STATISTICS: AN INTRODUCTION USING R

By M.J. Crawley

Exercises

11. ANALYSING COUNT DATA: POISSON ERRORS

Up to Practical 9, the data were all continuous measurements like weights, heights, lengths, temperatures, growth rates and so on. A great deal of the data collected by scientists, medical statisticians and economists, however, is in the form of *counts* (whole numbers or integers). The number of individuals that died, the number of firms going bankrupt, the number of days of frost, the number of red blood cells on a microscope slide, or the number of craters in a sector of lunar landscape are all potentially interesting variables for study. With count data, the number 0 often appears as a value of the response variable (consider, for example, what a 0 would mean in the context of the examples just listed).

For our present purposes, it is useful to think of count data as coming in four types:

- data on *frequencies*, where we count how many times something happened, but we have no way of knowing how often it did not happen (e.g. lightening strikes, bankruptcies, deaths, births)
- data on *proportions*, where both the number doing a particular thing, and the total group size are known (insects dying in an insecticide bioassay, sex ratios at birth, proportions responding in a questionnaire)
- *category* data, in which the response variable is a count distributed across a categorical variable with two or more levels
- *binary* response variables (dead or alive, solvent or insolvent, infected or immune)

We dealt with proportion data and binary response variables in Practical 10. Here we are concerned with pure counts rather than proportions. Straightforward linear regression methods (constant variance, normal errors) are not appropriate for count data for 5 main reasons:

- the linear model might lead to the prediction of negative counts
- the variance of the response variable is likely to increase with the mean
- the errors will not be normally distributed
- zeros are difficult to handle in transformations
- some distributions (e.g. log-normal or gamma) don't allow zeros

In S-Plus, count data are handled very elegantly in a **glm** by specifying **family=poisson** which sets errors = Poisson and link = log. The log link ensures that

all the fitted values are positive, while the Poisson errors take account of the fact that the data are integer and have variances equal to their means.

The Poisson distribution

The Poisson distribution is widely used for the description of count data that refer to cases where we know how many times something happened (e.g. kicks from cavalry horses, lightening strikes, bomb hits), but we have no way of knowing how many times it did not happen. This is in contrast to the binomial distribution (Practical 10) where we know how many times something did not happen as well as how often it did happen (e.g. if we got 6 heads out of 10 tosses of a coin, we must have got 4 tails).

The Poisson is a 1-parameter distribution, specified entirely by the mean. The variance is identical to the mean, so the variance/mean ratio is equal to one. Suppose we are studying the ecology of a leaf miner on birch trees, and our data consist of the numbers of mines per leaf (*x*). Many leaves have no mines at all, but some may have as many as 5 or 6 mines. If the mean number of mines per leaf is λ, then the probability of observing *x* mines per leaf is given by:

$$P(x) = \frac{e^{-\lambda}\lambda^{x}}{x!}$$

This can be calculated very simply on a hand calculator because:

$$P(x) = P(x-1)\frac{\lambda}{x}$$

This means that if you start with the *zero term*

$$P(0) = e^{-\lambda}$$

then each successive probability is obtained simply by multiplying by the mean and dividing by *x*.

Because the data are whole numbers (integers), it means that the residuals (y - the fitted values) can only take a restricted range of values. If the estimated mean was 0.5, for example, then the residuals for counts of 0, 1, 2, and 3 could only be -0.5, 0.5, 1.5 and 2.5. The normal distribution assumes that the residuals can take any values (it is a continuous distribution).

Similarly, the normal distribution allows for negative fitted values. Since we can not have negative counts, this is clearly not appropriate. SPlus deals with this by using the log link function when Poisson errors are specified

$$y = \exp(\sum \beta_i x_i)$$

so the fitted values are antilogs and can never go negative. Even if the linear predictor was $-\infty$ the fitted value would be exp(-∞) = 0 .

A further difference from the examples in previous chapters is that SPlus uses maximum likelihood methods other than least squares to estimate the parameters values and their standard errors. Given a set of data and a particular model, then the maximum likelihood estimates of the parameter values are *those values that make the observed data most likely* (hence the name maximum likelihood).

Because the Poisson is a one-parameter distribution (the mean is equal to the variance), SPlus does not attempt to estimate a scale parameter (it is set to a default value of 1.0). If the error structure of the data really is Poisson, then the ratio of residual deviance to degrees of freedom after model-fitting should be 1.0. If the ratio is substantially greater than one, then the data are said to show overdispersion, and remedial measures may need to be taken (e.g. the use of an empirical scale parameter or the specification of negative binomial errors).

With Poisson errors, the change in deviance attributable to a given factor is distributed asymptotically as chi-squared. This makes hypothesis testing extremely straightforward. We simply remove a given factor from the maximal model and note the resulting change in deviance and in the degrees of freedom. If the change in deviance is larger than the critical value of chi squared (**qchisq**) we retain the term in the model, while if the change in deviance is less than the value of chi squared in tables, the factor is insignificant and can be left out of the model.

Thus, the only important differences you will need to remember in modelling with Poisson rather than normal errors are:

- use **glm** rather than **aov** or **lm**
- the **family=poisson** directive must be specified (but the "family =" bit is optional)
- hypothesis testing involves deletion followed by chi-squared tests
- beware of overdispersion, and correct for it if necessary
- do not collapse contingency tables over explanatory variables

Deviance with Poisson errors

Up to this point, lack of fit has always been measured by SSE; the residual or error sum of squares

$$\text{SSE} = \sum (y - \hat{y})^2$$

where $\hat{y}$ are the fitted values estimated by the model. With **glm**'s SSE is only the maximum likelihood estimate of lack of fit when the model has normal errors and the identity link (in which case, you have to ask "why am I doing a glm?"). Generally, we use *deviance* to measure lack of fit in a **glm**. For Poisson errors the deviance is this:

$$\text{Poisson deviance} = 2\sum O \ln\left[\frac{O}{E}\right]$$

where O is the Observed count, and E is the Expected count as predicted by the current model. Let's see how this works. Suppose you have 4 counts, 2 from each of 2 locations. Location A produced counts of 3 and 6. Location B produced counts of 4

and 7. The total deviance (like SST) is based on the whole sample of 4 numbers. The expected count is the overall mean

```
mean(c(3,6,4,7))
```

```
[1] 5
```

This allows us to calculate the total deviance:

$$2 \times \left\{ 3 \times \ln\left[\frac{3}{5}\right] + 6 \times \ln\left[\frac{6}{5}\right] + 4 \times \ln\left[\frac{4}{5}\right] + 7 \times \ln\left[\frac{7}{5}\right] \right\}$$

Calculating the value gives

```
2*(3*log(3/5)+6*log(6/5)+4*log(4/5)+7*log(7/5))
```

```
[1] 2.048368
```

Now the 2 different location means are 9/2 = 4.5 and 11/2 =5.5 respectively. The residual deviance after fitting a 2–level factor for location should therefore be

$$2 \times \left\{ 3 \times \ln\left[\frac{3}{4.5}\right] + 6 \times \ln\left[\frac{6}{4.5}\right] + 4 \times \ln\left[\frac{4}{5.5}\right] + 7 \times \ln\left[\frac{7}{5.5}\right] \right\}$$

```
2*(3*log(3/4.5)+6*log(6/4.5)+4*log(4/5.5)+7*log(7/5.5))
```

```
[1] 1.848033
```

Given that the total deviance is 2.048 and the residual deviance is 1.848 we calculate the treatment deviance (due to differences between locations) as 2.048 – 1.848 = 0.2. Let's see if a **glm** with Poisson errors gives the same answers. Data entry first

```
y<-c(3,6,4,7)
```

```
location<-factor(c("A","A","B","B"))
```

now the statistical modelling

```
glm(y~location,poisson)
```

```
Degrees of Freedom: 3 Total (i.e. Null);  2 Residual
Null Deviance:      2.048
Residual Deviance: 1.848        AIC: 19.57
```

So far so good. The residual deviance (1.848) is as we calculated it to be. How about the total and treatment deviances ? We can get the total deviance if we fit the null model y ~ 1

```
glm(y~1,family=poisson)
```

```
Degrees of Freedom: 3 Total (i.e. Null);  3 Residual
Null Deviance:      2.048
Residual Deviance: 2.048        AIC: 17.77
```

So there is no mystery to the values of deviance. We could calculate them if we wanted to, at least for models as simple as this (more complex models require *iterative fits* to estimate the parameter values).

