshow test

This is our plan. We will simulate a covariate value x with 1000 elements, and we choose to do this from $R[0,1]$, but you might like to try other distributions.

We then choose values $a = 2, b = 1$ as our known parameter values, and compute

$$p_i = 1/1 + \exp(-a - bx_i)$$

for $i = 1, \cdots, 1000$ as our true probabilities. We then simulate the 'observation' vector y, such that $y_i = 1$ with probability p_i, $y_i = 0$ with probability $1 - p_i$.

Then we do the binary logistic regression of y on x. The first point that is of interest is to see how close the estimates of a, b are to the known true values.

The second point is to look at the deviance. Now, because we have **binary** observations, the deviance itself will not necessarily have a χ^2 distribution. Check that if you maximise the log-likelihood

$$\sum (y_i \log p_i + (1 - y_i) \log(1 - p_i))$$

with respect to $p_1, \cdots, p_{1000}$ then you will get the answer 0, whatever the values y. The number of parameters under the 'alternative' hypothesis is 1000, and this tends to infinity exactly as the number of observations (1000) tends to infinity.

Thus we cannot simply quote Wilks' theorem to give us an asymptotic χ^2 distribution for $2\times$ log(ratio of maximised likelihoods).

```
x =  runif(1000) ; a =  2; b=  1
p =  1/(1 + exp(-a -b*x))
u =  runif(1000) ; y =  (u<p)*1
plot(x,y)
first.glm =  glm(y~x,binomial) ; summary(first.glm)
plot(first.glm,ask=T) # but rather strange results!
pf =  first.glm$fitted.values ; hist(pf)
q =  quantile(pf,probs=seq(0,1,.1))# to define the deciles of risk
q =  round(q,3) ; q
z =  cut(pf, breaks=q); z # to group the fitted probabilities
table(y,z) # this is used for the Hosmer-Lemeshow goodness of fit test
pm =  tapply(pf,z,mean); round(pm,3)
```

In my simulation (where perhaps my choice of parameters could have been cleverer) this yielded

```
summary(glm.first)
Call: glm(formula = y ~ x, family = binomial)
Deviance Residuals:
      Min         1Q    Median        3Q        Max
 -2.336379 0.3812657 0.4080768 0.4383527 0.4711967
```

```
Coefficients:
               Value Std. Error  t value
(Intercept) 2.1418809  0.2189001 9.784740
        x   0.5309318  0.3945820 1.345555
(Dispersion Parameter for Binomial family taken to be 1 )
     Null Deviance: 572.0719 on 999 degrees of freedom
Residual Deviance: 570.271 on 998 degrees of freedom
Number of Fisher Scoring Iterations: 4
Correlation of Coefficients:
   (Intercept)
x   -0.8532743


table(y,z)
  0.895+ thru 0.900 0.900+ thru 0.905 0.905+ thru 0.909 0.909+ thru 0.914
0                 5                 8                12                13
1               104                83                81                99
  0.914+ thru 0.918 0.918+ thru 0.922 0.922+ thru 0.925 0.925+ thru 0.929
0                11                10                 9                 6
1                87                77                85               112
  0.929+ thru 0.933 0.933+ thru 0.935
0                 6                 3
1               111                68
# the above rounding has meant that we lost 10 observations out of the 1000

pm =  tapply(pf,z,mean) ; round(pm,3)
 0.895+ thru 0.900 0.900+ thru 0.905 0.905+ thru 0.909 0.909+ thru 0.914
           0.898             0.903             0.907             0.911
 0.914+ thru 0.918 0.918+ thru 0.922 0.922+ thru 0.925 0.925+ thru 0.929
           0.916              0.92             0.924             0.927
 0.929+ thru 0.933 0.933+ thru 0.935
           0.931             0.934
```

Thus we have computed (pm) the mean values of the fitted probabilities for each of the 10 groups. Denote these values as $\pi = k, \ k = 1, \cdots, 10$. The Hosmer-Lemeshow goodness of fit statistic for the underlying regression model is

$$\hat{C} = \sum \frac{(o_k - n_k \pi_k)^2}{n_k \pi_k (1 - \pi_k)}$$

where o_k is the 'observed' number of 1's in the kth group (ie the second row of the 2×10 table above), and n_k is the total number of observations in the kth group. Hosmer, Lemeshow and Klar(1988) showed that, using this grouping method based on the percentiles of the fitted probabilities, the null distribution of the goodness of fit statistic $\hat{C}$ is approximately χ^2 with $g - 2$ degrees of freedom, where g is the number of groups (so here $g = 10$).

Chapter 20

Logistic models for multinomial data

Here we use a dataset given in 'Optimum Experimental Designs for Multinomial Logistic Models' (1999) by S.S.Zocchi and A.C.Atkinson, in Biometrics 55, 437-444. To quote Zocchi and Atkinson 'In an experiment on the emergence of house flies.... seven sets of 500 pupae were exposed to one of several doses of radiation. Observations from each set of pupae after a period of time included the number of flies that died before the opening of the pupae (unopened pupae) (y_1),the number of flies that died before complete emergence (y_2), and the number of flies that completely emerged (y_3) from the set of $m = 500$ pupae. Given m, the response (y_1, y_2, y_3) is a trinomial random variable. In this experiment, the important proportions were y_1/m and $y_2/(m - y_1)$.'
Here is the data set.

```
     x   y1 y2  y3
1  80   62  5 433
2 100   94 24 382
3 120  179 60 261
4 140  335 80  85
5 160  432 46  22
6 180  487 11   2
7 200  498  2   0
```

Now, using an obvious notation, we see that the contribution to the log-likelihood of a given row of data is
$$\sum_i y_i \log \pi_i,$$
where
$$\sum_i \pi_i = 1.$$
The model we will fit is

$$\log(\pi_1/(1 - \pi_1)) = \theta_{10} + \theta_{11}x + \theta_{12}x^2, \text{ and} \log(\pi_2/\pi_3) = \theta_{20} + \theta_{20}x,$$

where we follow the notation of Zocchi and Atkinson.
Observe that the contribution to the log-likelihood from a single row can be written

$$[y_1 \log \pi_1 + (m - y_1) \log(1 - \pi_1)] + [y_2 \log(\pi_2/(\pi_2 + \pi_3)) + y_3 \log(\pi_3/(\pi_2 + \pi_3))].$$

Hence we can fit the model given above by combining two (independent) binomial logistic regressions, thus (with some of the output suppressed) we have

```
f.data =  read.table("flies", header=T)
attach(flies) ; m =  rep(500, times=7)
plot(x,y1/m)
```

```
plot(x, y2/(y2 +y3))
xx =  x*x
first.glm =  glm(y1/m ~ x+xx, binomial, weights=m)
summary(first.glm,cor=F)
Call: glm(formula = y1/m ~ x + xx, family = binomial, weights = m)

Coefficients:
                 Value    Std. Error    t value
(Intercept) -1.93464741 8.854925e-01 -2.184826
          x -0.02641690 1.441753e-02 -1.832277
         xx  0.00031744 5.716365e-05  5.553179

Residual Deviance: 3.164418 on 4 degrees of freedom

Number of Fisher Scoring Iterations: 6

second.glm =  glm(y2/(y2+y3)~ x, binomial, weights=y2+y3)
summary(second.glm,cor=F)
Call: glm(formula = y2/(y2 + y3) ~ x, family = binomial, weights = (y2 + y3))
Coefficients:
                 Value  Std. Error    t value
(Intercept) -9.15923692 0.522907974 -17.51596
          x  0.06386575 0.004126721  15.47615
Residual Deviance: 4.177344 on 5 degrees of freedom
Number of Fisher Scoring Iterations: 6
```

Thus we see, as Zocchi and Atkinson note, that the overall deviance measuring the fit of this model is $3.164 + 4.177$, with 9 degrees of freedom. Zocchi and Atkinson then derive the optimum *designs* for such experiments; loosely speaking, these designs tell us the doses (x) and corresponding numbers of observations required to achieve the most precise parameter estimates, for a given *total* number of observations.

It is also easy to fit a different logistic model, namely

$$\log(\pi_2/\pi_1) = \alpha_2 + \beta_2 x + \gamma_2 x^2, \text{ and } \log(\pi_3/\pi_1) = \alpha_3 + \beta_3 x + \gamma_3 x^2.$$

Try the following

```
library(MASS) ; library(nnet)
tp =  cbind(y1,y2,y3)
first.mult =  multinom(tp ~ x+xx,Hess=T); summary(first.mult)
```

Note that this model may be less suitable than the first model for the current dataset, which has a definite 'nested' structure.

Chapter 21

Binary Regression revisited; the Hosmer-Lemeshow test, the area under the ROC curve

We use the 'training' data set from 'Pattern Recognition and Neural Networks' (Cambridge University Press, 1996) by B.D.Ripley. This dataset may be accessed in R by

```
library(MASS)
data(Pima.tr)
```

To quote from Ripley's book, 'A population of women who were at least 21 years old, of Pima Indian heritage and living near Phoenix, Arizona, were tested for diabetes according to World Health Organisation criteria. The data were collected by the US National Institute of Diabetes and Digestive and Kidney Diseases, and are available from Murphy and Aha (1995)...... The reported variables are

