14

Count Data

Up to this point, the response variables have all been continuous measurements such as weights, heights, lengths, temperatures, and growth rates. 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, and 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). In this chapter we deal with 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. lightning strikes, bankruptcies, deaths, births). This is in contrast to count data on **proportions**, where we know the number doing a particular thing, but also the number not doing that thing (e.g. the proportion dying, sex ratios at birth, proportions of different groups responding to a questionnaire).

Straightforward linear regression methods (assuming constant variance, normal errors) are not appropriate for count data for four 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.

In R, count data are handled very elegantly in a generalized linear model by specifying family=poisson which sets errors = Poisson and link = log (see p. 515). 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 that are equal to their means.

A Regression with Poisson Errors

The following example has a count (the number of reported cancer cases per year per clinic) as the response variable, and a single continuous explanatory variable (the distance from a

nuclear plant to the clinic in km). The question is whether or not proximity to the reactor affects the number of cancer cases.

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

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

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

There seems to be a downward trend in cancer cases with distance (see the plot below). But is the trend significant? We do a regression of cases against distance, using a GLM with Poisson errors:

```
model1<-glm(Cancers~Distance,poisson)
summary(model1)
```

```
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  .

(Dispersion parameter for poisson family taken to be 1)

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

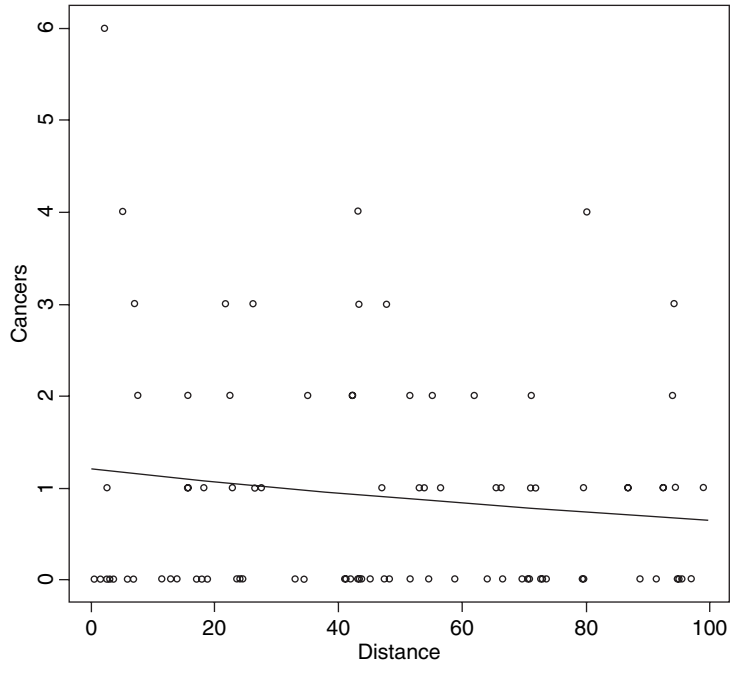
The trend does not look to be significant, but look at the residual deviance. It is assumed that this is the same as the residual degrees of freedom. The fact that residual deviance is larger than residual degrees of freedom indicates that we have overdispersion (extra, unexplained variation in the response). We compensate for the overdispersion by refitting the model using quasi-Poisson rather than Poisson errors:

```
model2<-glm(Cancers~Distance,quasipoisson)
summary(model2)
```

```
Coefficients:

             Estimate  Std. Error  t value  Pr(>|t|)
(Intercept)   0.186865   0.235341   0.794     0.429
Distance     -0.006138   0.004573  -1.342     0.183

(Dispersion parameter for quasipoisson family taken to be 1.555271)

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

Compensating for the overdispersion has increased the p value to 0.183, so there is no compelling evidence to support the existence of a trend in cancer incidence with distance from the nuclear plant. To draw the fitted model through the data, you need to understand that the GLM with Poisson errors uses the log link, so the parameter estimates and the predictions from the model (the 'linear predictor') are in logs, and need to be antilogged before the (non-significant) fitted line is drawn.

```
xv<-seq(0,100,.1
yv<-predict(model2,list(Distance=xv))
lines(xv,exp(yv))
```



Analysis of Deviance with Count Data

In our next example the response variable is a count of infected blood cells per mm^2 on microscope slides prepared from randomly selected individuals. The explanatory variables are smoker (logical, yes or no), age (three levels, under 20, 21 to 59, 60 and over), sex (male or female) and body mass score (three levels, normal, overweight, obese).

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

```
[1] "cells"  "smoker"  "age"  "sex"  "weight"
```

It is always a good idea with count data to get a feel for the overall frequency distribution of counts using table:

```
table(cells)
```

```
  0   1   2   3   4   5  6  7
314  75  50  32  18  13  7  2
```

Most subjects (314 of them) showed no damaged cells, and the maximum of 7 was observed in just two patients.

We begin data inspection by tabulating the main effect means:

```
tapply(cells,smoker,mean)
```

```
    FALSE       TRUE
0.5478723  1.9111111
```

```
tapply(cells,weight,mean)
```

```
   normal      obese       over
0.5833333  1.2814371  0.9357143
```

```
tapply(cells,sex,mean)
```

```
   female       male
0.6584507  1.2202643
```

```
tapply(cells,age,mean)
```

```
      mid        old      young
0.8676471  0.7835821  1.2710280
```

It looks as if smokers have a substantially higher mean count than non-smokers, that overweight and obese subjects had higher counts than normal weight, males had a higher count that females, and young subjects had a higher mean count than middle-aged or older people. We need to test whether any of these differences are significant and to assess whether there are interactions between the explanatory variables.

```
model1<-glm(cells~smoker*sex*age*weight,poisson)
summary(model1)
```

```
    Null deviance: 1052.95 on 510 degrees of freedom
Residual deviance: 736.33 on 477 degrees of freedom
AIC: 1318

Number of Fisher Scoring iterations: 6
```

