17
Modeling discrete outcomes

Contents

17.1 Chapter overview

The generalized linear model is an extension to the general linear model that will deal with discrete outcomes. This chapter provides a brief overview of the generalized linear model with particular focus on logistic regression for categorical outcomes and Poisson regression for count data. The links between logistic regression, Poisson regression and alternative models for categorical outcomes (multinomial, ordered logistic regression) and count data (negative binomial and zero-inflated models) are also reviewed. The chapter ends by considering the difficulty of modeling repeated measures data with discrete outcomes.

17.2 Modeling discrete outcomes in the general linear model

The general linear model is developed on the assumption that the outcome Y is continuous and unbounded. This limitation does not extend to the predictors in these models, which may be categorical or continuous. The solution to incorporating categorical predictors into regression models is to select an appropriate coding scheme (e.g., dummy or effect coding). Extending this approach to deal with a discrete outcome variable is less effective.

First, consider the case of a dichotomous variable. This could be coded as a continuous outcome taking either the value zero or the value one. The problem here is threefold: i) the usual linear regression model assumes Y is unbounded, ii) the predicted outcome is continuous and therefore can take values between zero and one and iii) distributional assumptions about the errors of the model will be violated. The first two problems are most relevant to prediction – as models that predict impossible values can be hard to interpret or lead to problems in application. On the other hand, if the focus is on predicting average rather than individual outcomes, the performance of the model could be satisfactory. Much will depend on the quality of the model (e.g., in terms of fit) and the precision you require. Recall also that, as a rule, it is the degree of distributional violations, not merely their presence, that matters. As each dichotomous observation is a Bernoulli trial, the outcome Y can be considered to have a binomial distribution (if independence of trials is plausible). Under circumstances where the normal approximation to the binomial is good, general linear models (e.g., t tests, regression and ANOVA) will tend to perform well. In practice, this is when the distribution is approximately symmetrical and sample sizes are moderate to large. Prediction using a general linear model will be poorest when the average value of Y approaches zero or one and best when it is close to .5.

An alternative way of conceptualizing the problem of dealing with discrete outcomes is to consider it an issue of linearity. A linear regression model predicts the same change in Y for a given change in X. If Y is bounded, then a linear model tends to run into problems. As Y approaches an upper or lower bound, a change in X that works well for mid-range values of Y may exceed the minimum or maximum possible value of Y.[1] Having a bounded range for the outcome variable is therefore a fundamental obstacle for a linear regression model. To correctly model a discrete outcome it is necessary to find a way to allow the effect of X to vary across the legitimate range of Y. In other words, having an upper or lower bound implies that the effect of X on Y is non-linear. Although a linear approximation will sometimes work well, the challenge is to find ways to capture the non-linear effect of X on Y explicitly. This does not necessarily require a non-linear regression model (e.g., transformations of X or Y may get around this restriction).

For count data, regression models pose similar challenges: i) counts are bounded at zero, ii) treating counts as continuous can lead to fractional rather than integer predictions and iii) the distributional assumptions of the model will be violated. Again, these problems are more pronounced for prediction than hypothesis testing. If the focus is hypothesis testing, a linear regression that assumes continuous responses will often suffice. This is particularly true for large samples where count data, often presumed to be Poisson distributed, are often well approximated by a normal distribution. Even in small samples the approximation may be reasonable and can be assisted by an appropriate transformation (e.g., a square root or Freeman-Tukey transformation).

The usual regression diagnostics for checking normality and homogeneity of variance help to determine the quality of the model. Even when the normal approximation is good, it will be wise to compare results with a model assuming a discrete outcome. Modeling a discrete outcome as continuous may make the results less persuasive (particularly as an appropriate discrete model is usually not difficult to fit). Furthermore, treating the outcome as discrete should lead to more precise estimates and accurate predictions (though these gains will sometimes be marginal). A corollary of this is that a regression model for discrete outcomes may also have greater statistical power. One strategy is to fit both kinds of models. If there are major discrepancies between the results, this is usually an indication that it is necessary to treat the outcome as discrete.[2]

17.3 Generalized linear models

17.3.1 A brief introduction to the generalized linear model

In a general linear model the mean of continuous outcome is modeled as an additive function of one or more predictors plus a random error component (assumed to have normal distribution). This approach to regression places restrictions on what can be modeled. In particular, the linearity of the model – the requirement to model the mean as an additive function of predictors – makes it difficult to cope with discrete, bounded outcomes. One solution is to apply a transformation (e.g., the square root for Poisson counts). This will sometimes work, but represents a somewhat *ad hoc* fix rather than a principled solution to the problem. A principled solution is to generalize the linear model to address both the problem that the mean is not a linear (i.e., additive) function of predictors and that the random component might not be a normal distribution. A *generalized linear model* does exactly this (Agresti, 1996).[3]

Agresti (*ibid.*) describes a generalized linear model in terms of three components. One component is the additive or linear combination of predictors (sometimes called the *systematic component*):

$$b_0 + b_1 X_1 + \ldots + b_q X_q$$

The systematic component corresponds to the right-hand side of the of a standard regression equation. The other two components are the *random component* and the *link*. The random component specifies the outcome variable Y and a probability distribution for it. It is analogous to the left-hand side of a standard regression equation. The binomial distribution is a typical choice of random component for a dichotomous outcome, while for count data the Poisson distribution

is a common default. Any distribution from the exponential family (a set of related distributions that includes the normal) can be selected as a random component.[4] The link is the component that determines the relationship between the systematic and random component. It works by specifying a function, the *link function* $g(.)$, which connects the random and systematic components. The link makes it possible to model any monotonic function of the mean of the original outcome. The expected value of the outcome $E(Y)$ is therefore related to the linear function of predictors via this link:

$$E(Y) = g(\mu) = b_0 + b_1X_1 + \ldots + b_qX_q \qquad \text{Equation 17.1}$$

This differs from a simple transformation of Y or X because the random component need not assume a normal distribution. This flexibility means that a transformation that produces an additive model doesn't also have to produce normality (provided its probability distribution can be captured by the chosen random component). Thus the link component separates the selection of a transformation from the choice of probability distribution.

The general linear model has a random component that is normal with a continuous, untransformed outcome. The generalized linear model therefore includes the general linear model as a special case. The link function for the general linear model is termed the *identity function*:

$$g(\mu) = \mu \qquad \text{Equation 17.2}$$

This is an example of a *canonical link function*. The definition of a canonical link function is somewhat technical, but it represents a natural or default choice associated with a particular random component. For a normal random component the identity function is a natural choice because it maps the range of the predictions of the regression onto the permitted range of Y. For a Poisson random component the canonical link function is the logarithm, while for the binomial distribution it is the logistic function. Although, the generalized linear model allows other random components (and permits non-canonical link functions), the following discussion is restricted to canonical link functions for the binomial, Poisson and a few closely related statistical models.

17.3.2 Estimation and inference

In the special case of a normal random component with an identity link function, the model can be fitted using least squares estimation. Otherwise generalized linear models require iterative methods to be fitted. Most software uses an iterative maximum likelihood (ML) estimation method (see Agresti, 1996), although Bayesian methods (e.g., MCMC estimation) are sometimes adopted (particularly where there are convergence problems). The algorithms for fitting these models are very similar to those used in obtaining parameter estimates for multilevel models (see Chapter 18).

Although the details of estimation for a generalized linear model are quite technical, the principle behind it is extremely simple. The model is first fitted with some vaguely plausible starting estimates for the parameters. The likelihood of the model with these parameter estimates is then calculated (i.e., a quantity proportional to the probability of these parameter estimates given the data at hand). The estimates are then altered and the likelihood recalculated. The process

continues until changing the parameter estimates has no discernable impact on the likelihood (e.g., to a predetermined number of decimal places). Fitting a model this way requires greater computing power than for least squares estimation and models sometimes fail to converge on an adequate solution. For this reason, non-iterative solutions are preferred if available. This applies for models with a normal random component and identity link (standard least squares parameter estimates are therefore also maximum likelihood estimates).

A by-product of ML estimation is the maximized loglikelihood of the model (i.e., the value of the loglikelihood at convergence). This can be used to derive a *deviance* statistic equal to -2 times the loglikelihood of the model: $-2\ln(\ell)$. This statistic has an approximate χ^2 distribution. The quality of the approximation varies according to the type of generalized linear model (*ibid.*). Deviance statistics make it possible to test nested models by referring the difference between models to a χ^2 distribution with degrees of freedom *(df)* equal to the change in the number of parameters. This test is often termed a *likelihood ratio test* (LRT), though it is a conventional null hypothesis significance test (NHST). For comparisons of models differing by only a small number of parameters these χ^2 tests are generally very accurate (though the overall test of deviance of an individual model may not be). It is also straightforward to use likelihood or information-theoretic approaches to compare nested and non-nested methods (e.g., using AIC).

Individual parameters can be tested this way, but it is also common to use a normal approximation in the form of a Wald test. This Wald test involves deriving a χ^2 test statistic from the ratio of a squared model coefficient to its squared standard error (*SE*):

$$\frac{b^2}{\hat{\sigma}_b^2} \approx \chi_1^2 \qquad \text{Equation 17.3}$$

The Wald test can be extended to test several parameters simultaneously (e.g., a set of dummy variables coding a categorical predictor), by summing the test statistics and comparing them to a χ^2 distribution. Thus for J predictors this test would take the form:

$$\sum_{j=1}^{J} \frac{b_j^2}{\hat{\sigma}_{b_j}^2} \approx \chi_J^2 \qquad \text{Equation 17.4}$$

In addition, for a single parameter only, a test equivalent to that described by Equation 17.3 can be obtained using the standard normal distribution:

$$\frac{b}{\hat{\sigma}_b} \approx z \sim N(0,1) \qquad \text{Equation 17.5}$$

These are examples of a class of procedure that are often labeled Wald statistics.[5] The z version of the Wald test is commonly employed to obtain a quick, approximate test or confidence interval (CI) for a regression coefficient, though more accurate deviance tests should be preferred for formal inference in generalized linear models. The deviance of the model can also be used to provide CIs that are superior to the Wald CI, through what is called *profiling*. This involves finding confidence limits by adjusting a single parameter estimate until the deviance of the model differs by the required appropriate critical value above and below its maximum (e.g., 1.92 or approximately half of $\chi^2_{1,.95}$). This is a profile likelihood approach because all other parameters (including nuisance parameters) are fixed at their maximum likelihood estimate (*MLE*).

Taking a regression approach to modeling discrete outcomes has further advantages. Familiar concepts such as prediction equations, model checking, coding of categorical predictors and introducing product terms for moderator or interaction analyses remain useful. However, there is a cost. The additional complexity of the approach presents particular problems that would be relatively easy to resolve in least squares models.

Effect size statistics based on standardized metrics are particularly problematic when computed for generalized linear models. Unstandardized effect size metrics are more widely employed for these models than for least squares models (even among researchers who routinely report standardized metrics for least squares models). The use of a link function adds a layer of difficulty to the choice of unstandardized effect size metric, but transforming the model estimates and reporting effects in terms of the discrete outcome (e.g., counts or probability of a success) can be very effective. Statistical power for generalized models can also be tricky. A very basic first approximation is often to calculate power for a similar least squares model (perhaps incorporating a transformation). This will be sufficient for some applications. Software such as G*Power also has options for logistic and Poisson regression power calculations (Faul *et al.*, 2007). For complex generalized linear models it may be necessary to use Monte Carlo methods for sample size or statistical power estimates.

17.4 Logistic regression

In some fields, notably in clinical or health research, many of the outcome measures are dichotomous (e.g., correct or incorrect; dead or alive). Logistic regression is often the analysis of choice for such data. A logistic regression is a generalized linear model with a binomial distribution as its random component and the logistic transformation as its link function. The logistic transformation may already be familiar from its application in forming a CI for an odds ratio. Thus logistic regression uses a logit link of form:

$$g(\mu) = \ln\left(\frac{\mu}{1-\mu}\right) \qquad \text{Equation 17.6}$$

The logistic transformation has several important properties. Put simply, it copes with a discrete, dichotomous outcome by modeling the probability of obtaining successes from a binomial distribution (sticking to the convention of labeling the outcome coded one as success and the outcome coded zero as failure). Logits (or log odds) are required to model these probabilities using an additive model of predictors (see Key Concept 17.1). The logistic regression equation relates the probability of success P for the i^{th} observation to a linear combination of predictors:

$$\ln\left(\frac{P_i}{1-P_i}\right) = b_0 + b_1 x_{1i} + \ldots + b_q x_{q,i} \qquad \text{Equation 17.7}$$

If the change in notation from μ to P is confusing, bear in mind that the expected value of the outcome Y is estimated by its mean (equivalent to the probability of success $\hat{P}$ for a binomial distribution).

Equation 17.7 expresses the relationship between the predictors and the outcome being modeled (the log odds of success), but for interpreting the parameter estimates it is typically

much easier to work with the equation in its odds or probability form. The same equation can be represented in odds form by taking the exponent of each side:

$$o_i = \frac{P_i}{1 - P_i} = e^{b_0 + b_1 x_{1i} + \ldots + b_q x_{q,i}} \qquad \text{Equation 17.8}$$

Going one step further, it is possible to represent the equation in terms of the probability of success:

$$P_i = \frac{o_i}{1 + o_i} = \frac{e^{b_0 + b_1 x_{1i} + \ldots + b_q x_{q,i}}}{1 + e^{b_0 + b_1 x_{1i} + \ldots + b_q x_{q,i}}} \qquad \text{Equation 17.9}$$

You may also encounter the equation in the following form (obtained by algebraic rearrangement):

$$P_i = \frac{1}{1 + e^{-\left(b_0 + b_1 x_{1i} + \ldots + b_q x_{q,i}\right)}} \qquad \text{Equation 17.10}$$

Although Equation 17.10 is arguably somewhat simpler, the earlier format will be used here because it provides a more direct connection to the odds and log odds (logit) versions of the equations. Using these formulas it is possible to produce prediction equations from a fitted model either in log odds, odds or probability form.

Logistic regression relaxes some assumptions of least squares regression. There is no requirement that the residuals are sampled from a normal distribution and no requirement for homogeneity of variances. It does, however, assume both independence and additivity of effects on the logit scale. Violations of independence, in particular, can cause major problems for logistic regression models.

KEY CONCEPT 17.1

The logistic transformation

The logistic transformation takes P, the probability of event with only two possible outcomes, and turns in into a logit (natural logarithm of the odds of the event or log odds). Switching from a dichotomous outcome to the probability of that outcome is a clever way to turn a discrete variable into a continuous one. At the same time, it retains a relatively clear interpretation in terms of the original context. The downside is that a probability is still bounded at both zero and at one. The inherent problem of modeling a bounded outcome with a linear function has not been addressed. What is required is a function that maps the effect of one or more predictors onto a bounded probability in a systematic but non-linear way.

Transforming a probability to odds offers a partial solution:

$$o = \frac{P}{1 - P}$$

Odds (the ratio of the probability of occurrence to non-occurrence or of success to failures) are continuous and scaled from zero to infinity. Switching to odds removes a boundary at one end of the number line, but not boundary at zero. This problem is resolved by using the logarithm of the odds (with the logarithm to base e, the natural logarithm, being the default choice):

$$\text{logit} = \ln\left(\frac{P}{1 - P}\right)$$

Because the odds are always greater than zero, the logarithm of the odds ranges from $-\infty$ to ∞. At one level the transformation is merely a mathematical trick that turns a discrete binary outcome into a continuous, unbounded outcome. Yet the choice is not completely arbitrary. Both odds and probabilities are themselves directly interpretable. While in some situations it is preferable to work with probabilities, odds and odds ratios are preferable in others (depending on your goal and on factors such as whether you wish to incorporate or strip out the influence of the base rates). Furthermore, using log odds in a linear regression model implies that, while the predictors have an additive relationship with respect to the log odds, their influence on the odds is multiplicative.

A final insight is that the logistic transformation maps differences in the predictors onto a non-linear function with a particular form. To see this relationship between probability and predictors, the inverse of the logistic function (i.e., its cumulative distribution function or *cdf*) needs to be plotted:

$$P = \frac{e^x}{1 + e^x}$$

The inverse of the logistic function has a sigmoidal (S-shaped) curve.