Using Poisson errors in modelling

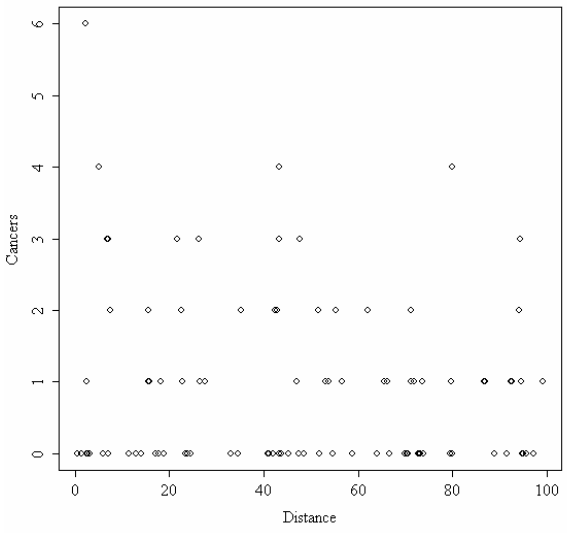
1) Continuous explanatory variables with count data: log-linear regression

This example involves counts of prostate cancer (cancer 'clusters') and distance from a nuclear processing plant. The response variable contains many zeros and is completely unsuitable for standard regression analysis. The single explanatory variable is continuous: distance to the nuclear plant from the clinic where the diagnosis was made (not from where the patients actually lived or worked). The issue is whether or not the data provide any evidence that the number of cancers increases with proximity to the plant.

```
clusters<-read.table("c:\\temp\\clusters.txt",header=T)
attach(clusters)
names(clusters)
```

```
[1] "Cancers"  "Distance"
```

```
plot(Distance,Cancers)
```



The first thing that you notice about plots of count data is that all the data are in sharp horizontal rows reflecting the integer values of the response. There are lots of zero's at all distances away from the nuclear plant. The largest value (the cluster of 6 cases) is close to the plant. But is there evidence for any trend in the number of cases as distance from the plant increases ?

The modelling is exactly like any other regression except that we replace **lm** by **glm**, and add the phrase family=poisson after the model formula (family= is optional):

```
model<-glm(Cancers~Distance,family=poisson)
summary(model)
```

```
Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)  0.186865   0.188728   0.990   0.3221
Distance    -0.006138   0.003667  -1.674   0.0941 .
```

There is a negative trend in the data (slope = -0.00614) but it is not significant under a 2-tailed test (t = 1.68). If you were desperate, you might say that a 1-tailed test is appropriate, because we expected *a priori* that the slope would be negative. I don't buy that. Anyway, we are not finished yet. Poisson errors is an assumption not a fact.

```
(Dispersion parameter for poisson family taken to be 1 )
```

We need to test for overdispersion. This would be evidenced if the residual deviance was larger than the residual degrees of freedom:

```
Null deviance: 149.48  on 93  degrees of freedom
Residual deviance: 146.64  on 92  degrees of freedom
AIC: 262.41
```

The dispersion parameter is actually 146.64 / 92 = 1.594 and we should make allowance for this overdispersion in our analysis. As a rule of thumb, the standard error of the parameters will increase as $\sqrt{\text{Dispersion parameter}}$ (i.e. they will be 1.26–fold larger in this case). A better way to adjust for overdispersion is to **use an F test with an empirical scale parameter instead of a chi square test**. This is carried out in R by using the family **quasipoisson** in place of poisson errors. We can accomplish this by deleting distance from the model (fitting only the intercept, ~1), then comparing the model fits using **anova** in which we specify test="F"

```
model<-glm(Cancers~Distance,family=quasipoisson)
model2<-glm(Cancers~1,family=quasipoisson)
anova(model,model2,test="F")
```

```
Analysis of Deviance Table

Model 1: Cancers ~ Distance
Model 2: Cancers ~ 1
  Resid. Df Resid. Dev Df Deviance      F Pr(>F)
1        92    146.643
2        93    149.484 -1   -2.841 1.8269 0.1798
```

There is no evidence for a decline in the number of cancers with distance. An F value as large as 1.83 will arise by chance alone with probability p = 0.18 when there is no trend in cancers with distance.

Analysis of deviance: categorical explanatory variables with count data

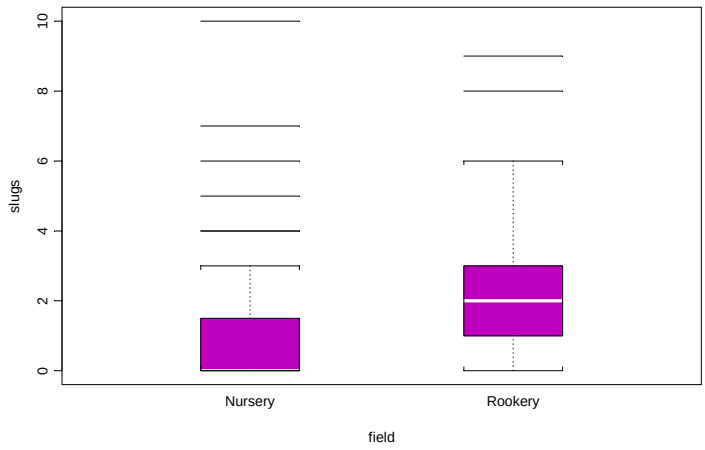
Count data were obtained on the numbers of slugs found beneath 40 tiles placed in a stratified random grid over each of two permanent grasslands. This was part of a pilot study in which the question was simply whether the mean slug density differs significantly between the two grasslands.

```
slugsurvey<-read.table("c:\\temp\\slugsurvey.txt",header=T)
attach(slugsurvey)
names(slugsurvey)
```

```
[1] "slugs" "field"
```

These count data seem ideally suited for analysis using **glm** with Poisson errors.

```
plot(field,slugs)
```



It certainly looks as if median slug numbers are higher in Rookery Field than in Nursery Field, but the range of counts is very large in both fields, so the significance of the difference must be in some doubt. We fit the model first as a simple 1-way analysis of deviance, using Poisson errors and the log link:

```
model<-glm(slugs~field,poisson)
```

```
summary(model)
```

```
Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)    0.2429     0.1399   1.737 0.082446 .
fieldRookery   0.5790     0.1748   3.312 0.000925 ***
```

The t-test for difference between fields is highly significant, so it looks like mean slug numbers really are different. But just a minute. We have not yet checked for overdispersion.

```
(Dispersion parameter for poisson family taken to be 1)
```

```
    Null deviance: 224.86  on 79  degrees of freedom
Residual deviance: 213.44  on 78  degrees of freedom
AIC: 346.26
```

Oops! The scale parameter (or Dispersion Parameter as it is labelled here) is 213.44/78 = 2.74, not 1 as was assumed by the model. The simplest thing we can do to compensate for this is to carry out an F test rather than a chi square test of deletion. The F test uses the empirical scale parameter as an estimate equivalent to the error variance, and performs a test much harsher than the chi square test. We can compare the two: chi square first:

```
model2<-update(model,~.-field)

anova(model,model2,test="Chi")
```

```
Analysis of Deviance Table

Model 1: slugs ~ field
Model 2: slugs ~ 1
  Resid. Df Resid. Dev Df Deviance P(>|Chi|)
1        78    213.438
2        79    224.859 -1  -11.422     0.001
```

This suggests that the difference between the fields is highly significant (p = 0.001). Now do exactly the same deletion, but use the F test with the empirical scale parameter. This requires that we re-fit the model using **family = quasipoisson**:

```
model<-glm(slugs~field,quasipoisson)
model2<-update(model,~.-field)
anova(model,model2,test="F")
```

```
Analysis of Deviance Table

Model 1: slugs ~ field
Model 2: slugs ~ 1

  Resid. Df Resid. Dev Df Deviance      F  Pr(>F)
1        78    213.438
2        79    224.859 -1  -11.422 3.6041 0.06134 .
```

A big change. Under the F test the difference in mean slug density is **not** significant.

We could try a parametric transformation, and analyse the data using a linear model (**aov**). For instance,

```
model3<-aov(log(slugs+1)~field)

summary(model3)
```

```
            Df Sum of Sq  Mean Sq  F Value    Pr(F)
```

```
    field  1    4.4750 4.475004 8.961176 0.003692791
Residuals 78   38.9514 0.499377
```

This says the difference is highly significant ($p < 0.004$), so clearly the adjustment for overdispersion in the F test is extremely unforgiving.

The alternative parametric transformation for count data is the square root. Here we compare a straightforward analysis of variance on the raw count data (normal errors and constant variance (wrongly) assumed) with a model based on the square root transformation.

```
model4<-aov(slugs~field)

anova(model4)

Analysis of Variance Table

Response: slugs

Terms added sequentially (first to last)
          Df Sum of Sq  Mean Sq  F Value      Pr(F)
    field  1     20.00 20.00000 3.900488 0.05181051
Residuals 78    399.95  5.12756
```

The untransformed anova suggests that the difference is not quite significant. Now for a square root transformation of the counts, assuming (now with much greater justification) that the errors are normal and the variance constant:

```
model5<-aov(slugs^0.5~field)

anova(model5)

Analysis of Variance Table
Response: slugs^0.5

          Df Sum of Sq  Mean Sq  F Value       Pr(F)
    field  1   7.44812 7.448123 9.312308 0.003110717
Residuals 78  62.38557 0.799815
```

So the linear model with square root transformed counts indicates that the difference in slug density between the two fields is highly significant.