The residual deviance (736.33) is much greater than the residual degrees of freedom (477), indicating overdispersion, so before interpreting any of the effects, we should refit the model using quasi-Poisson errors:

```
model2<-glm(cells~smoker*sex*age*weight,quasipoisson)
summary(model2)
```

```
Call:
glm(formula = cells ~ smoker * sex * age * weight, family = quasipoisson)

Deviance Residuals:
   Min      1Q  Median      3Q     Max
-2.236  -1.022  -0.851   0.520   3.760

Coefficients: (2 not defined because of singularities)
Estimate             Std. Error  t value  Pr(>|t|)
(Intercept)            -0.8329   0.4307    -1.934  0.0537  .
smokerTRUE             -0.1787   0.8057    -0.222  0.8246
sexmale                 0.1823   0.5831     0.313  0.7547
ageold                 -0.1830   0.5233    -0.350  0.7267
ageyoung                0.1398   0.6712     0.208  0.8351
weightobese             1.2384   0.8965     1.381  0.1678
weightover             -0.5534   1.4284    -0.387  0.6986
smokerTRUE:sexmale      0.8293   0.9630     0.861  0.3896
smokerTRUE:ageold      -1.7227   2.4243    -0.711  0.4777
```

```
smokerTRUE:ageyoung              1.1232  1.0584   1.061  0.2892
sexmale:ageold                  -0.2650  0.9445  -0.281  0.7791
sexmale:ageyoung                -0.2776  0.9879  -0.281  0.7788
smokerTRUE:weightobese           3.5689  1.9053   1.873  0.0617  .
smokerTRUE:weightover            2.2581  1.8524   1.219  0.2234
sexmale:weightobese             -1.1583  1.0493  -1.104  0.2702
sexmale:weightover               0.7985  1.5256   0.523  0.6009
ageold:weightobese              -0.9280  0.9687  -0.958  0.3386
ageyoung:weightobese            -1.2384  1.7098  -0.724  0.4693
ageold:weightover                1.0013  1.4776   0.678  0.4983
ageyoung:weightover              0.5534  1.7980   0.308  0.7584
smokerTRUE:sexmale:ageold        1.8342  2.1827   0.840  0.4011
smokerTRUE:sexmale:ageyoung     -0.8249  1.3558  -0.608  0.5432
smokerTRUE:sexmale:              -2.2379  1.7788  -1.258  0.2090
weightobese
smokerTRUE:sexmale:weightover   -2.5033  2.1120  -1.185  0.2365
smokerTRUE:ageold:               0.8298  3.3269   0.249  0.8031
weightobese
smokerTRUE:ageyoung:            -2.2108  1.0865  -2.035  0.0424  *
weightobese
smokerTRUE:ageold:               1.1275  1.6897   0.667  0.5049
weightover
smokerTRUE:ageyoung:weightover  -1.6156  2.2168  -0.729  0.4665
sexmale:ageold:weightobese       2.2210  1.3318   1.668  0.0960  .
sexmale:ageyoung:weightobese     2.5346  1.9488   1.301  0.1940
sexmale:ageold:weightover       -1.0641  1.9650  -0.542  0.5884
sexmale:ageyoung:weightover     -1.1087  2.1234  -0.522  0.6018
smokerTRUE:sexmale:ageold:      -1.6169  3.0561  -0.529  0.5970
weightobese
smokerTRUE:sexmale:ageyoung          NA      NA      NA      NA
weightobese
smokerTRUE:sexmale:ageold:           NA      NA      NA      NA
weightover
smokerTRUE:sexmale:ageyoung:     2.4160  2.6846   0.900  0.3686
weightover

(Dispersion parameter for quasipoisson family taken to be 1.854815)

    Null deviance: 1052.95 on 510 degrees of freedom
Residual deviance: 736.33 on 477 degrees of freedom
AIC: NA

Number of Fisher Scoring iterations: 6
```

There is an apparently significant three-way interaction between smoking, age and obesity ($p = 0.0424$). There were too few subjects to assess the four-way interaction (see the NAs in the table), so we begin model simplification by removing the highest-order interaction:

```
model3<-update(model2, ~. -smoker:sex:age:weight)
summary(model3)
```

```
Call:
glm(formula = cells ~ smoker + sex + age + weight + smoker:sex +
    smoker:age + sex:age + smoker:weight + sex:weight + age:weight +
```

```
    smoker:sex:age + smoker:sex:weight + smoker:age:weight +
    sex:age:weight, family = quasipoisson)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.2442  -1.0477  -0.8921   0.5195   3.7613

Coefficients:
                                 Estimate  Std. Error  t value  Pr(>|t|)
(Intercept)                     -0.897195   0.436988   -2.053   0.04060  *
smokerTRUE                       0.030263   0.735386    0.041   0.96719
sexmale                          0.297192   0.570009    0.521   0.60234
ageold                          -0.118726   0.528165   -0.225   0.82224
ageyoung                         0.289259   0.639618    0.452   0.65130
weightobese                      1.302660   0.898307    1.450   0.14768
weightover                      -0.005052   1.027198   -0.005   0.99608
smokerTRUE:sexmale               0.527345   0.867294    0.608   0.54345
smokerTRUE:ageold               -0.566584   1.700590   -0.333   0.73915
smokerTRUE:ageyoung              0.757297   0.939746    0.806   0.42073
sexmale:ageold                  -0.379884   0.935365   -0.406   0.68483
sexmale:ageyoung                -0.610703   0.920969   -0.663   0.50758
smokerTRUE:weightobese           3.924591   1.475476    2.660   0.00808  **
smokerTRUE:weightover            1.192159   1.259888    0.946   0.34450
sexmale:weightobese             -1.273202   1.040701   -1.223   0.22178
sexmale:weightover               0.154097   1.098781    0.140   0.88853
ageold:weightobese              -0.993355   0.970484   -1.024   0.30656
ageyoung:weightobese            -1.346913   1.459454   -0.923   0.35653
ageold:weightover                0.454217   1.090260    0.417   0.67715
ageyoung:weightover             -0.483955   1.300866   -0.372   0.71004
smokerTRUE:sexmale:ageold        0.771116   1.451512    0.531   0.59549
smokerTRUE:sexmale:ageyoung     -0.210317   1.140384   -0.184   0.85376
smokerTRUE:sexmale:weightobese  -2.500668   1.369941   -1.825   0.06857  .
smokerTRUE:sexmale:weightover   -1.110222   1.217531   -0.912   0.36230
smokerTRUE:ageold:weightobese   -0.882951   1.187871   -0.743   0.45766
smokerTRUE:ageyoung:weightobese -2.453315   1.047067   -2.343   0.01954  *
smokerTRUE:ageold:weightover     0.823018   1.528233    0.539   0.59045
smokerTRUE:ageyoung:weightover   0.040795   1.223664    0.033   0.97342
sexmale:ageold:weightobese       2.338617   1.324805    1.765   0.07816  .
sexmale:ageyoung:weightobese     2.822032   1.623849    1.738   0.08288  .
sexmale:ageold:weightover       -0.442066   1.545451   -0.286   0.77497
sexmale:ageyoung:weightover      0.357807   1.291194    0.277   0.78181

(Dispersion parameter for quasipoisson family taken to be 1.847991)

    Null deviance: 1052.95 on 510 degrees of freedom
Residual deviance: 737.87 on 479 degrees of freedom
AIC: NA

Number of Fisher Scoring iterations: 6
```