Figure 17.1 depicts the inverse of a logistic function relating the aggregated effect of one or more predictors on the *x*-axis to the predicted probability on the *y*-axis. The curve has an almost linear section in the middle where a normal linear regression model would provide a good fit (e.g., within the range $.2 < P < .8$). It curves sharply at the extremes and converges either on zero or on one. This behavior neatly captures the required non-linearity of effects at the boundary. The function itself can be shifted up or

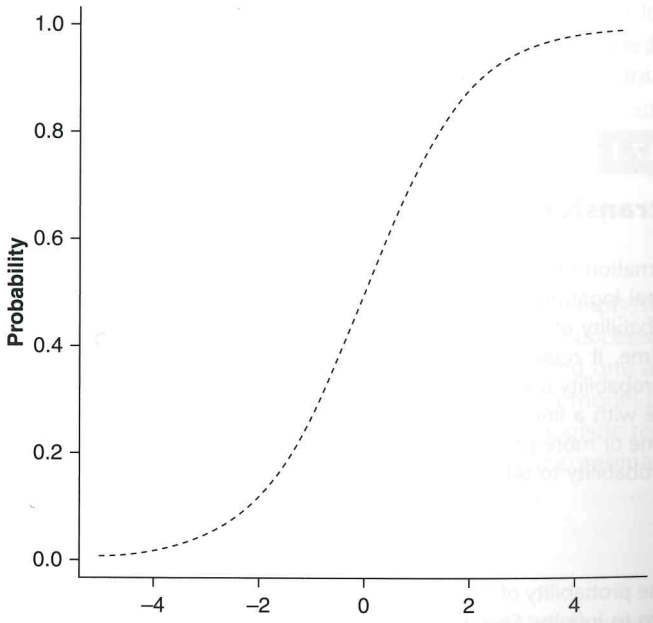

Figure 17.1 A sigmoidal curve produced by the inverse of the logistic function (the *cdf* of the logistic distribution)

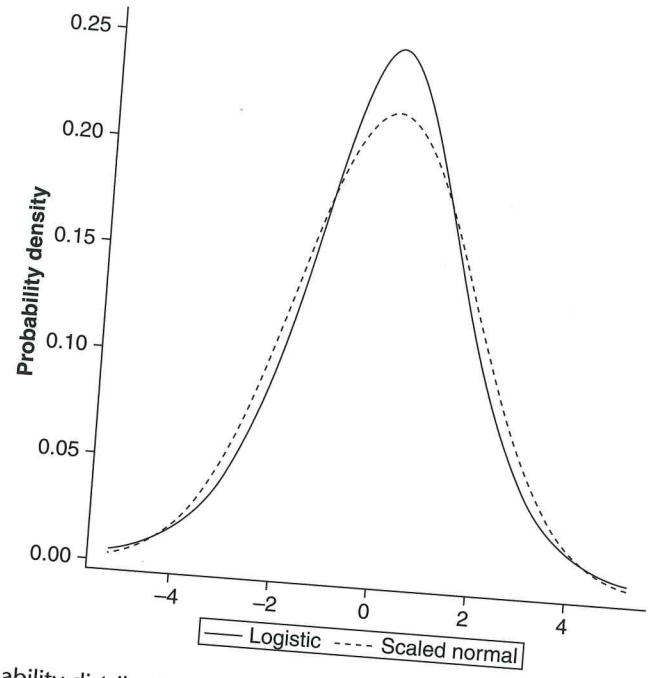

Figure 17.2 Probability distribution functions comparing the logistic distribution with a scaled normal distribution

down the *x*-axis by adding a constant to the equation (e.g., the intercept in systematic component) and the curve can become steeper or shallower by multiplying the log odds by a constant (e.g., changing the slope in a regression model). Changing the sign of the slope will change the direction of the slope so that the sigmoidal curve slopes down as it travels from left to right (required if the probability of success decreases as *X* increases).

The logistic function is not the only plausible sigmoidal function that might be adopted, but it is sufficiently flexible to work for many applications. The probability distribution for the log odds is symmetrical and approximately normal (see Section 7.5.6). This is very convenient for constructing CIs and tests. Figure 17.2 shows the *pdf* for the logistic distribution alongside a scaled normal distribution (scaled with parameters $\mu = 0$ and $\sigma = \pi/\sqrt{3}$ to match the standard logistic distribution). The rescaling is necessary because the standard logistic distribution has a variance of $\pi^2/3$, whereas the standard normal distribution has a variance of one).*

Of the other potential link functions for dichotomous data, the best known is the *probit*. The probit function maps a dichotomous outcome onto a standard normal (z) distribution (see Agresti, 1996). It does this by using the Φ function encountered in earlier chapters (or more formally, the *cdf* for the standard normal distribution). The sigmoidal form of the normal *cdf* is evident from Figure 2.8b. Logistic and probit regression produce very similar outputs for most applications (though probit regression should be superior if the dichotomous variable is derived from applying a cut-off to a continuous normal outcome).

*The logistic distribution can be altered in terms of 'spread' using a scale parameter. For many statistical applications – and all those considered here – this scale parameter is fixed at one and the variance is $\pi^2/3$.

Example 17.1 Subbotsky (2009, Experiment 3) reports an experiment looking at the impact of a suggested magical intervention on the content of dreams. Participants chose a target dream to focus on for three successive nights (e.g., involving them in some attractive role or activity). Some participants were offered a magical suggestion (a 'magic spell' to help them achieve their target dream) and some were not. One outcome of interest was the content of non-target dreams in the subsequent period. In the magical suggestion condition seven out of 21 non-target dreams were classified as 'scary', while in the no magical suggestion condition one out of 26 non-target dreams were classified as 'scary'. The total sample size (N) is therefore 47. Subsequent examples refer to this data set as the dream data.

It is possible to analyze these data via logistic regression. As the experimental condition is a categorical predictor it can be dummy coded (with magical suggestion coded one and no suggestion coded zero). Dream content is the outcome (with a scary dream coded as one and an ordinary dream as zero). The prediction equation is:

$$\ln\left(\frac{\hat{P}}{1-\hat{P}}\right) = -3.22 + 2.53 \times group$$

The deviance for this model is 35.211 and AIC is 39.211 (35.211 plus two times the total number of parameters $k = 2$). The model has residual df equal to $N - q - 1 = 45$. The term q is the number of predictors in the model (exactly as it would be in multiple regression).

The goodness-of-fit of the model could be assessed against a χ^2 distribution with 45 df. The NHST of goodness-of-fit is non-significant (as $df > \chi^2$). A non-significant goodness-of-fit test supposedly indicates a good fit (i.e., one not significantly different from a perfect fit), but the usual problems apply. In large samples the test will reject H_0 when there are only small departures from a model, while small samples will often fail to reject H_0 when the fit is relatively poor. Add to this the fact that the distribution of the deviance may only be poorly approximated by χ^2. The overall test of the deviance of the model is generally regarded as an unsatisfactory approach. A model comparison approach is preferred for inference.

For an individual predictor (such as the effect of group here) it is common to calculate a Wald test or CI. The SE for the intercept is 1.02 and the group slope is 1.12. A significance test of the intercept is uninteresting, but an interval estimate might be useful. With the dummy codes defined as above, the intercept is the estimate of the logits of a scary dream for the no magical suggestion condition. A 95% Wald CI for the intercept is:

$$b_0 \pm z_{.975} \times \hat{\sigma}_{b_0} \approx -3.2 \pm 1.96 \times 1.02 = -3.2 \pm 1.99$$

The Wald CI is not very accurate (tending to be too narrow). A more accurate approach is to use the profile likelihood: 95% CI [−6.10, −1.67]. There are several ways to get the CI for the other condition, but an easy way is to reverse the group coding (so that the magical suggestion condition is coded zero). Profile likelihood then gives a 95% CI of [−1.66, 0.18] for the magical suggestion condition. The interpretation of these point and interval estimates is quite tricky because they are on a logit scale. The issue of interpretation is explored in a later section (see Example 17.2). For the moment I'll just point out that when the odds are one the probability is .5 and the log odds are zero. As neither of the intervals include zero this suggests that the probability of a scary dream in either group is lower than .5 (as you would expect from the data).

The group effect (regardless of coding) is the difference in the logits for the two conditions. The Wald z test statistic is 2.53/1.12, and therefore $z = 2.26$, $p < .05$. Alternatively χ^2 $(1, N = 47) = 2.26^2 \approx 5.1$, $p < .05$. (The value of $p = .024$ is identical because these tests are equivalent).

An alternative test is obtained by comparing deviances for a model with and without the group predictor. The test of the difference in deviance is known as *likelihood ratio test* (sometimes labeled G^2 to distinguish it from the usual Pearson χ^2 test). The difference between the Pearson and likelihood statistics is discussed in Box 17.1. The model without the group predictor (an intercept-only model) has a deviance of 42.885. Subtracting 32.211 from 42.885 gives 7.674. The likelihood ratio test is therefore $G^2\ (1, N=47) = 7.67$, $p = .006$. Using profile likelihood, the corresponding 95% CI is [0.67, 5.51]. As this interval excludes zero it suggests that the odds of a scary dream are greater than one for the magical suggestion group. Out of interest, the Pearson chi-square test gives: $\chi^2\ (1, N=47) = 7.15$, $p = .007$.

The likelihood ratio, Wald and Pearson χ^2 tests are asymptotically equivalent and should produce very similar results in large samples. In generalized linear models with small samples, the likelihood ratio test is typically more accurate than the Wald test (and should be preferred to it). In situations where both the Pearson χ^2 test of independence and the likelihood ratio test can be computed, the former is superior when some cells have small expected values and the latter when no cells have small expected values (see Box 17.1). Although the Pearson χ^2 test could have been applied here for identical or near-identical results, the logistic regression model is more flexible. It would be easy to add other categorical or continuous predictors or interactions to the model.

Box 17.1 Pearson versus likelihood χ^2 test statistics

The Pearson χ^2 statistic is best known for its application in the χ^2 test of independence, but has applications in other situations (e.g., goodness-of-fit tests). It is calculated by: i) obtaining the residual deviations of the observed cell counts from the expected cell counts for a statistical model and ii) summing the ratio of the squared deviations to the expected counts for all cells. A generic form (e.g., ignoring the subscripts) is:

$$\chi^2 = \sum \frac{(O-E)^2}{E}$$

The corresponding generic formula for the likelihood ratio chi-square is:

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

They are goodness-of-fit statistics because they increase as the difference between observed and expected values increases. Small test statistics therefore suggest a good fit to the assumed model (e.g., the model defined by a null hypothesis). Both have an asymptotic χ^2 distribution under H_0, but the Pearson chi-square converges on the χ^2 distribution more rapidly than G^2. Agresti (1996) suggests that the approximation for G^2 is poor when the ratio of total observations to cells is less than five. It tends to produce conservative inferences when data are very sparse (e.g., expected counts are less than .5) and liberal inferences when expected counts are between .5 and five. In contrast, the Pearson chi-square statistic is slightly more robust to small expected values (provided overall N is not too small and no expected counts fall below one).

The Pearson chi-square statistic is therefore typically preferred for tests in small samples. The likelihood ratio test also has properties that make it attractive – especially in fields where larger samples are more common. First, the likelihood χ^2 is the deviance (strictly the residual deviance)

of the statistical model. It is therefore often employed because of its conceptual link to generalized linear models. A second useful property is that, because deviance is equal to $-2\ln(\ell)$, G^2 has a direct connection to likelihood inference (e.g., the calculation of information criteria such as AIC, AIC_C or BIC).

17.4.1 Interpreting parameter estimates in logistic regression

Interpretation of parameters in logistic regression can become convoluted. It helps to start by considering a model with only two predictors:

$$\ln\left(\frac{\hat{P}_i}{1-\hat{P}_i}\right) = b_0 + b_1 x_{1i} + b_2 x_{2i} \qquad \text{Equation 17.11}$$

The coefficient b_0 is the intercept of the generalized linear regression equation and determines the logits (log odds) when all predictors are zero. The coefficient b_1 is the slope, expressed in logits, of X_1 when X_2 is held constant. This is directly comparable to the interpretation of a coefficient in multiple regression. A one-unit increase in X_1 is associated with a b_1 increase in the log odds of success (assuming that X_2 is unchanged). If the predictors are orthogonal (or correlations between predictors are very low) the coefficients can reasonably be interpreted in isolation. Otherwise collinearity and multicollinearity present the usual difficulties for teasing apart effects of individual predictors (though, as usual, the overall fit and model predictions are unaffected as long as parameters can be estimated).

A logistic regression in logit form can be difficult to interpret (though it gets easier with experience). The slope of the inverse logistic curve (e.g., see Figure 17.1) for a given probability P and predictor is:

$$b_q P(1-P) \qquad \text{Equation 17.12}$$

For values close to $P = .5$ the slope of the predictor expressed on a probability scale is around 0.25 times the coefficient on the logit scale. This leads to an approximation using a 'divide by four rule' (Gelman and Hill, 2007). Using the rule, you can interpret the maximum effect of a predictor on the probability of an outcome by dividing its slope by four. This will be accurate for probabilities near .5, but will overestimate the impact of the predictor for probabilities approaching zero or one.

It is typically easier to interpret the regression coefficients in terms of their impact on the odds or probabilities of success. The odds of the logistic regression model with two predictors are obtained by applying the function e^x to each side of the logistic regression to give:

$$\hat{O}_i = \frac{\hat{P}_i}{1-\hat{P}_i} = e^{b_0 + b_1 x_{1i} + b_2 x_{2i}} \qquad \text{Equation 17.13}$$

This is a prediction equation for the odds of success. A useful feature is that each of the coefficients (including the intercept) has a direct interpretation in terms of odds. The intercept e^{b_0} is the odds of success when all the other predictors are coded zero. The slope e^{b_1} is the factor by which the odds of success are expected to increase for each unit increase in X_1 (with

X_2 held constant). Likewise, e^{b_2} is the multiplier for the odds of success associated with a one-unit increase in X_2. Thus the slopes in the odds form of the regression equation are a form of odds ratio (*OR*). The *OR* is therefore a useful effect size estimate in logistic regression (e.g., for situations where results from other studies with dichotomous outcomes are compared with results from logistic regression). The *OR* is particularly easy to interpret for categorical predictors with dummy coding. Under this parameterization, e^{b_q} would represent the *OR* for the category coded one relative to the category coded zero. The odds for the category coded zero would be represented by the intercept (assuming no other predictors in the model). The interpretation of predictors under effect coding is less transparent. The *OR* for an effect coded categorical predictor represents the square root of the change in odds of success.[6]

Comparing the *OR* for continuous predictors with categorical ones (on either coding scheme) should be done with caution. The *OR* for a continuous predictor will often be small in magnitude relative to those of categorical predictors. But, as in multiple regression, a one-unit change in X might be only a fraction of the possible change (in terms of the range in the sample or in the population). If you need to compare the two, make the comparison for a substantial change in the continuous predictor (e.g., two *SD* or the maximum possible change) or rescale the continuous predictor to facilitate the comparison you wish to make. Coefficients smaller than 0.1 or 0.01 tend to be awkward to work with (e.g., because software reports results to only two or three decimal places). Rescaling predictors can remove this sort of problem and make parameter estimates easier to interpret (without changing the fundamental model). CIs for the odds ratios are obtained by separately transforming the lower and upper bounds of the CI in log odds form, moving from a symmetrical interval estimate to an asymmetrical one.

A final option is to present the equation in probability form. Many people will find predicted probabilities intuitively more appealing than odds (and certainly than log odds). On the other hand, the mathematical properties of odds are sometimes very convenient for researchers. Effects on the odds scale are insensitive to the base rates of the outcome (i.e., to the proportion of successes). This makes them better for comparing effects that may have different base rates. The probability scale is better if you want to factor in the base rate (e.g., to show the impact of an effect for a specific individual or group). For this reason it is sensible to become familiar with both forms of the equation. The predicted probabilities for a model with two predictors are:

$$\hat{P}_i = \frac{e^{b_0+b_1x_{1i}+b_2x_{2i}}}{1+e^{b_0+b_1x_{1i}+b_2x_{2i}}} \qquad \text{Equation 17.14}$$

It is possible to interpret either effect on the probability scale by 'plugging' in particular values of other predictors. The obvious choices are the mean for continuous predictors and zero for dummy coded categorical predictors. However, it is important to realize that these choices can have a substantial impact on the predictions – because, in effect, they involve a shift in the base rates. For instance, a continuous predictor will produce a smaller absolute increase in the predicted probability of success for a group with a high average probability of success (e.g., .80) than for one with a low average probability of success (e.g., 0.20).

One solution to presenting such predictions is to plot the predicted probability as a function of one or more effects of interest. The prediction equation can be used to plot a sigmoidal logistic function relating the effects of a predictor (or combination of predictors) to the probability of success or failure. The observed or predicted values can also be added to the plot. Plotting the data on log odds scales (e.g., with confidence bands) helps to assess the linear fit of the

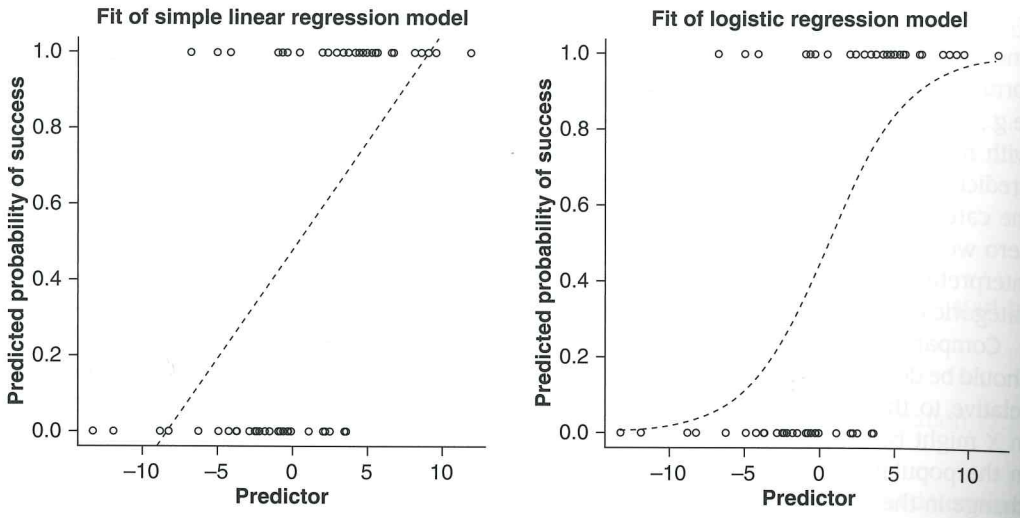

Figure 17.3 Comparing linear and logistic regression fits for a simulated data set with a single continuous predictor

systematic component (very much as you would in linear regression). Gelman and Hill (2007) provide examples of other ways to plot data from a logistic regression.