The next option in dealing with overdispersion is to use **quasi** to define a different family of error structures. For instance, we might retain the log link (this is good for constraining the predicted counts to be non-negative), but we might allow that the variance increases with the square of the mean (like a discrete version of a gamma distribution), rather than as the mean (as assumed by Poisson errors). This is how we write the model :

```
model6<-glm(slugs~field,family=quasi(link=log,variance=mu^2))
```

The software recognises “mu” as the mean of the distribution. The result of the fit is this:

```
summary(model6)
```

```
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)    0.2429     0.2300   1.056   0.2941
fieldRookery   0.5790     0.3252   1.780   0.0789 .

Dispersion parameter for quasi family taken to be 2.115615)
```

Note the estimate of the Dispersion Parameter, above. The t-test does not indicate a significant difference between fields. Model checking plots indicate that the errors are much more nearly normal under the log(slugs+1) transformation than in the quasi model.

```
    Null deviance: 40.144  on 79  degrees of freedom
Residual deviance: 45.606  on 78  degrees of freedom
AIC: NA
```

Let’s check the deletion test.

```
model7<-glm(slugs~1,family=quasi(link=log,variance=mu^2))
anova(model6,model7,test="F")
```

```
Analysis of Deviance Table

Model 1: slugs ~ field
Model 2: slugs ~ 1
  Resid. Df Resid. Dev Df Deviance      F Pr(>F)
1        78     45.606
2        79     40.144 -1    5.462 2.5815 0.1122
```

Not significant. What about a non-parametric test? We can use the Wilcoxon rank sum test to compare the two (unpaired) samples. Load the “classic tests” library:

```
library(ctest)
```

```
wilcox.test(slugs[field=="Nursery"],slugs[field=="Rookery"])
```

Notice that we need to split the response variable into two separate vectors, one for each field, in order to use this function (in general, the 2 vectors might be of different lengths).

```
Wilcoxon rank-sum test

data:slugs[field=="Nursery"]and slugs[field=="Rookery"]

rank-sum normal statistic with correction Z = -3.0581,
p-value = 0.0022
alternative hypothesis: true mu is not equal to 0
```

The non-parametric test says that mean slug numbers are significantly different in the 2 fields ($p = 0.0022$). **This p value is not particularly reliable, however, because there are so many ties in the data** (all those zeros).

Thus, several of the tests suggest that the differences between the mean slug densities are highly significant, while other tests suggest that the difference is insignificant. What is clear is that there is more to this than mere differences between the fields. Differences between the means explain only 5% of the variation in slug counts from tile to tile (deviance change of 11.42 out of 224.86). Within fields it is clear that the data are highly aggregated. It is very common for field data to have this kind of overdispersed, spatially aggregated structure, and it would be naive of us to assume that simple error structures will always work perfectly.

This kind of problem, where one kind of test says a difference is significant and another test says the same difference is not significant, comes up all the time when dealing with data with a low mean and a high variance. There is no obvious right answer, but the analyses correcting for overdispersion are clearly signalling that the result, so significant with linear models, should be treated with a more than usual degree of circumspection.

Summary of Overdispersion with Poisson errors

Overdispersion occurs when the residual deviance is greater than the residual degrees of freedom once the minimal adequate model has been fit to the data. It means that the errors are not, in fact, Poisson (i.e. equal to the mean) but are actually greater than assumed. This means that the estimated standard errors are too small, and the significance of model terms is more or less severely overestimated. We need to take care of overdispersion, and there are several options we can follow. In order of simplicity, these are

1) carry out significance tests using "**F**" rather than "**Chi**" in the **anova** directive after specifying family = **quasipoisson** rather than poisson

2) use **family = quasi** instead of family = poisson and specify a variance function

3) use negative binomial errors

ANCOVA with count data: categorical and continuous explanatory variables.

A long-term agricultural experiment had 90 grassland plots, each 25m x 25m, differing in biomass, soil pH and species richness (the count of species in the whole plot). It is well known that species richness declines with increasing biomass, but the question addressed here is whether the slope of that relationship differs with soil pH. The plots were classified according to a 3-level factor as high, medium or low pH with 30 plots in each level. The response variable is the count of species, so a **glm** with Poisson errors is a sensible choice. The continuous explanatory variable is long-term average biomass measured in June, and the categorical explanatory variable is soil pH. With a mixture of continuous and categorical explanatory variables, analysis of covariance is the appropriate method.

```
species<-read.table("c:\\temp\\species.txt",header=T)
attach(species)
names(species)
```

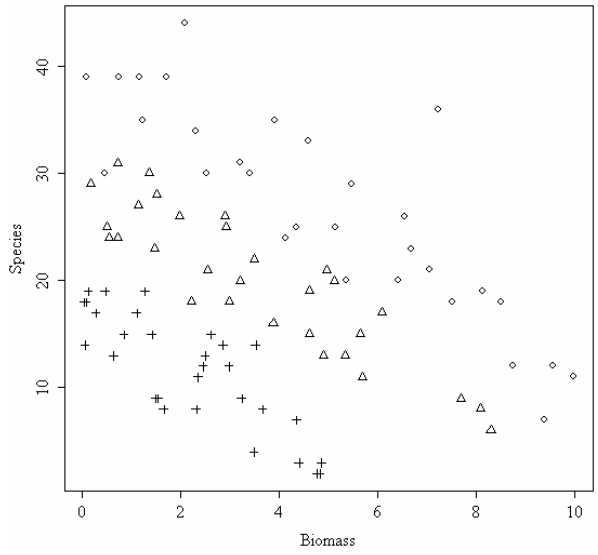
```
[1] "pH"      "Biomass" "Species"
```

We begin by plotting the data, using different symbols for each level of soil pH. It is important in cases like this to ensure that the axes in the first plot directive are scaled appropriately to accommodate all of the data. The trick here is to plot the axes with nothing between them: the **type="n"** option:

```
plot(Biomass,Species,type="n")
```

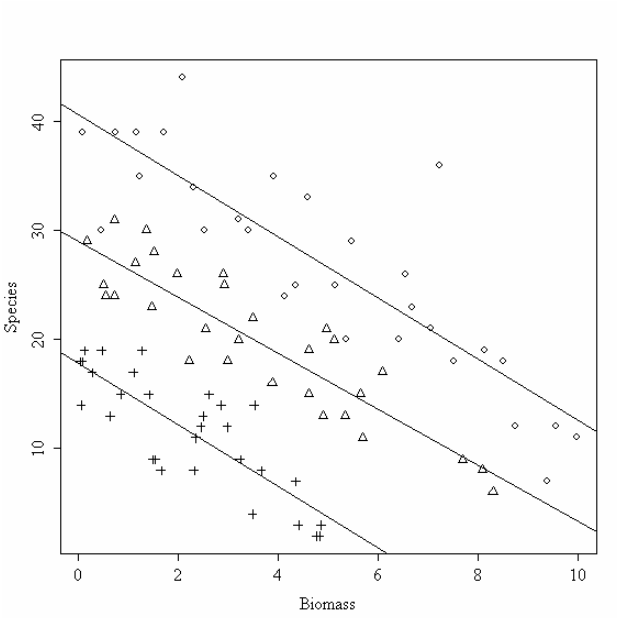
Now we can add a scatterplot of points for each level of soil pH, using different plotting characters **pch** for each:

```
points(Biomass[pH=="high"],Species[pH=="high"])
points(Biomass[pH=="mid"],Species[pH=="mid"],pch=2)
points(Biomass[pH=="low"],Species[pH=="low"],pch=3)
```



To help interpret the plot, we can fit linear regression lines through the clouds of points for each level of soil pH using **abline**. We shall not use linear regression in the statistical modelling, if only because a linear model would predict nonsense values (e.g. negative counts of species at high biomass). Note that the order of the variables is reversed in the lm directive (y then x) and that the variables are separated by ~ (tilde).

```
abline(lm(Species[pH=="high"]~Biomass[pH=="high"]))
abline(lm(Species[pH=="mid"]~Biomass[pH=="mid"]))
abline(lm(Species[pH=="low"]~Biomass[pH=="low"]))
```



There is certainly a clear difference in mean species richness with declining soil pH, but there is little evidence of any substantial difference in the slope of the relationship between species richness and biomass on soils of differing pH. Now we shall do the modelling properly, as a **log-linear analysis of covariance.**

The procedure is exactly the same as in standard analysis of covariance: we fit a full model with different slopes and intercepts for each factor level, then we simplify the model by assessing whether a common slope would describe the data equally well. The only difference is that we replace **lm** by **glm** and add the phrase family=poisson after the model formula:

```
model1<-glm(Species~pH*Biomass,poisson)
model2<-update(model1,~.-pH:Biomass)
```

Don’t forget the punctuation in update: comma tilde dot minus. Now we can get a full anova table comparing the 2 different models, with and without the interaction term (differences between slopes):

```
anova(model1,model2,test="Chi")
```

```
Analysis of Deviance Table

Model 1: Species ~ pH * Biomass
Model 2: Species ~ pH + Biomass
  Resid. Df Resid. Dev Df Deviance P(>|Chi|)
1        84     83.201
2        86     99.242 -2  -16.040 0.0003288
```

Our initial impression that the slopes were the same was completely wrong: in fact, the slopes are very significantly different ($p < 0.00035$). So we need to retain a model with different slopes. Note that the model containing different slopes (Model 1) is not overdispersed (residual deviance = 83.2 on 84 d.f.), so we do not need to correct by using quasipoisson.

