

Count Data: Log-Linear Models

Introduction

In the previous two topics we used generalised linear models with binomial error structures to analyse binary data - where all values are ones or zeros - and proportion data - where we know the number of events and how many were successes and failures. With count data we know how many times something happened but not how many times it did not. The data are integers (whole numbers) and so the residuals will be restricted in the values they can take accordingly. In many types of count data zeros are often common. We can analyse counts in relation to continuous explanatory variables or to categorical variables, in which case the data take the form of tables of frequencies. We can analyse count data using so-called log-linear models: GLMs with a Poisson error structure and a log link function. As we saw in the introduction, the Poisson distribution is a one parameter distribution, where the variance is assumed to equal the mean, which can be used to take account of the non-normal nature of much count data. For a series of counts with mean λ then the probability of observing a certain frequency x is:

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

And, because,

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

then successive probabilities can be calculated by taking the preceeding probability, multiplying by the mean and dividing by x , starting with the zero term as follows:

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

The log link function prevents predictions of negative counts since the fitted values are antilogs and must be positive. As with the analysis of proportion data the residual deviance is assumed to equal the residual degrees of freedom and the scale parameter is set as one, although we must check for over-dispersion and deal with it in a similar way. The deviance is once again estimated by an iterative weighted least squares maximum likelihood procedure with its distribution approximately following the chi-squared distribution. In GLMs with Poisson errors the deviance is:

$$2 \sum O \ln \left(\frac{O}{E} \right)$$

where O is the observed count and E is the expected count based on the statistical model. Imagine you had two counts from each of two areas: 3 and 6 in Location one and 4 and 7 in location two. Are the areas different? The expected count is the overall mean:

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

```
[1] 5
```

And the total deviance can then be calculated as:

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

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

```
[1] 2.048368
```

The means for the two locations are $9/2 = 4.5$ and $11/2 = 5.5$ respectively. The residual deviance after fitting 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
```

The treatment deviance is the difference:

$$2.048368 - 1.848033$$

```
[1] 0.200335
```

We can perform the same analysis in R using a GLM with Poisson errors as follows:

```
y <- c(3, 6, 4, 7) ;
location <- factor(c(1,1,2,2)) ;
model <- glm(y ~ location, poisson) ; summary(model)
```

Coefficients:

	Estimate	Std. Error	z value	Pr(> z)
(Intercept)	1.5041	0.3331	4.516	6.32e-06 ***
location2	0.2007	0.4492	0.447	0.655

Null deviance: 2.0484 on 3 degrees of freedom
Residual deviance: 1.8480 on 2 degrees of freedom

The total deviance of the null model and the residual deviance of the model including location match. Calculations for more elaborate examples are more complex as they are iterative estimations but this simple example hopefully serves to demystify deviance.

Analysis of count data with log-linear models in R

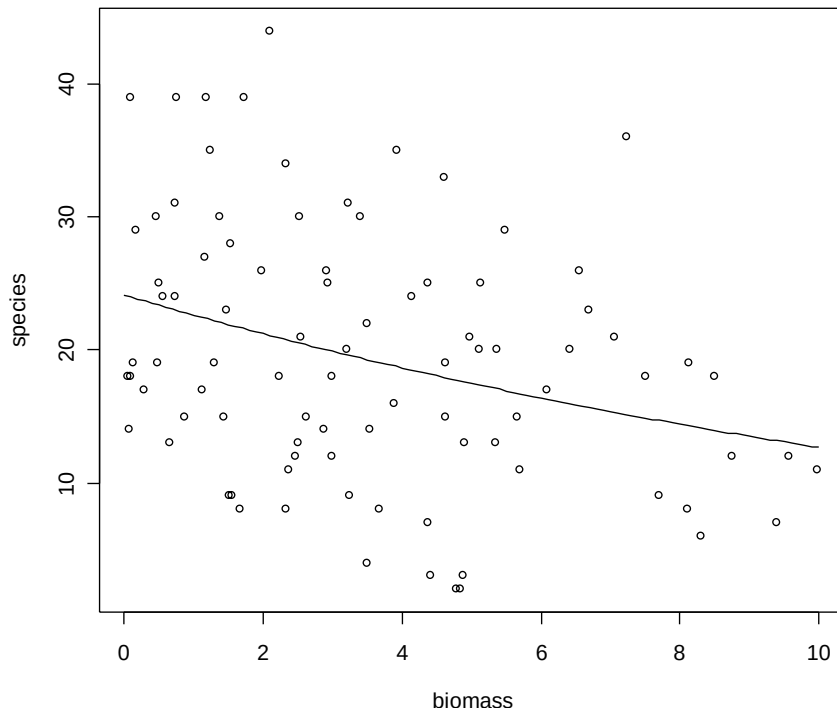
This example, taken from Crawley (2002), looks at numbers of species found in grassland plots fertilised with different amounts and types of nutrients. The data come from the Park Grass experiment at Rothamstead Experimental Station in S.E. England which is famous as the longest running ecological study (it was initially set up to test whether inorganic fertilizers worked in comparison with traditional methods). First load and inspect the data:

```
parkgrass <-  
read.table("D:\\Andy\\_Teaching\\Statistics\\_R.GLMs\\poisson.diversity.crawley.t  
xt", header=T) ; attach(parkgrass) ; names(parkgrass) ;
```