Figure 17.3 plots the predicted probabilities for a logistic regression involving a single continuous predictor (using simulated data). The panel on the left shows the line of best fit for a simple linear regression to the same data, while the panel on the right gives the fitted curve from a logistic regression. The observations are also plotted. These points fall either on one or zero (with darker circles indicating where more points have landed). The fit of the linear regression is poor at the extremes, but corresponds rather closely to the logistic fit for predicted probabilities in the middle of the distribution. This illustrates how a linear regression can produce satisfactory results when the average probability is close to .5.

Figure 17.3 also underlines an important property of the predictions. They are not predictions of the individual outcomes (successes or fails). They are predictive probabilities: predictions of the probability of an outcome (in this case the probability of a success). This is a subtle distinction, but one that is very important. Had the predicted probabilities been plotted, the points would all have fallen exactly on the regression line. Thus plotting the predictions can be a useful way to illustrate the fit of the model with continuous predictors.

Some authors (e.g., Gelman and Hill, 2007) advocate standardizing continuous predictors in logistic regression to aid interpretation. This can be done in the usual fashion (subtracting the mean and dividing by the *SD* prior to entering them in the regression). My own preference is to avoid standardization in most situations. For logistic regression this rescaling won't change the tests, but will change the coefficients (because a one-unit change will represent a one *SD* change in the predictor).

In a logistic regression with standardized dichotomous predictors, it may make sense to use effect coding for categorical predictors. Doing this ensures that, provided −1 and 1 codes are equally (or near equally) prevalent, the *SD* of the dichotomous predictor will be close to one. An equivalent option advocated by Gelman (2008c) involves dummy coding categorical

predictors, and then rescaling continuous predictors by centering and dividing by twice the *SD*. This works because dummy coded predictors have an *SD* of .5 when there are equal numbers of zero and one codes. Because only the predictor has been standardized, this form of standardization does not produce standardized regression coefficients (beta weights) like those in multiple linear regression. For a review of standardized coefficients analogous to those in least squares models consult Menard (2004).

If you decide not to standardize continuous predictors, it frequently aids interpretation to center those that are not on ratio scales. Rescaling continuous predictors (e.g., multiplying or dividing by a constant) is also sensible if it produces coefficients that are easier to work with. Rescaling effect coded predictors by dividing by two, or (in equal *n* situations) centering dummy coded predictors produces the codes −.5 and .5. This has some of the desirable properties of dummy coding (e.g., the slope of the effect represents the difference in groups) and some of those of effect coding. Last of all, although the systematic component of a generalized linear model is additive, the usual options for dealing with non-additivity of predictor effects exist. The predictors can be transformed (e.g., taking logarithms if their effects are multiplicative) and interactions between predictors or polynomial terms can be included. As with linear regression, extra care must be taken if standardization is applied in moderated logistic or polynomial logistic regression models.

Example 17.2 In Example 17.1 logistic regression was applied to the dream data where both predictor and outcome were discrete. A big advantage of logistic regression is that the predictors can also be continuous. To illustrate this, consider a data set collated by blogger Mark Thompson.* During the recent UK parliamentary expenses scandal the *Daily Telegraph* published a series of articles (starting 8 July 2008) reporting alleged abuses of the parliamentary expenses system by individual members of parliament (MPs). Thompson looked to see if MPs reported as having a problem with their expenses (coded *problem* in the data set) were likely to come from safe seats. He used the parliamentary majority at the last general election (the number of votes exceeding that of his or her nearest rival) as a measure of 'safeness' and found a correlation between the two (using Pearson's *r*). Subsequent examples will refer to this data set as the expenses data.

The parliamentary majority is a count variable, but because of its range (from 37 to 19,519) can be considered continuous for many purposes. Even so, there may be advantages to analyzing the data as a logistic regression. We'll first fit a model with a single predictor: *majority* (scaled in units of 10,000 votes to make the coefficients easier to work with). The outcome is *problem* (zero for no reported expenses problem and one for reports of an expenses problem). The fitted regression equation for this model is:

$$\ln\left(\frac{\hat{P}_i}{1-\hat{P}_i}\right) = -1.039 + 0.6878 \times majority_i$$

Expressed in odds form it is:

$$\hat{o}_i = \frac{\hat{P}_i}{1-\hat{P}_i} = e^{-1.039+0.6878\times majority}$$

The *OR* for majority is therefore $e^{0.6878} = 1.99$. An approximate 95% CI using the Wald method is [1.38, 2.87]. The more accurate profile likelihood CI is [1.38, 2.88]. The close agreement is not

surprising with such a large sample size, but for smaller samples the choice of CI could be important. As the majority is scaled in units of 10,000 votes, every 10,000 increase in parliamentary majority is associated with a near doubling of the odds of an expenses problem. Going from the smallest to largest majority is roughly an increase of 20,000 votes, so the *OR* for the most marginal to most safe seat is around $2 \times 2 = 4$. Putting it on a probability scale:

$$\hat{P}(problem) = \frac{e^{-1.039+0.6878 \times majority}}{1+e^{-1.039+0.6878 \times majority}}$$

The probability of an expenses problem for a person in a seat with zero majority (which can arise if the election is a dead heat and decided by coin toss) would be:

$$\hat{P}(problem \,|\, majority = 0) = \frac{e^{-1.039}}{1+e^{-1.039}} = \frac{0.3538}{1+0.3538} = .261$$

A similar calculation for $\hat{P}(problem \,|\, majority = 10,000) = .413$ and for $\hat{P}(problem \,|\, majority = 20,000) = .583$.

Figure 17.4 plots predictive probability of an expenses problem for this model against parliamentary majority (with approximate 95% confidence bands) and reveals a very clear pattern. The likelihood ratio test of the model (and therefore of the effect of majority) is statistically significant, $G^2 = 13.9$, $p = .002$, and $\Delta AIC = 11.9$ ($LR_{AIC} = 384$) favors the model containing the majority predictor. At this stage it is worth considering other possible predictors. Adding political party to the model doesn't improve the model fit once the 17 extra parameters are accounted for ($\Delta\chi^2_{17} = -29.0$, $\Delta AIC = 5.0$).

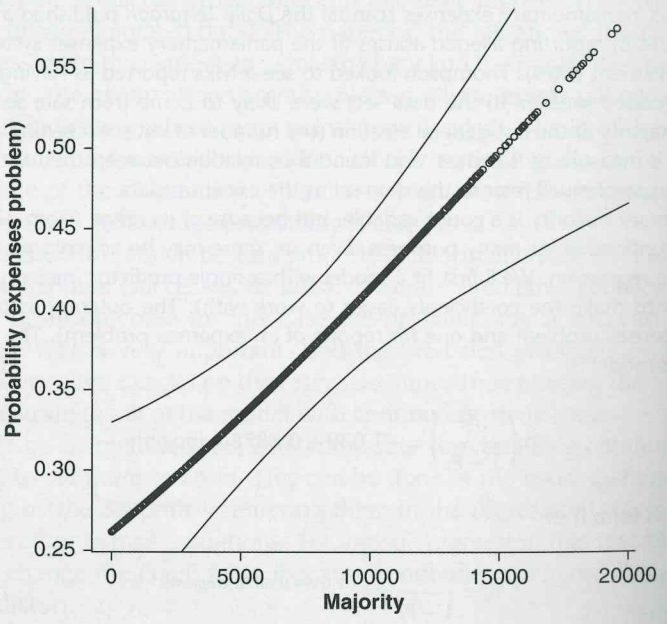

Figure 17.4 Predicted probability of expenses problem by parliamentary majority (with approximate 95% confidence bands) for the expenses data

A similar model (grouping minor parties together as 'Other') adds only four extra parameters, but barely improves the fit and is not more informative ($\Delta\chi^2_4 = -4.2$, $\Delta AIC = 3.8$). Further predictors, such as number of years sitting as an MP could be added to test additional hypotheses or refine the model (but are not considered here).

One of the striking features of Figure 17.4 is just how straight the sigmoidal curve is for these data. This suggests that the methods such as linear regression or correlation would give similar results (at least for inference). Although the simple linear regression and logistic regression lead to similar inferences, the logistic regression is the more correct model and should be first choice when working with continuous predictors and independent, dichotomous outcomes. Even when the overall findings are similar (as for the expenses data) the predictions and the interval estimates from the logistic model will be superior.

*http://markreckons.blogspot.com/

17.4.2 Deviance and model fit

An obvious route to assessing model fit in logistic regression (given that the models are fitted using ML methods) is to determine the deviance of the model. If Y_i represents the i^{th} observed values (zero or one) for each of the N outcomes and $\hat{P}_i$ is its predicted probability, the deviance is:

$$Deviance = -2\ln(\ell) = \sum_{i=1}^{N} -2\left[Y_i \ln\left(\hat{P}_i\right) + (1 - Y_i)\ln\left(1 - \hat{P}_i\right)\right] \qquad \text{Equation 17.15}$$

The expression within the square brackets is the loglikelihood of the model (and is closely related to the binomial distribution and its likelihood function). The deviance therefore depends on the difference between the observed values and predicted probabilities, being small when the predicted probability is close to the observed outcome (e.g., if $Y = 1$ and $\hat{P} = .9$ the contribution from that observation to the deviance is $-2\ln\left(\hat{P}\right) = -0.211$). When the predicted probability is far from the observed outcome the deviance will be large (e.g., if $Y = 0$ and $\hat{P} = .95$ the contribution from that observation to the deviance is $-2\ln\left(1 - \hat{P}\right) = 5.99$). The deviance of the model – more properly the residual deviance of the model – therefore plays a role analogous to the residual sums of squares in a multiple linear regression. Larger deviance implies a worse fit. The deviance has an approximate χ^2 distribution with $N - q - 1$ *df* (where N is the total number of observations and q is the number of predictors). The approximation is not necessarily good when continuous predictors are in the model (Agresti, 1996). A significance test of the overall model should be avoided in any case. It is better to compare the change in deviance between models (see Example 17.1).

Another way to compare the fit of the model is relative to the deviance of the null model. The null deviance is obtained from an intercept-only model with no predictors. In a balanced design with categorical predictors, the null deviance can also be calculated by using the grand mean of P via Equation 17.15. As the null model has no predictors, its deviance can be used as a baseline for comparison (not unlike the SS_{total} in multiple regression). The deviance of the *saturated model* could also be used as a comparator. The saturated model is one with a predictor

for each observation, thus resulting in perfect prediction. The deviance of a saturated model is therefore always exactly zero, and there is no need to fit the saturated model directly. Attempting to fit a saturated model will also produce errors in some software. Because its deviance is zero, this implies that the deviance of any other model is expressed relative to the saturated model (e.g., if D_S is the deviance of the saturated model and D_0 is the deviance of the null model, $D_0 = D_0 - D_S$).

The link between the residual deviance of a generalized linear model and $SS_{residual}$ makes it tempting to characterize the fit of the model in terms of the proportion of deviance or variance it explains. Several different forms of R^2 analog have been proposed for generalized linear models. These are known collectively as pseudo-R^2 measures (because none perfectly mimic the properties of R^2). All pseudo-R^2 measures have major limitations (in addition to the problems inherent in the use of standardized effect sizes – see Chapter 7). A major difficulty is that the total variance to be explained in a generalized linear model tends to vary as a function of the mean – and thus the proportion of variance explained is an ill-defined quantity (except in very restricted circumstances). Ultimately, using a predictive power measure – based on the correlation between observed and predictive values on the untransformed scale – is suggested (Zheng and Agresti, 2000). This measure has similar limitations to those of common pseudo-R^2 measures, can be applied to other generalized linear models, and has a clear interpretation in terms of the predictions of a model. For logistic regression it may sometimes also be useful to look at the percentage of outcomes (successes and fails) classified correctly by the model, though this approach has its drawbacks as well.[7]

17.4.3 Model checking and logistic regression diagnostics

It is one thing to assess the fit of a model, but another to detect or deal with any problems. In this section we'll briefly address residuals (and related leverage and influence statistics), collinearity diagnostics, sparse data and the phenomenon of *complete separation*. One further issue – *overdispersion* – will also be mentioned, but will be explored in more detail in relation to Poisson regression models.

A number of different regression diagnostics can be computed for generalized linear models. The two main types are *deviance residuals* and *Pearson residuals* (related to the fit of the likelihood and Pearson χ^2 statistics respectively). The raw residuals on the logit scale are usually avoided because they do not have particularly good distributional properties (but can be used to assess linearity of the systematic component). Pearson residuals are calculated from the difference between observed and predicted probabilities. This is scaled by the estimate of the standard deviation of the binomial distribution for each value. As the variance of a single Bernoulli trial from a binomial distribution is $P(1-P)$, this produces the equation:

$$e_{P_i} = \frac{Y_i - \hat{P}_i}{\sqrt{\hat{P}\left(1-\hat{P}\right)}} \qquad \text{Equation 17.16}$$

The formula for deviance residuals is:

$$e_{D_i} = \left(\frac{Y_i - \hat{P}_i}{\left|Y_i - \hat{P}\right|}\right)\sqrt{-2\left[Y_i \ln\left(\hat{P}_i\right) + (1-Y_i)\ln\left(1-\hat{P}_i\right)\right]} \qquad \text{Equation 17.17}$$

The right-hand term represents the square root of each observation's contribution to the overall deviance statistic. The sign of the residual depends on the direction of the discrepancy between Y_i and P_i and this is captured by the left-hand term (which evaluates as -1 if $Y_i = 0$ and 1 for if $Y_i = 1$).

Both types of residuals can be adjusted for leverage. The resulting quantities are termed either adjusted residuals or standardized residuals (and have an approximate *SD* of one). Standardizing involves dividing e_{Di} or e_{Pi} by $\sqrt{1-h_i}$ (where, as in linear regression, h_i is the leverage of the observation). It is also possible to calculate studentized residuals using estimates from a regression in which the i^{th} point has been deleted (again matching the practice in linear regression).

The preference in the literature is for deviance residuals over Pearson residuals. The deviance residuals have an asymptotic normal distribution and tend to be more stable than Pearson residuals. The variance of both Pearson and deviance residuals tends to be lower than for a normal distribution, but large residuals can be used to identify extreme observations, with studentized deviance residuals being most useful for this. On the other hand, a normal probability plot of residuals is unlikely to be very useful for assessing the distributional assumptions of the model in small samples, though it may help reveal extreme or particularly influential observations. Using leverage and either Pearson or deviance residuals it is also possible to calculate a generalized version of Cook's distance. Influence measures such as this can help you to avoid drawing strong conclusions from a model where parameter estimates and inference depends strongly on one or two highly influential observations.

Collinearity or multicollinearity of predictors is problematic for any regression model that aims to investigate the individual effects of predictors. If the focus is on overall prediction or model fit then correlations between predictors may not matter (though perfect collinearity will prevent the model from being fitted). As collinearity and multicollinearity are defined in terms of the predictors, the nature of the response is not so important and familiar diagnostics from multiple linear regression such as tolerance and VIF can be computed. The range of solutions for dealing with correlated predictors is also unchanged. The best solutions are either to design a study to avoid collinearity, or to increase sample size to compensate for decreased precision. Where collinearity is a consequence of the structure of the model, arising from fitting polynomial or interaction terms, centering or other forms of rescaling may aid interpretation of the model. Combining, transforming or dropping predictors may also be sensible, depending on context.

Logistic regression can perform badly if data are sparse. Sparseness is a loosely defined term, but applies when there is a very high proportion of either failures or successes. When discrete data are sparse then statistical power and precision of parameter estimation tends to be poor. This should not be surprising, as in such a model there is very little information to use for estimation. Dichotomous data contain less information per observation than continuous outcomes (that can assume a range of intermediate outcomes). Estimation is difficult when modeling sparse data and ML estimates may not converge. Simulation and exact methods are popular alternatives for sparse data (e.g., methods such as MCMC estimation), though they may be very demanding in terms of computing power. The most obvious solution for sparse data is to increase the sample size, but it may also be possible to design the study to avoid or decrease sparseness.