number of pregnancies
plasma glucose concentration in oral glucose tolerance test
diastolic blood pressure (mm Hg)
triceps skinfold thickness (mm)
serum insulin (μU/ml)
body mass index (weight in kg/(height in m)^2)
diabetes pedigree function
age in years.'
(The serum insulin measurement is omitted from the data below, as it had so many missing values.)

```
 npreg glu bp skin bmi ped age type
 5   86  68 28 30.2 0.364 24 No
 7 195   70 33 25.1 0.163 55 Yes
 5   77  82 41 35.8 0.156 35 No
 0 165   76 43 47.9 0.259 26 No
 0 107   60 25 26.4 0.133 23 No
 5   97  76 27 35.6 0.378 52 Yes
 3   83  58 31 34.3 0.336 25 No
 1 193   50 16 25.9 0.655 24 No
 3 142   80 15 32.4 0.200 63 No
 2 128   78 37 43.3 1.224 31 Yes
 0 137   40 35 43.1 2.288 33 Yes
 9 154   78 30 30.9 0.164 45 No
 1 189   60 23 30.1 0.398 59 Yes
12  92   62  7 27.6 0.926 44 Yes
 1   86  66 52 41.3 0.917 29 No
```

```
 4   99  76 15 23.2 0.223 21 No
 1  109  60  8 25.4 0.947 21 No
11  143  94 33 36.6 0.254 51 Yes
 1  149  68 29 29.3 0.349 42 Yes
 0  139  62 17 22.1 0.207 21 No
 2   99  70 16 20.4 0.235 27 No
 1  100  66 29 32.0 0.444 42 No
 4   83  86 19 29.3 0.317 34 No
 0  101  64 17 21.0 0.252 21 No
 1   87  68 34 37.6 0.401 24 No
 9  164  84 21 30.8 0.831 32 Yes
 1   99  58 10 25.4 0.551 21 No
 0  140  65 26 42.6 0.431 24 Yes
 5  108  72 43 36.1 0.263 33 No
 2  110  74 29 32.4 0.698 27 No
 1   79  60 42 43.5 0.678 23 No
 3  148  66 25 32.5 0.256 22 No
 0  121  66 30 34.3 0.203 33 Yes
 3  158  64 13 31.2 0.295 24 No
 2  105  80 45 33.7 0.711 29 Yes
13  145  82 19 22.2 0.245 57 No
 1   79  80 25 25.4 0.583 22 No
 1   71  48 18 20.4 0.323 22 No
 0  102  86 17 29.3 0.695 27 No
 0  119  66 27 38.8 0.259 22 No
 8  176  90 34 33.7 0.467 58 Yes
 1   97  68 21 27.2 1.095 22 No
 4  129  60 12 27.5 0.527 31 No
 1   97  64 19 18.2 0.299 21 No
 0   86  68 32 35.8 0.238 25 No
 2  125  60 20 33.8 0.088 31 No
 5  123  74 40 34.1 0.269 28 No
 2   92  76 20 24.2 1.698 28 No
 3  171  72 33 33.3 0.199 24 Yes
 1  199  76 43 42.9 1.394 22 Yes
 3  116  74 15 26.3 0.107 24 No
 2   83  66 23 32.2 0.497 22 No
 8  154  78 32 32.4 0.443 45 Yes
 1  114  66 36 38.1 0.289 21 No
 1  106  70 28 34.2 0.142 22 No
 4  127  88 11 34.5 0.598 28 No
 1  124  74 36 27.8 0.100 30 No
 1  109  38 18 23.1 0.407 26 No
 2  123  48 32 42.1 0.520 26 No
 8  167 106 46 37.6 0.165 43 Yes
 7  184  84 33 35.5 0.355 41 Yes
 1   96  64 27 33.2 0.289 21 No
10  129  76 28 35.9 0.280 39 No
 6   92  62 32 32.0 0.085 46 No
 6  109  60 27 25.0 0.206 27 No
 5  139  80 35 31.6 0.361 25 Yes
 6  134  70 23 35.4 0.542 29 Yes
 3  106  54 21 30.9 0.292 24 No
 0  131  66 40 34.3 0.196 22 Yes
 0  135  94 46 40.6 0.284 26 No
 5  158  84 41 39.4 0.395 29 Yes
 3  112  74 30 31.6 0.197 25 Yes
```

```
 8 181  68 36 30.1 0.615 60 Yes
 2 121  70 32 39.1 0.886 23 No
 1 168  88 29 35.0 0.905 52 Yes
 1 144  82 46 46.1 0.335 46 Yes
 2 101  58 17 24.2 0.614 23 No
 2  96  68 13 21.1 0.647 26 No
 3 107  62 13 22.9 0.678 23 Yes
12 121  78 17 26.5 0.259 62 No
 2 100  64 23 29.7 0.368 21 No
 4 154  72 29 31.3 0.338 37 No
 6 125  78 31 27.6 0.565 49 Yes
10 125  70 26 31.1 0.205 41 Yes
 2 122  76 27 35.9 0.483 26 No
 2 114  68 22 28.7 0.092 25 No
 1 115  70 30 34.6 0.529 32 Yes
 7 114  76 17 23.8 0.466 31 No
 2 115  64 22 30.8 0.421 21 No
 1 130  60 23 28.6 0.692 21 No
 1  79  75 30 32.0 0.396 22 No
 4 112  78 40 39.4 0.236 38 No
 7 150  78 29 35.2 0.692 54 Yes
 1  91  54 25 25.2 0.234 23 No
 1 100  72 12 25.3 0.658 28 No
12 140  82 43 39.2 0.528 58 Yes
 4 110  76 20 28.4 0.118 27 No
 2  94  76 18 31.6 0.649 23 No
 2  84  50 23 30.4 0.968 21 No
10 148  84 48 37.6 1.001 51 Yes
 3  61  82 28 34.4 0.243 46 No
 4 117  62 12 29.7 0.380 30 Yes
 3  99  80 11 19.3 0.284 30 No
 3  80  82 31 34.2 1.292 27 Yes
 4 154  62 31 32.8 0.237 23 No
 6 103  72 32 37.7 0.324 55 No
 6 111  64 39 34.2 0.260 24 No
 0 124  70 20 27.4 0.254 36 Yes
 1 143  74 22 26.2 0.256 21 No
 1  81  74 41 46.3 1.096 32 No
 4 189 110 31 28.5 0.680 37 No
 4 116  72 12 22.1 0.463 37 No
 7 103  66 32 39.1 0.344 31 Yes
 8 124  76 24 28.7 0.687 52 Yes
 1  71  78 50 33.2 0.422 21 No
 0 137  84 27 27.3 0.231 59 No
 9 112  82 32 34.2 0.260 36 Yes
 4 148  60 27 30.9 0.150 29 Yes
 1 136  74 50 37.4 0.399 24 No
 9 145  80 46 37.9 0.637 40 Yes
 1  93  56 11 22.5 0.417 22 No
 1 107  72 30 30.8 0.821 24 No
12 151  70 40 41.8 0.742 38 Yes
 1  97  70 40 38.1 0.218 30 No
 5 144  82 26 32.0 0.452 58 Yes
 2 112  86 42 38.4 0.246 28 No
 2  99  52 15 24.6 0.637 21 No
 1 109  56 21 25.2 0.833 23 No
 1 120  80 48 38.9 1.162 41 No
```

```
 7 187   68 39 37.7 0.254 41 Yes
 3 129   92 49 36.4 0.968 32 Yes
 7 179   95 31 34.2 0.164 60 No
 6  80   66 30 26.2 0.313 41 No
 2 105   58 40 34.9 0.225 25 No
 3 191   68 15 30.9 0.299 34 No
 0  95   80 45 36.5 0.330 26 No
 4  99   72 17 25.6 0.294 28 No
 0 137   68 14 24.8 0.143 21 No
 1  97   70 15 18.2 0.147 21 No
 0 100   88 60 46.8 0.962 31 No
 1 167   74 17 23.4 0.447 33 Yes
 0 180   90 26 36.5 0.314 35 Yes
 2 122   70 27 36.8 0.340 27 No
 1  90   62 12 27.2 0.580 24 No
 3 120   70 30 42.9 0.452 30 No
 6 154   78 41 46.1 0.571 27 No
 2  56   56 28 24.2 0.332 22 No
 0 177   60 29 34.6 1.072 21 Yes
 3 124   80 33 33.2 0.305 26 No
 8  85   55 20 24.4 0.136 42 No
12  88   74 40 35.3 0.378 48 No
 9 152   78 34 34.2 0.893 33 Yes
 0 198   66 32 41.3 0.502 28 Yes
 0 188   82 14 32.0 0.682 22 Yes
 5 139   64 35 28.6 0.411 26 No
 7 168   88 42 38.2 0.787 40 Yes
 2 197   70 99 34.7 0.575 62 Yes
 2 142   82 18 24.7 0.761 21 No
 8 126   74 38 25.9 0.162 39 No
 3 158   76 36 31.6 0.851 28 Yes
 3 130   78 23 28.4 0.323 34 Yes
 2 100   54 28 37.8 0.498 24 No
 1 164   82 43 32.8 0.341 50 No
 4  95   60 32 35.4 0.284 28 No
 2 122   52 43 36.2 0.816 28 No
 4  85   58 22 27.8 0.306 28 No
 0 151   90 46 42.1 0.371 21 Yes
 6 144   72 27 33.9 0.255 40 No
 3 111   90 12 28.4 0.495 29 No
 1 107   68 19 26.5 0.165 24 No
 6 115   60 39 33.7 0.245 40 Yes
 5 105   72 29 36.9 0.159 28 No
 7 194   68 28 35.9 0.745 41 Yes
 4 184   78 39 37.0 0.264 31 Yes
 0  95   85 25 37.4 0.247 24 Yes
 7 124   70 33 25.5 0.161 37 No
 1 111   62 13 24.0 0.138 23 No
 7 137   90 41 32.0 0.391 39 No
 9  57   80 37 32.8 0.096 41 No
 2 157   74 35 39.4 0.134 30 No
 2  95   54 14 26.1 0.748 22 No
12 140   85 33 37.4 0.244 41 No
 0 117   66 31 30.8 0.493 22 No
 8 100   74 40 39.4 0.661 43 Yes
 9 123   70 44 33.1 0.374 40 No
 0 138   60 35 34.6 0.534 21 Yes
```

```
14 100  78 25 36.6 0.412 46 Yes
14 175  62 30 33.6 0.212 38 Yes
 0  74  52 10 27.8 0.269 22 No
 1 133 102 28 32.8 0.234 45 Yes
 0 119  64 18 34.9 0.725 23 No
 5 155  84 44 38.7 0.619 34 No
 1 128  48 45 40.5 0.613 24 Yes
 2 112  68 22 34.1 0.315 26 No
 1 140  74 26 24.1 0.828 23 No
 2 141  58 34 25.4 0.699 24 No
 7 129  68 49 38.5 0.439 43 Yes
 0 106  70 37 39.4 0.605 22 No
 1 118  58 36 33.3 0.261 23 No
 8 155  62 26 34.0 0.543 46 Yes
```

How can we best predict Type (Yes or No) from the remaining variables? First we fit a logistic regression model with all of the 7 variables npreg,..., age as covariates, and then use the step.glm to eliminate unnecessary variables. This removes just bp, skin from the model, leaving us with the following model:

```
sec.glm =  glm( type ~ npreg + glu + bmi + ped + age, family = binomial)
summary(sec.glm,cor=F)
Call: glm(formula = type ~ npreg + glu + bmi + ped + age, family = binomial)
Deviance Residuals:
      Min         1Q     Median         3Q       Max
 -2.000913 -0.6815805 -0.3663931 0.6466932 2.289803


Coefficients:
                 Value Std. Error    t value
(Intercept) -9.93778446 1.53499540 -6.474146
      npreg  0.10313994 0.06441686  1.601133
        glu  0.03180805 0.00665075  4.782625
        bmi  0.07966944 0.03257669  2.445597
        ped  1.81135402 0.65934884  2.747186
        age  0.03928511 0.02093333  1.876678


(Dispersion Parameter for Binomial family taken to be 1 )

    Null Deviance: 256.4142 on 199 degrees of freedom

Residual Deviance: 178.4705 on 194 degrees of freedom

Number of Fisher Scoring Iterations: 4
plot(sec.glm, ask=T)
```

These diagnostic plots are inevitably not very helpful.
Let us look at the distribution of the 'linear predictor' in the 2 groups

```
lp =  sec.glm$linear.predictors
plot(type,lp)
p =  sec.glm$fitted.values # for fitted probabilities
```

Now we compute the 'deciles' of the fitted probabilities, ready for the Hosmer-Lemeshow test.

```
q =  quantile(p,probs=seq(0,1,.1)); q
        0%        10%        20%        30%        40%        50%        60%
 0.01038852 0.05656136 0.07409683 0.1156414 0.1611173 0.2576172 0.3371531
       70%        80%        90%       100%
 0.4563907 0.6459527 0.8088184 0.9645068
```

```
q[1] =  0; q[11] =  1 # tidying up the end points for q
y =  cut(p, breaks=q)
a =  table(type,y) ; a # will be the whole table, but in too much detail
```

Here is the 2×10 Table 21.1, in a more elegant format. The final row, $EYes$, is the 'expected' number of Yes's in each of the 10 cells. The mean value of p in each of the 10 classes is computed

p	.032	.064	.093	.141	.206	.287	.402	.557	.733	.885
No	20	19	18	17	16	11	12	11	7	1
Yes	0	1	2	3	4	9	8	9	13	19
$EYes$	0.64	1.28	1.86	2.82	4.12	5.74	8.04	11.14	14.66	17.70

Table 21.1: Table for Hosmer-Lemeshow test

as follows:

```
pm =  tapply(p,y,mean) ; round(pm,3)
```

This provides the top row of the above table. Now, as an exercise, you can compute (using an obvious notation)

$$\sum_i (r_i - n_i p_i)^2 / n_i p_i (1 - p_i),$$

the Hosmer-Lemeshow statistic for assessing the fit of the binary logistic model. You should refer this to $\chi^2 = 8$.

What happens if we classify all those with $p < .056$ as 'No', and all those with $p > .056$ as 'Yes'? Then, for the current data-set, we correctly identify 20 out of 132 as 'true negatives', and 68 out of 68 as 'true positives'. For this particular threshold value therefore, we have a 'true negative' rate of 20/132, and a 'true positive' rate of 68/68. We can repeat this for the remaining 9 threshold values. The plot of one error rate, here $1 - tn$, against the other,$1 - tp$, shows us that we can only reduce one error at the expense of increasing the other. The plot of $y = tp$ (also referred to as 'sensitivity') against $x = (1 - tn)$ (also referred to as 1 - 'specificity'), which is not shown here, is known in Signal Detection Theory as the ROC (Receiver Operating Characteristic) curve. Its shape shows us how well we can discriminate between the 2 groups of patients, 'No' and 'Yes', on the basis of p, the fitted probabilities, or equivalently on the basis of logit(p), the linear predictors, and is shown here as Figure 21.1.

```
tn =  cumsum(a[1,])/132 # 'true negatives'
tp =  1 - cumsum(a[2,])/68 # 'true positives'
plot(1-tn,tp, type="l", xlim=c(0,1), ylim=c(0,1))
```

Clearly, we could get a more accurate graph by taking more than 10 threshold values. Note that the current graph is actually formed by the lines joining the 10 points. However many points we take we will always get a *concave* curve.
We now tabulate tp and tn, for the various values of q.

```
cbind(tn,tp)
                                  tn          tp
0.00000000+ thru 0.05656136 0.1515152 1.0000000
0.05656136+ thru 0.07409683 0.2954545 0.9852941
0.07409683+ thru 0.11564137 0.4318182 0.9558824
0.11564137+ thru 0.16111733 0.5606061 0.9117647
0.16111733+ thru 0.25761717 0.6818182 0.8529412
0.25761717+ thru 0.33715310 0.7651515 0.7205882
0.33715310+ thru 0.45639067 0.8560606 0.6029412
0.45639067+ thru 0.64595269 0.9393939 0.4705882
0.64595269+ thru 0.80881836 0.9924242 0.2794118
0.80881836+ thru 1.00000000 1.0000000 0.0000000
```

The Area under the ROC curve is also called the AUC: it may be computed in R by
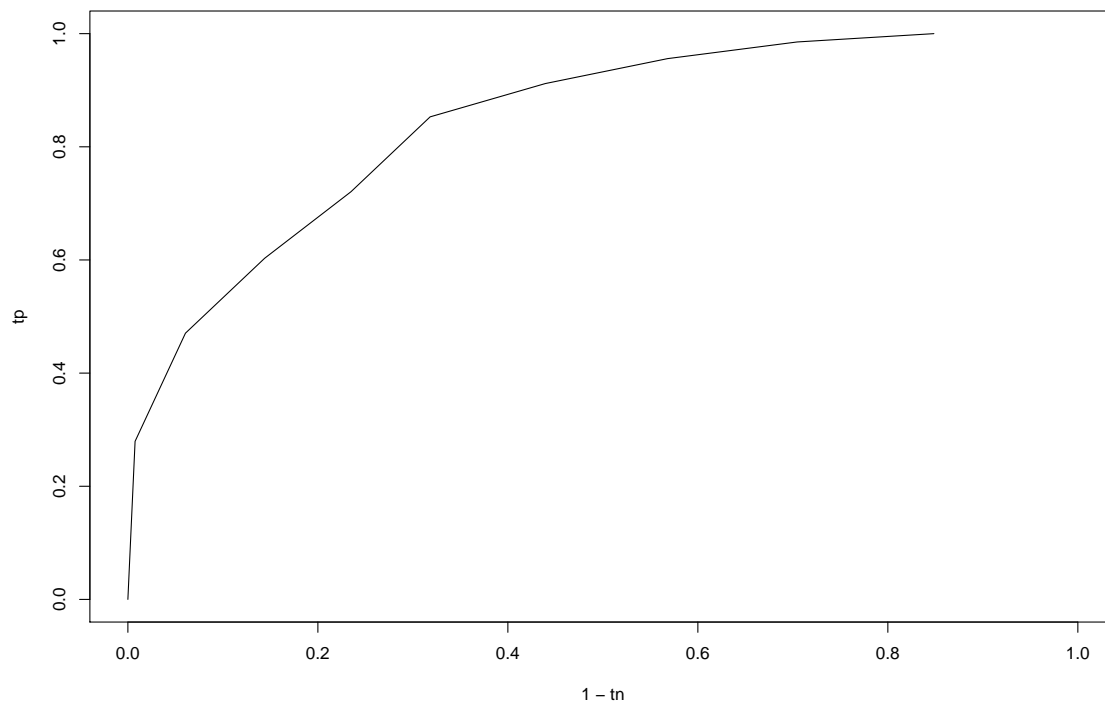
Figure 21.1: ROC curve

```
library(Hmisc)
?somers2
```

Note that Kosinski, Stillwell and Graepel 'Private traits and attributes are predictable from digital records of human behavior' in PNAS, 2013, use AUC as a measure of accuracy of prediction of individual traits in their analysis of a large dataset from Facebook, given at `www.mypersonality. org/wiki`.