The remaining model simplification is left to you as an exercise. Your minimal adequate model might look something like this:

```
summary(model18)

Call:
glm(formula = cells ~ smoker + weight + smoker:weight, family =
quasipoisson)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.6511  -1.1742  -0.9148   0.5533   3.6436
```

```
Coefficients:
                       Estimate  Std. Error  t value  Pr(>|t|)
(Intercept)             -0.8712      0.1760   -4.950  1.01e-06  ***
smokerTRUE               0.8224      0.2479    3.318  0.000973  ***
weightobese              0.4993      0.2260    2.209  0.027598  *
weightover               0.2618      0.2522    1.038  0.299723
smokerTRUE:weightobese   0.8063      0.3105    2.597  0.009675  **
smokerTRUE:weightover    0.4935      0.3442    1.434  0.152226

(Dispersion parameter for quasipoisson family taken to be 1.827927)

    Null deviance: 1052.95 on 510 degrees of freedom
Residual deviance: 737.87 on 479 degrees of freedom
AIC: NA
Number of Fisher Scoring iterations: 6
```

This model shows a highly significant interaction between smoking and weight in determining the number of damaged cells, but there are no convincing effects of age or sex. In a case like this, it is useful to produce a summary table to highlight the effects:

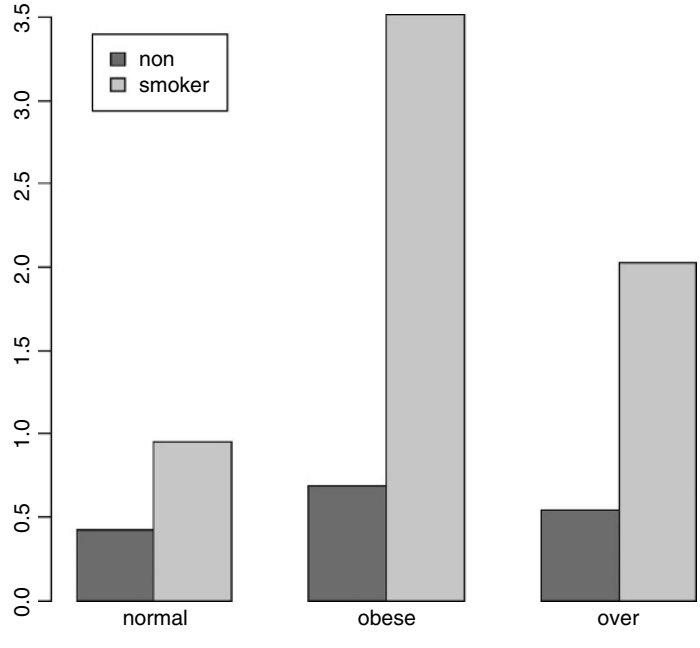
```
tapply (cells,list(smoker,weight),mean)
```

```
          normal     obese       over
FALSE  0.4184397  0.689394  0.5436893
TRUE   0.9523810  3.514286  2.0270270
```

The interaction arises because the response to smoking depends on body weight: smoking adds a mean of about 0.5 damaged cells for individuals with normal body weight, but adds 2.8 damaged cells for obese people.

It is straightforward to turn the summary table into a barplot:

```
barplot(tapply(cells,list(smoker,weight),mean),col=c(2,7),beside=T)
legend(1.2,3.4,c("non","smoker"),fill=c(2,7))
```



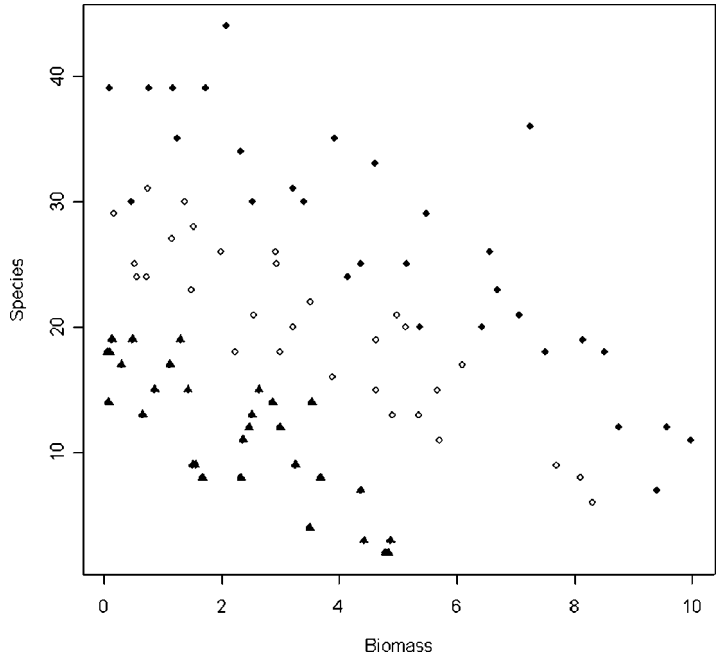
Analysis of Covariance with Count Data

In this next example the response is a count of the number of plant species on plots that have different biomass (a continuous explanatory variable) and different soil pH (a categorical variable with three levels: high, mid and low).

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

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

```
plot(Biomass,Species,type="n")
spp<-split(Species,pH)
bio<-split(Biomass,pH)
points(bio[[1]],spp[[1]],pch=16)
points(bio[[2]],spp[[2]],pch=17)
points(bio[[3]],spp[[3]])
```



Note the use of split to create separate lists of plotting coordinates for the three levels of pH. It is clear that Species declines with Biomass, and that soil pH has a big effect on Species, but does the slope of the relationship between Species and Biomass depend on pH? The lines look reasonably parallel from the scatterplot. This is a question about interaction effects, and in analysis of covariance, interaction effects are about differences between slopes:

```
model1<-glm(Species~ Biomass*pH,poisson)
summary(model1)
```

```
Coefficients:
             Estimate  Std. Error  z value  Pr(>|z|)
(Intercept)   3.76812     0.06153   61.240    < 2e-16  ***
Biomass      -0.10713     0.01249   -8.577    < 2e-16  ***
pHlow        -0.81557     0.10284   -7.931  2.18e-15  ***
pHmid        -0.33146     0.09217   -3.596  0.000323  ***
Biomass:pHlow -0.15503    0.04003   -3.873  0.000108  ***
Biomass:pHmid -0.03189    0.02308   -1.382  0.166954
(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: 4
```