Complete separation is a technical term applied to some logistic regression models. The problem is not unique to logistic regression. It arises when a predictor or combination of predictors perfectly predicts the observed outcomes. The name arises because the predictors can be used to completely separate the predicted outcomes into groups associated with distinct

predictor values. While this seems like a good thing, it is not. Perfect prediction means that the model cannot be fitted and no parameter estimates can be obtained. This can happen for any regression model, but is unlikely for a truly continuous response (and rare for counts). In contrast, complete separation can easily happen when sampling dichotomous outcomes in a small sample or if too many predictors are added to a model and it becomes saturated. The same solutions that can work for sparse data (altering the study design or collecting additional observations) may also prevent complete separation. If complete separation arises due to over-fitting then it will make sense to reduce the number of predictors, though collecting more data may be the only viable option in small samples. *Quasi-complete separation* is also possible. This happens if subsets of values for one predictor are tied on the outcome variable (e.g., all values coded one for a predictor are associated with zero on the outcome variable). Quasi-complete separation is harder to detect and leads to large and unstable *SE*s (similar to the effects of collinearity). It is worth checking whenever the *SE*s are unreasonably large (and collecting more data is again often the best solution). Adding information to the model in the form of a Bayesian prior may also resolve separation problems, for the same reason that adding new data may provide a fix (Gelman and Hill, 2007).

Overdispersion is a potential problem if the probability distribution of the generalized linear model random component has a variance that is a function of the mean. Logistic and Poisson regression are good examples. For the binomial distribution used in logistic regression, the mean is $\hat{P}$ and the variance is $\hat{P}\left(1-\hat{P}\right)$. As a consequence, the expected residual deviance of the model D_M is equal to $\nu = N - q - 1$ (the residual *df* of the model). This implies the ratio $D_M/\nu = 1$ (though some variability is to be expected due to sampling error). Either the Pearson χ^2 or model deviance can be used to detect overdispersion using the estimate D_M/ν. This quantity is an estimate of the overdispersion parameter φ.[8] Overdispersion causes the *SE*s to be too small (and hence interval estimates will be too narrow and tests too liberal). It is also possible, if somewhat unusual, to get underdispersion (and hence *SE*s that are too large). Overdispersion parameters can be used to correct inferences (see Box 17.2) though this may not be the best way of dealing with the problem.

If overdispersion is suspected, try first to gauge the extent of the problem. If the estimate of φ is much greater than one then it is likely that the problem is serious enough to distort the analysis. A number of things can cause overdispersion in a logistic regression model. These include a poorly fitting model (e.g., missing one or more important predictors), not including interactions between predictors, and lack of independence between observations. Finding a better-fitting model by adding predictors, removing predictors or including interaction terms may help resolve the problem. If lack of independence is suspected (e.g., in repeated measures or clustered data), then it may be possible to model the extra random variation between observations (e.g., switching to a different random component or to a multilevel logistic regression model).

Example 17.3 Many of the methods for interrogating residuals in linear regression are available for logistic regression, but may be hard to interpret. Plots of residuals or influence (e.g., Cook's distance) are always worth checking. For the expenses data the residuals are not particularly extreme (studentized residuals range from −1.29 to 1.64). The influence statistics also suggest no reason for concern (all Cook's distances are lower than 0.01). Assessing overdispersion requires calculating an overdispersion parameter. This both provides an estimate of the dispersion parameter and provides a

correction to *SE*s and tests. Doing so for the expenses data estimates the overdispersion parameter at 1.002876 (close to the desired value of one). The standard errors and tests are virtually unchanged. This should not be surprising, because the correction factor for the *SE*s is $\sqrt{1.002876} = 1.001437$ (see Box 17.2). Thus each *SE* is only 0.14% larger after the correction.

17.4.4 Multinomial logistic regression

An attractive feature of logistic regression is that is can be extended to deal with discrete data with more than two possible outcomes. This can be handled in two different ways. The simplest method is applicable only when the J outcomes are structured as a nested hierarchy of dichotomous outcomes or *nested dichotomies*. An example arises if several participants in an attention experiment were asked to respond if they had detected a target or not ('yes', 'no'). If they reported detecting a target they could then be asked if it was 'red' or 'blue'. This produces three possible outcomes 'yes-blue', 'yes-red' and 'no'. The effects of one or more predictor could be used to predict the probability of these outcomes by running two logistic regression models.[9] The first model would predict the probability of a 'yes' response among all N participants, while the second model (restricted only to the n 'yes' responses) would predict the probability of 'red' or 'blue'. In addition to separate tests provided within each of the nested logistic regressions it is possible to combine the deviance and df of each model. This gives an overall test of model fit. This is legitimate because each nested dichotomy is orthogonal to dichotomies nested elsewhere in the hierarchy and therefore deviances and df can be summed across sub-models (which are independent). This kind of model can be useful where responses are naturally nested, and is flexible enough that multiple branches can be fitted (e.g., 'no' responses could be also be split by a further dichotomous outcome). The obvious drawback of this type of design is that, as outcomes split, n for each model decreases. The precision of estimates depends on n within each sub-model (requiring N to be very large to maintain statistical power for all effects).

If the outcomes cannot be represented as nested dichotomies, a *multinomial logistic regression* (or *polychotomous logistic regression*) model can be employed. In this model outcomes are assumed to have a multinomial distribution in which each independent trial has a fixed probability P_j of falling into one of J unordered outcome categories. As the probabilities P_1 to P_J exhaust all possible outcomes under consideration, their sum must equal one:

$$\sum_{j=1}^{J} P_j = 1$$

The multinomial distribution reduces to the binomial when $J = 2$ and the model becomes a regular logistic regression (though when $J = 2$ the outcomes can be ordered or unordered).

Fitting a multinomial logistic regression is a bit like fitting $J - 1$ separate logistic regression models (Agresti, 1996). The principle here is similar to that of using indicator variables to code categorical predictors in a statistical model: it takes $J - 1$ variables to represent J categories. Unlike the nested dichotomies approach it is necessary to fit the $J - 1$ sets of parameter estimates simultaneously (maximizing the likelihood of the joint equations). This is more efficient and produces more accurate parameter estimates than fitting the models separately. One outcome

category (often the last outcome J) is chosen as a reference category and one or more slopes are estimated for the remaining $J-1$ outcomes. This produces $J-1$ equations of the form:

$$\ln\left(\frac{\hat{P}_{j,i}}{\hat{P}_{J,i}}\right) = b_0 + b_1 x_{1,i} + \ldots + b_q x_{q,i} \qquad \text{Equation 17.18}$$

This models each of the $J-1$ outcomes in terms of its log odds relative to an intercept. The subscript i is the index for all $i=1$ to N observations (but to simplify presentation of the equations the i subscript will be dropped for the rest of this discussion). No equation is required for outcome J because its predicted probability is a function of the probabilities of the other outcomes:

$$\hat{P}_J = 1 - \sum_{1}^{J-1} \hat{P}_j$$

This is identical to logistic regression when $J=2$ and where $\hat{P}_J = 1 - \hat{P}_1$ (the predicted probability of a failure, if $\hat{P}_1$ is a success).

For a model with two predictors and three outcome categories (labeled a, b and c) the $J-1=2$ regression equations are:

$$\ln\left(\frac{\hat{P}_a}{\hat{P}_c}\right) = b_{0_a} + b_{1_a} x_1 + b_{2_a} x_2$$

$$\ln\left(\frac{\hat{P}_b}{\hat{P}_c}\right) = b_{0_b} + b_{1_b} x_1 + b_{2_b} x_2 \qquad \text{Equation 17.19}$$

This defines the log odds of outcomes a and b relative to the reference outcome category c. What if you are interested in the log odds of a relative to b? Agresti (*ibid.*) explains how the log odds of any two categories can be determined from the differences in their coefficients. By this method, the log odds of a relative to b are:

$$\ln\left(\frac{\hat{P}_a}{\hat{P}_b}\right) = \ln\left(\frac{\hat{P}_a}{\hat{P}_c}\right) - \ln\left(\frac{\hat{P}_b}{\hat{P}_c}\right) = (b_{0_a} - b_{0_b}) + (b_{1_a} - b_{1_b}) x_1 + (b_{2_a} - b_{2_b}) x_2$$

Equation 17.20

This follows because odds are ratios of probabilities. Both the odds $\hat{P}_a/\hat{P}_c$ and $\hat{P}_b/\hat{P}_c$ have a common numerator (and division on the untransformed scale is equivalent to subtraction on the log scale). It is entirely legitimate to use a formula such as that in Equation 17.20 to extract individual coefficients. It is also possible to switch to another reference category (and this approach is superior if you want the *SE*s of the coefficients). This merely produces a change in sign of the log odds. The log odds of b relative to a for predictor X_2 are therefore equal to $b_{2b} - b_{2a}$. This is a property of all multinomial logistic regression models. In a model with six outcomes (labeled a to f) the intercept for outcome d relative to outcome a would be $b_{0d} - b_{0a}$.

The log odds forms of the equations are interpreted largely as they would be for logistic regression. The odds form of the equation can be obtained by applying the exponent function to both sides of the equation for each of the $J-1$ outcomes:

$$\left(\frac{\hat{P}_j}{\hat{P}_J}\right) = e^{b_0 + b_1 x_1 + \ldots + b_q x_q} \qquad \text{Equation 17.21}$$

These $J-1$ equations provide the *OR* for a unit increase in each predictor and the *OR* for the intercept. This has the usual interpretation, except that each outcome is represented relative to the reference outcome category (rather than to the probability of its own non-occurrence $1-\hat{P}_j$). The predictive probabilities can also be obtained. For an arbitrary outcome category a, the predictive probability is:

$$\hat{P}_a = \frac{e^{b_{0_a}+b_{1_a}x_1+\ldots+b_{q_a}x_q}}{1+\sum_{1=j}^{J-1} e^{b_{0_j}+b_{1_j}x_1+\ldots+b_{q_j}x_q}} \qquad \text{Equation 17.22}$$

The term $\sum_{1=j}^{J-1} e^{b_{0_j}+b_{1_j}x_1+\ldots+b_{q_j}x_q}$ is the sum of $e^{b_{0_j}+b_{1_j}x_1+\ldots+b_{q_j}x_q}$ for all of the outcome categories excluding the reference category J. This reduces to Equation 17.9 when $J=2$.

Returning to a study with $J=3$ outcomes and two predictors, the respective probabilities for outcome a would be

$$\hat{P}_a = \frac{e^{b_{0_a}+b_{1_a}x_1+b_{2_a}x_2}}{1+e^{b_{0_a}+b_{1_a}x_1+b_{2_a}x_2}+e^{b_{0_b}+b_{1_b}x_1+b_{2_b}x_2}}, \qquad \text{Equation 17.23}$$

while for outcome b they would be

$$\hat{P}_b = \frac{e^{b_{0_b}+b_{1_b}x_1+b_{2_b}x_2}}{1+e^{b_{0_a}+b_{1_a}x_1+b_{2_a}x_2}+e^{b_{0_b}+b_{1_b}x_1+b_{2_b}x_2}}, \qquad \text{Equation 17.24}$$

and for the reference category c they would be

$$\hat{P}_c = \frac{1}{1+e^{b_{0_a}+b_{1_a}x_1+b_{2_a}x_2}+e^{b_{0_b}+b_{1_b}x_1+b_{2_b}x_2}} \qquad \text{Equation 17.25}$$

The numerator for Equation 17.25 is 1 because it represents the odds of the reference category relative to itself.

Multinomial logistic regression, like its simpler namesake, can incorporate both categorical and continuous predictors. Categorical predictors are coded in the usual way. This typically involves dummy coding, but ANOVA- and ANCOVA-style models (e.g., using effect coding or a cell means model) are also possible. If all predictors in the model are categorical, the multinomial logistic regression (and indeed logistic regression) turn out to be a special case of a generalized linear model known as a loglinear regression model. It will often be a good idea to consider both logistic and loglinear regression models as candidates when all predictors are categorical.[10]

Multinomial logistic regression has three main restrictions. First, it assumes that the categories are unordered (and ignores order if it exists). Second, it is limited to situations where predictors take the same value for each outcome. This may be unduly restrictive. In a domain such as consumer choice, a researcher will be very interested in the factors that determine which product is chosen, but the potential outcomes (e.g., brand a, brand b and brand c) will have different values on some or all of the key predictors (e.g., price). These can be considered outcome-varying covariates (analogous to time-varying covariates in repeated measures).

Third, the model treats observations as independent. Models such as ordered logistic regression and generalized multilevel models relax these assumptions. The latter potentially allows all three limitations of a standard multinomial logistic regression model to be addressed (though other approaches are possible).

17.4.5 Ordered logistic regression

If categorical outcomes are ordered, fitting a multinomial logistic regression model for unordered outcomes ignores potentially crucial information about the data. Agresti (1996) discusses several approaches to the analysis of ordered categorical data. Of these, the *proportional odds model* is the most widely adopted model for bounded ordinal data.[11] The proportional odds model assumes that N independent observations fall into a set of J ordered categories. These are usually coded as a set of sequential integers (e.g., one to J). The coding of order should preserve the natural or logical order of the outcomes, but there is no requirement for the intervals to be equally spaced. Typical examples include grading of academic performance (e.g., where it is not reasonable to assume that the difference between a grade 'A' and a grade 'B' is the same as between 'E' and 'F') or Likert-style rating scales. Although ordinal outcomes lend themselves to ordered logistic regression, a least squares regression model with a good fit may well produce similar results (especially if the goal is hypothesis testing rather than prediction).

For discrete, ordinal outcomes the aim is to find a way of representing this order within the logistic regression framework. Logistic regression is ideally suited to dealing with dichotomous outcomes, and multinomial logistic regression extends this to polychotomous outcomes, by breaking down the model into pairs of dichotomous outcomes. The same solution can be adapted to deal with ordered categories. The key distinction is that ordered outcomes have greater constraints on them than unordered outcomes. These constraints are met by modeling the cumulative probability of each outcome rather than the probability of each separate outcome.

To see why using cumulative probability preserves the ordinal information in the data it helps to look at a simple case. For a rating scale with $J = 3$ possible responses, the outcomes could be 'disagree', 'neutral' and 'agree'. The cumulative probability of the first response is simply the probability that someone disagrees $\hat{P}_{disagree}$. The cumulative probability of the second response is $\hat{P}_{disagree}$ plus $\hat{P}_{neutral}$ and the cumulative probability of the third response is $\hat{P}_{disagree} + \hat{P}_{neutral} + \hat{P}_{agree} = 1$.[12] The constraint here is that the cumulative probability increases across the ordered outcomes; changes in a predictor either increase or decrease the probability of greater agreement (they cannot do both). For instance, a change in X could not increase $\hat{P}_{disagree}$ and $\hat{P}_{agree}$ at the expense of $\hat{P}_{neutral}$. Such a pattern could easily be modeled in an unordered multinomial regression (that treats agreement and disagreement as separate categories rather than two ends of a continuum).

The proportional odds model therefore treats ordered categories as if they were a series of logistic regression models for the $J - 1$ cumulative probabilities. As with unordered multinomial logistic regression, these logistic regression models are estimated simultaneously (with the constraint that the cumulative probability for the last category P_J equals one). In terms of equations, the J cumulative probabilities can be represented as:

$$\hat{P}(Y \leq j) = \hat{P}_1 + \ldots \hat{P}_j \qquad \text{Equation 17.26}$$

These probabilities will not sum to one, but the cumulative probability of the preceding outcome is constrained to be equal or lower than its successor:

$$\hat{P}_1 = \hat{P}(Y=1) \leq \ldots \leq \hat{P}(Y \leq j) \leq \hat{P}_J = 1 \qquad \text{Equation 17.27}$$

The cumulative logits (cumulative log odds) of the model are:

$$\ln\left(\frac{\hat{P}_1 + \cdots + \hat{P}_j}{\hat{P}_{j+1} + \cdots + \hat{P}_J}\right) = \ln\left(\frac{\hat{P}(Y \leq j)}{1 - \hat{P}(Y \leq j)}\right) \quad j = 1, \ldots, J-1 \qquad \text{Equation 17.28}$$

An important feature of this notation is the distinction between the predictive probability of an outcome $\hat{P}_j$ and its cumulative predictive probability $\hat{P}(Y \leq j)$. These are only identical for the first outcome $\hat{P}_1$ (assuming that the probabilities are cumulated from lowest to highest). At the level of the cumulative logits, the model is just a series of logistic regressions with dichotomous outcomes (just as if outcomes one to j had been coded as failure and outcomes $j+1$ to J had been coded as success).

The proportional odds model has several interesting features. The regression equation (excluding the subscript indexing the N independent observations) is:

$$\ln\left(\frac{\hat{P}(Y \leq j)}{1 - \hat{P}(Y \leq j)}\right) = b_{0_j} + b_1 x_1 + \ldots + b_q x_q \quad j = 1, \ldots, J-1 \qquad \text{Equation 17.29}$$

The first feature to note is that there are multiple intercepts represented by the b_{0_j} term. Each of the $J-1$ cut-points for the regression has its own intercept. If $J=3$ the two intercepts would represent the cut-points for a logistic regression equation comparing $j=1$ against $j>1$ and the cut-point for comparing $j<3$ with $j=3$. In an intercept-only model (or a model with centered predictors) these intercepts can be interpreted directly as average values for the sample. The second feature is that the coefficients for the q predictors X_1 to X_q are identical for all the cut-points. The log odds are therefore constant for all cut-points. This in turn implies that the effect of a predictor on the odds is to multiply them by a constant proportion, leading to the designation of the model as a proportional odds model. Each unit increase in a predictor X is associated with an increase of odds of the next highest outcome by a factor of e^{b_k}.