Chapter 22

Using a random-effects model in binomial regression

In her 2004-5 MPhil project, Hayley Jones studied data provided by the criminologist Prof David Farrington, and an example of such a dataset is provided below. We are trying to assess the effects of CCTV in crime reduction, and have data from 19 separate studies, denoted Airdrie, ..., Sutton, below. Each study contributes its own 2×2 contingency table, consisting of (eg for Airdrie)

```
              Before  After  Total
 Experimental a= 3007 b=1687  4694
 Control      c= 3793 d=3802  7595
```

Thus we see intuitively that CCTV is successful in the Airdrie study if $b/a < d/c$, equivalently, if $(ad/bc) > 1$, equivalently, if $log(ad/bc) > 0$.

We know that (subject to the appropriate binomial assumption, or equivalently with a, b, c, d treated as independent Poisson variables) log(ad/bc) is approximately normally distributed, with variance estimated as $(1/a + 1/b + 1/c + 1/d)$. Here, for Airdrie, $log(ad/bc) = 0.5803611$, and $(1/a + 1/b + 1/c + 1/d) = (0.03810497)^2$, so that the z-value for testing effectiveness of CCTV in Airdrie is $0.5803611/0.03810497 = 15.23059$. Clearly this is way out in the tail of the $N(0, 1)$ distribution, showing us that in Airdrie, we have a significant positive effect of CCTV.

The difficulty is that we don't get exactly the same story from the other 18 studies. The worksheet that follows describes how we address this problem, by 3 different methods

i) finding a weighted average of the $log(ad/bc)$ terms

ii) using glm, with a binomial distribution, and taking 'study' as a factor with a separate effect, and then allowing for over-dispersion by suitably inflating the se's of the parameter estimates by sqrt(deviance/df).

This is a standard 'fixup' for this problem.

iii) somewhat more sophisticated (but resulting in rather similar conclusions) is to fit a 'random-effects' model to the study effects. This may be achieved by the function glmmPQL() in library(MASS)

Here is the datafile 'Hdata'

```
Study               a        b        c        d
Airdrie          3007     1687     3793     3802
Newcastle        8918     4035    17576     7125
Cambridge        2600     2242     1324      968
Birmingham        163      156       59      108
Doncaster        5832     4591     1789     2002
Burnley          1805     1410     6242     6180
Cincinnati-N    89830    23842     3589      936
Cincinnati-H   105881    20172     2193      381
Cincinnati-F   106731    17860    11788     1976
```

```
New_York_City           32        29        26        21
Underground-S          252        73       548       409
Underground-N          244        52       359       101
Underground-OC         963      1251       652       755
Montreal               905       724      1376      1124
Guildford               73         8        39         1
Hartlepool             138       145       160       299
Bradford                55        23        94       105
Coventry               367       146       206       160
Sutton                 349       149      2367      1504
```

and here is what will do with it, using Splus6

```
> cctv = read.table("Hdata", header=T)
> attach(cctv)
> mu = log((a*d)/(b*c))
# Now compute the variances of the components of mu
> v = (1/a + 1/b + 1/c + 1/d)
# Now compute the weighted average of the components of mu,
# weighted inversely in proportion to their variances
> sum(mu/v) /sum(1/v)
[1] 0.08594243          # call this tr1
>  sqrt(1/(sum(1/v)))
[1] 0.01154468 # being the corresponding se: call is se1
# The estimate tr1 is based on the approximation that
# mu1, ..., mu19 are independent Normal variables, each with
# common mean, but with variances v1,...,v19.

> bef = c(a,c) ; aft = c(b,d)
> tr = c(rep("exp", times=19), rep("con", times=19))
>tr = factor(tr)
> Study = row.names(cctv)
>study = c(Study, Study)
> study = factor(study)
> n = bef + aft
```

We fit the model $bef_{ij} \sim Binomial(n_{ij}, \pi_{ij})$ where $i = 1, 2$ for exp, con respectively,
and $j = 1, \ldots, 19$ for studies $1, \ldots, 19$ and $logit(\pi_{1j}) = \mu + \theta + \lambda_j$,
$logit(\pi_{2j}) = \mu + \lambda_j$.
The λ_j are nuisance parameters.
Observe that the default with the 'corner-point constraints' in S-Plus is to set the 'con' parameter
to zero, since S-Plus works alphabetically, and 'con' precedes 'exp' in the alphabet. Similarly, the
parameter for Airdrie λ_1 is taken as 0

```
>  first.glm = glm(bef/n ~ tr + study, binomial, weights=n)
> summary(first.glm, cor=F)
Call: glm(formula = bef/n ~ tr + study, family = binomial, weights = n)
Deviance Residuals:
      Min        1Q     Median        3Q       Max
 -7.981211 -1.696361 0.02375559 2.766822 10.3125


Coefficients:
                        Value Std. Error     t value
       (Intercept)  0.180913355 0.01865923   9.6956497
               tr   0.087343075 0.01141402   7.6522609
   studyBirmingham -0.411593380 0.09292691  -4.4292165
     studyBradford -0.053528799 0.12190218  -0.4391127
      studyBurnley -0.140379988 0.02427887  -5.7819809
```

```
       studyCambridge -0.039275336 0.03012482  -1.3037536
    studyCincinnati-F  1.528253229 0.02056214  74.3236293
    studyCincinnati-H  1.393351953 0.02083398  66.8788336
    studyCincinnati-N  1.062307496 0.02059870  51.5715718
        studyCoventry  0.395692242 0.07314208   5.4099121
       studyDoncaster -0.100013183 0.02507148  -3.9891221
       studyGuildford  2.282494761 0.34450702   6.6253941
       studyHartlepool -0.613131565 0.07706999  -7.9555163
         studyMontreal -0.004764277 0.03618533  -0.1316632
   studyNew_York_City -0.081758293 0.19388794  -0.4216781
       studyNewcastle  0.653975401 0.02137831  30.5905980
          studySutton  0.305796472 0.03622242   8.4421886
    studyUnderground-N  1.156908809 0.09233618  12.5293119
   studyUnderground-OC -0.451226255 0.03814088 -11.8305161
    studyUnderground-S  0.303795149 0.06047648   5.0233601
```

(Dispersion Parameter for Binomial family taken to be 1)

```
    Null Deviance: 28934.9 on 37 degrees of freedom

Residual Deviance: 461.5323 on 18 degrees of freedom

Number of Fisher Scoring Iterations: 4
```

This gives the treatment effect, θ as 0.087343075, with se= 0.01141402. 0.087343075 is not that different from our first estimate, tr1: its se is a little smaller ($0.01141402 < se1$), but we are not happy about this se, since it's clear that the model fails to fit. Since we have over-dispersion with respect to the binomial (461.5323 being much bigger than 18), we first do the quick-fix correction to this, by estimating the dispersion parameter ϕ as 461.5323/18, thus inflating the se's for all the parameter estimates by the factor $\sqrt{\phi}$.

```
> summary(first.glm, dispersion = 461.5323/18, cor=F)
Call: glm(formula = bef/n ~ tr + study, family = binomial, weights = n)
Deviance Residuals:
      Min        1Q     Median        3Q       Max
 -7.981211 -1.696361 0.02375559 2.766822 10.3125

Coefficients:
                          Value Std. Error      t value
        (Intercept)  0.180913355 0.09448405   1.91475018
                 tr  0.087343075 0.05779676   1.51121053
    studyBirmingham -0.411593380 0.47055054  -0.87470601
      studyBradford -0.053528799 0.61727156  -0.08671839
       studyBurnley -0.140379988 0.12294004  -1.14185737
     studyCambridge -0.039275336 0.15254191  -0.25747242
  studyCincinnati-F  1.528253229 0.10411977  14.67783871
  studyCincinnati-H  1.393351953 0.10549623  13.20759955
  studyCincinnati-N  1.062307496 0.10430489  10.18463738
       studyCoventry  0.395692242 0.37036686   1.06837917
      studyDoncaster -0.100013183 0.12695351  -0.78779376
      studyGuildford  2.282494761 1.74446744   1.30841924
     studyHartlepool -0.613131565 0.39025646  -1.57109907
        studyMontreal -0.004764277 0.18323033  -0.02600157
   studyNew_York_City -0.081758293 0.98178318  -0.08327530
       studyNewcastle  0.653975401 0.10825258   6.04119938
          studySutton  0.305796472 0.18341811   1.66720980
   studyUnderground-N  1.156908809 0.46755931   2.47435736
  studyUnderground-OC -0.451226255 0.19313255  -2.33635532
```

```
   studyUnderground-S   0.303795149 0.30623253    0.99204075
```

(Dispersion Parameter for Binomial family taken to be 25.64068)

```
    Null Deviance: 28934.9 on 37 degrees of freedom
```

Residual Deviance: 461.5323 on 18 degrees of freedom

Number of Fisher Scoring Iterations: 4

So this leaves the estimate as 0.087343075 (tr2 say) but increases its se to 0.05779676, se2 say)
Can we do more realistic modelling?
We take a random effects model for the 19 studies, assuming that the λ_j parameters form a random
sample from $N(0, \sigma^2)$. Hence the loglikelihood function must now be found by integration, and
maximized numerically. Agresti (2002, p523) gives further details.
We use S-Plus rather than R for the glmmPQL function, since we found that the R version
went straight to a parameter boundary in its first iteration, and stuck there, giving us what was
obviously an incorrect answer for the treatment effect. (The R-help pages showed that this is a
known problem, and probably results from having to maximise a log-likelihood function which is
not unimodal. You might find a suitable fixup by changing the starting-point of the iteration.)

```
> library(MASS)
> model.PQL = glmmPQL(bef/n ~ tr, random=~1|study, family=binomial, weights=n)
iteration 1
iteration 2
iteration 3
iteration 4

Linear mixed-effects model fit by maximum likelihood
 Data: NULL
       AIC      BIC     logLik
  68.74376 75.2941 -30.37188

Random effects:
 Formula:   ~ 1 | study
        (Intercept) Residual
StdDev:   0.6048569  4.80589

Variance function:
 Structure: fixed weights
 Formula:   ~ invwt
Fixed effects: bef/n ~ tr
                Value Std.Error DF  t-value p-value
(Intercept) 0.5130222 0.1609987 18 3.186499  0.0051
        tr 0.0994761 0.0567193 18 1.753833  0.0965

Standardized Within-Group Residuals:
      Min          Q1         Med         Q3        Max
 -1.46372 -0.2451681 0.07496415 0.5907415 2.030163

Number of Observations: 38
Number of Groups: 19
```

So we see that this model gives the treatment effect as say tr3=0.0994761 (se = 0.0567193). (Com-
pare Hayley's p52.) On a 2-tailed test, we can say that CCTV gives an improvement which is
significant at 10% but not at 5%.
Warning: Agresti (2002, p524) points out that 'Unfortunately PQL methods can perform poorly
relative to ML.....where possible we recommend using ML rather than PQL'. Trying ML (eg via

glmmML() in R) is something I have yet to do.

A fourth possible approach would be to condition on the marginal totals $(a+c, b+d)$ in each 2×2 table, to give a conditional likelihood function (based on the generalized hypergeometric) for θ, the parameter of interest. The product of all such 19 conditional likelihood functions will then be a function of θ only, and can be maximized in the usual way.

Agresti (2002, p232 and p508) discusses a dataset with the same structure as the one given here. He describes the use of the Cochran-Mantel-Haenszel statistic, (compare the hypergeometric distribution).

In our notation this is is

$$CMH = (\Sigma(a - e))^2 / \Sigma V$$

where $e = (a + b)(a + c)/(a + b + c + d)$ (these are the 'null-hypothesis' expected values) and $V = (a + b)(c + d)(a + c)(b + d)/(a + b + c + d)^3$ (the 'null-hypothesis' variances).

This gives a value of 7.65^2 here, and 7.65 is very close to the initial z-value that we obtained on p2, of tr1/se1. This is to be expected. (For a 2×2 table, the test of independence of rows and columns based on $(log(ad/bc))^2$ is asymptotically equivalent to that based on $(a - e)^2$.)