We can test for the need for different slopes by comparing this maximal model (with six parameters) with a simpler model with different intercepts but the same slope (four parameters):

```
model2<-glm(Species~Biomass+pH,poisson)
anova(model1,model2,test="Chi")
```

```
Analysis of Deviance Table

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

The slopes are very significantly different ($p = 0.000\,33$), so we are justified in retaining the more complicated model1.

Finally, we draw the fitted lines through the scatterplot, using predict:

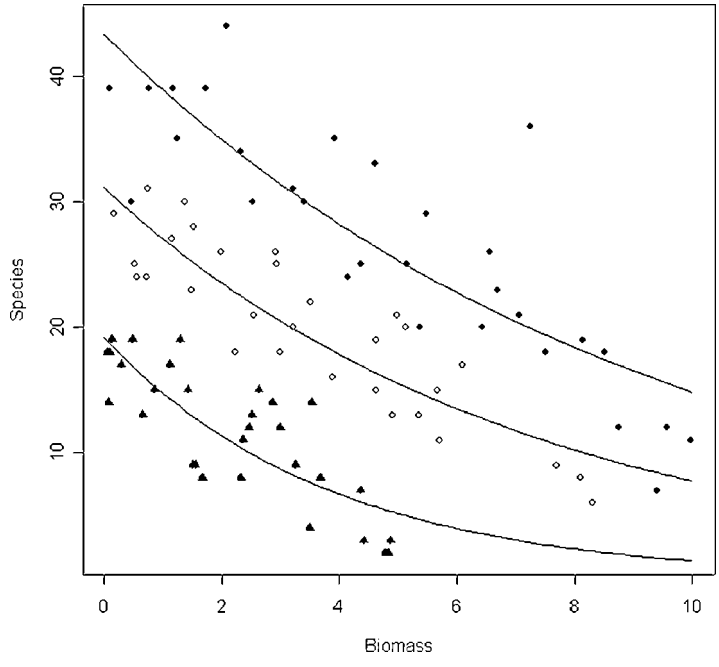
```
xv<-seq(0,10,0.1)
levels(pH)
```

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

```
length(xv)
```

```
[1] 101
```

```
phv<-rep("high",101)
yv<-predict(model1,list(pH=factor(phv),Biomass=xv),type="response")
lines(xv,yv)
phv<-rep("mid",101)
yv<-predict(model1,list(pH=factor(phv),Biomass=xv),type="response")
lines(xv,yv)
phv<-rep("low",101)
yv<-predict(model1,list(pH=factor(phv),Biomass=xv),type="response")
lines(xv,yv)
```



Note the use of type="response" in the predict function. This ensures that *yv* is calculated as Species rather than log(Species), and means we do not need to back-transform using antilogs before drawing the lines (compare with the example on p. 579). You could make the R code more elegant by writing a function to plot any number of lines, depending on the number of levels of the factor (three levels of pH in this case).

Frequency Distributions

Here are data on the numbers of bankruptcies in 80 districts. The question is whether there is any evidence that some districts show greater than expected numbers of cases. What would we expect? Of course we should expect some variation, but how much, exactly? Well that depends on our model of the process. Perhaps the simplest model is that absolutely nothing is going on, and that every singly bankruptcy case is absolutely independent of every other. That leads to the prediction that the numbers of cases per district will follow a Poisson process, a distribution in which the variance is equal to the mean (see p. 250). Let's see what the data show.

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

```
[1] "cases"
```

First we need to count the numbers of districts with no cases, one case, two cases, and so on. The R function that does this is called table:

```
frequencies<-table(cases)
frequencies
```

```
cases
 0   1   2  3  4  5  6  7  8  9  10
34  14  10  7  4  5  2  1  1  1   1
```

There were no cases at all in 34 districts, but one district had 10 cases. A good way to proceed is to compare our distribution (called frequencies) with the distribution that would be observed if the data really did come from a Poisson distribution as postulated by our model. We can use the R function dpois to compute the probability density of each of the 11 frequencies from 0 to 10 (we multiply the probability produced by dpois by the total sample of 80 to obtain the predicted frequencies). We need to calculate the mean number of cases per district: this is the Poisson distribution's only parameter:

```
mean(cases)
```

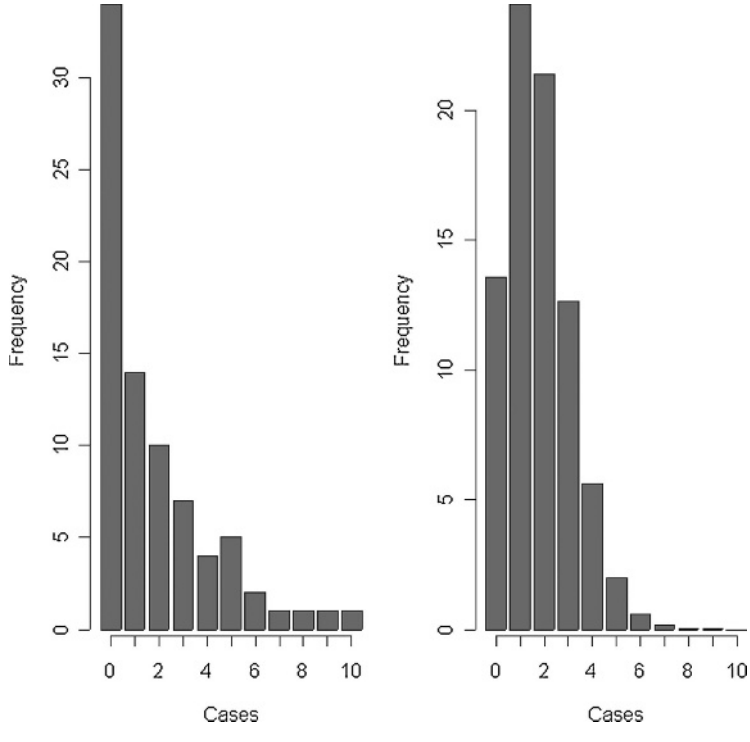
```
[1] 1.775
```

The plan is to draw two distributions side by side, so we set up the plotting region:

```
par(mfrow=c(1,2))
```

Now we plot the observed frequencies in the left-hand panel and the predicted, Poisson frequencies in the right-hand panel:

```
barplot(frequencies,ylab="Frequency",xlab="Cases",col="red")
barplot(dpois(0:10,1.775)*80,names=as.character(0:10),
                              ylab="Frequency",xlab="Cases",col="red")
```



The distributions are very different: the mode of the observed data is 0, but the mode of the Poisson with the same mean is 1; the observed data contained examples of 8, 9 and 10 cases, but these would be highly unlikely under a Poisson process. We would say that the observed data are highly **aggregated**; they have a variance–mean ratio much greater than 1 (the Poisson distribution, of course, has a variance–mean ratio of 1):