The cumulative odds and cumulative probabilities can be obtained by applying the familiar transformations to the cumulative logit form of the ordered logistic regression in Equation 17.29. The cumulative odds are given by

$$\frac{\hat{P}(Y \leq j)}{1 - \hat{P}(Y \leq j)} = e^{b_{0_j} + b_1 x_1 + \ldots + b_q x_q} \quad j = 1, \ldots, J-1 \qquad \text{Equation 17.30}$$

while the cumulative predictive probabilities are:

$$\hat{P}(Y \leq j) = \frac{e^{b_{0_j} + b_1 x_1 + \ldots + b_q x_q}}{1 + e^{b_{0_j} + b_1 x_1 + \ldots + b_q x_q}} \quad j = 1, \ldots, J-1 \qquad \text{Equation 17.31}$$

The log odds and odds ratios for the predictors can be interpreted without too much difficulty. They refer to the log odds and odds of shifting from a lower to higher outcome category based

on a one-unit increase in X. The cumulative probabilities are harder to interpret (e.g., the cumulative probability for outcome J, the last outcome is one). Instead, the predictive probabilities of the separate outcomes can be derived with a little arithmetic. The predictive probability of outcome j can be defined as:

$$P(Y=j)=P(Y\leq j)-P(Y\leq j-1) \qquad \text{Equation 17.32}$$

The predictive probability for an outcome is therefore the cumulative probability for outcome j minus the cumulative probability for the preceding outcome (if there is one). If $J=3$ (with $j=$ coded 1, 2 or 3) and the predicted cumulative probabilities are .36 and .82, the predictive probability of j_1 is .36. It follows that the probability of $j_2=.80-.36=.44$ and the probability of $j_3=1-.80=.20$.

The most obvious limitation of this form of ordered logistic regression model is the proportional odds restriction itself. Sometimes the effect of a predictor will vary between cut-points and a more flexible model is required. Often, even though the proportional odds assumption is unlikely to be met exactly, the odds will be sufficiently similar for each outcome that the proportional odds model provides a pretty good fit. Some software (e.g., SPSS) will provide a null hypothesis test of the proportional odds assumption, but often with large samples these tests tend to reject the assumption even when it holds up quite well (Harrell, 2001). A better check is to compare the coefficients between separate logistic regression models with identical predictors for each cut-point. The proportional odds assumption implies that effects of a predictor on the logit scale should be similar for each of these models. Modeling the effects of predictors at different cut-points substantially increases the number of parameters to be fitted when J is large (up to an additional $J-2$ effects per predictor). The assumption of proportional odds is rather like fitting a straight line to a messy X-Y relationship. Looking for a curve that fits better than the straight line might lead to over-fitting (unless there are ways to narrow down the scope of possible models).

The independence assumption can also be a major limitation (e.g., for rating data with repeated measures models). Some specialist software (e.g., MLwiN) permits multilevel proportional odds models to be fitted for repeated measures data and also allows separate slopes to be fitted for each intercept. Other approaches can also be used to relax the proportional odds assumption (see Agresti, 1996; Yee, 2010).

Example 17.4 Underwood *et al.* (2007) investigated perceptions of road safety in 119 children from two UK schools in years three, five and seven (roughly seven, nine and eleven years old). They looked at a number of measures, including a free sort task involving 20 pictures of road scenes. Responses on the free sort task were coded and analyzed in several different ways. One analysis used multiple regression to predict the use of a 'safe to cross' code based on gender, year group and two other codes: 'visibility' (*vis*) and 'other road users' (*oru*). This analysis suggested that the safe to cross codes were predicted by other road users for girls, but not boys. However, the safe to cross code occurred infrequently (either zero, one, two or three times). Because zero was the most common response the data are sparse (and less than ideal for multiple regression). One possible alternative is ordered logistic regression.

A first step is to fit a main effects only proportional odds logistic regression model with safe to cross (*stc*) as the outcome and year group (three, five or seven), female, *vis* and *oru* as predictors. The

improvement in deviance of this model over the null model is statistically significant, $G^2 = 13.7$, $p = .018$ ($\Delta AIC = 3.7$). However, the *SE* for the predictor *vis* is very high, suggesting quasi-complete separation. Only two of the children have visibility codes that are non-zero. Without additional data, the best option may be to drop *vis* from the model. The resulting model is also statistically significant, $G^2 = 11.6$, $p = .020$ ($\Delta AIC = 3.6$ versus the null model). Particular interest rests on differences between girls and boys (because proportionately more boys than girls are killed or injured in road traffic accidents). Underwood *et al.* (2007) fitted separate regressions for boys and girls and found *oru* predicted *stc* for girls, but not boys. A better model fits a *female* by *oru* interaction. The interaction model provides a marked improvement in fit relative to the main effects model: $G^2 = 17.0$, $p = .00046$ ($\Delta AIC = 7.0$). The coefficients *SE*s on the logit scale, Wald statistics as well as the OR and its 95% CI, are reported in Table 17.1.

Table 17.1 Regression coefficients, standard errors, Wald statistics and odds ratios for ordered logistic regression of the traffic data (including 95% CI for the *OR*)

	b	*SE*	Wald *z*	*OR*	95% CI (Wald)	
					Lower	Upper
stc > 0	−1.026	0.530	−1.94	0.358	0.127	1.128
stc > 1	−2.240	0.613	−3.66	0.106	0.032	0.354
stc > 2	−4.769	1.330	−3.58	0.008	0.001	0.115
year 5	−2.873	1.124	−0.75	0.057	0.006	0.512
year 7	−0.263	0.572	−2.55	0.769	0.251	2.358
female	−0.472	0.631	−0.46	0.624	0.181	2.146
oru	−0.091	0.353	−0.26	0.913	0.457	1.825
female × *oru*	1.403	0.607	2.31	4.066	2.146	13.348

Note: The coefficients and inferences arise from a model including year group, gender, other road users and a gender by other road user interaction.

The intercepts are the thresholds for the cumulative log odds of the model when all predictors are zero (i.e., for a year three male child with $oru = 0$). Here the first outcome ($stc = 0$) is the reference category and so the first threshold defines the log odds of $stc > 0$ versus 0. The middle threshold defines log odds for $stc = 2$ or 3 versus 0 or 1 and the last threshold splits $stc = 3$ from $stc < 3$. As the coefficients are cumulative log odds, the cumulative odds for each threshold are $e^{-1.026}$, $e^{-2.240}$ and $e^{-4.769}$ or 0.358, 0.106 and 0.00849. The respective cumulative probabilities defined by $\hat{P} = \hat{o}/(1 + \hat{o})$ are:

$stc > 0$	.26377
$stc > 1$	.09619
$stc > 2$	.00842

The conditional probability that $stc = 0$ is 1 minus the sum of these probabilities. By subtraction, the probabilities of each outcome (conditional on the other predictors being zero) are therefore:

$stc = 0$	.73624
$stc = 1$	.16757
$stc = 2$	.08777
$stc = 3$	.00842

Cumulative probabilities incorporating predictor effects can be obtained from the regression equation in probability form:

$$P(Y \leq stc) = \frac{e^{b0_{stc} - 2.873 year5 - 0.263 year7 - 0.472\ female - 0.091 oru + 1.403\ female \times oru}}{1 + e^{b0_{stc} - 2.873 year5 - 0.263 year7 - 0.472\ female - 0.091 oru + 1.403\ female \times oru}}$$

The effect of particular interest here is the *female* × *oru* interaction. This indicates that the *oru* effect is negligible for males ($OR = e^{-0.907} = 0.91$) but positive for females ($OR = e^{1.403} = 4.07$). So the odds of a moving up a threshold (to the next highest *stc* value) increase by roughly four for a one-unit increase in *oru*, but only for girls. Thus the 'other road user' codes in the free sort predict safe to cross responses in girls, but not boys. Figure 17.5 shows the predicted probability of zero, one, two or three safe to cross codes as a function of the sex of a child and the number of *oru* codes for year three children. The other year groups show a similar pattern.

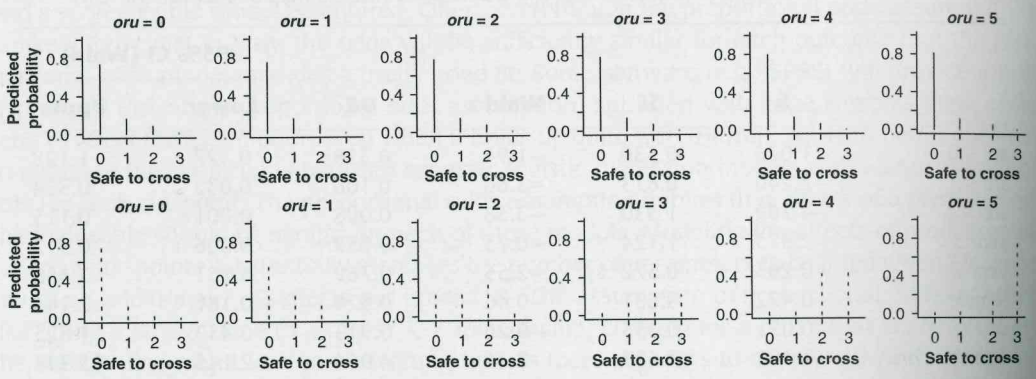

Figure 17.5 The predicted probability of zero, one, two or three safe to cross codes as a function of other road user (*oru*) codes by sex of child
Note: The upper panels show the predicted probabilities for boys (solid lines). The lower panels show the predicted probabilities for girls (dashed lines).

This illustrates the potential impact of the interaction effect. For boys (upper panel, solid lines) the presence of an *oru* code has little impact on a safe to cross response, whereas the probability of such a code increases markedly for girls (lower panel, dashed lines) as the number of *oru* codes increases.

Although this model produces broadly similar results to the multiple regression it would probably not be a good idea to rely on either analysis too heavily. The findings suggest an interesting and potentially important difference in how the boys and girls assess road conditions, but the data are too sparse to warrant high confidence in these conclusions. The proportional odds assumption could be checked for fitting separate logistic regression models for each cut-point (see Harrell, 2001).

17.5 Modeling count data

There are several potentially reasonable approaches when an outcome variable consists of independent counts. The starting point for modeling count data is very often the choice between a least squares linear regression model and Poisson regression. The residuals of least squares

regression with count data are often well approximated by a normal distribution in large samples, and a transformation will often produce an adequate model for moderate sample sizes. Poisson regression (a generalized linear model with a Poisson random component) is a logical alternative when the normal approximation is poor and may well produce a better predictive model. Models specialized for count data may also have superior statistical power and precision relative to least squares alternatives (see Atkins and Gallup 2007; Hilbe, 2007).

Because Poisson regression is somewhat restrictive, alternative models such as quasipoisson and negative binomial regression may be employed. Together these options provide great flexibility with pure count data. Sometimes, however, count data may arise from what can be considered a mixture of different processes. One process is dichotomous and determines whether an event occurs, while a second process determines the frequency of the event. Count data from this kind of mixture of processes may have an excess of zeroes relative to a Poisson or negative binomial model. Models for this kind of data are termed *zero-inflated* regression models and present additional difficulties to modelers. A related approach, that does not assume a mixture of processes, is the *hurdle* model in which all zeroes (excess or otherwise) are modeled in one component of the model and non-zero counts in another component. Count data with excess zeroes will rarely be adequately modeled using normal linear regression models (with or without a transformation) and present a particularly difficult challenge for researchers.

Box 17.2 Overdispersion parameters and corrected standard errors

The overdispersion parameter φ represents the degree to which data are overdispersed in the population being modeled (relative to variance implied by the choice of random component in a generalized linear model). This parameter is the ratio of the modeled variance to the expected population variance. It is therefore an indication of the degree to which the model underestimates the true variance. Overdispersion is meaningful only for statistical models in which the variance is a function of the mean. It is not relevant if the mean and variance are modeled by separate free parameters (e.g., in a model with normal distributed errors). A sample estimate of ϕ can be calculated from either the Pearson χ^2 or the residual deviance of the model (dividing them by the residual degrees of freedom ν). The estimate using deviance is therefore

$$\hat{\varphi} = \frac{D_M}{\nu},$$

and the estimate using the Pearson χ^2 is:

$$\hat{\varphi} = \frac{\chi^2_\nu}{\nu}$$

Both estimates should be similar in magnitude.

The expected value of $\hat{\varphi}$ in a model that is neither overdispersed nor underdispersed is one. Owing to sampling error, it will hardly ever be exactly one for real data. Values greater than one suggest overdispersion, while values less than one suggest underdispersion (rare for logistic regression, but a plausible, if infrequent, outcome in Poisson regression). Some authors (e.g., Field, 2009) recommend the cautious approach of picking the more extreme statistic to assess overdispersion. More commonly, people report the Pearson χ^2 estimate of φ (and the choice of statistic is rarely

critical). It is even possible to construct a NHST for the overdispersion parameter, though this is generally undesirable; what matters is the degree of overdispersion (or underdispersion) and not whether it is present.

The effects of overdispersion are to underestimate the variance and therefore the *SE*s, while underdispersion overestimates them. This is one reason why more emphasis is placed on the former rather than the latter. The overdispersion parameter, being an estimate of the ratio of the true variance to the model estimate can be used to correct the inferences. This is usually accomplished by adjusting the *SE*s. The corrected *SE* is $\sqrt{\hat{\varphi}}$ larger than that in the overdispersed or underdispersed model, because the sampling variance of the statistic is proportional to the variance of the model. If the uncorrected *SE* for a parameter estimate is $\hat{\sigma}_{\hat{\theta}}$, the corrected *SE* is therefore:

$$\hat{\sigma}_{\hat{\theta}}\sqrt{\hat{\varphi}}$$

This should explain why $\hat{\varphi} \geq 2$ is considered a serious problem. If $\hat{\varphi} = 2$ then the $\sigma_{\hat{\theta}}$ will be too small by a factor of $1/\sqrt{2} \approx .71$. This is nearly 30% smaller than it should be. Even if φ is as low as 1.2 the uncorrected *SE*s will be almost 10% smaller than required. Many researchers are cautious about overdispersion and take steps to deal with it. Correcting the *SE*s manually is one method, but can be cumbersome for complex models. You could also incorporate the overdispersion parameter (or an equivalent parameter) within your statistical model (e.g., using a quasipoisson or negative binomial random component in place of the Poisson distribution).

Overdispersion or underdispersion is a difficulty for all forms of inference (not merely NHSTs), and the overdispersion parameter can be used to correct interval estimates (e.g., by adjusting the *SE* for a Wald CI). A correction to AIC *quasi-AIC* (qAIC or $qAIC_C$) has also been proposed (Lebreton *et al.*, 1992; Bolker *et al.*, 2009). This uses φ to rescale the deviance. Thus qAIC would be computed as:

$$qAIC = \frac{-2\ln(\ell)}{\hat{\varphi}_c} + 2k \qquad \text{Equation 17.33}$$

The extension to $qAIC_C$ (for small samples) is straightforward:

$$qAIC_C = \frac{-2\ln(\ell)}{\hat{\varphi}_c} + 2k + \frac{2k(k+1)}{N-k-1} \qquad \text{Equation 17.34}$$

Here $\hat{\varphi}_c$ is the best available estimate of the dispersion parameter – usually the estimate from the most complex model under consideration other than the saturated model (Richards, 2008).

17.5.1 Poisson regression

Outcomes in Poisson regression are assumed to be independent counts with a Poisson distribution. This distribution has a single rate parameter λ (lambda) that is both its mean and its variance. In a Poisson regression model, the goal is to model λ as a linear function of the predictors. The main complication is that count data are bounded at zero (they can not be negative) and, particularly when counts are small, a linear (additive) function for predicting λ is problematic.