Chapter 23

Generalized Estimating Equations(gee) for Dr Rosanna Breen's binary data

Tom Fanshawe used this dataset in his MPhil Project, 2002-3.

In September 2002, Dr Rosanna Breen asked me about the use of Generalized Estimating Equations for her data from an educational experiment, where the response variable y was binary.

The dataset is given below: you will see that there are 1127 rows of data, with data from 9 separate subjects, whose id's are $78700, \ldots, 99032045$. Each subject is asked (almost all) of 46 different 'items', for which he/she gives either the correct response, $y = 1$, or the incorrect response, $y = 0$. In addition that subject gives the item a score, ranging from 0 to 4, according to how interesting, how difficult, and how 'relevant' is that item, (thus for example, diff=4 corresponds to the most difficult). These result in the column int, diff, rel, in the dataset. (Ignore the final 2 columns.)

	id	item	y	int	diff	rel	ec	perc
1	78700	1	1	2	1	2	2	48.78
2	78700	2	1	2	1	2	2	48.78
3	78700	3	0	2	4	3	2	48.78
4	78700	4	1	2	0	2	2	48.78
5	78700	5	1	0	0	0	2	48.78
6	78700	6	1	2	1	1	2	48.78
7	78700	7	0	2	3	1	2	48.78
8	78700	8	1	3	0	0	2	48.78
9	78700	9	0	0	4	1	2	48.78
10	78700	11	1	1	4	1	2	48.78
11	78700	12	1	3	2	1	2	48.78
12	78700	13	1	0	1	1	2	48.78
13	78700	14	0	2	4	0	2	48.78
14	78700	15	1	3	2	1	2	48.78
15	78700	16	0	2	3	1	2	48.78
16	78700	17	0	2	3	1	2	48.78
17	78700	18	0	3	4	1	2	48.78
18	78700	19	0	2	2	1	2	48.78
19	78700	22	1	3	1	1	2	48.78
20	78700	23	1	2	4	0	2	48.78
21	78700	24	0	1	1	1	2	48.78
22	78700	25	1	2	2	1	2	48.78
23	78700	26	1	3	2	1	2	48.78
24	78700	27	0	3	2	1	2	48.78
25	78700	28	1	2	3	1	2	48.78
26	78700	29	0	2	3	1	2	48.78

```
27        78700    30    0    1    3    1    2 48.78
28        78700    32    0    2    2    1    2 48.78
29        78700    33    1    0    4    0    2 48.78
30        78700    34    0    3    3    1    2 48.78
31        78700    35    1    2    2    1    2 48.78
32        78700    37    0    2    4    1    2 48.78
33        78700    38    1    2    2    1    2 48.78
34        78700    39    0    0    3    0    2 48.78
35        78700    40    0    0    3    0    2 48.78
36        78700    41    1    3    0    2    2 48.78
37        78700    42    0    2    2    2    2 48.78
38        78700    44    0    3    3    1    2 48.78
39        78700    45    0    2    4    1    2 48.78
40        78700    46    0    1    3    1    2 48.78
41        40343     1    1    2    2    4    2 34.15
42        40343     2    0    2    2    4    2 34.15
43        40343     3    0    1    0    0    2 34.15
44        40343     5    1    2    2    2    2 34.15
45        40343     7    0    2    3    2    2 34.15
46        40343     8    1    0    0    0    2 34.15
47        40343     9    0    1    2    3    2 34.15
48        40343    11    0    0    3    1    2 34.15
49        40343    12    1    2    2    2    2 34.15
50        40343    13    1    1    1    1    2 34.15
51        40343    16    0    0    1    0    2 34.15
52        40343    17    0    0    1    0    2 34.15
53        40343    18    0    2    1    1    2 34.15
54        40343    19    0    3    1    3    2 34.15
55        40343    20    0    3    1    3    2 34.15
56        40343    21    0    1    1    1    2 34.15
57        40343    22    0    1    1    0    2 34.15
58        40343    23    1    1    1    0    2 34.15
59        40343    24    1    2    1    1    2 34.15
60        40343    25    1    3    2    3    2 34.15
61        40343    26    0    3    2    4    2 34.15
62        40343    27    1    2    1    2    2 34.15
63        40343    28    0    1    2    2    2 34.15
64        40343    29    0    3    2    3    2 34.15
65        40343    30    0    1    1    1    2 34.15
66        40343    32    1    2    1    3    2 34.15
67        40343    33    0    1    4    0    2 34.15
68        40343    34    0    3    2    3    2 34.15
69        40343    35    1    3    1    1    2 34.15
70        40343    36    1    2    2    2    2 34.15
71        40343    37    0    2    1    0    2 34.15
72        40343    38    1    4    2    4    2 34.15
73        40343    39    0    2    1    0    2 34.15
74        40343    40    0    1    1    0    2 34.15
75        40343    41    0    3    1    3    2 34.15
76        40343    42    0    2    2    2    2 34.15
77        40343    43    0    2    1    0    2 34.15
78        40343    44    0    2    1    3    2 34.15
79        40343    45    0    0    0    0    2 34.15
80        40343    46    0    1    2    1    2 34.15
81        43744     1    1    4    0    4    2 31.71
82        43744     2    0    4    1    4    2 31.71
83        43744     3    0    3    1    1    2 31.71
```

```
84      43744    5   0   0   3   0   2 31.71
85      43744    6   0   0   2   2   2 31.71
86      43744    7   0   4   4   3   2 31.71
87      43744    8   1   4   0   0   2 31.71
88      43744    9   0   0   3   0   2 31.71
89      43744   11   0   0   3   0   2 31.71
90      43744   12   0   4   2   3   2 31.71
91      43744   13   1   4   0   3   2 31.71
92      43744   14   1   4   0   4   2 31.71
93      43744   16   0   0   0   1   2 31.71
94      43744   17   0   0   2   2   2 31.71
 95     43744   18   0   4   4   2   2 31.71
 96     43744   20   1   4   2   4   2 31.71
 97     43744   21   0   0   2   0   2 31.71
 98     43744   22   1   3   0   0   2 31.71
 99     43744   23   0   2   4   0   2 31.71
100     43744   24   0   3   2   2   2 31.71
101     43744   25   1   3   0   1   2 31.71
102     43744   26   1   4   2   4   2 31.71
103     43744   27   1   4   0   4   2 31.71
104     43744   28   0   0   3   0   2 31.71
105     43744   29   0   4   4   4   2 31.71
106     43744   30   0   4   3   3   2 31.71
107     43744   32   0   3   3   3   2 31.71
108     43744   33   0   0   4   0   2 31.71
109     43744   34   0   3   3   3   2 31.71
110     43744   35   1   3   0   2   2 31.71
111     43744   36   0   0   3   2   2 31.71
112     43744   37   0   2   2   3   2 31.71
113     43744   38   0   4   3   4   2 31.71
114     43744   39   0   2   2   2   2 31.71
115     43744   40   0   0   3   2   2 31.71
116     43744   41   0   4   1   4   2 31.71
117     43744   42   1   4   1   4   2 31.71
118     43744   44   1   4   0   4   2 31.71
119     43744   45   0   0   4   0   2 31.71
120     43744   46   0   0   3   0   2 31.71
121     49950    2   0   2   2   2   2 51.29
122     49950    3   1   3   2   3   2 51.29
123     49950    5   1   3   2   2   2 51.29
124     49950    6   0   3   3   2   2 51.29
125     49950    7   0   2   3   2   2 51.29
126     49950    8   1   3   2   2   2 51.29
127     49950    9   0   1   3   1   2 51.29
128     49950   12   1   2   2   2   2 51.29
129     49950   13   1   0   0   1   2 51.29
130     49950   14   1   3   2   3   2 51.29
131     49950   15   1   3   3   2   2 51.29
132     49950   16   1   1   1   1   2 51.29
133     49950   18   0   4   3  NA   2 51.29
134     49950   19   0   3   2   2   2 51.29
135     49950   20   1   2   2   2   2 51.29
136     49950   21   0   3   2   2   2 51.29
137     49950   22   1   2   1   1   2 51.29
138     49950   23   0   1   3   3   2 51.29
139     49950   24   1   1   2  NA   2 51.29
140     49950   25   1   2   1   1   2 51.29
```

141	49950	26	1	3	2	2	2 51.29
142	49950	27	1	3	2	2	2 51.29
143	49950	28	1	0	3	3	2 51.29
144	49950	29	0	2	3	3	2 51.29
145	49950	30	0	2	3	3	2 51.29
146	49950	32	1	2	3	3	2 51.29
147	49950	33	0	2	2	2	2 51.29
148	49950	34	0	3	3	NA	2 51.29
149	49950	35	0	2	3	3	2 51.29
150	49950	36	1	3	2	2	2 51.29
151	49950	37	0	2	3	3	2 51.29
152	49950	38	1	3	2	2	2 51.29
153	49950	40	0	2	3	3	2 51.29
154	49950	41	0	2	1	1	2 51.29
155	49950	42	1	3	2	2	2 51.29
156	49950	43	0	3	3	3	2 51.29
157	49950	44	1	3	2	2	2 51.29
158	49950	45	0	2	2	2	2 51.29
159	49950	46	1	3	2	2	2 51.29
160	45074	1	1	4	0	2	1 41.86
161	45074	2	1	4	0	4	1 41.86
162	45074	3	0	0	2	0	1 41.86
163	45074	4	1	0	4	0	1 41.86
164	45074	5	1	4	0	2	1 41.86
165	45074	6	0	2	2	4	1 41.86
166	45074	7	1	4	0	4	1 41.86
167	45074	8	1	4	4	0	1 41.86
168	45074	9	0	2	2	2	1 41.86
169	45074	11	0	0	4	0	1 41.86
170	45074	12	0	4	0	0	1 41.86
171	45074	13	1	4	0	0	1 41.86
172	45074	14	1	4	0	4	1 41.86
173	45074	15	0	4	2	2	1 41.86
174	45074	16	1	4	2	0	1 41.86
175	45074	17	0	2	2	0	1 41.86
176	45074	18	0	0	2	4	1 41.86
177	45074	20	1	4	0	4	1 41.86
178	45074	21	0	0	4	2	1 41.86
179	45074	22	0	0	4	0	1 41.86
180	45074	23	1	2	0	2	1 41.86
181	45074	24	1	2	1	0	1 41.86
182	45074	27	1	4	2	4	1 41.86
183	45074	28	1	2	4	4	1 41.86
184	45074	29	0	0	4	0	1 41.86
185	45074	30	0	2	2	2	1 41.86
186	45074	32	0	0	3	0	1 41.86
187	45074	33	0	0	4	0	1 41.86
188	45074	34	1	2	1	0	1 41.86
189	45074	35	0	2	4	2	1 41.86
190	45074	36	0	4	2	2	1 41.86
191	45074	37	0	4	4	2	1 41.86
192	45074	38	0	4	2	2	1 41.86
193	45074	39	0	4	4	0	1 41.86
194	45074	40	0	2	4	0	1 41.86
195	45074	41	1	4	2	2	1 41.86
196	45074	42	0	4	2	2	1 41.86
197	45074	43	1	4	0	2	1 41.86