Plotting the fitted values as smooth lines, separately for each level of soil pH, demonstrates the boundedness of log linear models. Our first plot, using linear regression, absurdly predicted negative values for species richness at high biomass. In contrast, the **glm** predicts species richness declining asymptotically towards low values. In reality of course, it is impossible that plant species richness could ever fall below 1 so long as the plots were vegetated.

```
plot(Biomass,Species,type="n")
```

Now we can add a scatterplot of points for each level of soil pH, using different plotting characters pch for each:

```
points(Biomass[pH=="high"],Species[pH=="high"])
points(Biomass[pH=="mid"],Species[pH=="mid"],pch=2)
points(Biomass[pH=="low"],Species[pH=="low"],pch=3)
```

The curve fitting requires that we make a data frame to contain values for all of the explanatory variables (the continuous variable Biomass for the x axis and the categorical explanatory variable pH for the 3 different graphs). First we generate the values for the x axis

```
x<-seq(0,10,0.1)
length(x)
```

```
[1] 101
```

This means that we need to generate101 repeats of each of the factor levels for soil pH

```
levels(pH)
```

```
[1] "high" "low"  "mid"
```

We created a new vector of length 3 x 101 to contain the factor levels:

```
acid<-factor(c(rep("low",101),rep("mid",101),rep("high",101)))
```

and now we need to make the vector of x values (Biomass) the same length

```
x<-c(x,x,x)
```

Finally, we use **predict** to predict the fitted values. We could back transform the predicted values manually (using the antilog function exp) but here we use type="response" to do this for us.

```
yv<-predict(model1,list(Biomass=x,pH=acid),type="response")
```

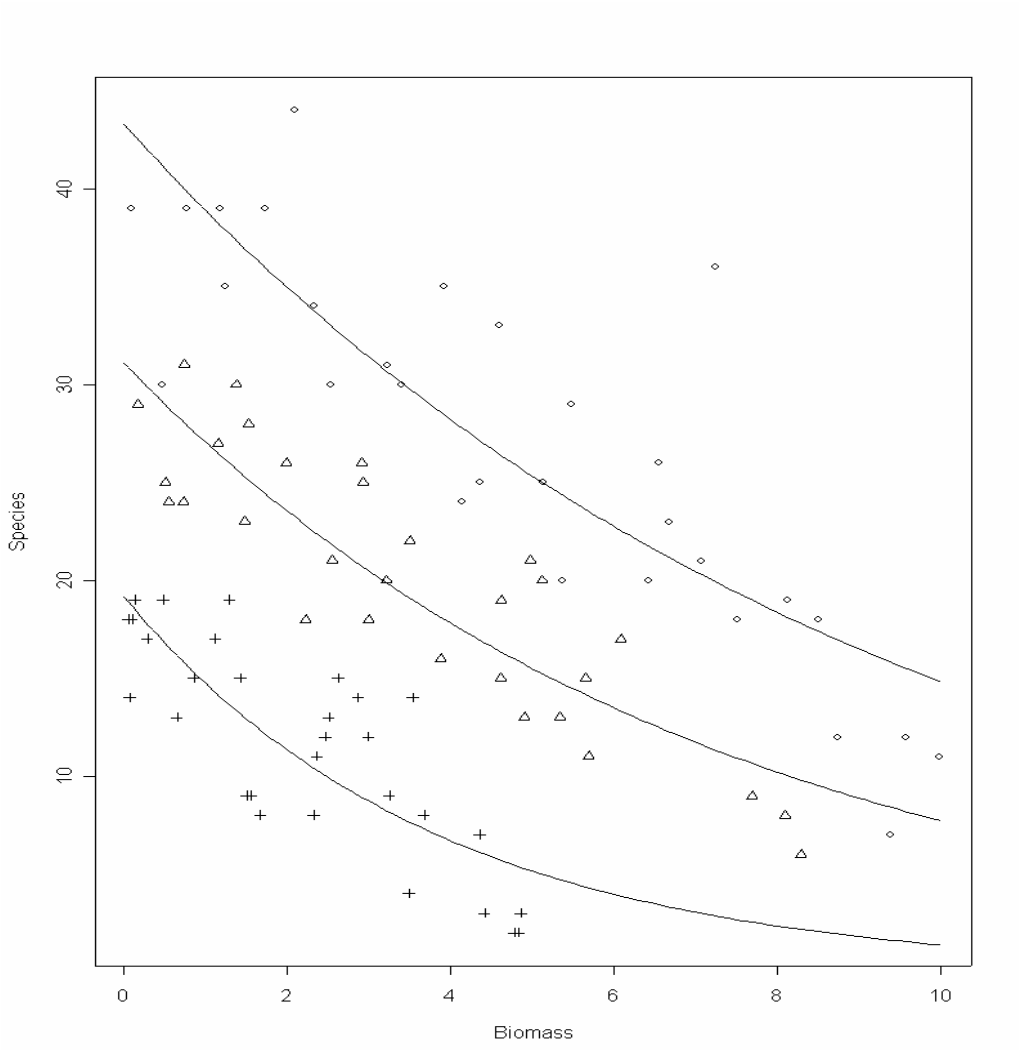
With complicated plots like this, where we want to fit different curves for different levels of one or more factors, it is useful to employ the split function to reduce the y-values and x-values into separate lists (one for each fitted line)

```
yvs<-split(yv,acid)
bvs<-split(x,acid)
```

Now we can fit the three lines. Note the use of double subscripts, [[1]], because the split function produces **lists** not vectors:

```
lines(bvs[[1]],yvs[[1]])
lines(bvs[[2]],yvs[[2]])
lines(bvs[[3]],yvs[[3]])
```

```
detach(species)
rm(species)
```



3) Categorical explanatory variables with count data: Contingency Tables

a) Fisher's Exact Test

This test is used for the analysis of contingency tables in which **one or more expected frequencies is less than 5**. The individual counts are a, b, c and d like this:

2x2 Table	Col. 1	Col. 2	Row totals
Row 1	a	b	a+b
Row 2	c	d	c+d
Column Totals	A+c	b+d	n

The probability of any one outcome is this

$$p = \frac{(a+b)!(c+d)!(a+c)!(b+d)!}{a!b!c!d!n!}$$

where n is the grand total. Our example concerns a test with ant colonies forming nests (nor not) on two species of trees:

	Tree A	Tree B	Row totals
With ants	6	2	8
Without ants	4	8	12
Column totals	10	10	20

It is easy to calculate the probability of this particular outcome (but **not in R** because, for some obscure reason, it doesn't have a factorial function!). This means that we need to start by writing a factorial function of our own:

```
factorial<-function(x) max(cumprod(1:x))
```

Now we can work out the probabilities for Fisher's Exact Test:

```
factorial(8)*factorial(12)*factorial(10)*factorial(10) /
    (factorial(6)*factorial(2)*factorial(4)*factorial(8)*factorial(20))
```

```
[1] 0.07501786
```

but that is only part of the story. We need to compute the probability of outcomes more extreme than this. There are two of them. Suppose only 1 ant colony had been found on Tree B. Then the table values would be 7, 1, 3, 9 but the row and column totals would be exactly the same (*the marginal totals are constrained*). The numerator always stays the same, so this case has probability

```
factorial(8)*factorial(12)*factorial(10)*factorial(10)/
    (factorial(7)*factorial(3)*factorial(1)*factorial(9)*factorial(20))
```

```
[1] 0.009526078
```

There is an even more extreme case if no ant colonies at all had been found on Tree B. Now the table elements become 8, 0, 2, 10 with probability

```
factorial(8)*factorial(12)*factorial(10)*factorial(10)/
    (factorial(8)*factorial(2)*factorial(0)*factorial(10)*factorial(20))
```

```
[1] 0.0003572279
```

and we need to add these 3 probabilities together

```
0.07501786+0.009526078+0.000352279
```

```
[1] 0.08489622
```

But there was no *a priori* reason for expecting the result to be in this particular direction (more ants on A than B). We need to allow for extreme counts in the opposite direction (more ants on B than on A), by doubling this probability (in fact, all Fisher's Exact Tests are 2-tailed).

```
2*(0.07501786+0.009526078+0.000352279)
```

```
[1] 0.1697924
```

We conclude that there is no evidence of any association between Tree Type and Ant Colonies. The observed pattern could have arisen by chance alone with probability = 0.17.

There is a built in function called **fisher.test**, which saves us all this computation. First, however, we need get the "classical tests" from the library called "ctest"

```
library(ctest)
```

Fishers exact test takes as its argument a 2x2 matrix containing the counts of the 4 contingencies. We make the matrix (column-wise) like this

```
x<-as.matrix(c(6,4,2,8))
```

```
dim(x)<-c(2,2)
```

To see what the matrix looks like type x

```
     [,1] [,2]
[1,]    6    2
[2,]    4    8
```

and run the test like this

```
fisher.test(x)
```

```
     Fisher's exact test

data:  x
p-value = 0.1698
alternative hypothesis: two.sided
```

It gives the same non-significant *p* value that we calculated longhand (above).

b) A simple 2x2 contingency table: the G-test

The simplest analysis of count data is the 2-by-2 contingency table. The traditional analysis of such tables used Pearson's chi-square to test the null hypothesis that two factors were independent in their effects on the response variable. Here, we show the alternative (often called the G-test) which uses log linear models to address the same question. As with Pearson, the test statistic is chi square with 1 degree of freedom.