```
var(cases)/mean(cases)
```

```
[1] 2.99483
```

So, if the data are not Poisson distributed, how are they distributed? A good candidate distribution where the variance–mean ratio is this big (*c*. 3.0) is the negative binomial distribution (see p. 252). This is a two-paramter distribution: the first parameter is the mean number of cases (1.775), and the second is called the clumping parameter, k (measuring the degree of aggregation in the data: small values of $k(k < 1)$ show high aggregation, while large values of $k(k > 5)$ show randomness). We can get an approximate estimate of the magnitude of k from

$$\hat{k} = \frac{\overline{x}^2}{s^2 - \overline{x}}.$$

We can work this out:

```
mean(cases)^2/(var(cases)-mean(cases))
```

```
[1] 0.8898003
```

so we shall work with $k = 0.89$. How do we compute the expected frequencies? The density function for the negative binomial distribution is dnbinom and it has three arguments: the frequency for which we want the probability (in our case 0 to 10), the number of successes (in our case 1), and the mean number of cases (1.775); we multiply by the total number of cases (80) to obtain the expected frequencies

```
exp<-dnbinom(0:10,1,mu=1.775)*80
```

We will draw a single figure in which the observed and expected frequencies are drawn side by side. The trick is to produce a new vector (called both) which is twice as long as the observed and expected frequency vectors ($2 \times 11 = 22$). Then, we put the observed frequencies in the odd-numbered elements (using modulo 2 to calculate the values of the subscripts), and the expected frequencies in the even-numbered elements:

```
both<-numeric(22)
both[1:22 %% 2 != 0]<-frequencies
both[1:22 %% 2 == 0]<-exp
```

On the x axis, we intend to label only every other bar:

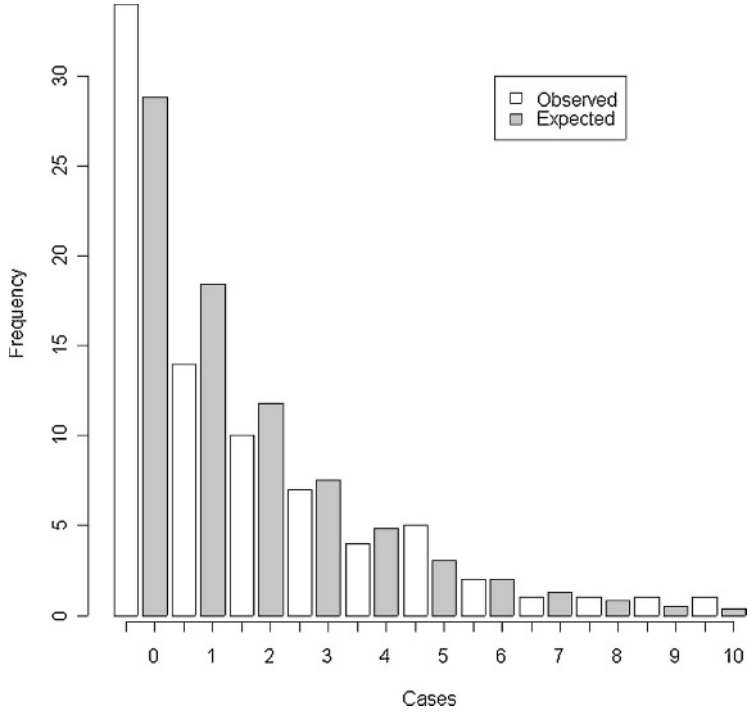
```
labels<-character(22)
labels[1:22 %% 2 == 0]<-as.character(0:10)
```

Now we can produce the barplot, using white for the observed frequencies and grey for the negative binomial frequencies:

```
par(mfrow=c(1,1))
barplot(both,col=rep(c("white","grey"),11),names=labels,ylab="Frequency",
        xlab="Cases")
```

Now we need to add a legend to show what the two colours of the bars mean. You can locate the legend by trial and error, or by left-clicking mouse when the cursor is in the correct position, using the locator(1) function (see p. 257):

```
legend(16,30,c("Observed","Expected"), fill=c("white","grey"))
```



The fit to the negative binomial distribution is much better than it was with the Poisson distribution, especially in the right-hand tail. But the observed data have too many 0s and too few 1s to be represented perfectly by a negative binomial distribution. If you want to quantify the lack of fit between the observed and expected frequency distributions, you can calculate Pearson's chi-squared $\sum (O-E)^2/E$ based on the number of comparisons that have expected frequency greater than 4:

```
exp
```

```
[1]  28.8288288  18.4400617  11.7949944  7.5445460  4.8257907  3.0867670
[7]   1.9744185   1.2629164   0.8078114  0.5167082  0.3305070
```

If we accumulate the rightmost six frequencies, then all the values of exp will be bigger than 4. The degrees of freedom are then given by the number of comparisons (6) - the number of parameters estimated from the data (2 in our case) -1 (for contingency, because the total frequency must add up to 80) $= 3$. We use a gets arrow to reduce the lengths of the observed and expected vectors, creating an upper interval called 5+ for '5 or more':

```
cs<-factor(0:10)
levels(cs)[6:11]<-"5+"
levels(cs)
```

```
[1] "0"  "1"  "2"  "3"  "4"  "5+"
```

Now make the two shorter vectors 'of' and 'ef' (for observed and expected frequencies):

```
ef<-as.vector(tapply(exp,cs,sum))
of<-as.vector(tapply(frequencies,cs,sum))
```

Finally we can compute the chi-squared value measuring the difference between the observed and expected frequency distributions, and use 1-pchisq to work out the p value:

```
sum((of-ef)^2/ef)
```

```
[1] 3.594145
```

```
1-pchisq(3.594145,3)
```

```
[1] 0.3087555
```

We conclude that a negative binomial description of these data is reasonable (the observed and expected distributions are not significantly different; $p = 0.31$).

Overdispersion in Log-linear Models

The data analysed in this section refer to children from Walgett, New South Wales, Australia, who were classified by sex (with two levels: male (M) and female (F)), culture (also with two levels: Aboriginal (A) and not (N)), age group (with four levels: F0 (primary), F1, F2 and F3) and learner status (with two levels: average (AL) and slow (SL)). The response variable is a count of the number of days absent from school in a particular school year.

