13

Count Data

Up to this point, the response variables have all been 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, are 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). 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. lightening strikes, bankruptcies, deaths, births). This is in contrast with 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, gender 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, and
- zeros are difficult to handle in transformations.

In R, count data are handled very elegantly in a `glm` by specifying `family = poisson` which sets errors = Poisson and link = log (see Chapter 7). The log link ensures that all the fitted values are positive, while the Poisson errors take account of the fact that the data are integers and have variances that are equal to their means.

A Regression with Poisson Errors

This 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

Statistics: An Introduction using R M. J. Crawley
 ISBNs: 0-470-02298-1 (PBK); 0-470-02297-3 (PPC)

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 first 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 re-fitting the model using quasipoisson rather than Poisson errors:

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

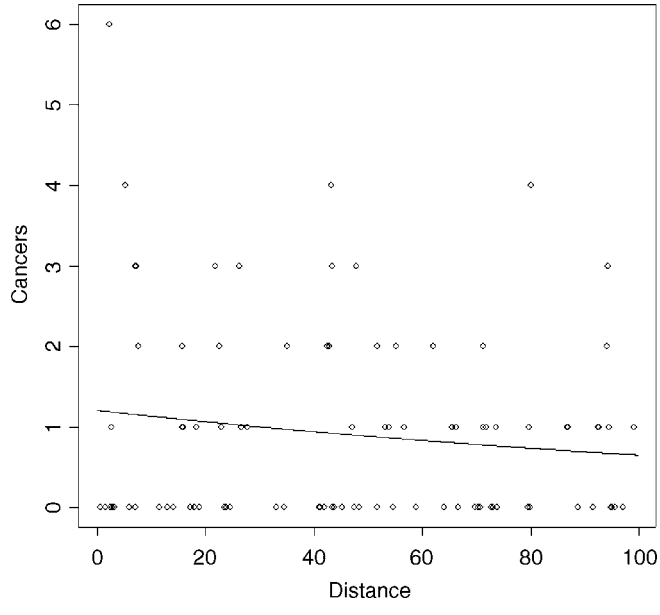
```
Coefficients:
             Estimate  Std. Error  t value Pr(>|t|)
(Intercept)   0.186865   0.235341   0.794    0.429
Distance     -0.006138   0.004573  -1.342    0.183

(Dispersion parameter for quasipoisson family taken to be
1.554966)

    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

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), gender (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"    "gender"    "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 seven 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,gender,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*gender*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 re-fit the model using quasipoisson errors:

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

Call:
glm(formula = cells ~ smoker * gender * 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
gendermale                                  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:gendermale                       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
gendermale:ageold                          -0.2650  0.9445 -0.281   0.7791
gendermale: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
gendermale:weightobese                     -1.1583  1.0493 -1.104   0.2702
gendermale: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:gendermale:ageold                1.8342  2.1827  0.840   0.4011
smokerTRUE:gendermale:ageyoung             -0.8249  1.3558 -0.608   0.5432
smokerTRUE:gendermale:weightobese          -2.2379  1.7788 -1.258   0.2090
smokerTRUE:gendermale:weightover           -2.5033  2.1120 -1.185   0.2365
smokerTRUE:ageold:weightobese               0.8298  3.3269  0.249   0.8031
smokerTRUE:ageyoung:weightobese            -2.2108  1.0865 -2.035   0.0424 *
smokerTRUE:ageold:weightover                1.1275  1.6897  0.667   0.5049
smokerTRUE:ageyoung:weightover             -1.6156  2.2168 -0.729   0.4665
gendermale:ageold:weightobese               2.2210  1.3318  1.668   0.0960 .
gendermale:ageyoung:weightobese             2.5346  1.9488  1.301   0.1940
gendermale:ageold:weightover               -1.0641  1.9650 -0.542   0.5884
gendermale:ageyoung:weightover             -1.1087  2.1234 -0.522   0.6018
smokerTRUE:gendermale:ageold:weightobese   -1.6169  3.0561 -0.529   0.5970
smokerTRUE:gendermale:ageyoung:weightobese      NA      NA      NA       NA
smokerTRUE:gendermale:ageold:weightover         NA      NA      NA       NA
smokerTRUE:gendermale:ageyoung:weightover   2.4160  2.6846  0.900   0.3686
- - -
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for quasipoisson family taken to be 1.854809)

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

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:gender:age:weight)
summary(model3)
```

```
Call:
glm(formula = cells ~ smoker + gender + age + weight + smoker:gender +
smoker:age + gender:age + smoker:weight + gender:weight + age:weight +
smoker:gender:age + smoker:gender:weight + smoker:age:weight +
gender: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   0.436987  -2.053  0.04060 *
smokerTRUE                         0.030263   0.735384   0.041  0.96719
gendermale                         0.297192   0.570008   0.521  0.60234
ageold                            -0.118726   0.528164  -0.225  0.82224
ageyoung                           0.289259   0.639617   0.452  0.65130
weightobese                        1.302660   0.898306   1.450  0.14768
weightover                        -0.005052   1.027197  -0.005  0.99608
smokerTRUE:gendermale              0.527345   0.867292   0.608  0.54345
smokerTRUE:ageold                 -0.566584   1.700587  -0.333  0.73915
smokerTRUE:ageyoung                0.757297   0.939745   0.806  0.42073
gendermale:ageold                 -0.379884   0.935363  -0.406  0.68482
gendermale:ageyoung               -0.610703   0.920967  -0.663  0.50758
smokerTRUE:weightobese             3.924591   1.475474   2.660  0.00808 **
smokerTRUE:weightover              1.192159   1.259886   0.946  0.34450
gendermale:weightobese            -1.273202   1.040700  -1.223  0.22178
gendermale:weightover              0.154097   1.098779   0.140  0.88853
ageold:weightobese                -0.993355   0.970483  -1.024  0.30656
ageyoung:weightobese              -1.346913   1.459452  -0.923  0.35653
ageold:weightover                  0.454217   1.090258   0.417  0.67715
ageyoung:weightover               -0.483955   1.300863  -0.372  0.71004
smokerTRUE:gendermale:ageold       0.771116   1.451509   0.531  0.59549
smokerTRUE:gendermale:ageyoung    -0.210317   1.140383  -0.184  0.85376
smokerTRUE:gendermale:weightobese -2.500668   1.369939  -1.825  0.06857 .
smokerTRUE:gendermale:weightover  -1.110222   1.217529  -0.912  0.36230
smokerTRUE:ageold:weightobese     -0.882951   1.187869  -0.743  0.45766
smokerTRUE:ageyoung:weightobese   -2.453315   1.047065  -2.343  0.01954 *
smokerTRUE:ageold:weightover       0.823018   1.528230   0.539  0.59045
smokerTRUE:ageyoung:weightover     0.040795   1.223662   0.033  0.97342
gendermale:ageold:weightobese      2.338617   1.324803   1.765  0.07816 .
gendermale:ageyoung:weightobese    2.822032   1.623846   1.738  0.08288 .
gendermale:ageold:weightover      -0.442066   1.545449  -0.286  0.77497
gendermale:ageyoung:weightover     0.357807   1.291192   0.277  0.78181
```