```
[1] "ph"          "biomass"     "species"
```

A count of the number of species per plot is given by "species", soil pH by "ph" and aboveground biomass production by "biomass". We can begin by looking at diversity in relation to biomass production:

```
plot(species ~ biomass)
```



It looks as if diversity is negatively related to biomass production (the code for plotting the line is given below). We can fit a log-linear model as follows:

```
model1 <- glm(species ~ biomass, poisson(link = log)) ; summary(model1)
```

Call:

```
glm(formula = species ~ biomass, family = poisson)
```

Deviance Residuals:

Min	1Q	Median	3Q	Max
-4.7703	-1.5994	-0.1025	1.2722	4.5433

Coefficients:

	Estimate	Std. Error	z value	Pr(> z)	
(Intercept)	3.184094	0.039159	81.31	< 2e-16	***
biomass	-0.064441	0.009838	-6.55	5.74e-11	***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for poisson family taken to be 1)

Null deviance: 452.35 on 89 degrees of freedom
 Residual deviance: 407.67 on 88 degrees of freedom
 AIC: 830.86

Number of Fisher Scoring iterations: 4

As usual, the output gives a reminder of the fitted model, some information on the residuals and then the regression coefficients and deviances. The negative effect of biomass production on diversity is highly significant. The negative log-linear regression slope was added to the graph above in the usual way (generate a sequence of a hundred or more x-values and use **predict** to generate the respective values for the curve):

```
xseq <- seq(0,10, 0.1)
fitted.curve <- predict(model1, list(biomass = xseq), type="response")
lines(xseq, fitted.curve)
```

However, the current model is over-dispersed since the residual deviance is much larger than the residual degrees of freedom. We may be able to account for some of this variation if we examine the effect of our second response variable, soil pH. However, for demonstration purposes, let's see how to correct for overdispersion with Poisson models. The procedure is the same as for binomial analysis of proportions except we now use quasipoisson:

```
modelq <- glm(species ~ biomass, quasipoisson) ; summary(modelq)
```

Call:

```
glm(formula = species ~ biomass, family = quasipoisson)
```

Deviance Residuals:

Min	1Q	Median	3Q	Max
-4.7703	-1.5994	-0.1025	1.2722	4.5433

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	3.18409	0.08191	38.871	< 2e-16 ***
biomass	-0.06444	0.02058	-3.131	0.00236 **

(Dispersion parameter for quasipoisson family taken to be 4.375742)

Null deviance: 452.35 on 89 degrees of freedom
Residual deviance: 407.67 on 88 degrees of freedom
AIC: NA

Number of Fisher Scoring iterations: 4

```
anova(modelq, test="F")
```

Analysis of Deviance Table

Model: quasipoisson, link: log

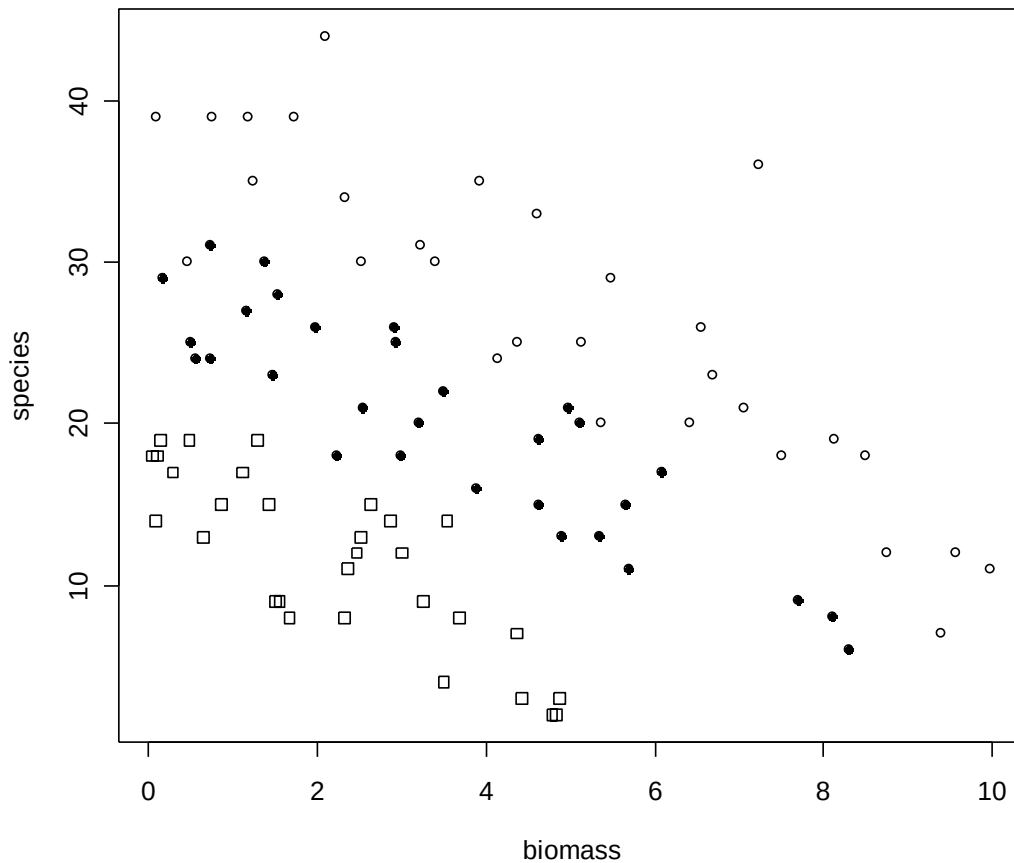
Response: species

Terms added sequentially (first to last)

	Df	Deviance	Resid.	Df	Resid. Dev	F	Pr(>F)
NULL				89	452.35		
biomass	1	44.67		88	407.67	10.209	0.001940 **

In both the summary and anova output we can see that the family is now quasipoisson and the dispersion parameter for quasipoisson family is taken to be 4.375742 rather than 1. However, in this case we may not need to correct for over-dispersion if we can find a model that fits the Poisson distribution more closely. We can return to the initial model and try including pH and see if this improves things. First, re-plot the graph but now using type=null to plot an empty frame and then adding the different pH sets of points one at a time by using the square indexing brackets:

```
plot(species ~ biomass, type="n")
points(species[pH=="high"] ~ biomass[pH=="high"])
points(species[pH=="mid"] ~ biomass[pH=="mid"], pch=16)
points(species[pH=="low"] ~ biomass[pH=="low"], pch=0)
```



It seems pretty clear that pH does have an additional effect. We can modify the original regression to an analysis of covariance as follows:

```
model2 <- glm(species ~ ph * biomass, poisson) ; summary(model2) ;
```

Call:

```
glm(formula = species ~ ph * biomass, family = poisson)
```

Deviance Residuals:

Min	1Q	Median	3Q	Max
-2.49780	-0.74845	-0.04023	0.55745	3.22974

Coefficients:

	Estimate	Std. Error	z value	Pr(> z)	
(Intercept)	3.76812	0.06153	61.242	< 2e-16	***
phlow	-0.81558	0.10282	-7.932	2.16e-15	***
phmid	-0.33146	0.09216	-3.596	0.000323	***
biomass	-0.10713	0.01249	-8.579	< 2e-16	***
phlow:biomass	-0.15502	0.03996	-3.880	0.000105	***

```
phmid:biomass -0.03189    0.02307   -1.382  0.166885
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

(Dispersion parameter for poisson family taken to be 1)

```
Null deviance: 452.346 on 89 degrees of freedom
Residual deviance: 83.201 on 84 degrees of freedom
AIC: 514.39
```

Number of Fisher Scoring iterations: 3

This model is much improved: the residual deviance and residual degrees of freedom are now equal indicating that the model is appropriate. We can test the interaction term by using the anova command to compare models with and without it:

```
model3 <- update(model2, ~. -ph:biomass)
anova(model2, model3, 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	83.20			
2	86	99.24	-2	-16.04	3.288e-04

Simplifying from model 2 (with the interaction) to model 3 (only main effects) shows that the interaction term is highly significant and needs to be retained in the model. We can finish by adding the negative slopes to our figure for the different pH classes. Because we want to plot a curve for each pH we need to generate a factor, 'PH', with three levels, each with as many repeats as we have values in xseq, and we need to make a vector that contains xseq repeated three times, once for each level of PH:

```
length(xseq)
```

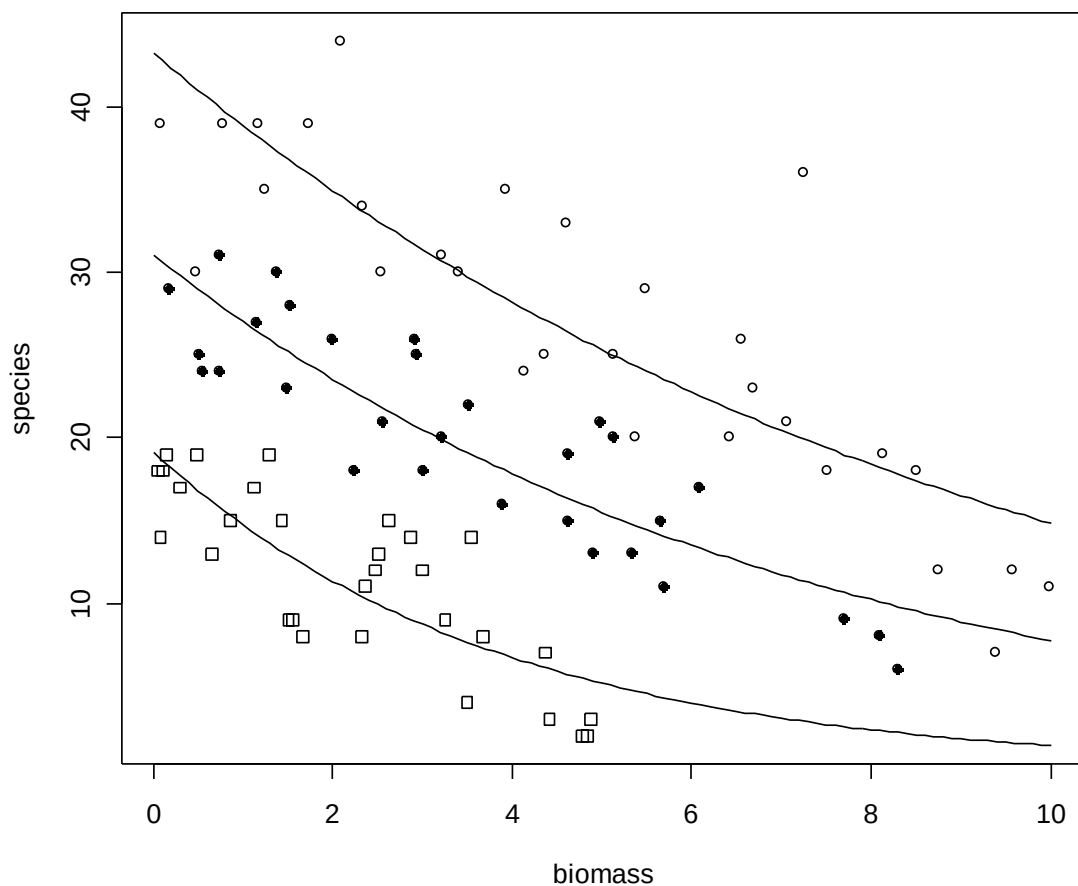
```
[1] 101
```

```
PH <- factor(c(rep("low", 101), rep("mid", 101), rep("high", 101)))
x <- c(xseq, xseq, xseq)
```

Now we can use the **lines** function to plot the curves for each level of PH. The code is slightly different to earlier examples. First, instead of running **predict** and **lines** as two sequential calls we can nest predict within the lines function. Instead of specifying just the x-sequence values with the **list** argument as in earlier examples we now use

data.frame to specify that the model should use the long variables `x` and `PH` for prediction. We need to use the square indexing brackets to restrict each of the repeated lines commands to one level of `PH` restricting both `x` and `PH`:

```
lines(x[PH == "high"], predict(model2, type = "response", data.frame(biomass =
x, ph = PH))[PH == "high"])
lines(x[PH == "mid"], predict(model2, type = "response", data.frame(biomass = x,
ph = PH))[PH == "mid"])
lines(x[PH == "low"], predict(model2, type = "response", data.frame(biomass = x,
ph = PH))[PH == "low"])
```



```
detach(parkgrass)
```

A comparison of count of proportion analyses

In some cases the same data can be analysed with different approaches. In this example, taken from McCullagh & Nelder (1989), the same dataset is analysed first as a complex contingency table of counts and then later with a binomial analysis of proportions. The original data come from Schoener (1968) but were later re-analysed by others.

```
lizards<-
read.table("D:\\Andy\\_Teaching\\Statistics\\_R.GLMs\\poisson.lizards.schoener.t
xt", header=T) ; attach(lizards) ; names(lizards)
```

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

The data are counts (n) of the distribution of two species of Anolis lizard (opalinus and grahamii) in terms of when and where they were observed with regard to light level, height above the ground, perch thickness and time of day.

The first model to fit is one with a parameter for every data point (the names have replaced with shorter versions further down the table to save space):

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

Warning message:

```
Algorithm did not converge in: (if (is.empty.model(mt))
glm.fit.null else glm.fit)(x = X, y = Y,
```

Call:

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

Coefficients:

	Estimate	Std.Error	z-value	Pr(> z)
(Intercept)	1.386e+00	5.000e-01	2.773	0.006**
speciesOpalinus	6.818e-15	7.071e-01	9.64e-15	1.00
sunSun	9.163e-01	5.916e-01	1.549	0.12143
heightLow	-1.369e+01	2.847e+02	-0.048	0.9616
perchNarrow	-2.877e-01	7.638e-01	-0.377	0.7064
timeMid.day	-1.386e+00	1.118e+00	-1.240	0.2150
timeMorning	-6.931e-01	8.660e-01	-0.800	0.42349
sp0:sunS	5.878e-01	8.097e-01	0.726	0.46786
sp0:htL	1.479e+01	2.847e+02	0.052	0.95857
sunS:htL	1.248e+01	2.847e+02	0.044	0.96502
sp0:perNar	5.108e-01	1.017e+00	0.503	0.61530
sunS:perNar	6.454e-02	8.991e-01	0.072	0.94277
htL:perNar	1.259e+01	2.847e+02	0.044	0.96472
sp0:tMid	2.079e+00	1.275e+00	1.631	0.10284
sp0:tMorn	2.303e+00	1.025e+00	2.247	0.02463*
sunS:tMid	2.079e+00	1.183e+00	1.757	0.07884.
sunS:tMorn	7.885e-01	9.700e-01	0.813	0.41631

Statistics for free: Generalised Linear Modelling using R

htL:tMid	1.386e+00	4.026e+02	0.003	0.99725
htL:tMorn	6.931e-01	4.026e+02	0.002	0.99863
perNar:tMid	2.877e-01	1.607e+00	0.179	0.85795
perNar:tMorn	6.931e-01	1.190e+00	0.582	0.56032
sp0:sunS:htL	-1.391e+01	2.847e+02	-0.049	0.96103
sp0:sunS:perNar	-1.099e+00	1.200e+00	-0.916	0.35974
sp0:htL:perNar	-1.530e+01	2.847e+02	-0.054	0.95714
sunS:htL:perNar	-1.208e+01	2.847e+02	-0.042	0.96615
sp0:sunS:tMid	-1.429e+00	1.358e+00	-1.052	0.29283
sp0:sunS:tMorn	-1.762e+00	1.151e+00	-1.530	0.12598
sp0:htL:tMid	-2.485e+00	4.026e+02	-0.006	0.99508
sp0:htL:tMorn	-2.223e+00	4.026e+02	-0.006	0.99559
sunS:htL:tMid	-1.792e+00	4.026e+02	-0.004	0.99645
sunS:htL:tMorn	-2.776e-01	4.026e+02	-0.001	0.99945
sp0:perNar:tMid	-1.204e+00	1.846e+00	-0.652	0.51431
sp0:perNar:tMorn	-1.833e+00	1.429e+00	-1.283	0.19965
sunS:perNar:tMid	4.055e-01	1.700e+00	0.239	0.81147
sunS:perNar:tMorn	-1.598e-01	1.341e+00	-0.119	0.90514
htL:perNar:tMid	-1.259e+01	4.930e+02	-0.026	0.97963
htL:perNar:tMorn	-1.300e+01	4.930e+02	-0.026	0.97897
sp0:sunS:htL:perNar	1.442e+01	2.847e+02	0.051	0.95960
sp0:sunS:htL:tMid	2.989e+00	4.026e+02	0.007	0.99408
sp0:sunS:htL:tMorn	2.040e+00	4.026e+02	0.005	0.99596
sp0:sunS:perNar:tMid	1.182e+00	1.981e+00	0.597	0.55083
sp0:sunS:perNar:tMorn	1.417e+00	1.641e+00	0.864	0.38786
sp0:htL:perNar:tMid	1.609e+00	5.693e+02	0.003	0.99774
sp0:htL:perNar:tMorn	1.585e+01	4.930e+02	0.032	0.97436
sunS:htL:perNar:tMid	1.183e+01	4.930e+02	0.024	0.98085
sunS:htL:perNar:tMorn	1.057e+01	4.930e+02	0.021	0.98290
sp0:sunS:htL:pN:tMid	-1.307e+00	5.693e+02	-0.002	0.99817
sp0:sunS:htL:pN:tMor	-1.330e+01	4.931e+02	-0.027	0.97847

(Dispersion parameter for poisson family taken to be 1)

Null deviance: 7.3756e+02 on 47 degrees of freedom
 Residual deviance: 5.4480e-05 on 0 degrees of freedom
 AIC: 259.25

Number of Fisher Scoring iterations: 10

Because we have deliberately fitted a saturated model the model has no degrees of freedom and no explanatory power. We are interested in 15 interaction terms involving species and the other parameters are so-called nuisance parameters that must be retained to ensure the marginal totals are correctly constrained. Removing essential nuisance parameters is a danger of contingency table analysis and for that reason we also cannot use automated stepwise procedures since they could remove terms that must be retained.