198	45074	44	0	4	2	2	1	41.86
199	45074	45	0	0	4	0	1	41.86
200	45074	46	0	2	4	0	1	41.86
201	46063	1	1	1	2	0	2	29.55
202	46063	2	0	2	2	2	2	29.55
203	46063	3	0	3	3	3	2	29.55
204	46063	4	1	1	1	0	2	29.55
205	46063	5	0	0	3	0	2	29.55
206	46063	6	0	3	3	3	2	29.55
207	46063	7	0	0	4	1	2	29.55
208	46063	8	1	0	0	0	2	29.55
209	46063	9	0	2	3	3	2	29.55
210	46063	11	0	1	4	0	2	29.55
211	46063	12	1	2	0	0	2	29.55
212	46063	13	1	2	0	0	2	29.55
213	46063	14	1	1	0	0	2	29.55
214	46063	15	0	0	4	0	2	29.55
215	46063	16	0	0	4	0	2	29.55
216	46063	17	0	0	4	0	2	29.55
217	46063	18	0	3	3	4	2	29.55
218	46063	19	1	0	1	4	2	29.55
219	46063	20	1	3	2	0	2	29.55
220	46063	21	1	3	0	1	2	29.55
221	46063	22	1	1	3	0	2	29.55
222	46063	23	0	0	4	0	2	29.55
223	46063	24	0	0	3	0	2	29.55
224	46063	25	1	2	0	2	2	29.55
225	46063	26	0	1	2	0	2	29.55
226	46063	27	0	3	2	3	2	29.55
227	46063	28	0	0	3	0	2	29.55
228	46063	29	0	2	4	2	2	29.55
229	46063	30	0	0	4	1	2	29.55
230	46063	32	0	1	4	0	2	29.55
231	46063	33	0	2	4	2	2	29.55
232	46063	34	0	2	4	1	2	29.55
233	46063	36	0	1	3	0	2	29.55
234	46063	37	0	2	3	3	2	29.55
235	46063	38	0	2	2	3	2	29.55
236	46063	39	0	1	3	3	2	29.55
237	46063	40	0	2	3	3	2	29.55
238	46063	41	0	2	0	1	2	29.55
239	46063	42	0	3	1	4	2	29.55
240	46063	43	0	1	4	3	2	29.55
241	46063	44	1	3	0	4	2	29.55
242	46063	45	0	0	3	0	2	29.55
243	46063	46	0	0	4	0	2	29.55
244	58996	1	1	4	2	4	1	29.55
245	58996	2	0	3	4	4	1	29.55
246	58996	3	0	4	3	3	1	29.55
247	58996	4	0	2	2	4	1	29.55
248	58996	5	1	4	0	4	1	29.55
249	58996	6	0	2	3	4	1	29.55
250	58996	7	0	0	2	3	1	29.55
251	58996	8	0	4	1	2	1	29.55
252	58996	9	0	0	2	4	1	29.55
253	58996	11	0	2	2	0	1	29.55
254	58996	12	1	0	2	3	1	29.55

```
255      58996   13   1   4    2    1    1 29.55
256      58996   15   0   0    4    0    1 29.55
257      58996   16   0   0    3    1    1 29.55
258      58996   17   0   2    3    2    1 29.55
259      58996   18   0   3    2    2    1 29.55
260      58996   19   1   3    2    3    1 29.55
261      58996   20   1   4    4    3    1 29.55
262      58996   21   0   4    2    4    1 29.55
263      58996   22   1   4    2    4    1 29.55
264      58996   23   0   0    3    0    1 29.55
265      58996   24   0   0    3    0    1 29.55
266      58996   25   0   2    4    3    1 29.55
267      58996   26   1   4    4    4    1 29.55
268      58996   27   1   2    3    2    1 29.55
269      58996   28   1   2    2    3    1 29.55
270      58996   29   0   2    2    1    1 29.55
271      58996   30   0   1    2    1    1 29.55
272      58996   32   0   0    4    0    1 29.55
273      58996   33   0   1    2    1    1 29.55
274      58996   34   0   4    3    4    1 29.55
275      58996   35   0   4    3    2    1 29.55
276      58996   36   0   2    2    0    1 29.55
277      58996   37   0   4    0    4    1 29.55
278      58996   38   0   2    2    0    1 29.55
279      58996   39   0   1    2    2    1 29.55
280      58996   40   0   2    2    1    1 29.55
281      58996   41   0   4    2    1    1 29.55
282      58996   42   0   3    2    4    1 29.55
283      58996   43   1   2    3    4    1 29.55
284      58996   44   1   4    2    4    1 29.55
285      58996   45   1   2    3    4    1 29.55
286      58996   46   0   3    2    0    1 29.55
287      53327    1   0   0    3    0    1 31.12
288      53327    2   0   2    2    2    1 31.12
289      53327    3   0   2    2    3    1 31.12
290      53327    4   0   1    3    0    1 31.12
291      53327    5   0   1    1    1    1 31.12
292      53327    6   0   1    1    1    1 31.12
293      53327    7   0   1    1    1    1 31.12
294      53327    8   1   3    1    0    1 31.12
295      53327    9   0   3    3    1    1 31.12
296      53327   11   0   0    4    0    1 31.12
297      53327   12   1   3    0    0    1 31.12
298      53327   13   1   NA   NA   NA   1 31.12
299      53327   14   0   2    2    2    1 31.12
300      53327   15   1   3    1    2    1 31.12
301      53327   16   1   0    0    1    1 31.12
302      53327   17   0   2    2    1    1 31.12
303      53327   18   0   2    4    1    1 31.12
304      53327   19   0   3    3    2    1 31.12
305      53327   20   1   1    0    0    1 31.12
306      53327   21   0   2    0    1    1 31.12
307      53327   22   0   2    2    1    1 31.12
308      53327   23   0   2    2    2    1 31.12
309      53327   24   0   1    1    1    1 31.12
310      53327   25   1   1    2    0    1 31.12
311      53327   26   0   3    0    2    1 31.12
```

312	53327	27	1	2	2	2	1 31.12
313	53327	28	1	0	0	0	1 31.12
314	53327	29	0	2	2	2	1 31.12
315	53327	30	0	2	2	1	1 31.12
316	53327	32	0	1	1	0	1 31.12
317	53327	33	1	1	4	0	1 31.12
318	53327	34	1	2	1	2	1 31.12
319	53327	35	1	2	2	1	1 31.12
320	53327	36	0	0	0	0	1 31.12
321	53327	37	0	2	2	1	1 31.12
322	53327	38	0	2	1	2	1 31.12
323	53327	39	0	1	1	1	1 31.12
324	53327	40	0	2	3	1	1 31.12
325	53327	41	0	2	3	0	1 31.12
326	53327	42	0	1	2	1	1 31.12
327	53327	43	1	3	0	3	1 31.12
328	53327	44	1	0	0	1	1 31.12
329	53327	45	0	1	1	0	1 31.12
330	53327	46	0	2	1	1	1 31.12
331	40307	1	0	2	2	3	1 32.56
332	40307	2	0	3	3	3	1 32.56
333	40307	3	0	1	4	0	1 32.56
334	40307	4	0	1	4	0	1 32.56
335	40307	5	1	4	3	0	1 32.56
336	40307	6	1	4	2	1	1 32.56
337	40307	7	0	0	4	1	1 32.56
338	40307	8	1	0	0	0	1 32.56
339	40307	9	0	0	4	2	1 32.56
340	40307	11	0	0	4	0	1 32.56
341	40307	12	0	1	0	0	1 32.56
342	40307	13	1	1	0	4	1 32.56
343	40307	14	1	4	2	4	1 32.56
344	40307	15	0	4	0	1	1 32.56
345	40307	16	0	0	0	0	1 32.56
346	40307	17	0	0	2	1	1 32.56
347	40307	18	0	0	4	1	1 32.56
348	40307	19	1	4	0	4	1 32.56
349	40307	20	1	4	0	4	1 32.56
350	40307	21	0	0	4	0	1 32.56
351	40307	22	0	2	1	0	1 32.56
352	40307	23	0	0	4	3	1 32.56
353	40307	24	1	3	0	2	1 32.56
354	40307	26	1	4	2	3	1 32.56
355	40307	27	0	4	1	0	1 32.56
356	40307	28	1	3	2	4	1 32.56
357	40307	29	0	0	4	0	1 32.56
358	40307	30	0	3	3	1	1 32.56
359	40307	32	0	4	2	2	1 32.56
360	40307	33	0	1	4	0	1 32.56
361	40307	34	0	4	2	2	1 32.56
362	40307	36	1	4	0	3	1 32.56
363	40307	37	1	2	2	2	1 32.56
364	40307	38	0	2	3	3	1 32.56
365	40307	39	0	3	1	4	1 32.56
366	40307	40	0	2	2	3	1 32.56
367	40307	41	0	3	2	2	1 32.56
368	40307	42	0	2	4	3	1 32.56

```
369      40307    43    0    4    3    4    1 32.56
370      40307    44    1    4    0    4    1 32.56
371      40307    45    0    0    3    0    1 32.56
372      40307    46    0    1    3    3    1 32.56
373      39764     1    0    1    4    4    1 11.91
374      39764     2    0    4    3    3    1 11.91
375      39764     3    0    0    3    3    1 11.91
376      39764     5    0    0    3    0    1 11.91
377      39764     6    0    2    4    2    1 11.91
378      39764     7    0    2    1    0    1 11.91
379      39764     8    1    0    0    0    1 11.91
380      39764     9    0    0    4    3    1 11.91
381      39764    11    0    4    3    2    1 11.91
382      39764    12    0    2    2    0    1 11.91
383      39764    13    1    0    0    4    1 11.91
384      39764    14    0    2    3    4    1 11.91
385      39764    15    0    0    4    3    1 11.91
386      39764    16    0    4    0    4    1 11.91
387      39764    17    1    3    2    4    1 11.91
388      39764    18    0    4    0    2    1 11.91
389      39764    19    0    4    4    4    1 11.91
390      39764    20    0    4    0    4    1 11.91
391      39764    21    0    4    0    4    1 11.91
392      39764    22    0    3    0    2    1 11.91
393      39764    23    0    3    3    2    1 11.91
394      39764    24    0    2    2    0    1 11.91
395      39764    26    0    3    4    0    1 11.91
396      39764    27    0    2    2    4    1 11.91
397      39764    28    0    0    4    4    1 11.91
398      39764    29    0    0    4    0    1 11.91
399      39764    30    0    0    3    3    1 11.91
400      39764    32    0    4    3    3    1 11.91
401      39764    33    0    0    4    0    1 11.91
402      39764    34    0    4    3    0    1 11.91
403      39764    35    0    4    3    4    1 11.91
404      39764    36    0    4    2    0    1 11.91
405      39764    37    0    3    1    0    1 11.91
406      39764    38    0    1    3    4    1 11.91
407      39764    39    0    0    0    3    1 11.91
408      39764    40    0    0    2    3    1 11.91
409      39764    41    0    0    3    2    1 11.91
410      39764    42    0    0    4    0    1 11.91
411      39764    43    0    0    2    1    1 11.91
412      39764    44    1    2    0    4    1 11.91
413      39764    45    1    0    3    0    1 11.91
414      39764    46    0    3    1    2    1 11.91
415      77983     1    1    4    3    2    4 81.09
416      77983     2    1    4    4    4    4 81.09
417      77983     3    1    4    4    4    4 81.09
418      77983     6    1    2    3    1    4 81.09
419      77983     7    1    3    1    2    4 81.09
420      77983     8    1    0    4    1    4 81.09
421      77983     9    1    4    4    4    4 81.09
422      77983    11    1    2    2    3    4 81.09
423      77983    12    1    4    3    3    4 81.09
424      77983    14    1    4    4    2    4 81.09
425      77983    15    1    4    2    3    4 81.09
```

```
426        77983   16    1    4    4    4    4 81.09
427        77983   17    1    4    3    4    4 81.09
428        77983   18    0    3    2    3    4 81.09
429        77983   19    0    4    1    4    4 81.09
430        77983   20    1    4    4    4    4 81.09
431        77983   21    1    4    4    4    4 81.09
432        77983   22    1    4    3    2    4 81.09
433        77983   23    1    4    4    3    4 81.09
434        77983   24    1    4    4    3    4 81.09
435        77983   25    1    4    4    4    4 81.09
436        77983   26    1    4    3    4    4 81.09
437        77983   27    1    4    3    2    4 81.09
438        77983   29    0    4    3    3    4 81.09
439        77983   30    0    4    3    2    4 81.09
440        77983   32    1    4    4    4    4 81.09
441        77983   33    0    3    4    3    4 81.09
442        77983   34    1    4    3    4    4 81.09
443        77983   35    1    4    2    2    4 81.09
444        77983   36    1    4    4    3    4 81.09
445        77983   37    1    4    4    3    4 81.09
446        77983   38    0    4    3    4    4 81.09
447        77983   40    1    4    4    4    4 81.09
448        77983   41    1    4    3    3    4 81.09
449        77983   43    1    4    3    4    4 81.09
450        77983   44    1    2    2    1    4 81.09
451        77983   45    0    0    4    3    4 81.09
452        77983   46    0    2    4    2    4 81.09
453        56296    1    1    3    4    3    3 72.98
454        56296    2    0    3    3    3    3 72.98
455        56296    3    0    2    4    4    3 72.98
456        56296    5    1    1    2    1    3 72.98
457        56296    7    1    2    2    2    3 72.98
458        56296    8    1    3    1    3    3 72.98
459        56296    9    0    1    4    2    3 72.98
460        56296   12    1    1    2    2    3 72.98
461        56296   14    0    2    3    2    3 72.98
462        56296   16    1    1    2    3    3 72.98
463        56296   17    1    1    1    3    3 72.98
464        56296   18    1    3    3    3    3 72.98
465        56296   19    1    3    2    3    3 72.98
466        56296   20    1    2    3    3    3 72.98
467        56296   21    0    1    4    2    3 72.98
468        56296   22    1    3    3    4    3 72.98
469        56296   23    1    2    3    2    3 72.98
470        56296   24    1    1    3    2    3 72.98
471        56296   25    1    3    3    3    3 72.98
472        56296   26    0    3    3    3    3 72.98
473        56296   27    1    3    3    3    3 72.98
474        56296   29    0    2    4    2    3 72.98
475        56296   30    1    2    3    4    3 72.98
476        56296   32    1    2    3    2    3 72.98
477        56296   33    1    3    4    4    3 72.98
478        56296   34    1    2    4    2    3 72.98
479        56296   35    1    3    3    3    3 72.98
480        56296   36    1    1    1    2    3 72.98
481        56296   37    1    1    4    1    3 72.98
482        56296   38    0    2    2    1    3 72.98
```

```
483      56296    39    1    1    2    2    3 72.98
484      56296    40    1    2    3    1    3 72.98
485      56296    42    1    2    3    1    3 72.98
486      56296    43    0    3    3    2    3 72.98
487      56296    44    0    2    2    2    3 72.98
488      56296    45    1    2    2    3    3 72.98
489      56296    46    1    1    2    2    3 72.98
490      43555     1    1    3    0    3    3 52.28
491      43555     2    0    3    1    2    3 52.28
492      43555     3    0    1    1    1    3 52.28
493      43555     4    1    2    0    0    3 52.28
494      43555     5    1    0    0    1    3 52.28
495      43555     7    1    2    2    3    3 52.28
496      43555     8    1    1    1    1    3 52.28
497      43555     9    1    2    2    1    3 52.28
498      43555    11    0    3    2    3    3 52.28
499      43555    12    1    4    0    3    3 52.28
500      43555    14    1    1    0    2    3 52.28
501      43555    15    1    3    0    3    3 52.28
502      43555    16    1    1    0    2    3 52.28
503      43555    17    1    3    1    3    3 52.28
504      43555    18    0    2    3    2    3 52.28
505      43555    19    0    3    1    2    3 52.28
506      43555    20    1    3    1    3    3 52.28
507      43555    21    0    0    2    1    3 52.28
508      43555    22    1    2    1    0    3 52.28
509      43555    23    0    3    3    3    3 52.28
510      43555    24    0    3    2    3    3 52.28
511      43555    25    1    3    0    3    3 52.28
512      43555    26    0    4    2    3    3 52.28
513      43555    27    1    2    2    2    3 52.28
514      43555    28    1    0    3    1    3 52.28
515      43555    29    1    1    1    2    3 52.28
516      43555    30    1    2    2    3    3 52.28
517      43555    32    0    2    1    1    3 52.28
518      43555    33    1    1    2    1    3 52.28
519      43555    34    0    2    2    2    3 52.28
520      43555    35    0    1    1    1    3 52.28
521      43555    36    0    3    1    4    3 52.28
522      43555    37    0    1    1    0    3 52.28
523      43555    38    0    3    3    2    3 52.28
524      43555    39    1    3    2    2    3 52.28
525      43555    40    0    2    3    1    3 52.28
526      43555    41    1    3    1    3    3 52.28
527      43555    42    0    3    3    3    3 52.28
528      43555    43    0    2    2    2    3 52.28
529      43555    44    0    3    0    0    3 52.28
530      43555    45    0    2    0    0    3 52.28
531      43555    46    0    3    0    0    3 52.28
532      41829     1    1    1    4    3    3 50.00
533      41829     2    0    3    2    3    3 50.00
534      41829     3    0    1    4    1    3 50.00
535      41829     5    1    1    3    1    3 50.00
536      41829     6    0    2    3    2    3 50.00
537      41829     7    1    2    3    2    3 50.00
538      41829     8    1    2    1    3    3 50.00
539      41829     9    0    2    3    1    3 50.00
```

```
540        41829   11   0   3   3   2   3 50.00
541        41829   12   1   3   2   3   3 50.00
542        41829   13   1   3   1   4   3 50.00
543        41829   14   0   2   2   2   3 50.00
544        41829   15   0   2   2   2   3 50.00
545        41829   16   1   1   2   2   3 50.00
546        41829   17   1   3   1   4   3 50.00
547        41829   18   0   3   2   2   3 50.00
548        41829   19   1   3   1   4   3 50.00
549        41829   20   1   3   1   2   3 50.00
550        41829   21   1   1   2   1   3 50.00
551        41829   22   0   2   2   2   3 50.00
552        41829   23   1   1   2   3   3 50.00
553        41829   24   0   2   2   3   3 50.00
554        41829   26   0   3   2   3   3 50.00
555        41829   27   0   2   3   4   3 50.00
556        41829   28   1   3   3   3   3 50.00
557        41829   29   1   3   3   4   3 50.00
558        41829   30   0   3   2   3   3 50.00
559        41829   33   0   1   3   2   3 50.00
560        41829   34   0   2   3   2   3 50.00
561        41829   35   1   3   3   3   3 50.00
562        41829   36   1   2   1   3   3 50.00
563        41829   37   1   2   3   3   3 50.00
564        41829   38   1   3   2   3   3 50.00
565        41829   39   0   2   3   2   3 50.00
566        41829   40   0   1   4   1   3 50.00
567        41829   41   1   2   2   3   3 50.00
568        41829   42   0   2   2   3   3 50.00
569        41829   44   0   2   2   3   3 50.00
570        41829   45   1   3   2   3   3 50.00
571        41829   46   0   2   3   2   3 50.00
572        51556    1   1   2   3   2   3 56.10
573        51556    2   1   3   2   3   3 56.10
574        51556    3   1   2   3   3   3 56.10
575        51556    4   1   1   2   1   3 56.10
576        51556    5   0   2   2   2   3 56.10
577        51556    6   0   3   3   3   3 56.10
578        51556    7   1   3   2   4   3 56.10
579        51556    8   1   1   1   2   3 56.10
580        51556    9   0   1   2   1   3 56.10
581        51556   11   0   1   3   3   3 56.10
582        51556   12   1   2   1   2   3 56.10
583        51556   14   1   2   1   2   3 56.10
584        51556   15   0   3   2   4   3 56.10
585        51556   16   0   1   2   3   3 56.10
586        51556   18   1   2   2   3   3 56.10
587        51556   19   0   3   2   4   3 56.10
588        51556   20   0   1   2   1   3 56.10
589        51556   21   0   1   3   2   3 56.10
590        51556   22   1   2   2   2   3 56.10
591        51556   23   1   3   3   4   3 56.10
592        51556   24   1   2   2   2   3 56.10
593        51556   25   1   3   2   4   3 56.10
594        51556   26   1   3   2   3   3 56.10
595        51556   27   1   2   3   3   3 56.10
596        51556   28   0   1   3   3   3 56.10
```

```
597        51556   29   0   3   3   3   3 56.10
598        51556   30   0   2   1   3   3 56.10
599        51556   32   1   1   2   2   3 56.10
600        51556   33   0   2   2   3   3 56.10
601        51556   34   1   3   2   3   3 56.10
602        51556   35   1   1   1   3   3 56.10
603        51556   36   1   3   1   3   3 56.10
604        51556   37   1   2   3   2   3 56.10
605        51556   38   0   3   2   3   3 56.10
606        51556   40   0   3   3   3   3 56.10
607        51556   42   0   3   2   3   3 56.10
608        51556   43   0   2   3   3   3 56.10
609        51556   44   1   2   2   3   3 56.10
610        51556   45   1   3   3   3   3 56.10
611        51556   46   0   2   1   3   3 56.10
612        38080    1   0   3   4   1   4 17.08
613        38080    2   0   1   3   2   4 17.08
614        38080    3   0   4   2   4   4 17.08
615        38080    4   1   4   3   0   4 17.08
616        38080    5   0   0   1   0   4 17.08
617        38080    6   0   2   2   2   4 17.08
618        38080    7   0   1   3   2   4 17.08
619        38080    8   1   4   2   3   4 17.08
620        38080    9   0   1   4   1   4 17.08
621        38080   11   0   3   3   3   4 17.08
622        38080   12   0   3   1   3   4 17.08
623        38080   14   1   3   2   3   4 17.08
624        38080   15   0   3   2   2   4 17.08
625        38080   16   0   3   0   3   4 17.08
626        38080   17   0   3   0   3   4 17.08
627        38080   18   0   3   2   3   4 17.08
628        38080   19   0   2   0   2   4 17.08
629        38080   20   1   3   0   2   4 17.08
630        38080   21   0   1   1   1   4 17.08
631        38080   22   0   4   0   4   4 17.08
632        38080   23   0   2   2   2   4 17.08
633        38080   24   0   3   2   3   4 17.08
634        38080   25   1   3   1   2   4 17.08
635        38080   26   0   2   4   1   4 17.08
636        38080   27   0   4   2   4   4 17.08
637        38080   28   0   1   4   1   4 17.08
638        38080   29   0   2   4   2   4 17.08
639        38080   30   0   3   3   3   4 17.08
640        38080   32   0   1   3   1   4 17.08
641        38080   33   0   4   2   4   4 17.08
642        38080   34   0   1   3   1   4 17.08
643        38080   35   0   3   1   3   4 17.08
644        38080   36   0   3   0   3   4 17.08
645        38080   37   0   1   4   2   4 17.08
646        38080   38   0   3   2   3   4 17.08
647        38080   40   1   1   4   1   4 17.08
648        38080   42   0   2   4   2   4 17.08
649        38080   43   0   0   4   0   4 17.08
650        38080   44   0   2   0   1   4 17.08
651        38080   45   1   3   4   3   4 17.08
652        38080   46   0   2   3   2   4 17.08
653        38701    1   0   2   3   1   4 34.09
```