```
library(MASS)
data(quine)
attach(quine)
names(quine)
```

```
[1] "Eth"  "Sex"  "Age"  "Lrn"  "Days"
```

We begin with a log-linear model for the counts, and fit a maximal model containing all the factors and all their interactions:

```
model1<-glm(Days~Eth*Sex*Age*Lrn,poisson)
summary(model1)
```

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

    Null deviance: 2073.5 on 145 degrees of freedom
Residual deviance: 1173.9 on 118 degrees of freedom
AIC: 1818.4
```

Next, we check the residual deviance to see if there is overdispersion. Recall that the residual deviance should be equal to the residual degrees of freedom if the Poisson errors assumption is appropriate. Here it is 1173.9 on 118 d.f., indicating overdispersion by a factor of roughly 10. This is much too big to ignore, so before embarking on model simplification we try a different approach, using quasi-Poisson errors to account for the overdispersion:

```
model2<-glm(Days~Eth*Sex*Age*Lrn,quasipoisson)
summary(model2)
```

```
Deviance Residuals:
    Min       1Q   Median       3Q      Max
-7.3872  -2.5129  -0.4205   1.7424   6.6783

Coefficients: (4 not defined because of singularities)
                     Estimate  Std. Error  t value  Pr(>|t|)
(Intercept)            3.0564      0.3346    9.135  2.22e-15  ***
EthN                  -0.1386      0.4904   -0.283    0.7780
SexM                  -0.4914      0.5082   -0.967    0.3356
AgeF1                 -0.6227      0.5281   -1.179    0.2407
AgeF2                 -2.3632      2.2066   -1.071    0.2864
AgeF3                 -0.3784      0.4296   -0.881    0.3802
LrnSL                 -1.9577      1.8120   -1.080    0.2822
EthN:SexM             -0.7524      0.8272   -0.910    0.3649
EthN:AgeF1             0.1029      0.7427    0.139    0.8901
EthN:AgeF2            -0.5546      3.8094   -0.146    0.8845
EthN:AgeF3             0.0633      0.6194    0.102    0.9188
SexM:AgeF1             0.4092      0.9372    0.437    0.6632
SexM:AgeF2             3.1098      2.2506    1.382    0.1696
SexM:AgeF3             1.1145      0.6173    1.806    0.0735  .
EthN:LrnSL             2.2588      1.9474    1.160    0.2484
SexM:LrnSL             1.5900      1.9448    0.818    0.4152
AgeF1:LrnSL            2.6421      1.8688    1.414    0.1601
AgeF2:LrnSL            4.8585      2.8413    1.710    0.0899  .
AgeF3:LrnSL                NA          NA       NA        NA
EthN:SexM:AgeF1       -0.3105      1.6756   -0.185    0.8533
EthN:SexM:AgeF2        0.3469      3.8928    0.089    0.9291
EthN:SexM:AgeF3        0.8329      0.9629    0.865    0.3888
EthN:SexM:LrnSL       -0.1639      2.1666   -0.076    0.9398
EthN:AgeF1:LrnSL      -3.5493      2.0712   -1.714    0.0892  .
EthN:AgeF2:LrnSL      -3.3315      4.2739   -0.779    0.4373
EthN:AgeF3:LrnSL           NA          NA       NA        NA
SexM:AgeF1:LrnSL      -2.4285      2.1901   -1.109    0.2697
SexM:AgeF2:LrnSL      -4.1914      2.9472   -1.422    0.1576
SexM:AgeF3:LrnSL           NA          NA       NA        NA
EthN:SexM:AgeF1:LrnSL  2.1711      2.7527    0.789    0.4319
EthN:SexM:AgeF2:LrnSL  2.1029      4.4203    0.476    0.6351
EthN:SexM:AgeF3:LrnSL      NA          NA       NA        NA
- - -
Signif. codes: 0 '***'0.001 '**'0.01 '*'0.05 '.'0.1 ' '1
(Dispersion parameter for quasipoisson family taken to be 9.514226)

    Null deviance: 2073.5 on 145 degrees of freedom
Residual deviance: 1173.9 on 118 degrees of freedom
AIC: NA

Number of Fisher Scoring iterations: 5
```

Notice that certain interactions have not been estimated because of missing factor-level combinations, as indicated by the zeros in the following table:

```
ftable(table(Eth,Sex,Age,Lrn))
```

```
Eth  Sex  Age  Lrn  AL  SL
A    F    F0         4   1
          F1         5  10
          F2         1   8
          F3         9   0
     M    F0         5   3
          F1         2   3
          F2         7   4
          F3         7   0
N    F    F0         4   1
          F1         6  11
          F2         1   9
          F3        10   0
     M    F0         6   3
          F1         2   7
          F2         7   3
          F3         7   0
```

This occurs because slow learners never get into Form 3.

Unfortunately, AIC is not defined for this model, so we cannot automate the simplification using stepAIC. We need to do the model simplification long-hand, therefore, remembering to do F tests (not chi-squared) because of the overdispersion. Here is the last step of the simplification before obtaining the minimal adequate model. Do we need the age by learning interaction?

```
model4<-update(model3,~. - Age:Lrn)
anova(model3,model4,test="F")
```

```
Analysis of Deviance Table

   Resid. Df  Res.Dev  Df  Deviance       F  Pr(>-F)
1        127  1280.52
2        129  1301.08  -2    -20.56  1.0306   0.3598
```

No we don't. So here is the minimal adequate model with quasi-Poisson errors:

```
summary(model4)
```

```
Coefficients:
             Estimate  Std. Error  t value  Pr(>|t|)
(Intercept)   2.83161     0.30489    9.287  4.98e-16  ***
EthN          0.09821     0.38631    0.254   0.79973
SexM         -0.56268     0.38877   -1.447   0.15023
AgeF1        -0.20878     0.35933   -0.581   0.56223
AgeF2         0.16223     0.37481    0.433   0.66586
AgeF3        -0.25584     0.37855   -0.676   0.50036
LrnSL         0.50311     0.30798    1.634   0.10479
EthN:SexM    -0.24554     0.37347   -0.657   0.51206
EthN:AgeF1   -0.68742     0.46823   -1.468   0.14450
EthN:AgeF2   -1.07361     0.42449   -2.529   0.01264  *
EthN:AgeF3    0.01879     0.42914    0.044   0.96513
EthN:LrnSL   -0.65154     0.45857   -1.421   0.15778
SexM:AgeF1   -0.26358     0.50673   -0.520   0.60385
```

```
SexM:AgeF2          0.94531   0.43530    2.172   0.03171   *
SexM:AgeF3          1.35285   0.42933    3.151   0.00202   *
SexM:LrnSL         -0.29570   0.41144   -0.719   0.47363
EthN:SexM:LrnSL     1.60463   0.57112    2.810   0.00573   *

(Dispersion parameter for quasipoisson family taken to be 9.833426)

    Null deviance: 2073.5 on 145 degrees of freedom
Residual deviance: 1301.1 on 129 degrees of freedom
```

There is a very significant three-way interaction between ethnic origin, sex and learning difficulty; non-Aboriginal slow-learning boys were more likely to be absent than non-aboriginal boys without learning difficulties.

```
ftable(tapply(Days,list(Eth,Sex,Lrn),mean))
```

```
             AL        SL
A   F   14.47368  27.36842
    M   22.28571  20.20000
N   F   13.14286   7.00000
    M   13.36364  17.00000
```

Note, however, that amongst the pupils without learning difficulties it is the Aboriginal boys who miss the most days, and it is Aboriginal girls with learning difficulties who have the highest rate of absenteeism overall.