```
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

The single 5-way interaction is not significant so we can accept model2 and then proceed to test the 4-way interactions:

```
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

```
model4 <- update(model2, ~. -species:sun:height:time) ;
anova(model2, model4, test = "Chi")
```

	Resid.	Df	Resid. Dev	Df	Deviance	P(> Chi)
1	2		0.00010			
2	4		0.44172	-2	-0.44162	0.80187

```
model5 <- update(model2, ~. -species:height:perch:time) ;
anova(model2, model5, test = "Chi")
```

	Resid.	Df	Resid. Dev	Df	Deviance	P(> Chi)
1	2		0.0001			
2	4		3.2219	-2	-3.2218	0.1997

```
model6 <- update(model2, ~. -species:sun:perch:time) ;
anova(model2, model6, test = "Chi")
```

	Resid.	Df	Resid. Dev	Df	Deviance	P(> Chi)
1	2		0.00010			
2	4		0.81016	-2	-0.81006	0.66696

They are all non-significant so we can remove them all (to save on typing just cut and paste from the code above) and make a new model 7 against which we can test the 3-way interactions of interest in the same way:

```
model7 <- update(model2, ~. -species:sun:height:perch-species:sun:height:time-
species:height:perch:time-species:sun:perch:time) ;
```

```
model8 <- update(model7, ~. -species:sun:perch) ;
anova(model7, model8, test = "Chi")
```

	Resid.	Df	Resid. Dev	Df	Deviance	P(> Chi)
1		9	8.5734			
2		10	8.5878	-1	-0.0144	0.9046

```
model9 <- update(model7, ~. -species:sun:height) ;
anova(model7, model9, test = "Chi")
```

	Resid.	Df	Resid. Dev	Df	Deviance	P(> Chi)
1		9	8.5734			
2		10	11.8678	-1	-3.2944	0.0695

```
model10 <- update(model7, ~. -species:sun:time) ;
anova(model7, model10, test = "Chi")
```

	Resid.	Df	Resid. Dev	Df	Deviance	P(> Chi)
1		9	8.5734			
2		11	10.9035	-2	-2.3301	0.3119

```
model11 <- update(model7, ~. -species:perch:time) ;
anova(model7, model11, test = "Chi")
```

	Resid.	Df	Resid. Dev	Df	Deviance	P(> Chi)
1		9	8.5734			
2		11	8.5916	-2	-0.0182	0.9910

```
model12 <- update(model7, ~. -species:height:perch) ;
anova(model7, model12, test = "Chi")
```

	Resid.	Df	Resid. Dev	Df	Deviance	P(> Chi)
1		9	8.5734			
2		10	8.8756	-1	-0.3022	0.5825