654	38701	2	0	3	2	3	4	34.09
655	38701	3	0	2	3	1	4	34.09
656	38701	4	1	2	2	1	4	34.09
657	38701	5	1	3	2	2	4	34.09
658	38701	6	0	3	2	2	4	34.09
659	38701	7	0	2	3	1	4	34.09
660	38701	8	1	3	1	3	4	34.09
661	38701	9	0	2	3	1	4	34.09
662	38701	11	0	3	4	2	4	34.09
663	38701	12	1	4	2	4	4	34.09
664	38701	13	1	3	0	3	4	34.09
665	38701	14	0	3	3	3	4	34.09
666	38701	15	1	4	3	3	4	34.09
667	38701	16	0	4	1	3	4	34.09
668	38701	17	0	3	2	3	4	34.09
669	38701	18	0	3	4	2	4	34.09
670	38701	19	1	4	2	3	4	34.09
671	38701	20	1	3	2	2	4	34.09
672	38701	21	0	3	3	3	4	34.09
673	38701	22	0	3	2	1	4	34.09
674	38701	23	0	2	4	2	4	34.09
675	38701	24	1	3	2	2	4	34.09
676	38701	25	1	3	2	3	4	34.09
677	38701	26	0	3	3	2	4	34.09
678	38701	27	0	3	3	2	4	34.09
679	38701	28	0	1	4	1	4	34.09
680	38701	29	0	2	3	1	4	34.09
681	38701	30	0	2	3	1	4	34.09
682	38701	32	0	3	3	2	4	34.09
683	38701	33	0	2	3	1	4	34.09
684	38701	34	0	3	3	2	4	34.09
685	38701	35	1	4	2	3	4	34.09
686	38701	37	0	2	3	2	4	34.09
687	38701	38	0	3	3	3	4	34.09
688	38701	39	0	3	2	3	4	34.09
689	38701	40	0	2	3	1	4	34.09
690	38701	41	0	3	2	3	4	34.09
691	38701	42	0	3	3	2	4	34.09
692	38701	43	1	3	2	3	4	34.09
693	38701	44	1	3	2	2	4	34.09
694	38701	45	0	3	2	3	4	34.09
695	38701	46	1	2	2	2	4	34.09
696	5540	1	1	4	1	4	4	80.00
697	5540	2	1	4	1	4	4	80.00
698	5540	3	0	4	1	4	4	80.00
699	5540	4	0	2	1	1	4	80.00
700	5540	6	1	2	1	2	4	80.00
701	5540	7	0	2	1	1	4	80.00
702	5540	9	1	2	2	1	4	80.00
703	5540	11	1	3	1	3	4	80.00
704	5540	12	1	4	1	1	4	80.00
705	5540	14	1	3	1	2	4	80.00
706	5540	15	1	4	0	4	4	80.00
707	5540	16	1	1	1	1	4	80.00
708	5540	17	0	4	1	4	4	80.00
709	5540	18	1	4	2	4	4	80.00
710	5540	19	0	4	1	4	4	80.00

711	5540	20	0	4	1	2	4 80.00
712	5540	21	1	2	2	2	4 80.00
713	5540	22	1	3	1	3	4 80.00
714	5540	23	1	4	1	4	4 80.00
715	5540	24	1	2	1	1	4 80.00
716	5540	26	1	2	1	4	4 80.00
717	5540	27	1	3	1	1	4 80.00
718	5540	28	1	2	2	1	4 80.00
719	5540	29	1	2	1	1	4 80.00
720	5540	30	1	3	1	2	4 80.00
721	5540	32	1	3	1	1	4 80.00
722	5540	33	1	2	1	2	4 80.00
723	5540	34	1	4	1	3	4 80.00
724	5540	35	1	4	1	3	4 80.00
725	5540	36	1	2	1	2	4 80.00
726	5540	37	1	2	1	2	4 80.00
727	5540	38	1	2	1	3	4 80.00
728	5540	40	1	2	3	2	4 80.00
729	5540	42	1	3	2	4	4 80.00
730	5540	43	1	2	1	4	4 80.00
731	5540	44	1	3	1	4	4 80.00
732	5540	45	0	3	1	4	4 80.00
733	5540	46	0	2	1	2	4 80.00
734	57301	1	1	1	4	4	4 41.31
735	57301	2	0	4	2	4	4 41.31
736	57301	3	0	2	4	2	4 41.31
737	57301	4	1	2	2	3	4 41.31
738	57301	5	0	2	0	2	4 41.31
739	57301	6	0	2	4	2	4 41.31
740	57301	7	1	4	4	4	4 41.31
741	57301	9	0	2	4	2	4 41.31
742	57301	11	1	4	3	4	4 41.31
743	57301	12	1	4	1	4	4 41.31
744	57301	13	1	4	2	4	4 41.31
745	57301	14	1	2	2	4	4 41.31
746	57301	15	1	4	2	4	4 41.31
747	57301	16	1	2	2	4	4 41.31
748	57301	18	1	4	4	4	4 41.31
749	57301	19	0	4	0	4	4 41.31
750	57301	20	1	2	2	4	4 41.31
751	57301	22	1	3	3	3	4 41.31
752	57301	23	1	4	2	4	4 41.31
753	57301	24	0	2	4	3	4 41.31
754	57301	25	1	4	3	4	4 41.31
755	57301	26	0	4	0	4	4 41.31
756	57301	27	0	4	4	4	4 41.31
757	57301	28	0	1	4	2	4 41.31
758	57301	29	0	4	3	4	4 41.31
759	57301	30	0	4	2	4	4 41.31
760	57301	32	1	4	2	4	4 41.31
761	57301	33	0	3	4	3	4 41.31
762	57301	34	0	4	2	4	4 41.31
763	57301	35	1	4	4	4	4 41.31
764	57301	36	0	2	4	4	4 41.31
765	57301	37	1	4	4	3	4 41.31
766	57301	38	0	4	3	4	4 41.31
767	57301	39	0	4	3	4	4 41.31