Negative binomial errors

Instead of using quasi-Poisson errors (as above) we could use a negative binomial model. This is in the MASS library and involves the function glm.nb. The modelling proceeds in exactly the same way as with a typical GLM:

```
model.nb1<-glm.nb(Days~Eth*Sex*Age*Lrn)
summary(model.nb1,cor=F)
```

```
Call:
glm.nb(formula = Days ~ Eth * Sex * Age * Lrn, init.theta =
1.92836014510701, link = log)

(Dispersion parameter for Negative Binomial(1.9284) family taken to be 1)

    Null deviance: 272.29 on 145 degrees of freedom
Residual deviance: 167.45 on 118 degrees of freedom
AIC: 1097.3
             Theta: 1.928
         Std. Err.: 0.269
2 x log-likelihood: −1039.324
```

The output is slightly different than a conventional GLM: you see the estimated negative binomial parameter (here called theta, but known to us as k and equal to 1.928) and its approximate standard error (0.269) and 2 times the log-likelihood (contrast this with the residual deviance from our quasi-Poisson model, which was 1301.1; see above). Note that the residual deviance in the negative binomial model (167.45) is not 2 times the log-likelihood.

An advantage of the negative binomial model over the quasi-Poisson is that we can automate the model simplification with stepAIC:

```
model.nb2<-stepAIC(model.nb1)
summary(model.nb2,cor=F)
```

```
Coefficients: (3 not defined because of singularities)
                  Estimate  Std. Error  z value  Pr(>|z|)
(Intercept)         3.1693      0.3411    9.292    < 2e-16  ***
EthN               -0.3560      0.4210   -0.845   0.397848
SexM               -0.6920      0.4138   -1.672   0.094459  .
AgeF1              -0.6405      0.4638   -1.381   0.167329
AgeF2              -2.4576      0.8675   -2.833   0.004612  **
AgeF3              -0.5880      0.3973   -1.480   0.138885
LrnSL              -1.0264      0.7378   -1.391   0.164179
EthN:SexM          -0.3562      0.3854   -0.924   0.355364
EthN:AgeF1          0.1500      0.5644    0.266   0.790400
EthN:AgeF2         -0.3833      0.5640   -0.680   0.496746
EthN:AgeF3          0.4719      0.4542    1.039   0.298824
SexM:AgeF1          0.2985      0.6047    0.494   0.621597
SexM:AgeF2          3.2904      0.8941    3.680   0.000233  ***
SexM:AgeF3          1.5412      0.4548    3.389   0.000702  ***
EthN:LrnSL          0.9651      0.7753    1.245   0.213255
SexM:LrnSL          0.5457      0.8013    0.681   0.495873
AgeF1:LrnSL         1.6231      0.8222    1.974   0.048373  *
AgeF2:LrnSL         3.8321      1.1054    3.467   0.000527  ***
AgeF3:LrnSL             NA          NA       NA         NA
EthN:SexM:LrnSL     1.3578      0.5914    2.296   0.021684  *
EthN:AgeF1:LrnSL   -2.1013      0.8728   -2.408   0.016058  *
EthN:AgeF2:LrnSL   -1.8260      0.8774   -2.081   0.037426  *
EthN:AgeF3:LrnSL        NA          NA       NA         NA
SexM:AgeF1:LrnSL   -1.1086      0.9409   -1.178   0.238671
SexM:AgeF2:LrnSL   -2.8800      1.1550   -2.493   0.012651  *
SexM:AgeF3:LrnSL        NA          NA       NA         NA
(Dispersion parameter for Negative Binomial(1.8653) family taken to be 1)

    Null deviance: 265.27 on 145 degrees of freedom
Residual deviance: 167.44 on 123 degrees of freedom
AIC: 1091.4
            Theta: 1.865
        Std. Err.: 0.258
2 x log-likelihood: −1043.409
```

```
model.nb3<-update(model.nb2,~. - Sex:Age:Lrn)
anova(model.nb3,model.nb2)
```

```
Likelihood ratio tests of Negative Binomial Models

   theta Resid.    df   2 x log-lik.    Test   df   LR stat.      Pr(Chi)
1      1.789507   125     -1049.111
2      1.865343   123     -1043.409   1 vs 2    2   5.701942   0.05778817
```

The sex-by-age-by-learning interaction does not survive a deletion test ($p = 0.058$), nor does ethnic-origin-by-age-by-learning ($p = 0.115$) nor age-by-learning ($p = 0.150$):

```
model.nb4<-update(model.nb3,~. - Eth:Age:Lrn)
anova(model.nb3,model.nb4)
```

```
Likelihood ratio tests of Negative Binomial Models

      theta   Resid. df   2 x log-lik.    Test   df   LR stat.     Pr(Chi)
1  1.724987         127     -1053.431
2  1.789507         125     -1049.111   1 vs 2    2   4.320086   0.1153202
```

```
model.nb5<-update(model.nb4,~. - Age:Lrn)
anova(model.nb4,model.nb5)
```