```
model13 <- update(model7, ~. -species:height:time) ;
anova(model7, model13, test = "Chi")
```

	Resid.	Df	Resid. Dev	Df	Deviance	P(> Chi)
1		9	8.5734			
2		11	9.9712	-2	-1.3978	0.4971

One term was marginal but for this exercise let's decide to omit marginal terms. Again, we can remove all the 3-way interactions, make a new benchmark model14 (again save typing but copying from above) and test the 2-way interactions:

```
model14 <- update(model7, ~. -species:sun:perch -species:sun:height -
species:sun:time -species:perch:time -species:height:perch -species:height:time)
```

```
model15 <- update(model14, ~. -species:time)
anova(model14, model15, test = "Chi")
```

	Resid.	Df	Resid. Dev	Df	Deviance	P(> Chi)
1	18		14.205			
2	20		25.803	-2	-11.598	0.003

```
model16 <- update(model14, ~. -species:height)
anova(model14, model16, test = "Chi")
```

	Resid.	Df	Resid. Dev	Df	Deviance	P(> Chi)
1	18		14.205			
2	19		36.272	-1	-22.067	2.633e-06

```
model17 <- update(model14, ~. -species:sun)
anova(model14, model17, test = "Chi")
```

	Resid.	Df	Resid. Dev	Df	Deviance	P(> Chi)
1	18		14.2052			
2	19		21.8922	-1	-7.6870	0.0056

```
model18 <- update(model14, ~. -species:perch)
anova(model14, model18, test = "Chi")
```

	Resid.	Df	Resid. Dev	Df	Deviance	P(> Chi)
1	18		14.2052			
2	19		27.3353	-1	-13.1301	0.0003

The significant 2-way interactions between species and the four putative niche axes support the hypothesis that the lizards are indeed differentiated with respect to these factors.

An alternative analysis by logistic regression

An alternative approach is available when it is possible to express the data not as counts but as proportions. We can then use a GLM with binomial errors and a simpler model that omits the nuisance parameters necessary only to constrain the marginal totals of the contingency table of counts. In this example we can replace the counts of the two lizard species found under the different conditions with the proportion of all lizards belonging to one of the two species.

```
Ag <- as.vector(tapply(n, list(sun, height, perch, time, species),
sum))[species=="grahamii"]
Ao <- as.vector(tapply(n, list(sun, height, perch, time, species),
sum))[species=="opalinus"]
total <- as.vector(tapply(n, list(sun, height, perch, time), sum))
Ag; Ao; total
```

We calculate the counts of the two species and the total as a check that everything adds up, as it does. We can now analyse the proportions of *A. grahamii* and *A. opalinus* after binding the two together as a compound response variable and creating shorter versions of all the response variables (see **Proportion data: Logistic Regression**)

```
pq <- cbind(Ag, Ao)
s <- sun[species=="grahamii"]
h <- height[species=="grahamii"]
p <- perch[species=="grahamii"]
t <- time[species=="grahamii"]
```

```
modelb <- glm(pq ~ s*h*p*t, binomial) ; summary(modelb)
```

Coefficients:

	Estimate	Std. Error	z value
Pr(> z)			
(Intercept)	2.3026	0.7416	3.105
0.00190 **			
sSun	-1.3218	1.0042	-1.316
0.18808			
hLow	-1.1741	0.8187	-1.434
0.15155			
pNarrow	11.0509	133.5274	0.083
0.93404			
tMid.day	-0.2231	1.2942	-0.172
0.86311			
tMorning	-2.3026	1.0247	-2.247
0.02463 *			
sSun:hLow	0.3185	1.1199	0.284
0.77613			
sSun:pNarrow	0.6511	193.8662	0.003
0.99732			
hLow:pNarrow	-10.3549	133.5287	-0.078
0.93819			
sSun:tMid.day	0.6286	1.8394	0.342
0.73254			
sSun:tMorning	1.8326	1.4289	1.283
0.19965			
hLow:tMid.day	0.3331	1.3637	0.244
0.80706			
hLow:tMorning	1.7619	1.1515	1.530
0.12598			
pNarrow:tMid.day	-0.2063	191.5384	-0.001
0.99914			
pNarrow:tMorning	2.2302	189.1970	0.012
0.99060			
sSun:hLow:pNarrow	1.0126	193.8703	0.005

```

0.99583
sSun:hLow:tMid.day      -0.2351      1.9343   -0.122
0.90327
sSun:hLow:tMorning      -1.4171      1.6410   -0.864
0.38786
sSun:pNarrow:tMid.day   -2.2398      1.4797   -1.514
0.13011
sSun:pNarrow:tMorning   -14.4430     235.6959  -0.061
0.95114
hLow:pNarrow:tMid.day    0.8929      191.5402   0.005
0.99628
hLow:pNarrow:tMorning    -2.0477      189.1994  -0.011
0.99136
sSun:hLow:pNarrow:tMorning 11.9008      235.7020   0.050
0.95973
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

(Dispersion parameter for binomial family taken to be 1)

```

Null deviance: 7.0102e+01 on 22 degrees of freedom
Residual deviance: 1.5849e-04 on 0 degrees of freedom
AIC: 102.82

```

We are still faced with a complex multiple regression of observational data and one pragmatic option is to use the stepwise backwards deletion approach to model simplification (see *Statistics for Free I; Multiple Regression*) and, starting with the 4-way interaction, remove each term from the model in sequence while retaining all other terms at the same of lower levels. However, purely to save a bit of time here we can use the automated step function as a short cut (I have edited the output slightly as there is a lot of it):

```
step(modelb)
```