Suppose we have the following contingency table of counts of oak trees supporting two species of cynipid gall formers

	Andricus present	*Andricus* absent
Biorhiza present	13	44
Biorhiza absent	25	29

The response variable is a count of the number of trees falling into each of 4 categories: with and without the 2 gall formers, *Andricus* and *Biorhiza*. The question to be addressed is whether there is any evidence of ecological association or separation between these two insects on different oak trees. We set up the data like this. There are 3 variables: the response, which we'll call *count*, a 2-level factor (presence or absence) for *Biorhiza*, and a 2-level factor (presence or absence) for *Andricus*. The data frame is so small, we may as well type in the values directly:

```
count<-c(13,44,25,29)
Biorhiza<-factor(c("present","present","absent","absent"))
Andricus<-factor(c("present","absent","present","absent"))
```

Carrying out the log-linear modelling (the **glm** with Poisson errors) is straightforward. We give a name to the model object (*ct* stands for contingency table), then specify the model formula in the usual way. The only extra detail is that we need to specify family=poisson:

```
ct<-glm(count~Biorhiza+Andricus,poisson)
```

We shall not bother with the summary of *ct*, because most of the information is superfluous to present purposes. All we need is the value of the residual deviance:

```
ct
```

```
Call:  glm(formula = count ~ Biorhiza + Andricus, family
= poisson)


Degrees of Freedom: 3 Total (i.e. Null);  1 Residual
Null Deviance:           18.19
Residual Deviance: 6.878     AIC: 33.19
```

The coefficients are of no interest. All we want is the value of the residual deviance (6.878) to compare with chi square tables with 1 d.f. (3.841). Because the calculated value of chi square is larger than the value in tables, we reject the null hypothesis of independence in the distribution of these two gall formers.

Note that the test does not tell us whether there is a positive or a negative correlation between the 2 species' distributions. For this, we need to look at the data and the fitted values (these are the counts we would have expected if the distributions really had been independent):

```
fitted(ct)

    1         2         3         4
 19.51355 37.48649 18.48652 35.51351

count

[1] 13     44         25         29
```

This tells us that if the distributions had been independent, then we should have expected to find the two galls together on the same tree on more than 19 occasions. We actually found them together only 13 times, so the significant association between the two taxa is negative. Positive correlations would be indicated when the observed count of co-occurrence was greater than the expected frequency.

c) A complex contingency table: Schoener's lizards

It is all too easy to analyse complex contingency tables in the wrong way and to produce answers that are actually not supported by the data. Problems typically occur because people fail (or forget) to include all the interactions between the nuisance variables that are necessary to constrain the marginal totals. Note that the problems almost never arise if the same analysis can be structured so that it can be carried out as the analysis of proportion data using binomial errors (compare the example explained below with its re-analysis as proportion data later on, where there are no nuisance variables at all).

The example concerns niche differences between two lizard species:

```
lizards<-read.table("c:\\temp\\lizards.txt",header=T)
attach(lizards)
names(lizards)

[1] "n"    "sun"    "height"  "perch"    "time"     "species"
```

Schoener collected information on the distribution of two *Anolis* lizard species (*A. opalinus* and *A. grahamii*) to see if their ecological niches were different in terms of where and when they perched to prey on insects. Perches were classified by twig diameter, their height in the bush, whether the perch was in sun or shade when the lizard was counted, and the time of day at which they were foraging. The response variable is a count of the number of times a lizard of each species was seen in each of the contingencies. The modelling looks like this. First we fit a saturated model which has a parameter for every data point. There are no degrees of freedom and hence the model has no explanatory power.

```
model1<-glm(n~species*sun*height*perch*time,family=poisson)
```

Prepare yourself for a nasty shock !

```
summary(model1)
```

```
Coefficients:
                                          Estimate Std. Error    z value Pr(>|z|)
(Intercept)                            1.386e+000 5.000e-001      2.773  0.00556 **
species                               -1.000e-014 7.071e-001 -7.73e-015  1.00000
sun                                    9.163e-001 5.916e-001      1.549  0.12143
height                                -1.369e+001 2.847e+002     -0.048  0.96165
perch                                 -2.877e-001 7.638e-001     -0.377  0.70642
timeMid.day                           -1.386e+000 1.118e+000     -1.240  0.21500
timeMorning                           -6.931e-001 8.660e-001     -0.800  0.42349
species.sun                            5.878e-001 8.097e-001      0.726  0.46786
species.height                         1.479e+001 2.847e+002      0.052  0.95857
species.perch                          5.108e-001 1.017e+000      0.503  0.61530
species.timeMid.day                    2.079e+000 1.275e+000      1.631  0.10284
species.timeMorning                    2.303e+000 1.025e+000      2.247  0.02463 *
sun.height                             1.248e+001 2.847e+002      0.044  0.96502
sun.perch                              6.454e-002 8.991e-001      0.072  0.94277
sun.timeMid.day                        2.079e+000 1.183e+000      1.757  0.07884 .
sun.timeMorning                        7.885e-001 9.700e-001      0.813  0.41631
height.perch                           1.259e+001 2.847e+002      0.044  0.96472
height.timeMid.day                     1.386e+000 4.026e+002      0.003  0.99725
height.timeMorning                     6.931e-001 4.026e+002      0.002  0.99863
perch.timeMid.day                      2.877e-001 1.607e+000      0.179  0.85795
perch.timeMorning                      6.931e-001 1.190e+000      0.582  0.56032
species.sun.height                    -1.391e+001 2.847e+002     -0.049  0.96103
species.sun.perch                     -1.099e+000 1.200e+000     -0.916  0.35974
species.sun.timeMid.day               -1.429e+000 1.358e+000     -1.052  0.29283
species.sun.timeMorning               -1.762e+000 1.151e+000     -1.530  0.12598
species.height.perch                  -1.530e+001 2.847e+002     -0.054  0.95714
species.height.timeMid.day            -2.485e+000 4.026e+002     -0.006  0.99508
species.height.timeMorning            -2.223e+000 4.026e+002     -0.006  0.99559
species.perch.timeMid.day             -1.204e+000 1.846e+000     -0.652  0.51431
species.perch.timeMorning             -1.833e+000 1.429e+000     -1.283  0.19965
sun.height.perch                      -1.208e+001 2.847e+002     -0.042  0.96615
sun.height.timeMid.day                -1.792e+000 4.026e+002     -0.004  0.99645
sun.height.timeMorning                -2.776e-001 4.026e+002     -0.001  0.99945
sun.perch.timeMid.day                  4.055e-001 1.700e+000      0.239  0.81147
sun.perch.timeMorning                 -1.598e-001 1.341e+000     -0.119  0.90514
height.perch.timeMid.day              -1.259e+001 4.930e+002     -0.026  0.97963
height.perch.timeMorning              -1.300e+001 4.930e+002     -0.026  0.97897
species.sun.height.perch               1.442e+001 2.847e+002      0.051  0.95960
species.sun.height.timeMid.day         2.989e+000 4.026e+002      0.007  0.99408
species.sun.height.timeMorning         2.040e+000 4.026e+002      0.005  0.99596
species.sun.perch.timeMid.day          1.182e+000 1.981e+000      0.597  0.55083
species.sun.perch.timeMorning          1.417e+000 1.641e+000      0.864  0.38786
species.height.perch.timeMid.day       1.609e+000 5.693e+002      0.003  0.99774
species.height.perch.timeMorning       1.585e+001 4.930e+002      0.032  0.97436
sun.height.perch.timeMid.day           1.183e+001 4.930e+002      0.024  0.98085
sun.height.perch.timeMorning           1.057e+001 4.930e+002      0.021  0.98290
species.sun.height.perch.timeMid.day -1.307e+000 5.693e+002     -0.002  0.99817
species.sun.height.perch.timeMorning -1.330e+001 4.931e+002     -0.027  0.97847

 (Dispersion parameter for poisson family taken to be 1)

Null deviance: 7.3756e+002  on 47  degrees of freedom
Residual deviance: 5.4480e-005  on  0  degrees of freedom
AIC: 259.25
```

There are two key things to note here: (1) the residual deviance and degrees of freedom are both zero because we have intentionally fitted a *saturated model* which has as many parameters (48) as there are data points; and (2) all but 23 of the 48 estimated parameters are nuisance variables. The nuisance variables are of absolutely no scientific interest and are included only to **ensure that all of the marginal totals are correctly constrained.** The only parameters that are of scientific interest are 23 ***interaction terms involving species***.

The only way to model a data set like this successfully is to be fantastically well organised. Terms involving species are deleted stepwise from the current model, starting with the highest order interactions. Non-significant terms are left out. Significant terms are added back in to the model. We need to keep very accurate track of which terms we have deleted and which terms remain to be deleted. Remember,

you can't remove any 3-way interactions until the 4-way interactions containing the relevant factors have been removed.