Poisson regression is a form of generalized linear model in which a logarithmic link function is employed (Agresti, 1996). The random component consists of the Y counts assumed to have

a Poisson distribution. The canonical link function is:

$$g(\lambda) = \ln(Y)$$

Equation 17.35

Given that the rate parameter is the mean of a Poisson distribution it could also be expressed as $g(\mu) = \ln(Y)$. Putting this together with the systematic component gives a generalized linear regression equation:

$$\ln(y_i) = b_0 + b_1 x_{1,i} + \ldots + b_q x_{q,i}$$

Equation 17.36

As with logistic regression, this equation in the generalized linear equation can be rearranged to place the original untransformed outcome (the y_i counts) on the left-hand side. Applying the exponential function to both sides gives:

$$y_i = e^{b_0 + b_1 x_{1,i} + \ldots + b_q x_{q,i}}$$

Equation 17.37

Although Equation 17.37 expresses the formula in terms of the population, it is trivial to express this in terms of the expected or predicted counts:

$$\hat{Y} = e^{b_0 + b_1 X_1 + \ldots + b_q X_q}$$

Equation 17.38

This is a multiplicative model in the sense that a one-unit increase in X is associated with a fixed proportion change in Y. When a coefficient is zero the proportionate change is one and the expected count is unchanged; the predictor has no effect. The multiplicative relationship between predictor effects can be reflected directly in the regression or prediction equation. For instance, Equation 17.37 could be expressed as:

$$y_i = e^{b_0} \times e^{b_1 x_{1,i}} \times \ldots \times e^{b_q x_{q,i}}$$

Equation 17.39

The interpretation of Poisson coefficients is less complex than for logistic regression. The multiplicative effect of a coefficient b_q on the counts is given by e^{b_q}. If you prefer, this can be converted into a percentage or proportion increase. For example the percentage increase or decrease would be:

$$\Delta\hat{Y} = 100\left(e^{b_q} - 1\right)\%$$

Equation 17.40

For very small values of b_q it turns out that the value of the coefficient is approximately equal to the proportion of change (Gelman and Hill, 2007). A coefficient of $b_q = 0.08$ would be roughly equal to a 0.08 or 8% increase in Y for each unit increase in X. Thus if X increased by two you would expect the observed count to increase by about 16%. So a coefficient of -0.04 equates to roughly a 4% decrease in Y for a one-unit increase in X. This holds only for coefficients close to zero. As the coefficient departs from zero (e.g., above ±0.2) the underestimate of the change is progressively large (being substantial at ±0.5 and grossly inaccurate above this). If the time period or area of which counts are observed is fixed and known, each of the multiplicative coefficients e^{b_q} also can be interpreted as a risk or rate ratio (*RR*). If the time period or area varies between observations, the *RR* interpretation can sometimes be restored by including an *offset* (see Section 17.5.2).

A Poisson regression, like other generalized linear models, is fitted by ML estimation and supports model comparisons using deviance statistics or information criteria. CIs and large

sample tests can be obtained from the asymptotic Wald statistics (though profiling gives more accurate CIs in small samples). The deviance of a Poisson model is:

$$Deviance = -2\ln(\ell) = 2\sum_{i=1}^{N}\left(y_i \ln\left(\frac{y_i}{\hat{y}_i}\right) - (y_i - \hat{y}_i)\right) \qquad \text{Equation 17.41}$$

This calculation requires only y_i (the observed counts) and $\hat{y}_i$ (the predicted counts from the model). The difficulty therefore stems from estimating parameters that maximize the likelihood (as opposed to the formulas themselves). This expression implies that the deviance residuals, the square root of the contribution of each observation to the total deviance, take the form:

$$e_{D_i} = \left(\frac{y_i - \hat{y}_i}{|y_i - \hat{y}_i|}\right)\sqrt{2\left(y_i \ln\left(\frac{y_i}{\hat{y}_i}\right) - (y_i - \hat{y}_i)\right)} \qquad \text{Equation 17.42}$$

The Pearson residuals are the difference between the observed and predicted counts divided by their estimated *SD*. This *SD* is estimated from the square root of the estimated mean (which is also the variance of the Poisson distribution):

$$e_{P_i} = \frac{y_i - \hat{y}_i}{\sqrt{\hat{y}_i}} \qquad \text{Equation 17.43}$$

To obtain approximately constant variance for the residuals it is usual to get the adjusted or standardized residuals. Again this involves dividing by $\sqrt{1-h_i}$. The standardized residuals are preferred for assessing the distributional assumptions of Poisson regression, though studentized residuals (calculated in the usual way) are superior for detecting extreme observations. Corresponding influence measures such as Cook's distance can also be derived.

For model checking, deviance residuals tend to have better distributional properties than Pearson residuals. In large samples the approximate χ^2 distribution of the deviance and can be used to assess the goodness-of-fit of the model (though as always, comparisons between models are preferred). Plots of residuals can highlight extreme observations and sometimes reveal lack of independence (e.g., by plotting standardized residuals versus a potential source of dependency such as order of data collection or time of day), or departures from linearity (e.g., plotted against predictor or fitted values), but are less useful than for least squares models.

A serious concern for any Poisson regression model is overdispersion (see Box 17.2). Overdispersion (and sometimes underdispersion) occurs because a Poisson distribution has a single rate parameter that is both its mean and its variance. As in logistic regression, its variance is therefore a function of the mean. Poisson regression models can underestimate or overestimate the true variance if the population of counts being modeled is more or less variable than expected by its mean. For data where each observation is a count accumulated over the same units (e.g., a count of six represents six arguments for the same couple) overdispersion or underdispersion tends to be very common. Lindsey (1999) argues that correctly dispersed Poisson models are plausible only if the observations are accumulations of independent frequencies rather than counts *per se*. While overdispersion is common, it is still important to explore a number of models before concluding that overdispersion is both present and sufficient to distort the results. A poorly fitting model will make it difficult to gauge the degree of overdispersion or underdispersion correctly. For repeated measures data, multilevel Poisson or multilevel logistic regression should also be investigated.

If overdispersion is found to be a serious problem, then one option is to correct the standard errors using $\sqrt{\hat{\varphi}}$ (see Box 17.2). In recent years this approach has fallen slightly out of favor. Fitting an alternative model that allows both the variance and mean to vary is the preferred approach. In a multilevel model this can be accomplished indirectly (by modeling variation between higher-level units separately). In single-level models the preferred options are quasipoisson regression and negative binomial regression.

Example 17.5 This example again uses the surgical checklist data introduced in Example 7.7. These data came from eight hospitals in eight different countries. Factors other than the presence of the checklist might predict mortality. These include the quality of the healthcare in the hospital. This is hard to measure, but one possible proxy is the per capita GDP of the country the hospital is in. Haynes *et al.* (2009) coded this as a dichotomous predictor: *income* (low or high). Using Poisson regression it is possible to model the mortality count before and after introduction of the checklist as a function of *time* (pre or post), *op.k* (total number of operations in thousands) and *income*.

A main effects model with all three predictors has the prediction equation:

$$\ln(\textit{mortality}) = -1.113 + 0.752 \times \textit{pre} + 0.786 \times \textit{low} + 3.904 \times \textit{op.k}$$

Note that *time* and *income* are dummy coded with $pre = 1$ and $low = 1$ (hence the labeling of coefficients in the model). Scaling operations in thousands makes the operations coefficient easier to work with (e.g., 3.904 instead of 0.003904). There are 16 observations (two per hospital), so a main effects model with $16 - 3 - 1 = 12$ *df* can be compared with the (intercept-only) null model with $16 - 1 = 15$ *df*.

	df	*Deviance*	*AIC*
Null model	15	57.5	108.3
Main effects model	12	32.1	89.0

The change in deviance is $57.5 - 32.1 = 25.4$. The likelihood ratio test is statistically significant, $G^2 = 25.4$, $df = 3$, $p < .0001$, and the change in AIC substantial. A model with all two-way interactions requires three extra parameters, hardly improves model fit and slightly increases AIC ($\Delta df = 3$, $\Delta G^2 = -3.6$, $\Delta\text{AIC} = +2.4$).

The interpretation of coefficients is also clear. The predictors *pre* and *low* have positive coefficients. This indicates that mortality was higher before the checklist was introduced and is higher for low-income countries. The number of operations also increases mortality rates for trivial reasons; the fewer procedures, the fewer opportunities for patients to experience adverse health problems. The coefficients are on a log scale, so it is very helpful to reverse the transformation for interpretation. For *pre*, $e^{0.7524} = 2.1$, and it suggests the mortality rate was 2.1 times higher in hospitals before the checklist was introduced. Hospitals in low-income countries have mortality counts that are roughly 2.2 times higher than for hospitals in high-income countries, and every additional 1000 operations increases the mortality rate in a hospital by a factor of about 50. To gauge the impact of each predictor it is better to look at interval estimates rather than significance tests. Using profile likelihood, the 95% CIs (on the count scale) are:

pre	[1.34, 3.42]
op.k	[2.53, 1293.01]
low	[1.41, 3.49]

None of the intervals includes zero (though the effect of operations seems to be measured rather imprecisely).

There are several things worth checking at this stage. Several of the hospitals have quite influential points (e.g., Cook's distance of .53 and .64), but this is hardly surprising in a model with low residual *df*. What about overdispersion? The residual deviance is 32.8 with 12 *df*. This suggests $\hat{\varphi} = 32.1/12 = 2.68$. For Pearson χ^2, $\hat{\varphi} = 28.2/12 = 2.35$. Both statistics indicate substantial overdispersion (not too surprising given the paired mortality counts from within the same hospitals).

17.5.2 Offsets and rates

The Poisson distribution is often used to compare rates of occurrence of discrete events spread over time or over an area. Unlike a simple Poisson model, Poisson regression treats the rate as an additive function of a set of predictors. Modeling rates in this way is straightforward when the exposure to events is equivalent for all observations. Under such circumstances the model for the rates and the counts is equivalent. If a mean of ten counts were observed in a period of ten seconds, this is a rate of ten per ten seconds or (less clumsily) one event per second. In these situations it is possible to determine the expected rate from the predicted mean count after fitting the model. The rate can also be obtained directly from the model by adding an *offset*. If the opportunity to observe events is not equivalent for all observations, then adding an offset is a requirement.[13] From this perspective, an offset is merely an adjustment to a Poisson regression model that permits the predicted outcome to be interpreted as a rate rather than a count when exposure varies between units.[14]

To understand how an offset works, envisage a data set that consists of counts sampled over a period of ten days. These might represent the number of arguments for a married couple or accidents reported in a workplace. The data can therefore be represented as number of events (arguments, accidents) over a given time period (ten days). The number of events is the observed count Y and the time period is the *exposure* E. The rate is therefore defined as Y/E. Even in a laboratory experiment it may not be possible to fully control the exposure period. The exposure may vary naturally, by design or by misfortune (e.g., some workplaces or couples may contribute data for only seven or eight days rather than the full ten). To model the rate now requires a regression equation of the form:

$$\ln\left(\frac{y_i}{E_i}\right) = b_0 + b_1 x_{1,i} + \ldots + b_q x_{q,i} \qquad \text{Equation 17.44}$$

As the logarithm of a ratio is the difference between the logarithm of the numerator and denominator, this model can also be expressed as,

$$\ln(y_i) - \ln(E_i) = b_0 + b_1 x_{1,i} + \ldots + b_q x_{q,i} \qquad \text{Equation 17.45}$$

where the $-\ln(E_i)$ term is the offset. Most software for generalized linear models has options to include an offset. If the software doesn't, it can still incorporate an offset (provided it is

possible to place constraints on the parameter estimate). This is because Equation 17.45 can be rearranged (by subtracting the offset from both sides) to take the form:

$$\ln(y_i) = \ln(E_i) + b_0 + b_1 x_{1,i} + \ldots + b_q x_{q,i} \qquad \text{Equation 17.46}$$

This is a Poisson regression model in which the natural logarithm of the exposure has been added as a predictor, and where the slope of the predictor $\ln(E_i)$ has been forced to equal one. So an alternative (and more general) way of thinking about an offset is that it involves adding a predictor with a slope of exactly one into the model. In a sense, the exposure is a predictor with privileged status in the model – its effect is assumed to take precedence in adjusting the model (it doesn't compete with other predictors to explain variance and its coefficient is taken as given rather than estimated). This is a logical model to fit if the goal is to de-confound the effects of differential exposure from the outcome. There may be other occasions where it makes more sense to treat exposure as a regular predictor (e.g., if there is good reason to think that its effects vary from occasion to occasion) or if determining its influence on the outcome is part of the research. There may also be other situations in which it makes sense to treat a variable as an offset, rather than estimate its effects as an ordinary predictor.

Example 17.6 In Example 17.5 the outcome variable in the Poisson regression was the mortality count in eight hospitals before and after an intervention. In that model the number of operations in each hospital was used as a predictor in the regression. An alternative model, probably a better one, is to compare the mortality rates between hospitals. As the number of operations in each hospital varies considerably, the predicted outcomes cannot be converted to a rate per 1000 operations by dividing it by a common denominator. To model this mortality rate involves adding the logarithm of the number of operations (in 1000s) as an offset to the model. The model with no offset was:

$$\ln(mortality) = b_0 + b_1\, pre + b_2\, low + b_3\, op.k$$

The model with the offset becomes:

$$\ln(mortality) = b_0 + b_1\, pre + b_2\, low + \ln(op.k)$$

Fitting this model gives the prediction equation:

$$\ln(mortality) = 1.604 + 0.655\, pre + 0.768\, low + \ln(op.k)$$

Expressing this as a rate gives:

$$\frac{mortality}{op.k} = e^{1.604 + 0.655\, pre + 0.768\, low}$$

This model has deviance of 32.7 with 13 *df* and AIC = 87.5. Relative to the null model ($G^2 = 53.5$, $df = 15$, AIC = 104.3) the main effects model with the offset provides a superior fit ($\Delta df = 2$, $\Delta G^2 = -20.8$, ΔAIC = −16.8). The profile likelihood CIs for the rate ratios are:

pre	95% CI [1.25, 3.02]
low	95% CI [1.39, 3.43]

Although the coefficients have changed slightly, the biggest shift has been in the intercept.

The predicted mortality rate per 1000 operations for a hospital in a low-income country after the surgical checklist has been introduced is:

$$\frac{mortality}{op.k} = e^{1.604+0.655\times 0+0.768\times 1} = e^{1.604+0.768} = e^{2.372} = 10.7$$

For a similar hospital without the checklist, the rate would be 20.6. The overall ratio (risk ratio) for mortality post-checklist is .52, 95% CI [0.33, 0.80], indicating a substantial decrease in risk.

The overdispersion parameter (calculated from the Pearson χ^2) has also decreased slightly from 2.35 to 2.14. This is still a clear signal of overdispersion.

17.5.3 Negative binomial and quasipoisson regression

Adjusting standard errors for a fitted Poisson model using an overdispersion parameter can be time-consuming and rather inflexible. A more satisfactory approach is to model the dispersion of the data separately from the mean – using a quasipoisson or a negative binomial model.[15] Quasipoisson regression does this by fitting a model with a random component in which the variance is a linear function of the mean determined by an additional parameter:

$$\sigma^2_{Y_i} = \varphi\mu_i \qquad \text{Equation 17.47}$$

This parameter has been labeled φ because it is functionally equivalent to an overdispersion parameter (the difference being that it is incorporated directly into the model rather than being applied as a correction *post hoc*). A disadvantage is that software that fits a quasipoisson model can't readily assess its relative fit to the Poisson model. It provides the same parameter estimates, but with revised standard errors.

Quasipoisson isn't the only option. The negative binomial can be adopted as an alternative to the Poisson distribution. The negative binomial distribution is usually first encountered as a distribution for modeling n, the number of Bernoulli trials to observe r failures for a fixed probability P.[16] For this reason it is convenient to treat the parameters of the negative binomial in a generalized linear model as μ and σ^2 (which can be expressed as functions of P and r) and which are both constrained to be greater than zero. Cook (2009) shows how the mean of a negative binomial distribution can be written as:

$$\mu = r\frac{(1-P)}{P} \qquad \text{Equation 17.48}$$

For modeling the dispersion of count data, the crucial property of the negative binomial is that the variance can be expressed as:

$$\sigma^2 = r\frac{(1-P)}{P^2} = \mu + \frac{1}{r}\mu^2 \qquad \text{Equation 17.49}$$

Thus, at one level, the negative binomial distribution is just a convenient probability function with an extra parameter. What makes it particularly attractive is that when r is large, $1/r$ shrinks

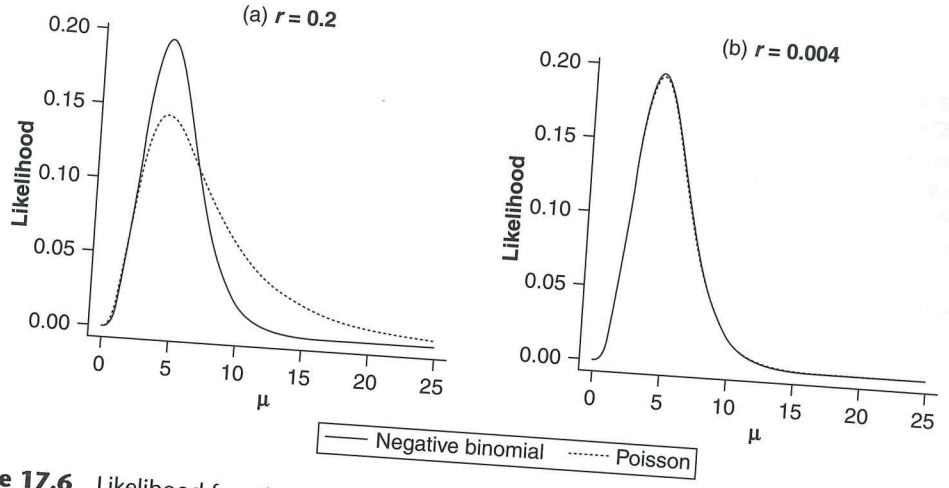

Figure 17.6 Likelihood functions for negative binomial and Poisson distributions, when $\hat{\lambda} = 4$ and (a) $r = 0.2$, or (b) $r = 0.004$

toward zero and the variance converges on μ (and therefore the probability mass function (*pmf*) of the negative binomial converges on the Poisson). Cook (*ibid.*) calls $1/r$ a 'clumping factor' that behaves like an inverted dispersion parameter. For large r, the counts clump together, but for small values they spread apart and hence overdispersion is observed. Figure 17.6 shows the relationship between the Poisson and negative binomial distributions. For convenience, the likelihood functions for the parameter μ of the Poisson and negative binomial distribution have been plotted for an observed count of $\hat{\lambda} = 4$.[17] Panel (a) indicates that even for moderately large values of r the probability distributions diverge and the Poisson is noticeably overdispersed relative to the negative binomial. In panel (b), when r is very small the probability distributions are not detectably different. This pattern is comforting because it indicates that the negative binomial will mimic the fit of the Poisson model if overdispersion is not present.