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  0.152225

(Dispersion parameter for quasipoisson family taken to be 1.827925)

    Null deviance:    1052.95 on 510 degrees of freedom
Residual deviance:     792.85 on 505 degrees of freedom
```

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 gender. In a case like this, it is useful to produce a summary table to highlight the effects:

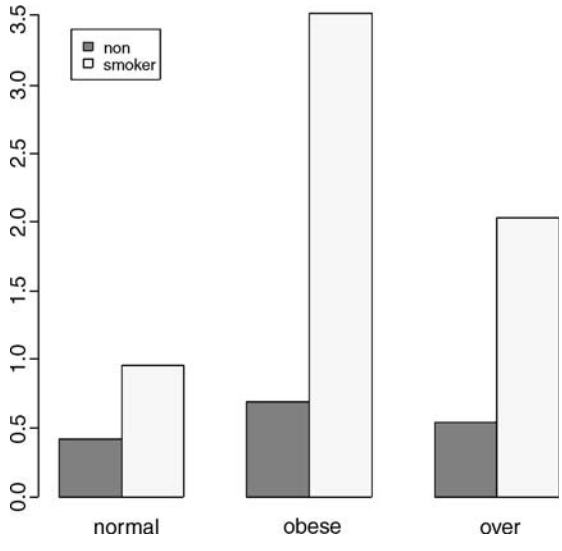
```
tapply(cells,list(smoker,weight),mean)
```