The 23 parameters to be tested are involved in the following 15 potentially interesting interaction terms that involve species:

```
species.sun
species.height
species.perch
species.timeMid.day
species.timeMorning
species.sun.height
species.sun.perch
species.sun.timeMid.day
species.sun.timeMorning
species.height.perch
species.height.timeMid.day
species.height.timeMorning
species.perch.timeMid.day
species.perch.timeMorning
species.sun.height.perch
species.sun.height.timeMid.day
species.sun.height.timeMorning
species.sun.perch.timeMid.day
species.sun.perch.timeMorning
species.height.perch.timeMid.day
species.height.perch.timeMorning
species.sun.height.perch.timeMid.day
species.sun.height.perch.timeMorning
```

I suggest that you start at the bottom of this list and tick off the interaction terms as you delete them. Once you have tested, say, all of the 4-way interactions involving species and found them to be non significant, then leave them all out (but leave the high order interactions involving the nuisance variables in the model).

Using automatic model simplification with **step** is not much use in a case like this because:

- **step** tends to be too lenient, and to leave non-significant terms in the model
- it is likely to remove interaction terms between nuisance variables and these must stay in the model in order to constrain the marginal totals properly

There is no way round it. We have to do the model simplification the long way, using **update** to delete terms and **anova** to test the significance of deleted terms. I have used cut and paste to make the following table of deletion tests:

```
model2<-update(model1,~.-species:sun:height:perch:time)
anova(model1,model2,test="Chi")
```

```
  Resid. Df Resid. Dev Df Deviance P(>|Chi|)
1         0    0.00005
2         2    0.00010 -2 -0.00004   0.99998
```

This means that we can adopt model2 as the base model and systematically delete the various 4-way interactions involving species:

```
model3<-update(model2,~.-species:sun:height:perch)
anova(model2,model3,test="Chi")
```

```
Resid. Df Resid. Dev Df Deviance P(>|Chi|)
1         2     0.0001
2         3     2.7089 -1  -2.7088    0.0998
```

And so on.

```
Resid. Df Resid. Dev   Test                            Df  Deviance      Pr(Chi)
       2 0.00014458   -species:sun:height:perch:time -2 -0.00004    0.9999766
       3   2.708911   -species:sun:height:perch      -1 -2.708766    0.09979817
       5   3.111413   -species:sun:height:time       -2 -0.4025021   0.8177071
       7   7.928383   -species:height:perch:time     -2 -4.816971    0.08995144
       9   8.573263   -species:sun:perch:time        -2 -0.6448801   0.7243793
      10   8.587620   -species:sun:perch             -1 -0.01435679  0.9046259
      11   11.97484   -species:sun:height            -1 -3.387218    0.06570373
      13   13.34891   -species:sun:time              -2 -1.374076    0.5030639
      15   13.37456   -species:perch:time            -2 -0.02564741  0.9872582
      16   13.68243   -species:height:perch          -1 -0.3078676   0.5789916
      18   14.20496   -species:height:time           -2 -0.522531    0.7700764
  *** 20   25.80257   -species:time                  -2 -11.59761   0.003031181
  *** 19   36.27283   -species:height                -1 -22.06787   2.6317e-006
  *** 19   21.89253   -species:sun                   -1 -7.687569   0.005560247
  *** 19   27.33579   -species:perch                 -1 -13.13083   0.000290476
```

There are no significant interactions between the explanatory variables that have a significant effect on the distribution of these two lizard species. All of the factors have highly significant main effects, however. The two lizard species utilise different parts of the bush, occupy perches of different diameters, are active at different times of day, and are found in differing levels of shade. The published analyses that report significant interactions between factors all made the mistake of unintentionally leaving out one or more interactions involving the nuisance variables. This is a very easy thing to do, unless a strict regime of *simplifying downwards from the saturated model* is followed religiously. If you always simplify down from a complex model to a simple model this problem will never arise, and you will save yourself from making embarrassing mistakes of interpretation. With this many comparisons being done, I would always work at $p = 0.01$ for significance, rather than $p = 0.05$.

d) The Danger of Contingency Tables

In observational studies we quantify only a limited number of explanatory variables. It is inevitable that we shall fail to note (or to measure) a number of factors that have an important influence on the ecological behaviour of the system in question. That's life, and given that we make every effort to note the important factors, there is little we can do about it. The problem comes when we ignore factors that have an important influence on ecological behaviour. This difficulty can be particularly acute if we *aggregate data over important explanatory variables*. An example should make this clear.

Suppose we are carrying out a study of induced defences in trees. A preliminary trial has suggested that early feeding on a leaf by aphids may cause chemical changes in the leaf which reduce the probability of that leaf being attacked later in the season by hole-making insects. To this end we mark a large cohort of leaves, then score whether they were infested by aphids early in the season and whether they were holed by

insects later in the year. The work was carried out on two different trees and the results were as follows:

Tree	Aphids	Holed	Intact	Total leaves	Proportion Holed
Tree 1	Without	35	1750	1785	0.0196
	With	23	1146	1169	0.0197
Tree 2	Without	146	1642	1788	0.0817
	With	30	333	363	0.0826

There are 4 variables: the response variable, count, with 8 values (highlighted above), a 2-level factor for late season feeding by caterpillars (holed or intact), a 2-level factor for early season aphid feeding (With or Without aphids) and a 2-level factor for tree (the observations come from two separate trees, imaginatively named Tree1 and Tree2).

```
induced<-read.table("c:\\temp\\induced.txt",header=T)
attach(induced)
names(induced)
```

```
[1] "Tree"        "Aphid"       "Caterpillar"   "Count"
```

We call the model *id* (for induced defences) and fit what is known as a **saturated model**. This is a curious thing, which has as many parameters as there are values of the response variable. The fit of the model is perfect, so there are no residual degrees of freedom and no residual deviance. The reason that we fit a saturated model is that it is always the best place to start modelling complex contingency tables. If we fit the saturated model, then there is no risk that we inadvertently leave out important interactions between the nuisance variables.

```
id<-glm(Count~Tree*Aphid*Caterpillar,family=poisson)
```

The asterisk notation ensures that the saturated model is fitted, because all of the main effects and 2-way interactions are fitted, along with the 3-way interaction *Tree* by *Aphid* by *Caterpillar*. The model fit involves the estimation of 2 x 2 x 2 = 8 parameters, and exactly matches the 8 values of the response variable, *Count*. There is no point looking at the saturated model in any detail, because the reams of information it contains are all superfluous. The first real step in the modelling is to use **update** to remove the 3-way interaction from the saturated model, and then to use **anova** to test whether the 3-way interaction is significant or not.

```
id2<-update(id , ~ . - Tree:Aphid:Caterpillar)
```

The punctuation here is very important (it is comma, tilde, dot, minus) and note the use of colons rather than asterisks to denote interaction terms rather than main effects plus interaction terms. Now we can see whether the 3-way interaction was significant by specifying test=”Chi” like this:

```
anova(id,id2,test="Chi")
```

```
Analysis of Deviance Table
Response: Count

  Resid. Df  Resid. Dev Df Deviance P(>|Chi|)
1         0 -3.975e-013
2         1     0.00079 -1 -0.00079   0.97756
```

This shows clearly that the interaction between caterpillar attack and leaf holing does not differ from tree to tree (p = 0.97756). Note that if this interaction had been significant, then we would have stopped the modelling at this stage. But it wasn't, so we leave it out and continue. What about the main question? Is there an interaction between caterpillar attack and leaf holing? To test this we delete the *Caterpillar:Aphid* interaction from the model, call it *id3*, and assess the results using **anova**:

```
id3<-update(id2 , ~ . - Aphid:Caterpillar)
anova(id2,id3,test="Chi")
```

```
Analysis of Deviance Table
Response: Count

  Resid. Df  Resid. Dev               Test Df    Deviance   Pr(Chi)
1         1 0.000791372
2         2 0.004085322 -Aphid:Caterpillar -1 -0.00329395 0.9542322
```

There is absolutely no hint of an interaction (p = 0.954). The interpretation is clear. This work provides no evidence for induced defences caused by early season caterpillar feeding.