When should negative binomial regression be preferred over quasipoisson? Negative binomial regression has some practical advantages. The deviance of the model can readily be compared to that of a Poisson regression and it is in some senses a more natural model (Atkins and Gallup, 2007). Ver Hoef and Boveng (2007) point out that a fundamental difference is that the quasipoisson assumes a linear relationship between mean and variance, while the negative binomial distribution assumes a quadratic relationship. They suggest a diagnostic plot of the squared residuals of the model versus the mean (this is best done by grouping or binning data). Alternatively, if there are theoretical reasons to believe that a linear or quadratic function is more plausible this could also motivate selection of the appropriate model.

Example 17.7 The Poisson regressions of the surgical checklist data in Example 17.5 and Example 17.6 indicated that the counts were overdispersed. One option for dealing with this is to manually correct the *SE*s of the coefficients using the square root of the dispersion parameter. For example, the *SE* for the *pre* effect is 0.224. The Pearson overdispersion parameter is 2.14 and so, correcting for overdispersion, the *SE* should be $\sqrt{2.14} \times 0.224 = 0.327$. This procedure is useful if

your software does not offer quasipoisson or negative binomial regression as an option. qAIC or $qAIC_C$ can also be obtained manually if required. For instance, qAIC for the main effects model with offset is 41.33, while qAIC for the two-way interaction model with offset is 43.33. This suggests that the simpler model is slightly more informative ($\Delta qAIC = 2.0$).

Running the quasipoisson regression in R produces the corrected *SE* for each coefficient automatically. Even better, it also provides profile likelihood CIs for the quasipoisson model. Fitting this model with the offset ln(*op.k*) will necessarily produce identical coefficients to those from the Poisson regression. The 95% profile likelihood CIs for mortality rate ratios from the quasipoisson model (corrected for overdispersion and including the offset) are:

pre [1.03, 3.75]
low [1.14, 4.30]

Both CIs are wider than before. In addition, the interval estimate for the pre-post difference in mortality now only barely excludes one. The *p* value from the Wald test is .067 and from the more accurate likelihood ratio test it is .041.

Fitting a negative binomial model is also an option here. The negative binomial regression is more than a linear adjustment to the variance – it fits a different distribution – and so may produce different parameter estimates. In a negative binomial model with the same predictors and an offset the prediction equation is:

$$\ln(mortality) = 1.613 + 0.653\,pre + 0.736\,low + \ln(op.k)$$

This model has deviance 20.6 with 13 *df* and AIC = 84.6. The coefficients are indeed slightly different (e.g., the difference between *pre* and *post* checklist mortality is a little larger). The profile likelihood CIs for the mortality rate ratios are:

pre [1.06, 3.52]
low [1.15, 3.85]

Both 95% CIs now exclude one and the likelihood ratio *p* value for the test of time is .031.

The choice between negative binomial and quasipoisson is not necessarily an easy one (though they often produce similar models). Given that the negative binomial is perhaps a more principled approach to overdispersion it would be my default choice. Here both models point to very similar models of the data (and it would be unwise to focus too much on the difference in *p* values). Diagnostics such as Cook's distance also hint at a marginal preference for the negative binomial model, which has slightly lower influence statistics (all now < .30).

17.5.4 Dealing with zero-inflated count data

A feature of count data is that zeroes are not infrequent, even if the rate at which events occur exceeds zero. Poisson and negative binomial models both assume a rate of occurrence greater than zero, so (in these models) the absence of events is either bad luck or indicates an exposure too narrow to observe them. This handles the presence of zero counts for some phenomena, but is unreasonable for others. What if some of the zero counts represent a true absence of the event; measurements on a unit that simply doesn't generate them? A well-known illustration

is for criminal behavior. A study might look at a large sample of teenagers to determine what factors influence violent criminal behavior. Some of the teenagers will commit one or more violent crimes, but the majority (if the sample is representative) won't. One way to consider this is to suppose that there are two subpopulations of teenagers – one with a zero rate of violent crime and one with a non-zero rate.[18] A potential solution is to fit separate statistical models: one to predict whether a teenager has a zero or non-zero rate of violent crime and one to predict the number of crimes for the latter. This strategy leads to a class of models termed 'mixture models' (Atkins and Gallup, 2007). For the first model a way of predicting the occurrence of an event is required. Coding occurrence as one and non-occurrence as zero leads to the problem of predicting a dichotomous outcome from a set of predictors, and thus a logistic regression model can be employed. To predict the non-zero crime counts several models could be considered, but Poisson and negative binomial regression are the obvious choices (depending on the dispersion of the counts).

A *zero-inflated Poisson regression* is therefore a mixture model with a Poisson regression nested within a logistic regression:

$$Y \sim f(y_i) \begin{cases} 0 & \text{with} \quad P(Y=0) = 1 - P_i \\ \sim Poisson\left(e^{b_0+b_1x_{1,i}+\ldots+b_qx_{q,i}}\right) & \text{with} \quad P(Y>0) = P_i \end{cases} \qquad \text{Equation 17.50}$$

A zero-inflated negative binomial regression is a mixture model of the form:

$$Y \sim f(y_i) \begin{cases} 0 & \text{with} \quad P(Y=0) = 1 - P_i \\ \sim NB\left(e^{b_0+b_1x_{1,i}+\ldots+b_qx_{q,i}}, \sigma^2\right) & \text{with} \quad P(Y>0) = P_i \end{cases} \qquad \text{Equation 17.51}$$

In both equations $f(y_i)$ designates a function of two other probability distributions – one producing zeroes (with a fixed probability $1 - P_i$) and one producing Y counts of the modeled outcome according to a Poisson or negative binomial distribution with fixed probability P_i.

Figure 17.7 shows the differences between Poisson, zero-inflated Poisson and zero-inflated negative binomial distributions for the same population mean. Each plot shows histogram of a simulated random sample of 10,000 observations for an expected mean count of seven. Panel (a) shows the Poisson distribution, (b) the zero-inflated Poisson when the probability of a zero (ignoring the Poisson counts) is $P = .2$ and (c) shows the negative binomial distribution for $P = .2$ and $r = 0.2$. The zero-inflated Poisson is very similar to the Poisson distribution, with about 20% of the observations 'shaved off' the distribution and allocated to zero. The zero-inflated negative binomial in panel (c) has greater dispersion and a more pronounced spike at zero (comprising nearly 25% of the observations). Because the population parameters are known for these simulated data, this is in excess of the 20% 'structural' zeroes expected when $P = .2$. For a real data set the probability would have to be estimated from what is likely to be a very noisy sample.

Mixture models are particularly difficult to fit (Atkins and Gallup, 2007). It isn't good enough to fit separate models to the zeroes and non-zeroes in the sample. The difficulty is that a zero could arise either from the logistic regression with probability $1 - P_i$, or it could arise from the Poisson or negative binomial count distribution. It may help to refer back to Figure 2.4a. This shows the *pmf* for a Poisson distribution with $\lambda = 2$. Zero counts are fairly common when λ, the population rate parameter, is small. Fitting zero-inflated count data therefore involves optimizing both fits simultaneously to provide the best overall explanation of the observed data. In addition, predictors may have different effects for the probability and count parts of the model.

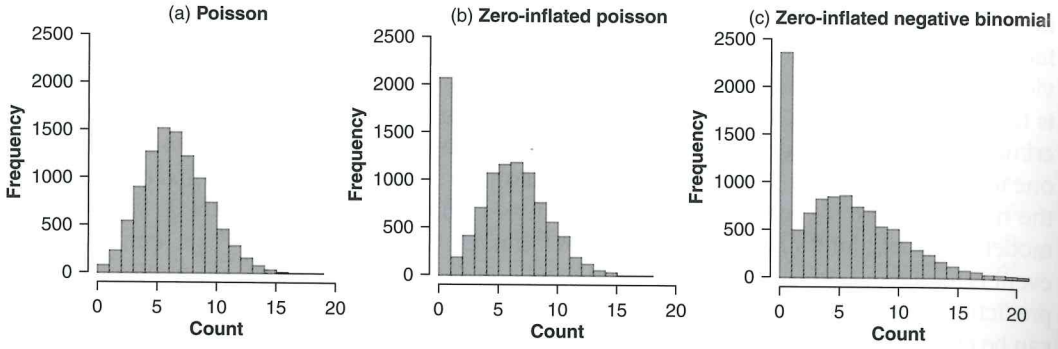

Figure 17.7 Counts sampled from simulated data, with $\mu = 7$ and either (a) a Poisson distribution, (b) a zero-inflated Poisson distribution with $P = .2$, or (c) a zero-inflated negative binomial distribution with $P = .2$ and $r = 0.2$

Modeling the data as a mixture in this way can lead to difficulties of interpretation. Perhaps the most important point is that it will always be safer to interpret the overall fit and predictions of the model rather than interpret the components separately. Zeroes in the model arise from both components, so interpreting the components in isolation could lead to misleading conclusions. An analogy here can be made with interpreting interaction terms in a regression model; interpretation of the product term in isolation will be misleading relative to plotting the predictions of the model. One situation in which the separate interpretation of the components can be defended is if there is strong theoretical justification for the view that a mixture of two populations is being modeled. If this mixed population interpretation is appropriate then inferences based on separate components can be restricted to the population of interest. If this interpretation is not justified there may be advantages to switching to a different approach such as a hurdle model. A hurdle model also separates performance into two components, but one component (typically a logistic regression) models all the zeroes, while the other component (a truncated Poisson or negative binomial) models the non-zero counts (see Zeileis *et al.*, 2008). Hurdle models and zero-inflated models tend to produce very similar overall fits, so the choice between them relates to the interpretation of the separate components. In a hurdle model it is easier to separate out the effects of predictors on zero responses and non-zero responses.

17.6 Modeling discrete outcomes with correlated measures

The generalized linear models considered so far have all assumed independent observations. The random component of the models has distribution such as the normal, binomial, Poisson or negative binomial that requires independence. In the case of a normal generalized linear model it is possible to develop models that explicitly account for the correlation and assume independence of residuals. In addition, for both logistic regression and Poisson regression, correlated observations can be a cause of overdispersion (or even underdispersion). Modeling the dispersion (e.g., using negative binomial regression) may also provide a partial solution to problems with correlated or repeated outcomes. A more principled approach is possible by setting up a *generalized estimating equation* (GEE) or a multilevel generalized linear model (see Chapter 18).

More restrictive, but simpler approaches also exist for some situations (see Agresti, 1996). One well-known approach, for working with paired dichotomies, is illustrated here.

17.6.1 Logistic regression with paired observations

A relatively simple approach to dealing with repeated or correlated measures where outcomes are dichotomous is possible for paired data (e.g., from matched pairs or repeated measures designs). The approach is known as *conditional maximum likelihood* or *conditional logistic regression*. Agresti (*ibid.*) shows how this is related to simpler procedures such as McNemar's test of change in 2×2 tables (and can be extended to more complex study designs).

For paired data, the model of interest is probably of the form:

$$\ln\left(\frac{\hat{P}_{ij}}{1-\hat{P}_{ij}}\right) = b_{0,j} + b_1 x_{1i,j} + \ldots + b_q x_{q,ij} \qquad \text{Equation 17.52}$$

This looks to be a regular logistic regression except that the intercept b_{0j} is a random variable equivalent to a subject term in a one-way repeated measures ANOVA. Unfortunately, fitting multiple intercepts causes estimation difficulties if there are large numbers of participants (*ibid.*). Conditional logistic regression works by eliminating the subject effect from the likelihood estimate entirely. The conditional maximum likelihood is therefore estimated for the parameters (in the sense that the estimates are all conditional on the subject term).

This can be done using specialized software, but it is also possible to arrive at the conditional estimates using standard logistic regression software. This is done by fitting a no-intercept model with a constant as the outcome (i.e., $Y = 1$ for all cases) and with the difference between the paired cases (a series of values equal to -1, 0 or 1) as the predictor. Applied to a 2×2 table, the difference between the deviance of the null and residual model would be roughly equivalent to the more familiar McNemar test. However, unlike the McNemar test, other predictors can be added to the model. These are included as additional predictors (along with the differences) in the usual way. If the cases differ in terms of some predictor value it is also possible to use the between-pair difference in predictor as an explanatory variable.

Although conditional logistic regression is useful, it can be awkward to implement (even using the shortcuts described here), and is inflexible. On the other hand, it can cope with paired repeated measures, with 1:1 matching or 1:x matching between cases and controls (where there are several controls for each case). Other forms of repeated or correlated measures designs require different solutions. A final issue is that the random effect is treated as a nuisance variable rather than being explicitly modeled (as it would be in a multilevel model).

Example 17.8 To illustrate how conditional logistic regression can be fitted, I'll use an imaginary case-control study (in which 30 cases are matched with 30 controls). The question of interest is whether a potential risk factor (*prf*) is associated with increased odds of being a case versus a control. The contingency table looks like this:

		Controls		
		no prf	*prf*	
Cases	*no prf*	8	4	*12*
	prf	17	1	*18*
		25	*5*	*30*

To run a conditional logistic regression using standard logistic regression software requires data arranged as two columns: one for cases and one for controls. As there are 30 case-control pairs, there will be 30 rows. Of these, one row will contain one for both cases and controls (indicating both have the potential risk factor). Seventeen rows will have one entered for cases and zero for controls. Four rows will have zero for case and one for controls. The remaining eight cases will have zero for both cases and controls. For the analysis you will also need a column of constants (with 30 rows containing one in each cell). Last, you will need a column of differences (created by subtracting the control column from the case column).

After setting the data up as described, the next step is to run a logistic regression with no intercept and with the column of constants the outcome. The model has one predictor: the difference scores (though it would be possible to add others). The cases and controls are not entered into the analysis directly – they are just used to create the difference scores. The log odds estimated from this model (for the difference predictor) are the log odds of cases having the risk factor relative to controls. These log odds are 1.45. The corresponding odds ratio is therefore $e^{1.45} = 4.25$, or .81 expressed as a probability. This odds ratio is identical to that obtained from the McNemar test ($17/4 = 4.25$). Using profile likelihood, the 95% CI for the odds ratio is [1.57, 14.77]. The likelihood ratio or deviance test is: $G^2\ (1, N = 30) = 8.66$, $p = .0032$.

This procedure ought to work with most logistic regression software (but may not in practice). It is also possible to 'trick' other procedures to run the analysis – including survival analysis in SPSS or R (though this requires data arranged in a slightly different form).

17.7 R code for Chapter 17

17.7.1 The logistic function

The logistic function can be plotted via several routes. Figure 17.1 used the `curve()` function and specified the equation directly:

```
curve(exp(x)/(1+exp(x)), ylab = 'Probability', xlab = NA,
  xlim=c(-5,5), lwd=2, lty=3)
```

This shows how the predictive probability depends on the predictors (in this case *x*). It is also instructive to see the probability density plotted against that of the normal distribution (as in Figure 17.2). The normal distribution needs to be scaled to have the same standard deviation (defined by the first command below). The remaining functions plot the density of a standard logistic distribution and add the scaled normal on top (with an appropriate legend).

```
logis.sd <- pi/3^0.5
curve(dlogis(x), xlim=c(-5,5), ylab = 'Probability density',
  lwd=1)
curve(dnorm(x,0,logis.sd), lty= 3, add = TRUE, lw=1.5)
legend(1.6, 0.23, legend = c('Logistic', 'Scaled normal'), lty
  =c(1,3), cex = 1, lwd=c(1.5,1))
```

Note that a similar figure to that in Figure 17.1 could have been obtained directly using the *cdf* or inverse *quantile* function for the logistic distribution. This is simpler, but hides the form of the equation (which is worth becoming familiar with):

```
curve(plogis(x), xlim =c(-5,5))
```

17.7.2 Logistic regression and χ^2 (Example 17.1)

For the dream data set it is quite easy to set up the data by creating a numeric vector with either zero or one for group membership or the 'scary dream' response. Using the `rep()` reduces the need to type in lots of numbers (e.g., `rep(0,26)` repeats the value zero 26 times). The first vector specifies the group (zero for no suggestion and one for suggestion). The second specifies the response (zero for no scary dream and one for scary dream).

```
group <- c(rep(0,26), rep(1,21))
scary <- c(1, rep(0,25), rep(1,7), rep(0,14))
```

The logistic regression is run using the `glm()` function. This specifies a general linear model using a formula similar to that of the `lm()` function. One difference is that a family is now specified for the random component of the generalized linear model. This defaults to `family=normal`. The `family()` function used as an argument to `glm()` also determines the link function (which defaults to the canonical link function and can therefore often be ignored). Thus the following are identical in output:

```
glm(scary ~ group, family=binomial(link = logit))
glm(scary ~ group, family=binomial)
```

If the family is not specified, `glm()` is equivalent to `lm()` for most purposes. For this reason `glm()` is often used as a convenient generic function for regression modeling. However, because the output from the functions is delivered slightly differently, there can be differences in how other functions act on objects (models) created by the two functions.