Start: AIC= 102.82

```
pq ~ s * h * p * t
```

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

Step: AIC= 100.82

```

pq ~ s + h + p + t + s:h + s:p + h:p + s:t + h:t + p:t +
s:h:p +
s:h:t + s:p:t + h:p:t

```

Statistics for free: Generalised Linear Modelling using R

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

Step: AIC= 97.27

pq ~ s + h + p + t + s:h + s:p + h:p + s:t + h:t + p:t +
s:h:p + s:h:t + s:p:t

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

Step: AIC= 93.9

pq ~ s + h + p + t + s:h + s:p + h:p + s:t + h:t + p:t +
s:h:p + s:p:t

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

Step: AIC= 92.16

pq ~ s + h + p + t + s:h + s:p + h:p + s:t + p:t + s:h:p +
s:p:t

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

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

output edited

Degrees of Freedom: 22 Total (i.e. Null); 7 Residual

Null Deviance: 70.1

Residual Deviance: 3.34 AIC: 92.16

Warning messages: # output edited

The automated step function removes several interaction terms (s:h:p:t, h:p:t,

s:h:t, h:t) but results in a quite complex model with two 3-way interactions and five 2-way interactions. We have noted before that model simplification using the AIC tends to be generous in leaving terms in the model. We can now pick up the procedure by hand and test the remaining terms by stepwise backwards deletion. We begin by copying the final model from the step function above as our modelb1 and testing the remaining 3-way interactions:

```
model1 <- glm(pq ~ s+h+p+t + s:h + s:p + h:p + s:t + p:t +s:h:p + s:p:t, binomial)
```

```
model2 <- update(model1, ~. -s:p:t)
model3 <- update(model1, ~. -s:h:p)
anova(model1, model2, test="Chi") ;
anova(model1, model3, test="Chi") ;
```

Analysis of Deviance Table

```
Model 1:pq ~ s+h+p+t +s:h +s:p +h:p +s:t +p:t +s:h:p +s:p:t
Model 2:pq ~ s +h +p +t +s:h +s:p +h:p +s:t +p:t +s:h:p
```

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

Analysis of Deviance Table

```
Model 1:pq ~ s+h+p+t +s:h +s:p +h:p +s:t +p:t +s:h:p +s:p:t
Model 2:pq ~ s +h +p +t +s:h +s:p +h:p +s:t +p:t +s:p:t
```

	Resid.Df	Resid.Dev	Df	Deviance	P(> Chi)
1	7	3.3405			
2	8	5.8273	-1	-2.4868	0.1148

The first interaction is marginal but for our purposes here let's decide to be strict and to drop both of the remaining 3-way interactions. We can now move on to test the 2-ways:

```
model4 <- glm(pq ~ s+h+p+t + s:h + s:p + h:p + s:t + p:t, binomial)
```

```
model5 <- update(model4, ~. -s:h)
model6 <- update(model4, ~. -s:p)
model7 <- update(model4, ~. -h:p)
model8 <- update(model4, ~. -s:t)
model9 <- update(model4, ~. -p:t)
```

```
anova(model4, model5, test="Chi") ;
anova(model4, model6, test="Chi") ;
anova(model4, model7, test="Chi") ;
```

```
anova(model4, model8, test="Chi") ;
anova(model4, model9, test="Chi") ;
```

Analysis of Deviance Table

```
Model 1: pq ~ s + h + p + t + s:h + s:p + h:p + s:t + p:t
Model 2: pq ~ s + h + p + t + s:p + h:p + s:t + p:t
```

	Resid.Df	Resid.Dev	Df	Deviance	P(> Chi)
1	10	10.9032			
2	11	10.9268	-1	-0.0236	0.8779

Analysis of Deviance Table

```
Model 1: pq ~ s + h + p + t + s:h + s:p + h:p + s:t + p:t
Model 2: pq ~ s + h + p + t + s:h + h:p + s:t + p:t
```

	Resid.Df	Resid.Dev	Df	Deviance	P(> Chi)
1	10	10.9032			
2	11	11.1432	-1	-0.2401	0.6242

Analysis of Deviance Table

```
Model 1: pq ~ s + h + p + t + s:h + s:p + h:p + s:t + p:t
Model 2: pq ~ s + h + p + t + s:h + s:p + s:t + p:t
```

	Resid.Df	Resid.Dev	Df	Deviance	P(> Chi)
1	10	10.9032			
2	11	13.2543	-1	-2.3511	0.1252

Analysis of Deviance Table

```
Model 1: pq ~ s + h + p + t + s:h + s:p + h:p + s:t + p:t
Model 2: pq ~ s + h + p + t + s:h + s:p + h:p + p:t
```

	Resid.Df	Resid.Dev	Df	Deviance	P(> Chi)
1	10	10.9032			
2	12	10.9090	-2	-0.0058	0.9971

Analysis of Deviance Table

```
Model 1: pq ~ s + h + p + t + s:h + s:p + h:p + s:t + p:t
Model 2: pq ~ s + h + p + t + s:h + s:p + h:p + s:t
```

	Resid. Df	Resid. Dev	Df	Deviance	P(> Chi)
1	10	10.9032			
2	12	11.7600	-2	-0.8568	0.6516