Now we shall do the analysis the wrong way, in order to show the danger of collapsing a contingency table over important explanatory variables. Suppose we went straight for the interaction of interest, *Aphid:Caterpillar*. We might proceed like this:

```
wrong<-glm(Count~Aphid*Caterpillar,family=poisson)
wrong1<-update(wrong,~. - Aphid:Caterpillar)
anova(wrong,wrong1,test="Chi")
```

```
Analysis of Deviance Table
Response: Count

  Resid. Df Resid. Dev               Test Df  Deviance     Pr(Chi)
1         4   550.1917
2         5   556.8511 -Aphid:Caterpillar -1 -6.659372 0.009863566
```

The *Aphid* by *Caterpillar* interaction is highly significant (p < 0.01) providing strong evidence for induced defences. **Wrong !!** By failing to include *Tree* in the model we have omitted an important explanatory variable. As it turns out, and we should really have determined by more thorough preliminary analysis, the trees differ enormously in their average levels of leaf holing:

as.vector(tapply(Count,list(Caterpillar,Tree),sum))[1]/ tapply(Count,Tree,sum) [1]

```
  Tree1
0.01963439
```

as.vector(tapply(Count,list(Caterpillar,Tree),sum))[3]/ tapply(Count,Tree,sum) [2]

```
   Tree2
0.08182241
```

Tree2 has more than 4 times the proportion of its leaves with holes made by caterpillars. If we had been paying more attention when we did the modelling the wrong way, we should have noticed that the model containing only Aphid and Caterpillar had massive **overdispersion**, and this should have alerted us that all was not well. The moral is simple and clear. Always fit a saturated model first, containing all the variables of interest and all the interactions involving the nuisance variables (*Tree* in this case). Only delete from the model those interactions that involve the variables of interest (*Aphid* and *Caterpillar* in this case). Main effects are meaningless in contingency tables, as are the model summaries. **Always test for overdispersion**. It will never be a problem if you follow the advice of simplifying down from a saturated model, because you only ever leave out non-significant terms.

4) Bird Ring Recoveries

Certain kinds of survival analysis involve the periodic recovery of small numbers of dead animals. Bird ringing records are a good example of this kind of data. Out of a reasonably large, known number of ringed birds, a usually small number of dead birds carrying rings is recovered each year. The number of rings recovered each year declines because the pool of ringed birds declines each year as a result of natural mortality. There are two important variables in this case: 1) the probability of a bird dying in a given time period; and 2) the probability that, having died, the ring will be discovered. It is possible that one or both of these parameters varies with the age of the bird (i.e. with the time elapsed since the ring was applied). We can use SPlus with Poisson errors to fit log-linear models to data like this, and to compare estimated survival and ring-recovery rates from different cohorts of animals.

The probability that a bird survives from ringing up to time t_{j-1} is the survivorship s_{j-1} . The number of ringed birds still alive at time t_{j-1} is therefore $N_o s_{j-1}$, where N_o is the number of birds initially ringed and released at time t_o. Thus, the number of birds that die in the present time interval D_j is:

$$D_j = N_0 s_{j-1} d_j$$

where d_j is the probability of an animal that survived to time t_{j-1} dying during the interval $(t_j - t_{j-1})$. Now, out of these D_j dead animals, we expect to recover a small proportion p_j, so the expected number of recoveries, R_j, is:

$$R_j = N_0 s_{j-1} d_j p_j$$

Maximum likelihood estimates of the death and recovery probabilities are possible only if assumptions are made about the way in which the two parameters change with age and with time after ringing (e.g. Seber, 1982). Common assumptions are that the probability of death is constant, once a given age has been reached, and that the probability of discovery of a dead bird is constant over time.

In order to analyse ring return data by log-linear modelling, we must combine the two probabilities of death and recovery into a single parameter z_j. This is the *probability of a death being recorded* in year j for an animal that was alive at the beginning of year j.

$$R_j = N_0 s_{j-1} z_j$$

Now, taking logs, rearranging and then taking antilogs we can write this expression as:

$$R_j = \exp[\ln(N_0) - \lambda_{j-1} t_{j-1} + \ln(z_j)]$$

replacing the survivorship term S_{j-1} by the proportion dying λt. Since we know the number of birds originally marked, we can use $\ln(N_0)$ as an offset. Then, using the log link, we are left with a graph of log of recoveries $\ln R_j$ against time t_j in which the intercept is given by the log of the recovery probability ($\ln(z_j)$) and the slope λ_{j-1} is the survival rate per unit time over the period 0 to t_{j-1}.

S-Plus can now be used to analyse count data on the number of recoveries of dead ringed birds, and to compare models based on different assumptions about survivorship and recovery:

- different cohorts have constant death rates and recovery rates, and these rates are the same in all cohorts (the null model);
- different cohorts have the same death rates but different recovery probabilities;
- different cohorts differ in both their death rates and their recovery rates;
- the death rates may be time dependent; or
- a variety of more complex models.

Suppose that tawny owls were ringed in three different kinds of woodland in the same county; oak, birch and mixed woodland. The numbers marked in the three habitats were 75, 49 and 128 respectively. The numbers of dead, ringed birds recovered in subsequent years were as follows:

```
owlrings<-read.table("c:\\temp\\owlrings.txt",header=T)
attach(owlrings)
names(owlrings)
```

```
[1] "recovered" "year"      "wood"      "marked"
```

The model has the number of rings recovered as the response variable (a count with Poisson errors, with the number of birds marked appearing in the model as an offset

(on the log scale of the linear predictor, because the log link is the default for Poisson models: hence the name, log-linear models).

```
model<-glm(recovered~wood*year+offset(log(marked)),family=poisson)

summary(model)
```

```
Coefficients:
               Estimate Std. Error z value Pr(>|z|)
(Intercept)    -2.34595    0.58719  -3.995 6.46e-05 ***
woodbirch       0.14340    0.70377   0.204   0.8385
woodoak        -0.08609    0.77883  -0.111   0.9120
year           -0.35839    0.18323  -1.956   0.0505 .
woodbirch:year -0.12476    0.22886  -0.545   0.5856
woodoak:year   -0.02906    0.24617  -0.118   0.9060

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 42.839  on 20  degrees of freedom
Residual deviance: 15.670  on 15  degrees of freedom
AIC: 69.859
```

There is clearly no justification for keeping different slopes for the different woodlands ($p > 0.5$), so we remove the interaction term, then compare the complex and simpler models using anova. Note the lack of overdispersion.

```
model2<-glm(recovered~wood+year+offset(log(marked)),family=poisson)

anova(model,model2,test="Chi")
```

```
Analysis of Deviance Table

Model 1: recovered ~ wood * year + offset(log(marked))
Model 2: recovered ~ wood + year + offset(log(marked))

  Resid. Df Resid. Dev Df Deviance P(>|Chi|)
1        15    15.6703
2        17    16.0353 -2  -0.3650    0.8332
```

This confirms that there is no justification for retaining different slopes in different woodlands. What about differences in the intercepts of different woodlands? We simplify model2, like this:

```
model3<-glm(recovered~year+offset(log(marked)),family=poisson)
anova(model2,model3,test="Chi")
```

```
Analysis of Deviance Table

Model 1: recovered ~ wood + year + offset(log(marked))
Model 2: recovered ~ year + offset(log(marked))

  Resid. Df Resid. Dev Df Deviance P(>|Chi|)
```

```
1        17    16.0353
2        19    16.2502 -2  -0.2148    0.8982
```

Not even close (p = 0.898). What about the slope?

```
model4<-glm(recovered~offset(log(marked)),family=poisson)
anova(model3,model4,test="Chi")
```

```
Analysis of Deviance Table

Model 1: recovered ~ year + offset(log(marked))
Model 2: recovered ~ offset(log(marked))
  Resid. Df Resid. Dev Df Deviance P(>|Chi|)
1        19     16.250
2        20     42.839 -1  -26.589 2.517e-07
```

No doubt about the significance of the slope. So model3 looks to be minimal adequate:

```
summary(model3)
```

```
Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -2.30547    0.27314  -8.441  < 2e-16 ***
year        -0.42522    0.09114  -4.665 3.08e-06 ***

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 42.839  on 20  degrees of freedom
Residual deviance: 16.250  on 19  degrees of freedom
AIC: 62.439
```

The final model is not overdispersed (16.25 on 19 d.f.). The interpretation is that the log of the recovery rate (the intercept) is –2.30547 and that the log of the survival rate (the slope) is – 0.42522. Taking antilogs we get

```
exp(-2.30547)
```

```
[1] 0.09971193
```

```
exp(-0.42522)
```

```
[1] 0.653626
```

We conclude that the annual survivorship of the owls is about 65% and that survival does not differ significantly from one woodland to another. The annual probability of recovery of a ring from a dead ringed bird is estimated to be about 10%.

Error checking by **plot(model3)** shows a few problems: the deviance declines markedly with the fitted values, and there is substantial non-normality in the error distribution, particularly for the largest negative residuals. But this will not affect the parameter estimates greatly, and parameter estimation is our main purpose here.

Reanalysis of Schoener's lizards as proportion data:
getting rid of the nuisance variables

The analysis of count data in complex contingency tables is difficult, tedious and very easy to get wrong. If the response can be reformulated as a proportion, then the analysis is much more straightforward, because there is no longer any need to include the vast numbers of nuisance parameters necessary to constrain the marginal totals.