```
              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),beside = T)
legend(1.2,3.4,c("non","smoker"),fill = c(2,7))
```



The Danger of Contingency Tables

We have already dealt with simple contingency tables and their analysis using Fisher's Exact Test or Pearson's chi-squared (see p. 90), but there is an important further issue to be dealt with. In observational studies we quantify only a limited number of explanatory variables. It is inevitable that we shall fail to measure a number of factors that have an important influence on the 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 the response variable. 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 four variables: the response variable, count, with eight values (in grey above), a two-level factor for late season feeding by caterpillars (holed or intact), a two-level

factor for early season aphid feeding (with or without aphids) and a two-level factor for tree (the observations come from two separate trees, imaginatively named Tree 1 and Tree 2).

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

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

We begin by fitting 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 so-called 'nuisance variables'. These are the parameters that need to be in the model to ensure that the marginal totals are properly constrained.

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

The asterisk notation ensures that the saturated model is fitted, because all of the main effects and two-way interactions are fitted, along with the three-way interaction Tree by Aphid by Caterpillar. The model fit involves the estimation of $2 \times 2 \times 2 = 8$ parameters, and exactly matches the eight 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 three-way interaction from the saturated model, and then to use anova to test whether the three-way interaction is significant or not.

```
model2<-update(model, ~. - 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 three-way interaction was significant by specifying test="Chi" like this:

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

```
Analysis of Deviance Table

Model 1: Count~Tree * Aphid * Caterpillar
Model 2: Count~Tree + Aphid + Caterpillar + Tree:Aphid +
  Tree:Caterpillar + Aphid:Caterpillar

Resid. Df     Resid. Dev    Df   Deviance  P(>|Chi|)
1       0     -9.97e-14
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, and assess the results using anova:

```
model3 < -update(model2, ~ . - Aphid:Caterpillar)
anova(model3,model2,test = "Chi")
```

```
Analysis of Deviance Table
Model 1: Count~Tree + Aphid + Caterpillar + Tree:Aphid +
  Tree:Caterpillar
Model 2: Count~Tree + Aphid + Caterpillar + Tree:Aphid +
  Tree:Caterpillar + Aphid:Caterpillar

Resid. Df    Resid. Dev    Df    Deviance    P(>|Chi|)
1       2        0.00409
2       1        0.00079     1    0.00329    0.95423
```

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.

However, look what happens when we do the modelling the wrong way. 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

Model 1: Count~Aphid * Caterpillar
Model 2: Count~Aphid + Caterpillar
  Resid. Df    Resid. Dev    Df   Deviance  P(>|Chi|)
1         4        550.19
2         5        556.85    -1      -6.66       0.01
```

The Aphid:Caterpillar interaction is highly significant ($p = 0.01$) providing strong evidence for induced defences. This is **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
```

Tree 2 has more than four times the proportion of its leaves holed 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, and you never delete terms involving any of the nuisance variables.

Analysis of Covariance with Count Data

In this 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).

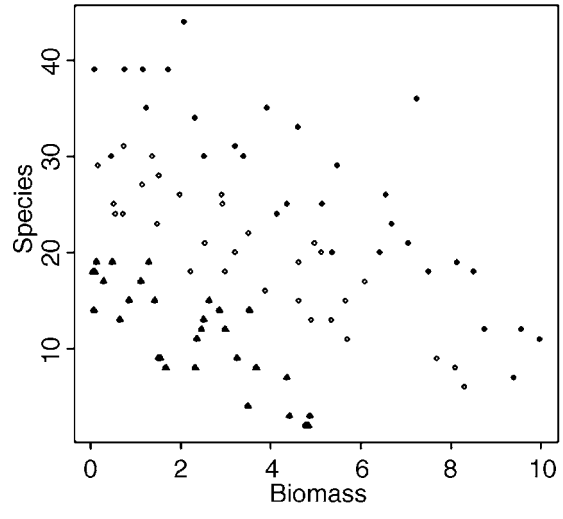
```
species <-read.table("c:\\temp\\species.txt",header = T)
attach(species)
names(species)
```