None of the 2-way interactions are significant so we can finish by testing the four main effects:

```
model10 <- glm(pq ~ s+h+p+t, binomial)
```

```
model11 <- update(model10, ~. -s)
model12 <- update(model10, ~. -h)
model13 <- update(model10, ~. -p)
model14 <- update(model10, ~. -t)
```

```
anova(model10, model11, test="Chi") ;
anova(model10, model12, test="Chi") ;
anova(model10, model13, test="Chi") ;
anova(model10, model14, test="Chi") ;
```

Analysis of Deviance Table

```
Model 1: pq ~ s + h + p + t
Model 2: pq ~ h + p + t
```

	Resid.Df	Resid.Dev	Df	Deviance	P(> Chi)
1	17	14.2046			
2	18	27.3346	-1	-13.1300	0.0003

Analysis of Deviance Table

```
Model 1: pq ~ s + h + p + t
Model 2: pq ~ s + p + t
```

	Resid.Df	Resid.Dev	Df	Deviance	P(> Chi)
1	17	14.2046			
2	18	21.8917	-1	-7.6871	0.0056

Analysis of Deviance Table

```
Model 1: pq ~ s + h + p + t
Model 2: pq ~ s + h + t
```

	Resid.Df	Resid.Dev	Df	Deviance	P(> Chi)
1	17	14.205			
2	18	36.271	-1	-22.066	2.634e-06

Analysis of Deviance Table

```
Model 1: pq ~ s + h + p + t
Model 2: pq ~ s + h + p
```

Statistics for free: Generalised Linear Modelling using R

	Resid.Df	Resid.Dev	Df	Deviance	P(> Chi)
1	17	14.205			
2	19	25.802	-2	-11.597	0.003

So, it turns out that the minimal adequate model produced by stepwise backwards deletion contains only the four main effects:

```
mam <- glm(pq ~ s+h+p+t, binomial) ; summary(mam)
```

Call:

```
glm(formula = pq ~ s + h + p + t, family = binomial)
```

Deviance Residuals:

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

Coefficients:

	Estimate	Std. Error	z value	Pr(> z)	
(Intercept)	1.9447	0.3413	5.698	1.22e-08	***
sS	-0.7626	0.2112	-3.610	0.000306	***
hLow	-0.8473	0.3222	-2.629	0.008554	**
pNarrow	1.1300	0.2570	4.397	1.10e-05	***
tMid.day	0.2271	0.2501	0.908	0.363913	
tMorning	-0.7368	0.2989	-2.465	0.013712	*

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(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

Number of Fisher Scoring iterations: 3

The residual deviance is not greater than the residual degrees of freedom so the model is not over-dispersed. If, following McCullagh & Nelder (1989), we want to take the model simplification process further we could see if we can remove the non-significant midday level of time (see the estimates above). We can try replacing time (t) with a new two-level factor, t2:

```
t2 <- factor(1 + (t=="Morning")) ; t2
alt.mam <- glm(pq ~ s+h+p+t2, binomial) ; summary(alt.mam)
anova(mam, alt.mam, test="Chi")
```

Call:

```
glm(formula = pq ~ s + h + p + t2, family = binomial)
```

Coefficients:

	Estimate	Std. Error	z value	Pr(> z)	
(Intercept)	2.0312	0.3299	6.156	7.45e-10	***
sS	-0.7485	0.2104	-3.557	0.000375	***
hLow	-0.7871	0.3158	-2.493	0.012682	*
pNarrow	1.1188	0.2565	4.362	1.29e-05	***
t22	-0.8717	0.2611	-3.339	0.000842	***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(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

Number of Fisher Scoring iterations: 3

Analysis of Deviance Table

Model 1: pq ~ s + h + p + t

Model 2: pq ~ s + h + p + t2

	Resid.Df	Resid.Dev	Df	Deviance	P(> Chi)
1	17	14.2046			
2	18	15.0232	-1	-0.8186	0.3656

The change in deviance due to the simplification is not significant and all parameters in the model are significant so if we wanted to follow the model simplification approach as far as we can then we could finish with this alternative minimal adequate model. The analysis suggests that the two Anolis lizards do differ in their niches as described by the explanatory variables but there were no complex combinations of factors (e.g. a switch in height depending on time of day) as shown by the lack of significant interactions.

```
detach(lizards)
```

References

- Crawley, M. J. 2002. Statistical computing: An introduction to data analysis using S-Plus.
- John Wiley and Sons Ltd.
- McCullagh, P. and Nelder, J. A. 1989. Generalized Linear Models. - Chapman and Hall.
- Krause, A. & Olsen, M. (2002) The basics of S-Plus Springer-Verlag, Berlin.
- Venables, W.N. & Ripley, B.D. (2002) Modern applied statistics with S Springer-Verlag,
Berlin.

Exercises

Log-linear regression:

Source: Trad. !

Data file: poisson.horsekicks.bortkewitsch.txt

Exercise: Did deaths from horse kicks vary randomly between cavalry corps, or were some more careless than others?

The following worked example is a real dataset analysed in Quinn & Keough (2002). Compare your analysis to that of Quinn and Keough and the original papers.

Log-linear regression:

Source: Sinclair & Arcese (1995)

Data file: poisson.wildebeest.sinclair.txt

Exercise: Do these counts wildebeest carcasses provide evidence for a link between probability of being predated and sex or bone marrow type?

Solution: Quinn & Keough pages 390 - 391.