The computational part of the exercise here is data compression. We need to reduce the data frame called lizards so that it has the proportion of all lizards that are *Anolis grahamii* as the response, with the same set of explanatory variables (but each with half as many rows as before). It is essential to ensure that all of the cases are in exactly the same sequence for both of the lizard species. To do this, we create a sorted version of the data frame:

```
sorted.lizards<-lizards[order(lizards[,2],lizards[,3],lizards[,4],lizards[,5],lizards[,6]),1:6]
```

So now we can save the ordered *grahamii* and *opalinus* vectors as *ng* and *no* and bind them together to use as the response vector in the binomial analysis

```
ng<-sorted.lizards[,1][sorted.lizards[,6]=="grahamii"]
no<-sorted.lizards[,1][sorted.lizards[,6]=="opalinus"]
y<-cbind(ng,no)
```

The 4 shortened explanatory variables (the factors s, h, p, and t) are now produced:

```
s<-sorted.lizards[,2][sorted.lizards[,6]=="grahamii"]
h<-sorted.lizards[,3][sorted.lizards[,6]=="grahamii"]
p<-sorted.lizards[,4][sorted.lizards[,6]=="grahamii"]
t<-sorted.lizards[,5][sorted.lizards[,6]=="grahamii"]
```

You should check that these have been automatically declared to be factors (hint: use **is.factor**). Now we are ready to fit the saturated model:

```
model<-glm(y ~ s*h*p*t , binomial)
```

Let's see how good **step** is at simplifying this saturated model:

```
step(model)
```

```
Start:  AIC= 102.82
 y ~ s + h + p + t + s:h + s:p + s:t + h:p + h:t + p:t + s:h:p +
    s:h:t + s:p:t + h:p:t + s:h:p:t
```

(out goes the 4-way interaction s:h:p:t)

```
          Df  Deviance    AIC
- s:h:p:t  1 0.0002621 100.82
<none>       0.0001585 102.82

Step:  AIC= 100.82
```

```
 y ~ s + h + p + t + s:h + s:p + s:t + h:p + h:t + p:t + s:h:p +
    s:h:t + s:p:t + h:p:t
```

(out goes the 3-way interaction s:h:t)

```
        Df  Deviance     AIC
- s:h:t  2     0.442  97.266
- s:p:t  2     0.810  97.635
- h:p:t  2     3.222 100.046
<none>     0.0002621 100.824
- s:h:p  1     2.709 101.533

Step:  AIC= 97.27
 y ~ s + h + p + t + s:h + s:p + s:t + h:p + h:t + p:t + s:h:p +
    s:p:t + h:p:t
```

(out goes the 3-way interaction s:p:t)

```
        Df Deviance    AIC
- s:p:t  2    1.071 93.896
<none>        0.442 97.266
- h:p:t  2    4.648 97.472
- s:h:p  1    3.111 97.936

Step:  AIC= 93.9
 y ~ s + h + p + t + s:h + s:p + s:t + h:p + h:t + p:t + s:h:p +
    h:p:t

        Df Deviance    AIC
- s:t    2    3.340 92.165
<none>        1.071 93.896
- s:h:p  1    3.302 94.126
- h:p:t  2    5.791 94.615

Step:  AIC= 92.16
 y ~ s + h + p + t + s:h + s:p + h:p + h:t + p:t + s:h:p + h:p:t
```

(AIC has *increased,* so this deletion was not accepted)

```
        Df Deviance    AIC
<none>        3.340 92.165
- s:h:p  1    5.827 92.651
- h:p:t  2    8.542 93.366

Degrees of Freedom: 22 Total (i.e. Null);  7 Residual
Null Deviance:      70.1
Residual Deviance: 3.34         AIC: 92.16
Start:  AIC= 48.0002
```

(it wants to keep the 3-way interaction h:p:t)

```
model1<-step(model)
summary(model1)
```

```
Coefficients:
                      Estimate Std. Error z value Pr(>|z|)
(Intercept)            -0.8297     0.5120  -1.620   0.1051
sSun                    0.4791     0.4744   1.010   0.3126
hLow                  -11.9069    72.7886  -0.164   0.8701
pNarrow                 0.1751     0.7481   0.234   0.8149
tMid.day               -0.9101     0.4272  -2.130   0.0331 *
tMorning               -0.9314     0.4637  -2.009   0.0445 *
sSun:hLow              10.7912    72.7884   0.148   0.8821
sSun:pNarrow            0.2416     0.6987   0.346   0.7295
hLow:pNarrow           12.1239    72.8016   0.167   0.8677
hLow:tMid.day          -0.2446     0.9278  -0.264   0.7920
hLow:tMorning           0.5732     0.9260   0.619   0.5359
pNarrow:tMid.day        0.2140     0.6486   0.330   0.7415
pNarrow:tMorning        0.7106     0.6980   1.018   0.3087
sSun:hLow:pNarrow     -10.9665    72.8028  -0.151   0.8803
hLow:pNarrow:tMid.day  -0.6021     1.3489  -0.446   0.6553
hLow:pNarrow:tMorning  -3.2054     1.6106  -1.990   0.0466 *
```

As often happens, **step** has left a rather complicated model with two 3-way interactions and six 2-way interactions in it. Let's see how we get on by hand in simplifying the reduced model (model1) that **step** has bequeathed us. We remove the terms in sequence and use **anova** to compare the simpler model with its more complex predecessor:

```
model2<-update(model1,~.-h:p:t)
anova(model1,model2,test="Chi")
```

```
Analysis of Deviance Table

  Resid. Df Resid. Dev Df Deviance P(>|Chi|)
1         7     3.3405
2         9     8.5422 -2  -5.2017    0.0742
```

This simplification caused a non-significant increase in deviance (p = 0.07) so we make model2 the current model, and remove the next 3-way interaction, s:h:p (see below). We keep removing terms until a significant term is discovered, as follows:

```
                          Resid. Df Resid. Dev Df Deviance P(>|Chi|)

model2 <-update(model1,~.-h:p:t)  9     8.5422 -2  -5.2017    0.0742
model3 <-update(model2,~.-s:h:p) 10    10.9032 -1  -2.3610    0.1244
model4 <-update(model3,~.-p:t)   12    10.9090 -2  -0.0058    0.9971
model5 <-update(model4,~.-h:t)   14    11.7667 -2  -0.8577    0.6513
model6 <-update(model5,~.-h:p)   15    11.9789 -1  -0.2122    0.6450
model7 <-update(model6,~.-s:p)   16    11.9843 -1  -0.0054    0.9414
model8 <-update(model7,~.-s:h)   17    14.2046 -1  -2.2203    0.1362
model9 <-update(model8,~.-t)     19     25.802 -2  -11.597     0.003 ***
model10<-update(model8,~.-p)     18    27.3346 -1 -13.1300    0.0003 ****
model11<-update(model8,~.-h)     18     36.271 -1  -22.066 2.634e-06 ****
model12<-update(model8,~.-s)     18    21.8917 -1  -7.6871    0.0056 ***
```

Note that because there was a significant increase in deviance when time was deleted from model8, we use model 8 not model9 in assessing the significance of perch diameter (and of the other two main effects as well). The main effect probabilities are identical to those obtained by contingency table analysis with Poisson errors (above). We need to look at the model summary for model8 to see whether all of the factor levels need to be retained:

```
summary(model8)
```

```
Call:
glm(formula = y ~ s + h + p + t, family = binomial)

Deviance Residuals:
     Min        1Q    Median        3Q       Max
-1.48718  -0.62644  -0.04488   0.37800   1.66015

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  -1.2079     0.3534  -3.418 0.000631 ***
sSun          0.8473     0.3222   2.629 0.008554 **
hLow         -1.1300     0.2570  -4.397 1.10e-05 ***
pNarrow       0.7626     0.2112   3.610 0.000306 ***
tMid.day     -0.9639     0.2815  -3.424 0.000618 ***
tMorning     -0.7368     0.2989  -2.465 0.013712 *

 (Dispersion parameter for binomial family taken to be 1)

    Null deviance: 70.102  on 22  degrees of freedom
Residual deviance: 14.205  on 17  degrees of freedom
AIC: 83.029
```

Mid.day and Morning are not significantly different from one another (their parameters are -0.96 and -0.74 with a standard error of a difference of 0.299), so we lump them together in a new factor called t3:

```
t3<-t
levels(t3)[c(2,3)]<-"other"
levels(t3)
```

```
[1] "Afternoon" "other"
```

```
model9<-glm(y~s+h+p+t3,binomial)
anova(model8,model9,test="Chi")
```

```
Analysis of Deviance Table

Response: y

          Terms Resid. Df Resid. Dev    Test Df   Deviance   Pr(Chi)
1  s + h + p + t       17   14.20457
2 s + h + p + t3       18   15.02320 1 vs. 2 -1 -0.8186255 0.3655824
```

This simplification was justified (p = 0.37), so we accept model9.

```
summary(model9)
```

```
Call:
glm(formula = y ~ s + h + p + t3, family = binomial)

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  -1.1595     0.3482  -3.330 0.000870 ***
sSun          0.7871     0.3158   2.493 0.012682 *
hLow         -1.1188     0.2565  -4.362 1.29e-05 ***
pNarrow       0.7485     0.2104   3.557 0.000375 ***
t3other      -0.8717     0.2611  -3.339 0.000842 ***
```

```
(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 70.102  on 22  degrees of freedom
Residual deviance: 15.023  on 18  degrees of freedom
AIC: 81.847
```

All the parameters are significant, so this is the minimal adequate model. There are just 5 parameters, and the model contains no nuisance variables (compare this with the massive contingency table model on p. 322). The ecological interpretation is straightforward: the two lizard species differ significantly in their niches on all the niche-axes that were measured. However, there were no significant interactions (nothing subtle was happening like swapping perch sizes at different times of day).