768	57301	40	0	4	2	4	4	41.31
769	57301	42	1	4	2	4	4	41.31
770	57301	43	1	4	2	4	4	41.31
771	57301	44	0	4	2	4	4	41.31
772	57301	45	0	4	2	4	4	41.31
773	57301	46	0	3	4	4	4	41.31
774	99024071	1	1	2	1	4	3	73.69
775	99024071	2	1	3	1	3	3	73.69
776	99024071	3	0	2	1	3	3	73.69
777	99024071	4	1	0	0	2	3	73.69
778	99024071	5	1	0	0	0	3	73.69
779	99024071	6	1	3	1	2	3	73.69
780	99024071	7	0	3	1	3	3	73.69
781	99024071	8	1	0	0	2	3	73.69
782	99024071	9	0	0	0	1	3	73.69
783	99024071	11	1	4	4	3	3	73.69
784	99024071	12	1	4	4	3	3	73.69
785	99024071	13	1	4	4	4	3	73.69
786	99024071	14	1	3	3	2	3	73.69
787	99024071	17	1	2	0	3	3	73.69
788	99024071	18	1	2	2	1	3	73.69
789	99024071	19	1	2	0	1	3	73.69
790	99024071	20	1	4	1	3	3	73.69
791	99024071	21	0	2	1	2	3	73.69
792	99024071	23	1	2	0	2	3	73.69
793	99024071	24	1	3	1	4	3	73.69
794	99024071	25	1	3	2	2	3	73.69
795	99024071	26	0	4	3	3	3	73.69
796	99024071	27	1	3	3	2	3	73.69
797	99024071	28	1	1	2	0	3	73.69
798	99024071	29	0	4	2	2	3	73.69
799	99024071	30	1	4	2	3	3	73.69
800	99024071	32	1	4	2	3	3	73.69
801	99024071	33	1	4	2	3	3	73.69
802	99024071	34	0	4	3	4	3	73.69
803	99024071	35	1	3	1	4	3	73.69
804	99024071	36	1	4	1	3	3	73.69
805	99024071	37	1	2	1	0	3	73.69
806	99024071	38	0	2	1	4	3	73.69
807	99024071	40	0	3	1	2	3	73.69
808	99024071	42	1	0	1	2	3	73.69
809	99024071	43	0	3	0	2	3	73.69
810	99024071	44	1	1	1	2	3	73.69
811	99024071	45	1	2	1	3	3	73.69
812	98061846	1	1	2	2	3	4	65.00
813	98061846	2	0	3	3	3	4	65.00
814	98061846	3	1	0	3	3	4	65.00
815	98061846	4	1	2	2	2	4	65.00
816	98061846	5	1	1	1	3	4	65.00
817	98061846	6	0	2	1	0	4	65.00
818	98061846	7	1	2	1	3	4	65.00
819	98061846	8	1	1	1	2	4	65.00
820	98061846	9	0	2	3	1	4	65.00
821	98061846	11	1	2	3	3	4	65.00
822	98061846	12	1	2	2	4	4	65.00
823	98061846	13	1	1	1	2	4	65.00
824	98061846	14	1	1	1	4	4	65.00

```
825    98061846    16    1    1    1    3    4 65.00
826    98061846    17    1    1    2    3    4 65.00
827    98061846    18    0    1    4    2    4 65.00
828    98061846    19    1    3    2    3    4 65.00
829    98061846    21    0    2    2    2    4 65.00
830    98061846    22    1    3    3    3    4 65.00
831    98061846    23    1    4    3    4    4 65.00
832    98061846    24    0    3    2    3    4 65.00
833    98061846    25    1    3    2    3    4 65.00
834    98061846    26    1    3    3    4    4 65.00
835    98061846    27    1    3    3    3    4 65.00
836    98061846    28    1    1    1    1    4 65.00
837    98061846    29    0    3    4    3    4 65.00
838    98061846    30    1    2    2    3    4 65.00
839    98061846    33    0    0    3    2    4 65.00
840    98061846    34    0    2    3    2    4 65.00
841    98061846    36    1    2    3    3    4 65.00
842    98061846    37    0    1    4    0    4 65.00
843    98061846    38    1    3    2    3    4 65.00
844    98061846    39    1    2    2    1    4 65.00
845    98061846    40    0    0    1    0    4 65.00
846    98061846    41    1    1    2    1    4 65.00
847    98061846    42    0    2    2    3    4 65.00
848    98061846    43    1    2    2    3    4 65.00
849    98061846    44    1    2    2    2    4 65.00
850    98061846    45    0    3    3    3    4 65.00
851    98061846    46    0    2    3    3    4 65.00
852    99042656     1    0    2    3    4    4 48.65
853    99042656     2    0    0    3    3    4 48.65
854    99042656     3    1    2    2    4    4 48.65
855    99042656     4    1    0    0    3    4 48.65
856    99042656     5    0    0    2    3    4 48.65
857    99042656     6    0    2    1    4    4 48.65
858    99042656     7    0    1    3    3    4 48.65
859    99042656     9    1    0    0    2    4 48.65
860    99042656    11    1    1    3    4    4 48.65
861    99042656    12    0    0    1    4    4 48.65
862    99042656    14    1    3    3    4    4 48.65
863    99042656    15    0    0    3    2    4 48.65
864    99042656    17    1    2    2    4    4 48.65
865    99042656    18    1    3    2    4    4 48.65
866    99042656    21    0    0    3    2    4 48.65
867    99042656    22    1    1    1    3    4 48.65
868    99042656    23    0    3    0    4    4 48.65
869    99042656    24    1    1    0    3    4 48.65
870    99042656    26    0    2    2    4    4 48.65
871    99042656    27    1    2    4    4    4 48.65
872    99042656    28    0    1    3    3    4 48.65
873    99042656    29    1    1    2    3    4 48.65
874    99042656    30    1    2    3    3    4 48.65
875    99042656    32    1    0    3    3    4 48.65
876    99042656    33    1    1    2    2    4 48.65
877    99042656    34    0    0    3    3    4 48.65
878    99042656    35    0    0    0    2    4 48.65
879    99042656    36    1    3    2    4    4 48.65
880    99042656    37    0    0    3    2    4 48.65
881    99042656    38    1    2    3    4    4 48.65
```

882	99042656	39	0	0	3	4	4 48.65
883	99042656	40	0	0	4	1	4 48.65
884	99042656	42	0	1	3	4	4 48.65
885	99042656	43	0	1	3	3	4 48.65
886	99042656	44	0	0	0	4	4 48.65
887	99042656	45	1	2	3	3	4 48.65
888	99042656	46	1	3	3	4	4 48.65
889	99031056	1	1	3	2	3	3 55.27
890	99031056	2	0	2	3	2	3 55.27
891	99031056	3	1	2	2	2	3 55.27
892	99031056	4	1	3	2	3	3 55.27
893	99031056	5	1	1	1	1	3 55.27
894	99031056	7	1	3	2	2	3 55.27
895	99031056	9	1	2	3	1	3 55.27
896	99031056	12	1	2	1	3	3 55.27
897	99031056	13	0	2	1	2	3 55.27
898	99031056	14	1	3	2	3	3 55.27
899	99031056	17	1	2	1	3	3 55.27
900	99031056	18	0	1	3	1	3 55.27
901	99031056	19	0	1	1	1	3 55.27
902	99031056	20	1	3	1	4	3 55.27
903	99031056	21	1	1	2	2	3 55.27
904	99031056	22	1	2	2	2	3 55.27
905	99031056	23	1	3	4	3	3 55.27
906	99031056	24	1	3	3	3	3 55.27
907	99031056	25	1	3	1	4	3 55.27
908	99031056	26	1	2	3	2	3 55.27
909	99031056	27	0	3	3	3	3 55.27
910	99031056	28	0	1	2	1	3 55.27
911	99031056	29	0	1	3	1	3 55.27
912	99031056	30	1	3	1	3	3 55.27
913	99031056	32	1	3	2	3	3 55.27
914	99031056	33	1	3	4	3	3 55.27
915	99031056	34	0	3	4	3	3 55.27
916	99031056	35	0	1	2	2	3 55.27
917	99031056	36	1	4	1	4	3 55.27
918	99031056	37	0	2	2	1	3 55.27
919	99031056	38	0	3	2	4	3 55.27
920	99031056	39	0	3	2	2	3 55.27
921	99031056	40	0	2	2	2	3 55.27
922	99031056	41	1	3	1	3	3 55.27
923	99031056	42	0	2	2	2	3 55.27
924	99031056	43	0	1	2	2	3 55.27
925	99031056	44	0	2	2	3	3 55.27
926	99031056	45	0	3	2	4	3 55.27
927	99029081	1	0	0	3	1	2 57.15
928	99029081	2	0	2	2	4	2 57.15
929	99029081	3	0	2	3	3	2 57.15
930	99029081	4	0	0	2	1	2 57.15
931	99029081	5	1	2	1	2	2 57.15
932	99029081	6	1	2	2	2	2 57.15
933	99029081	7	0	1	3	1	2 57.15
934	99029081	9	0	0	1	2	2 57.15
935	99029081	11	0	1	3	2	2 57.15
936	99029081	12	1	2	1	2	2 57.15
937	99029081	13	1	2	0	2	2 57.15
938	99029081	14	0	1	3	2	2 57.15

```
939    99029081   15   1   1   2   2    2 57.15
940    99029081   17   1   2   1   2    2 57.15
941    99029081   18   1   2   2   3    2 57.15
942    99029081   19   1   3   1   3    2 57.15
943    99029081   20   0   2   2   1    2 57.15
944    99029081   21   0   2   2   1    2 57.15
945    99029081   22   1   3   1   4    2 57.15
946    99029081   23   1   2   3   3    2 57.15
947    99029081   24   1   2   1   4    2 57.15
948    99029081   25   1   2   2   2    2 57.15
949    99029081   26   1   2   3   1    2 57.15
950    99029081   27   0   2   1   2    2 57.15
951    99029081   28   1   0   1   2    2 57.15
952    99029081   29   1   2   3   3    2 57.15
953    99029081   30   0   1   2   2    2 57.15
954    99029081   32   0   1   2   1    2 57.15
955    99029081   33   1   3   2   3    2 57.15
956    99029081   34   0   1   3   3    2 57.15
957    99029081   35   0   2   2   2    2 57.15
958    99029081   36   1   2   2   3    2 57.15
959    99029081   37   1   2   1   2    2 57.15
960    99029081   38   1   2   2   3    2 57.15
961    99029081   39   1   2   1   2    2 57.15
962    99029081   40   0   2   2   2    2 57.15
963    99029081   41   0   2   1   3    2 57.15
964    99029081   42   1   2   2   3    2 57.15
965    99029081   43   0   2   2   2    2 57.15
966    99029081   45   1   0   2   2    2 57.15
967    99029081   46   1   1   2   2    2 57.15
968    99050099    1   0   1   1   1    1 87.50
969    99050099    2   1   2   1   3    1 87.50
970    99050099    3   1   1   3   0    1 87.50
971    99050099    4   1   1   1   1    1 87.50
972    99050099    5   1   2   2   3    1 87.50
973    99050099    6   0   2   3   3    1 87.50
974    99050099    7   1   3   2   4    1 87.50
975    99050099    8   1   3   1   4    1 87.50
976    99050099    9   1   4   3   4    1 87.50
977    99050099   11   1   3   2   4    1 87.50
978    99050099   12   1   4   1   4    1 87.50
979    99050099   14   0   1   3   2    1 87.50
980    99050099   15   1   4   2   4    1 87.50
981    99050099   16   1   4   1   4    1 87.50
982    99050099   18   1   2   2   2    1 87.50
983    99050099   19   1   4   1   3    1 87.50
984    99050099   20   1   3   2   3    1 87.50
985    99050099   21   1   3   1   3    1 87.50
986    99050099   22   1   1   2   0    1 87.50
987    99050099   23   1   3   1   4    1 87.50
988    99050099   24   1   3   1   3    1 87.50
989    99050099   26   1   2   1   3    1 87.50
990    99050099   27   1   1   1   1    1 87.50
991    99050099   28   1   1   2   1    1 87.50
992    99050099   29   1   2   2   2    1 87.50
993    99050099   30   1   2   2   2    1 87.50
994    99050099   32   1   1   3   3    1 87.50
995    99050099   33   1   4   1   4    1 87.50
```