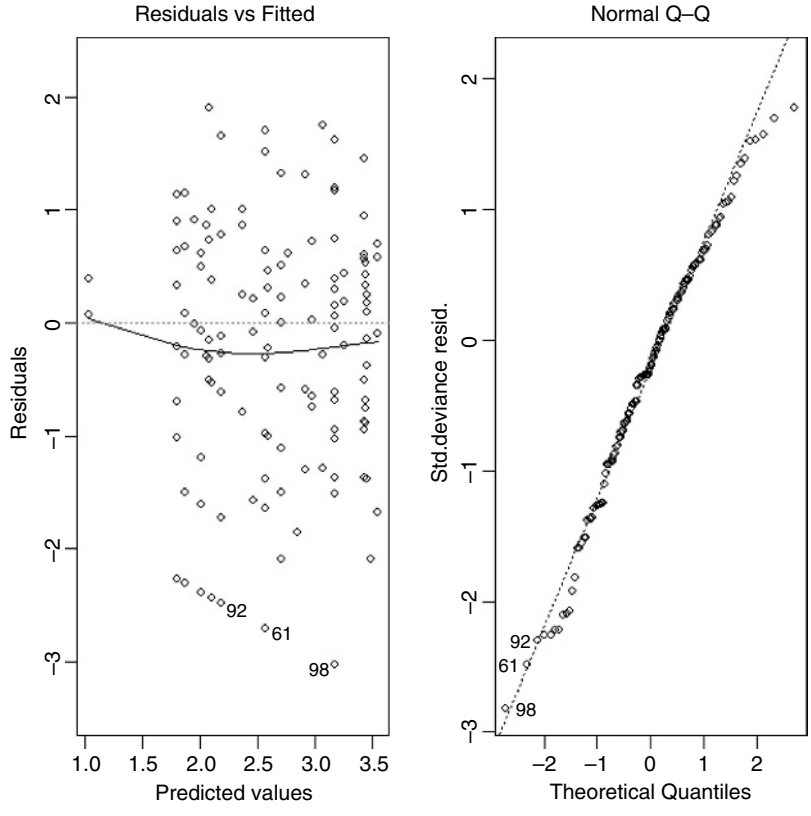
```
Likelihood ratio tests of Negative Binomial Models

   theta Resid.    df   2 x log-lik.    Test   df   LR stat.    Pr(Chi)
1      1.678620   129     -1057.219
2      1.724987   127     -1053.431   1 vs 2    2   3.787823   0.150482
```

```
summary(model.nb5,cor=F)
```

```
Coefficients:

                 Estimate  Std. Error  z value  Pr(>|z|)
(Intercept)       2.91755     0.32626    8.942    < 2e-16  ***
EthN              0.05666     0.39515    0.143    0.88598
SexM             -0.55047     0.39014   -1.411    0.15825
AgeF1            -0.32379     0.38373   -0.844    0.39878
AgeF2            -0.06383     0.42046   -0.152    0.87933
AgeF3            -0.34854     0.39128   -0.891    0.37305
LrnSL             0.57697     0.33382    1.728    0.08392  .
EthN:SexM        -0.41608     0.37491   -1.110    0.26708
EthN:AgeF1       -0.56613     0.43162   -1.312    0.18965
EthN:AgeF2       -0.89577     0.42950   -2.086    0.03702  *
EthN:AgeF3        0.08467     0.44010    0.192    0.84744
SexM:AgeF1       -0.08459     0.45324   -0.187    0.85195
SexM:AgeF2        1.13752     0.45192    2.517    0.01183  *
SexM:AgeF3        1.43124     0.44365    3.226    0.00126  **
EthN:LrnSL       -0.78724     0.43058   -1.828    0.06750  .
SexM:LrnSL       -0.47437     0.45908   -1.033    0.30147
EthN:SexM:LrnSL   1.75289     0.58341    3.005    0.00266  **
(Dispersion parameter for Negative Binomial(1.6786) family taken to be 1)

    Null deviance: 243.98 on 145 degrees of freedom
Residual deviance: 168.03 on 129 degrees of freedom
AIC: 1093.2
             Theta: 1.679
         Std. Err.: 0.22
2 x log-likelihood: −1057.219
```

The minimal adequate model, therefore, contains exactly the same terms as we obtained with quasi-Poisson, but the significance levels are higher (e.g. the three-way interaction has $p = 0.002\,66$ compared with $p = 0.005\,73$). We need to plot the model to check assumptions:

```
par(mfrow=c(1,2))
plot(model.nb5)
par(mfrow=c(1,1))
```



The variance is well behaved and the residuals are close to normally distributed. The combination of low p values plus the ability to use stepAIC makes glm.nb a very useful modelling function for count data such as these.

Use of lmer with Complex Nesting

In this section we have count data (snails) so we want to use family = poisson. But we have complicated spatial pseudoreplication arising from a split-plot design, so we cannot use a GLM. The answer is to use generalized mixed models, lmer. The default method for a generalized linear model fit with lmer has been switched from PQL to the Laplace method. The Laplace method is more reliable than PQL, and is not so much slower to as to preclude its routine use (Doug Bates, personal communication).

The syntax is extended in the usual way to accommodate the random effects (Chapter 19), with slashes showing the nesting of the random effects, and with the factor associated with the largest plot size on the left and the smallest on the right. We revisit the split-plot experiment on biomass (p. 469) and analyse the count data on snails captured from each plot. The model we want to fit is a generalized mixed model with Poisson errors (because the data are counts) with complex nesting to take account of the four-level split-plot design (Rabbit exclusion within Blocks, Lime treatment within Rabbit plots,

3 Competition treatments within each Lime plot and 4 nutrient regimes within each Competition plot):

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

```
[1]  "vals"         "Block"     "Rabbit"   "Lime"
[5]  "Competition"  "Nutrient"
```

The syntax within lmer is very straightforward: fixed effects after the tilde ~, then random effects inside brackets, then the GLM family:

```
library(lme4)
model<-
lmer(vals~Nutrient+(1|Block/Rabbit/Lime/Competition),family=poisson)
summary(model)
```

```
Generalized linear mixed model fit using Laplace
Formula: vals ~ Nutrient + (1 | Block/Rabbit/Lime/Competition)
 Family: poisson(log link)
  AIC    BIC  logLik  deviance
420.2  451.8  -202.1     404.2

Random effects:

Groups                              Name          Variance   Std.Dev.
Competition:(Lime:(Rabbit:Block)) (Intercept) 2.2660e-03 4.7603e-02
Lime:(Rabbit:Block)               (Intercept) 5.0000e-10 2.2361e-05
Rabbit:Block                      (Intercept) 5.0000e-10 2.2361e-05
Block                             (Intercept) 5.0000e-10 2.2361e-05
number of obs: 384, groups: Competition:(Lime:(Rabbit:Block)),96;
Lime:(Rabbit:Block), 32; Rabbit:Block, 16; Block, 8
Estimated scale (compare to 1) 0.974339

Fixed effects:
            Estimate  Std. Error  z value  Pr(>|z|)
(Intercept)   1.10794     0.05885  18.826    <2e-16  ***
NutrientNP    0.11654     0.08063   1.445     0.148
NutrientO    -0.02094     0.08338  -0.251     0.802
NutrientP    -0.01047     0.08316  -0.126     0.900

Correlation of Fixed Effects:

            (Intr)   NtrnNP   NtrntO
NutrientNP  -0.725
NutrientO   -0.701    0.512
NutrientP   -0.703    0.513    0.496
```

There are no significant differences in snail density under any of the four nutrient treatments (Fixed effects, minimum $p = 0.148$) and only Competition within Lime within Rabbit within Block has an appreciable variance component (standard deviation 0.047 603). Note that because we are using Poisson errors, the fixed effects are on the log scale (the scale of the linear predictor; see p. 513). You might want to compare these best linear unbiased predictors with the logs of the arithmetic mean snail counts:

```
log(tapply(vals,Nutrient,mean))
```

```
       N         NP          O          P
1.108975   1.225612   1.088141   1.098612
```

The values are so close because in this case the random effects are so slight (see p. 627). Note, too, that there is no evidence of overdispersion once the random effects have been incorporated, and the estimated scale parameter is 0.974 339 (it would be 1 in a perfect Poisson world).