```
[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 scatter plot. 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    61.242      <2e-16***
Biomass         -0.10713      0.01249    -8.579      <2e-16***
pHlow           -0.81558      0.10282    -7.932    2.16e-15***
pHmid           -0.33146      0.09216    -3.596    0.000323***
Biomass:pHlow   -0.15502      0.03996    -3.880    0.000105***
Biomass:pHmid   -0.03189      0.02307    -1.382    0.166885

(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
```

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 ($p = 0.00033$), so we are justified in retaining the more complicated model 1. Finally, we draw the fitted lines through the scatterplot, using predict:

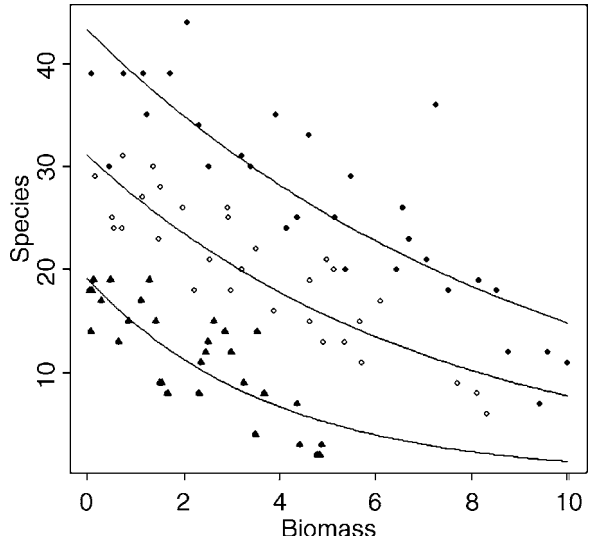
```
xv <-seq(0,10,0.1)
levels(pH)
```

```
[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. 229). 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 single 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 (Box 13.1).

Box 13.1. The Poisson distribution

The Poisson distribution is widely used for the description of count data. 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. The Poisson is a one-parameter distribution, specified entirely by the mean. The variance is identical to the mean (λ), so the variance/mean ratio is equal to one. The probability of observing a count of x 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.

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 ten 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:

```
barplot(frequencies,ylab="Frequency",xlab="Cases",col="red")
```

and the predicted, Poisson frequencies in the right-hand panel

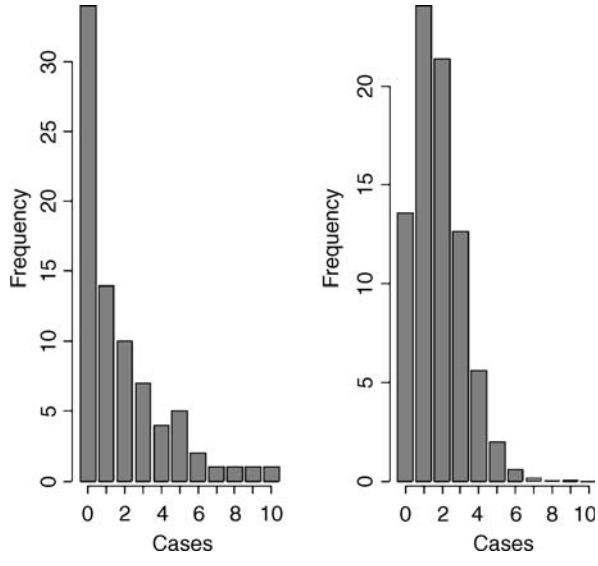
```
barplot(dpois(0:10,1.775)*80,names=as.character(0:10),ylab="Frequency",
  xlab="Cases",col="red")
```

The distributions are very different (p. 242) the mode of the observed data is zero, but the mode of the Poisson with the same mean is one; the observed data contained examples of eight, nine and ten 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 one (the Poisson, of course, has variance/mean ratio = 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 (about 3.0) is the negative binomial distribution (Box 13.2).