996	99050099	34	1	2	1	2	1 87.50
997	99050099	35	1	2	1	2	1 87.50
998	99050099	36	1	3	1	3	1 87.50
999	99050099	37	1	3	2	3	1 87.50
1000	99050099	38	1	4	2	4	1 87.50
1001	99050099	40	1	3	3	3	1 87.50
1002	99050099	41	1	3	1	2	1 87.50
1003	99050099	42	0	3	3	2	1 87.50
1004	99050099	43	0	3	3	2	1 87.50
1005	99050099	44	1	3	1	3	1 87.50
1006	99050099	45	1	3	1	4	1 87.50
1007	99050099	46	1	4	1	4	1 87.50
1008	99033052	1	1	3	2	4	1 74.36
1009	99033052	2	0	3	2	2	1 74.36
1010	99033052	3	1	4	1	2	1 74.36
1011	99033052	4	1	4	0	1	1 74.36
1012	99033052	5	1	3	0	4	1 74.36
1013	99033052	7	0	3	3	4	1 74.36
1014	99033052	8	1	3	0	4	1 74.36
1015	99033052	9	1	3	0	4	1 74.36
1016	99033052	11	1	2	2	4	1 74.36
1017	99033052	12	1	3	0	4	1 74.36
1018	99033052	13	1	3	0	4	1 74.36
1019	99033052	14	0	2	2	1	1 74.36
1020	99033052	15	1	3	0	4	1 74.36
1021	99033052	17	1	4	0	4	1 74.36
1022	99033052	18	0	1	4	3	1 74.36
1023	99033052	19	1	3	2	2	1 74.36
1024	99033052	20	1	2	2	2	1 74.36
1025	99033052	21	1	3	0	4	1 74.36
1026	99033052	23	1	3	2	4	1 74.36
1027	99033052	24	1	3	1	3	1 74.36
1028	99033052	25	1	3	0	4	1 74.36
1029	99033052	26	0	2	1	3	1 74.36
1030	99033052	27	0	2	4	1	1 74.36
1031	99033052	28	1	3	0	4	1 74.36
1032	99033052	29	1	3	1	3	1 74.36
1033	99033052	30	1	4	0	4	1 74.36
1034	99033052	32	1	3	1	0	1 74.36
1035	99033052	33	1	4	0	4	1 74.36
1036	99033052	34	0	3	3	4	1 74.36
1037	99033052	36	1	1	2	4	1 74.36
1038	99033052	37	1	2	2	4	1 74.36
1039	99033052	38	1	2	1	4	1 74.36
1040	99033052	39	1	3	0	4	1 74.36
1041	99033052	40	1	3	1	2	1 74.36
1042	99033052	41	1	2	0	4	1 74.36
1043	99033052	42	0	1	3	2	1 74.36
1044	99033052	43	0	1	3	1	1 74.36
1045	99033052	44	0	2	2	1	1 74.36
1046	99033052	45	1	3	1	4	1 74.36
1047	98066053	1	0	3	1	3	3 34.89
1048	98066053	2	1	3	1	3	3 34.89
1049	98066053	3	0	2	2	2	3 34.89
1050	98066053	4	1	2	2	2	3 34.89
1051	98066053	5	0	1	1	2	3 34.89
1052	98066053	6	1	3	1	3	3 34.89

1053	98066053	7	1	3	2	3	3	34.89
1054	98066053	8	1	2	4	3	3	34.89
1055	98066053	9	0	1	2	2	3	34.89
1056	98066053	11	0	2	3	3	3	34.89
1057	98066053	12	1	3	1	3	3	34.89
1058	98066053	13	1	3	0	2	3	34.89
1059	98066053	16	0	2	0	2	3	34.89
1060	98066053	17	0	3	2	3	3	34.89
1061	98066053	18	0	3	1	4	3	34.89
1062	98066053	19	1	4	0	3	3	34.89
1063	98066053	20	1	3	1	3	3	34.89
1064	98066053	21	0	2	2	2	3	34.89
1065	98066053	22	1	2	2	1	3	34.89
1066	98066053	23	0	3	1	2	3	34.89
1067	98066053	24	0	2	2	2	3	34.89
1068	98066053	25	1	4	1	3	3	34.89
1069	98066053	26	0	4	2	3	3	34.89
1070	98066053	27	1	3	2	3	3	34.89
1071	98066053	28	0	1	2	2	3	34.89
1072	98066053	29	0	3	2	3	3	34.89
1073	98066053	30	0	3	2	3	3	34.89
1074	98066053	32	0	1	2	2	3	34.89
1075	98066053	33	0	1	3	3	3	34.89
1076	98066053	34	0	2	2	3	3	34.89
1077	98066053	35	1	3	1	3	3	34.89
1078	98066053	36	1	3	3	3	3	34.89
1079	98066053	37	0	2	3	3	3	34.89
1080	98066053	38	1	3	2	3	3	34.89
1081	98066053	39	0	2	2	2	3	34.89
1082	98066053	40	0	1	2	2	3	34.89
1083	98066053	41	0	4	2	3	3	34.89
1084	98066053	42	0	2	2	2	3	34.89
1085	98066053	43	0	3	1	3	3	34.89
1086	98066053	44	0	2	2	2	3	34.89
1087	98066053	45	0	2	2	2	3	34.89
1088	98066053	46	0	2	2	2	3	34.89
1089	99032045	1	1	2	2	1	2	43.59
1090	99032045	2	0	3	2	3	2	43.59
1091	99032045	3	0	0	2	0	2	43.59
1092	99032045	6	1	4	2	3	2	43.59
1093	99032045	7	0	3	3	4	2	43.59
1094	99032045	8	1	0	0	0	2	43.59
1095	99032045	9	0	3	4	3	2	43.59
1096	99032045	11	0	1	4	0	2	43.59
1097	99032045	12	1	3	2	0	2	43.59
1098	99032045	13	1	1	2	0	2	43.59
1099	99032045	14	0	3	3	2	2	43.59
1100	99032045	15	0	3	4	3	2	43.59
1101	99032045	16	0	2	3	1	2	43.59
1102	99032045	17	0	2	3	1	2	43.59
1103	99032045	18	0	2	3	2	2	43.59
1104	99032045	19	0	3	2	3	2	43.59
1105	99032045	20	1	3	2	2	2	43.59
1106	99032045	21	0	2	3	1	2	43.59
1107	99032045	22	1	2	1	0	2	43.59
1108	99032045	23	0	4	4	0	2	43.59
1109	99032045	24	1	2	1	0	2	43.59

```
1110     99032045   26   1   3   3   3   2 43.59
1111     99032045   27   1   3   3   2   2 43.59
1112     99032045   29   0   4   4   4   2 43.59
1113     99032045   30   0   4   4   3   2 43.59
1114     99032045   32   1   3   2   0   2 43.59
1115     99032045   33   1   2   4   0   2 43.59
1116     99032045   34   0   2   4   0   2 43.59
1117     99032045   35   0   3   3   3   2 43.59
1118     99032045   36   1   3   1   1   2 43.59
1119     99032045   37   0   2   2   1   2 43.59
1120     99032045   38   1   3   2   2   2 43.59
1121     99032045   39   1   2   2   0   2 43.59
1122     99032045   40   0   0   3   0   2 43.59
1123     99032045   42   1   3   2   2   2 43.59
1124     99032045   43   0   3   2   4   2 43.59
1125     99032045   44   1   2   3   3   2 43.59
1126     99032045   45   0   0   4   0   2 43.59
1127     99032045   46   0   1   3   0   2 43.59
```

It is easy to do a rather simple-minded analysis of the data thus:

```
data.RB = read.table("data.RB", header=T)
attach(data.RB)
Item = factor(item)
first.glm = glm(y~ int+ diff+ rel + Item, binomial)
summary(first.glm)
```

showing, for example, that P(correct response) decreases with difficulty, as we might expect. There is possibly an interaction between the terms 'int' and 'diff', so we also try

```
intdif = int*diff
next.glm= glm(y~ int+ diff+ rel + intdif + Item, binomial)
```

However, this simple analysis fails to allow for the fact that for any given subject(=id) we have repeated observations. We use GEE (generalized linear models for dependent data) to allow for this dependence. GEE allows several possibilities for the correlation structure for observations on the same id: here the symmetric or "exchangeable" correlation structure seemed to make the most sense.

```
library(gee)
first.gee = gee(y~ int+ diff+ rel + intdif + Item, id=id, family=binomial,
corstr="exchangeable")
summary(first.gee)

 GEE:   GENERALIZED LINEAR MODELS FOR DEPENDENT DATA
 gee S-function, version 4.13 modified 98/01/27 (1998)

Model:
 Link:           Logit
 Variance to Mean Relation: Binomial
 Correlation Structure:    Exchangeable
Call:
gee(formula = y ~ int + diff + rel + intdif + Item, id = id,
    family = binomial, corstr = "exchangeable")

Summary of Residuals:
        Min           1Q       Median           3Q          Max
-0.97977860 -0.35293189 -0.07409786   0.37101927   0.93037501
```

```
Coefficients:
              Estimate Naive S.E.      Naive z Robust S.E.     Robust z
(Intercept)  1.04651203 0.56367424  1.85659014  0.66920602  1.56381145
int          0.05290849 0.12721274  0.41590555  0.16664054  0.31750069
diff        -0.63719390 0.13131320 -4.85247411  0.11111541 -5.73452341
rel          0.25935821 0.06846523  3.78817428  0.06888366  3.76516299
intdif       0.09117772 0.04811922  1.89482954  0.05776439  1.57844178
Item2       -2.12353541 0.58919062 -3.60415682  0.61580870 -3.44836863
Item3       -1.60457927 0.59178711 -2.71141299  0.63752687 -2.51688099
Item4        0.67227515 0.69304014  0.97003783  0.71076962  0.94584115
Item5       -0.05051358 0.61105810 -0.08266576  0.59697950 -0.08461527
Item6       -1.41348532 0.59827969 -2.36258283  0.61234697 -2.30830784
Item7       -1.22023473 0.57294908 -2.12974376  0.51610458 -2.36431682
Item8        2.37620560 1.14228412  2.08022292  1.31834567  1.80241469
Item9       -1.59714005 0.61248758 -2.60762846  0.63174671 -2.52813356
Item11      -0.86053608 0.60210189 -1.42922003  0.53696247 -1.60260005
Item12       0.15666588 0.62206226  0.25184919  0.49567156  0.31606792
Item13       1.88333671 1.20970636  1.55685444  1.46330374  1.28704428
Item14      -0.62466393 0.58473354 -1.06828818  0.51959403 -1.20221538
Item15      -0.79635129 0.62743127 -1.26922475  0.62727073 -1.26954958
Item16      -1.00351791 0.60786778 -1.65088189  0.51224007 -1.95907731
Item17      -1.13536537 0.59534863 -1.90705968  0.55405975 -2.04917496
Item18      -1.62325778 0.59245051 -2.73990442  0.59502491 -2.72805012
Item19      -1.55958697 0.58131975 -2.68283844  0.64915323 -2.40249435
Item20       0.19048107 0.64993477  0.29307721  0.65373417  0.29137389
Item21      -1.78990738 0.61526946 -2.90914388  0.51731809 -3.45997445
Item22       0.06679524 0.60778493  0.10989947  0.53346648  0.12520982
Item23      -0.51638764 0.57748013 -0.89420850  0.40166595 -1.28561467
Item24      -0.63402813 0.57494816 -1.10275704  0.57524629 -1.10218551
Item25       1.97870643 1.13100218  1.74951602  1.10716698  1.78717977
Item26      -1.18263118 0.57259040 -2.06540517  0.59570855 -1.98525131
Item27      -0.48542943 0.56718574 -0.85585619  0.58668256 -0.82741411
Item28       0.15535457 0.59127175  0.26274648  0.54025995  0.28755523
Item29      -1.79948383 0.60695465 -2.96477477  0.69219207 -2.59968861
Item30      -1.65259386 0.58159700 -2.84147589  0.61997454 -2.66558342
Item32      -0.54146773 0.57914603 -0.93494163  0.46782749 -1.15740898
Item33      -0.55546596 0.58585654 -0.94812624  0.63555286 -0.87398860
Item34      -1.85835707 0.60366346 -3.07846537  0.59690067 -3.11334394
Item35      -0.71756744 0.58770289 -1.22096973  0.52825988 -1.35836064
Item36      -0.38239264 0.59052338 -0.64754869  0.57865355 -0.66083175
Item37      -1.01061935 0.57040134 -1.77176888  0.54820320 -1.84351230
Item38      -1.42172418 0.56313801 -2.52464609  0.62076912 -2.29026241
Item39      -1.60327397 0.62962473 -2.54639612  0.56144088 -2.85564166
Item40      -1.76717825 0.62111229 -2.84518320  0.67133137 -2.63234867
Item41      -1.35568057 0.61811797 -2.19323920  0.53842086 -2.51788267
Item42      -1.72702558 0.59055074 -2.92443217  0.55396678 -3.11756163
Item43      -1.76212821 0.60221782 -2.92606453  0.54223136 -3.24977183
Item44      -1.03071019 0.57401780 -1.79560668  0.60385833 -1.70687417
Item45      -1.08658147 0.58352706 -1.86209269  0.69032967 -1.57400372
Item46      -1.71396426 0.62319290 -2.75029490  0.70800230 -2.42084562

Estimated Scale Parameter:  1.028398
Number of Iterations:  5
Working Correlation
 [1,] 1.0000000 0.1223918 0.1223918 0.1223918 ...
 [2,] 0.1223918 1.0000000 0.1223918 0.1223918 ...
 [3,] 0.1223918 0.1223918 1.0000000 0.1223918 ...
```

```
[4,] 0.1223918 0.1223918 0.1223918 1.0000000 ...
[5,] 0.1223918 0.1223918 0.1223918 0.1223918 ...
   ..................................etc...................................
```

where we have NOT given every element of this 46×46 correlation matrix, since it clearly has all diagonal elements 1, and all off-diagonal elements $= 0.1223918$. This is indicating only modest positive dependence between responses for the same subject. You will also see that our 'Estimated Scale Parameter' is only 1.028398: this corresponds to the estimate of the factor ϕ in the formula $var(y) = \phi\pi(1 - \pi)$, where $E(y) = \pi$.

```
interaction.plot(diff,int, y)
table(diff,int, y)
```

This is a pretty good way to explain the interaction between 'interest' and 'difficulty': can you put this into words?

Reference Hardin, J.W. and Hilbe, J.M. (2003) *Generalized Estimating Equations.* Chapman and Hall/CRC

Index