Box 13.2. Negative binomial distribution

This discrete, two-parameter distribution is useful for describing the distribution of count data, where the variance is often much greater than the mean. The two parameters are the mean μ and the clumping parameter k. The smaller the value of k, the greater the degree of clumping. The density function is

$$p(x) = \left(1 + \frac{\mu}{k}\right)^{-k} \frac{\Gamma(k+x)}{x!\Gamma(k)} \left(\frac{\mu}{\mu+k}\right)^{x}$$

where Γ is the gamma function. The zero term is found by setting $x = 0$ and simplifying:

$$p(0) = \left(1 + \frac{\mu}{k}\right)^{-k}$$

and successive terms in the distribution can be computed iteratively from

$$p(x) = p(x-1)\left(\frac{k+x-1}{x}\right)\left(\frac{\mu}{\mu+k}\right).$$

An initial estimate of the value of k can be obtained from the sample mean and variance

$$k \approx \frac{\mu^2}{s^2 - \mu}.$$

Since k cannot be negative, it is clear that the negative binomial distribution should not be fitted to data where the variance is less than the mean (use a binomial distribution instead). The precise maximum likelihood estimate of k is found numerically, by iterating progressively more fine-tuned values of k until the left- and right-hand sides of the following equation are equal:

$$n \ln\left(1 + \frac{\mu}{k}\right) = \sum_{x=0}^{\max} \left(\frac{A_{(x)}}{k + x}\right)$$

where the vector $A(x)$ contains the total frequency of values **greater** than x.
You could create a function to work out the probability densities like this:

```
factorial <-function(x) max(cumprod(1:x))
negbin <-function(x,u,k) (1+u/k)^(-k)*(u/(u+k))^x*gamma(k+x)/(factorial(x)
  *gamma(k))
```

then use the function to produce a barplot of probability densities for a range of x values (say 0 to 10, for a distribution with specified mean and aggregation parameter (say $\mu = 0.8, k = 0.2$) like this

```
xf <-numeric(11)
for (i in 0:10) xf[i+1] <-negbin(i,0.8,0.2)
barplot(xf)
```

This is a two-parameter 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

$$k = \frac{\mu^2}{s^2 - \mu}.$$

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
```

The plan is to 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:

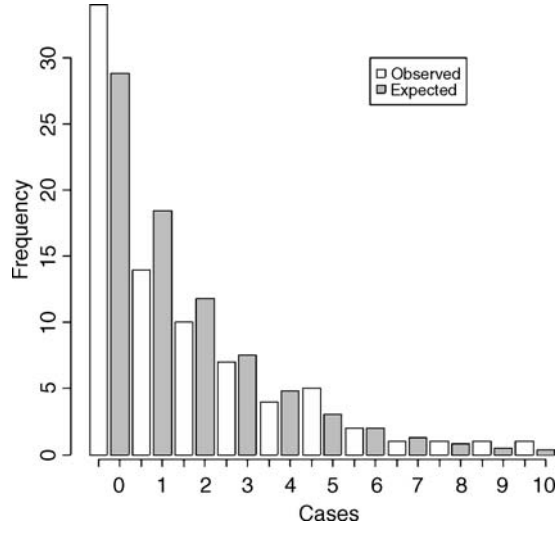
```
labels<-character(22)
labels[1:22 %% 2 == 0]<-as.character(0:10)
```

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

```
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. 126):

```
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 zeros and too few ones 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 square $\sum (O-E)^2/E$ based on the number of comparisons that have expected frequency > 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 four. The degrees of freedom are then given by the number of comparisons (six) – the number of parameters estimated from the data (two in our case) – 1 (for contingency, because the total frequency must add up to 80) = three degrees of freedom. We use 'levels gets' 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 square 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; $p = 0.31$).