Wald tests are given by the `summary()` function for the model:

```
subbot.mod <- glm(scary ~ group, family=binomial)
summary(subbot.mod)
```

The change in deviance for the group effect can be obtained by the `anova()` function. The `drop1()` function also gives the change in deviance, AIC and, if requested, the likelihood ratio test of the change in deviance:

```
anova(subbot.mod)
drop1(subbot.mod, test = 'Chisq')
```

The `confint()` command gives the profile likelihood CIs for generalized linear models.

```
confint(subbot.mod)
```

As dummy coding is being used for the grouping variable it is also very simple to reverse the coding. The main reason to do this is simply to change the intercept to obtain the CI for the other group:

```
summary(glm(scary ~ I(1 - group), family=binomial))
confint(glm(scary ~ I(1 - group), family=binomial))
```

The usual Pearson χ^2 test of independence can be obtained by creating a contingency table and using `chisq.test()`:

```
ctable <- matrix(nrow=2, ncol=2)
ctable[1:4] <- c(1, 7, 25, 14)
ctable
chisq.test(ctable, correct=FALSE)
```

17.7.3 Interpreting logistic regression coefficients (Example 17.2)

Example 17.2 modeled the effect of a continuous predictor – the electoral majority of a UK member of parliament – on a discrete outcome (whether there was a reported allegation of expenses abuse). As the majority data range from 37 to nearly 20,000 it is convenient to rescale the majorities by expressing them in units of 10,000:

```
expenses <- read.csv('expenses.csv')
majority.10k <- expenses$majority/10000
```

The model with majority as predictor and problem as outcome can then be fitted using `glm()`.

```
model.10k <- glm(problem ~ majority.10k, family=binomial,
  data = expenses)
summary(model.10k)
```

Wald CIs on the logit scale are computed by adding or subtracting the appropriate margin of error (e.g., ±1.96 SE for a 95% CI). However, the profile indicates the likelihood CIs that should be more accurate:

```
confint(model.10k)
```

In most cases the odds ratios are easier to interpret than the logit scale coefficients. The CIs on the odds scale can be obtained by exponentiation:

```
exp(model.10k$coefficients)
exp(confint(model.10k))
```

The predictions for each MP on the logit scale are easily obtained using the `predict()` function. To change the scale to the predicted probabilities (by using the inverse of the link function), `type='response'` can be specified. Alternatively the fitted values of the model can be used (as these are calculated on the untransformed response scale). Compare the following outputs:

```
predict(model.10k)
predict(model.10k, type='response')
model.10k$fitted.values
```

Plotting predicted values on the probability scale versus a predictor such as majority is now easy:

```
plot(expenses$majority,model.10k$fitted.values,
  ylab='Probability(Expenses problem)', xlab='Majority')
```

This is similar to the plots in Figures 17.3 and 17.4. Figure 17.4 also adds approximate Wald confidence bands. This is done by using the predict function to obtain standard errors for 20,000 or so predicted new values (with a range of zero to two for the majority variable scaled in units of 10,000) and using these to calculate a margin of error at each point. Adding or subtracting these to the fitted values gives upper or lower bounds for the log odds.

```
maj <- data.frame(majority.10k = seq(0, 2, 1/10000))
moe <- predict(model.10k, newdata=maj, se.fit=TRUE)[[2]] *
  qnorm(.975)
ub <- predict(model.10k, newdata=maj, se.fit=FALSE) + moe
lb <- predict(model.10k, newdata=maj, se.fit=FALSE) - moe
```

Adding these to a plot `lines()` joins the points and gives the appearance of a smooth function:

```
lines(c(0:20000),exp(lb)/(1+exp(lb)), col='dark gray')
lines(c(0:20000),exp(ub)/(1+exp(ub)), col='dark gray')
```

The rest of the example considers the deviance and AIC for the model:

```
anova(model.10k, test = 'Chisq')
drop1(model.10k, test = 'Chisq')

model.null <- glm(problem ~ 1, family=binomial, data =
  expenses)

AIC(model.10k) - AIC(model.null)

LR.aic <- 1/exp((AIC(model.10k) - AIC(model.null))/2)
LR.aic
```

The model isn't improved by adding party affiliation (given that this requires an extra 17 *df*) or the extra four parameters when minor parties are categorized as 'other'.

```
all.parties <- glm(problem ~ majority.10k + factor(Party),
  family=binomial, data = expenses)
drop1(all.parties)
main.parties <- glm(problem ~ majority.10k + Lab + Con +
  LibDem + SNP + Other, family=binomial, data = expenses)
drop1(main.parties)
```

Note that if the conventional significance test is applied for the `all.parties` model the more complex model is a statistically significantly better fit:

```
drop1(all.parties, test = 'Chisq')
```

This is a good example where throwing predictors at a model will often produce a substantial shift in fit. This is a doubly bad model; because it ignores the problem of over-fitting and because some of the smaller 'party' labels are for MPs who had been ejected from a main party because of expenses allegations.

17.7.4 Model checking in logistic regression (Example 17.3)

Quantities such as the standardized and unstandardized residuals, Cook's distance and leverage can be obtained from a `glm()` model in the same way as for a linear regression model.

```
cooks.distance(model.10k)
resid(model.10k)
```

By default the residuals are the working residuals of the model (on the transformed scale – not in terms of the untransformed response). The standardized and studentized residuals may also be useful and can be obtained with familiar commands:

```
rstandard(model.10k)
rstudent(model.10k)
```

The summary command also provides the dispersion parameter used for the model. This should be one for a logistic regression, and refitting the model as a quasipoisson model gives a simple method to extract the dispersion parameter:

```
summary(model.10k)$dispersion
model.10k.q <- glm(problem ~ majority.10k,
  family=quasibinomial, data = expenses)
summary(model.10k.q)$dispersion
```

17.7.5 Ordered logistic regression (Example 17.4)

The traffic data used in this example can be read into R from an SPSS file.

```
library(foreign)
traffic <- read.spss('traffic.sav', to.data.frame=TRUE)
```

To run an ordered logistic regression there are several options including `lrm()` in the `rms` package and `polr()` in `MASS`. The data here are very sparse and difficult to fit. The `lrm()` function seems to cope best. The following model fits indicate the problem with the visibility predictor `vis`.

```
install.packages('rms')
library(rms)

traf.me <- lrm(stc ~ female + year + oru + vis, data=traffic)
traf.me
traf.null <- lrm(stc ~ 1, data=traffic)
traf.null
```

A better model drops `vis` and there is also some indication that the fit is improved by adding the `female:oru` interaction:

```
traf.me2 <- lrm(stc ~ female + year + oru, data=traffic)
traf.int <- lrm(stc ~ female + year + oru + female:oru,
  data=traffic)
traf.me2
traf.int
```

The formula for the `traf.int` model could be represented more succinctly as stc ~ year + female*oru.

17.7.6 Poisson regression (Example 17.5)

Example 17.5 returns to the surgical checklist data. It models the mortality rates across the eight different hospitals, using data in the 'checklist.csv' data file. Again, it helps to rescale some of the predictors. In this case the number of operations is scaled in terms of units of 1000. A main effects model adds predictors for the number of operations and dummy indicators for time (pre or post checklist) and income (low or high income per capita countries).

```
checklist <- read.csv('checklist.csv')
op.k <- checklist$operations/1000

mort.me <- glm(mortality ~ time + op.k + income,
  data=checklist, family=poisson)

summary(mort.me)
```

For the difference in deviance and likelihood ratio test versus the null model use the `anova()` function:

```
anova(mort.me, update(mort.me, ~ 1), test = 'Chisq')
```

The model with all two-way interactions can be tested in same way:

```
mort.int <- glm(mortality ~ (time + operations + income)^2,
  data=checklist, family= poisson)

anova(mort.me, mort.int, test = 'Chisq')
AIC(mort.int) - AIC(mort.me)
```

The CIs (using profile methods) for the log scale and the count scale are given by:

```
confint(mort.me)
exp(confint(mort.me))
```

Residuals and Cook's distance can be obtained with the usual commands:

```
residuals(mort.me)
cooks.distance(mort.me)
```

The residual deviance obtained directly from the model object could also be calculated from the residuals directly:

```
mort.me$deviance
sum(residuals(mort.me)^2)
```

The Pearson χ^2 statistic of 28.2 is also easy to calculate from residuals, but it is necessary to use the Pearson residuals:

```
sum(residuals(mort.me, type='pearson')^2)
```

Either quantity can be used to estimate the dispersion parameter by dividing by the residual *df*, though the Pearson χ^2 produces larger estimates and tends to be preferred.

```
sum(residuals(mort.me)^2)/12
sum(residuals(mort.me, type='pearson')^2)/12
```

The latter is the parameter reported by R for a fitted quasipoisson model:

```
summary(update(mort.me, family = quasipoisson))$dispersion
```

17.7.7 Offsets and rates (Example 17.6)

Treating the number of operations in the previous example as a predictor is probably not the best approach. A better approach, that allows the Poisson model to treat the mortality as a rate per operation (strictly per 1000 operations) is to enter the number of operations as an offset.

```
mort.off <- glm(mortality ~ time + income, data=checklist,
  offset=log(op.k), family=poisson)
```

An equivalent way to express the model with the offset explicit in the formula is also sometimes useful:

```
glm(mortality ~ time + income + offset(log(op.k)),
  data=checklist, family=poisson)
```

The CIs for the main predictors are broadly similar in this instance, but the overdispersion is slightly less severe:

```
exp(confint(mort.off))
sum(resid(mort.off, type = 'pearson')^2)/13
```

The `drop1()` function is also useful for investigating the effects of individual predictors with AIC or with an NHST of the change in deviance:

```
drop1(mort.off, test = 'Chisq')
```

17.7.8 Negative binomial and quasipoisson regression (Example 17.7)

Figure 17.6 shows how the negative binomial distribution and the Poisson are almost indistinguishable when the 'clumping factor' r is very small (and overdispersion negligible). A continuous likelihood function (rather than discrete *pmf*) is used to make the plots clearer (and easier to produce). The likelihoods are generated using the `dpois()` and `dnbinom()` functions. The `dnbinom()` function uses a slightly different parameterization than that in Equation 17.48, using a 'size' parameter equal to $1/r$. The following plot matches panel (b) of Figure 17.6 and shows the similarity of the two distributions when 'clumpiness' is low.

```
siz <- 250
curve(dpois(4, x), xlim=c(0,25), xlab=expression(mu),
  main=expression(paste('(b) ',italic(r),'= 0.004')),
  cex.main =0.95, ylab = 'Likelihood')
curve(dnbinom(4, size=siz, mu=x), add = TRUE, lty = 3)
legend(12.5, 0.175, legend = c('negative binomial','Poisson'),
  lty=c(1,3), cex = 0.8)
```

To show the potential difference in the distribution as 'clumpiness' increases, compare it with panel (a):

```
siz <- 5
curve(dpois(4, x), xlim=c(0,25), lty = 1, xlab=expression(mu),
  main=expression(paste('(a)', italic(r),'= 0.2')), cex.main
  =0.95, ylab = 'Likelihood')
curve(dnbinom(4, size=siz, mu=x), add = TRUE, lty = 3)
legend(12.5,.175, legend = c('negative binomial','Poisson'),
  lty=c(1,3), cex = 0.8)
```

To deal with the overdispersion in the checklist data, one option is to adjust the *SE*s using the overdispersion parameter. R makes this option simpler by fitting a quasipoisson model that also supports profile likelihood CIs. The `drop1()` function also provides accurate NHSTs (but not AIC).

```
mort.oq <- glm(mortality~time+income, data=checklist,
  offset=log(op.k), family=quasipoisson)
summary(mort.oq)

exp(confint(mort.oq))
drop1(mort.oq, test = 'Chisq')
```

The `bbmle` package provides functions to compute qAIC and $qAIC_C$. These require the estimated dispersion parameter as input. Here the dispersion parameter is extracted from a quasipoisson model and qAIC computed for the two-way interaction model (the most complex under consideration):

```
library(bbmle)
disp <- summary(glm(mortality ~ time * income, data=checklist,
  offset=log(op.k), family=quasipoisson))$dispersion

mort.off.int <- glm(mortality ~ time * income, data=checklist,
  offset=log(op.k), family=poisson)

qAIC(mort.off, dispersion = disp)
qAIC(mort.off.int, dispersion = disp)
qAIC(mort.off, dispersion = disp) - qAIC(mort.off.int,
  dispersion = disp)
```

As qAIC is rather simple to compute, it may be easier to do it directly from the dispersion and loglikelihood. This calculation should match for the `mort.off.int` model above:

```
(logLik(mort.off.int)[1]*-2)/disp + 4 * 2
```

The negative binomial model is less limited than the quasipoisson, but requires a package such as MASS to fit it. MASS provides the `glm.nb()` function to fit negative binomial generalized linear models. This has a default 'log' link function and works slightly differently from `glm()` (e.g., in requiring an offset to be part of the formula). Refitting the checklist main effect model with an offset is done as follows:

```
library(MASS)
mort.nb <- glm.nb(mortality ~ time + income +
  offset(log(op.k)), data=checklist)
summary(mort.nb)
```

Again the profile likelihood CI and tests of the predictors are obtained using familiar functions:

```
mort.nb <- glm.nb(mortality ~ time + income +
  offset(log(op.k)), data=checklist)
summary(mort.nb)
```

```
exp(confint(mort.nb))
drop1(mort.nb, test = 'Chisq')
```

Residuals, Cook's distance and so forth are also provided:

```
resid(mort.nb, type ='pearson')
predict(mort.nb, type = 'response')
cooks.distance(mort.nb)
```

17.7.9 Modeling zero-inflated count data

Zero-inflated Poisson, negative binomial models and other mixture models are not yet routinely available in statistical software. Functions for working with these models are found in a number of R packages. The plots in Figure 17.7 use the `rzipois()` function from the `VGAM` package and the `rzinbinom()` function from the `emdbook` package.[19] Using these functions, Figure 17.7 is:

```
install.packages('emdbook')
install.packages('VGAM')

library(emdbook)
library(VGAM)

count <- 7
prob <- .2
n <- 10000
size <- 5

par(mfrow=c(1,3), mar=c(4,4,2,1)+.1, pty='s', cex=1)
hist(rpois(n, count), xlim=c(0,20), ylim=c(0,2500), main='(a)
  Poisson', cex.main=1.1, xlab = 'Count', col='light gray')
hist(rzipois(n, count, prob), xlim=c(0,20), ylim=c(0,2500),
  main ='(b) Zero-inflated Poisson', cex.main=1.1, xlab =
  'Count', col='light gray')
hist(rzinbinom(n, count, size, prob), xlim=c(0,20),
  ylim=c(0,2500), breaks = 25, main='(c) Zero-inflated
  negative binomial', cex.main = 1.1, xlab='Count', col='light
  gray')
```

`VGAM` can be used to fit zero-inflated count models with the `vglm()` function. Although `vglm()` is very powerful, a more user-friendly option for extending Poisson and negative binomial regression models is the `zeroinfl()` function from the `pscl` package. For instance, to fit an intercept only zero-inflated Poisson or negative binomial model for the checklist data you'd specify the formula like this:

```
install.packages('pscl')
library(pscl)
```

```
zeroinfl(mortality ~ 1, dist='poisson', link = 'logit',
  data=checklist)

zeroinfl(mortality ~ 1, dist='negbin', link = 'logit',
  data=checklist)
```

The `dist` argument indicates the family used for the count part of the mixture model (the link function here is always the log and need not be specified). The `link` argument specifies the link function for the zero-inflation part of the mixture model. The defaults are `dist='poisson'` and `link = 'logit'`. The following model adds `income` as a predictor to both parts of the model:

```
zeroinfl(mortality ~ income, data=checklist)
```

Sometimes it makes sense to add predictors only to the count part of the model (or possibly only the zero-inflation part). Here, although a zero-inflation model doesn't look that plausible, it is more plausible (and fits better) if the predictors (including offset) are added to the count part of the model only. This is achieved by separating the predictors in each component using | (not to be confused with its role in other functions such as multilevel models).

```
zeroinfl(mortality ~ time + income + offset(log(op.k))|1,
  data=checklist)
```

You may prefer to fit a hurdle model in place of the zero-inflated model. This is also possible within the `pscl` package by using the `hurdle()` function with the same formula structure:

```
hurdle(mortality ~ time + income + offset(log(op.k))|1,
  data=checklist)
```

In general the models will have very similar overall fits and predictions (but slightly different parameter estimates because of the way excess zeroes are modeled). Thus the choice between them rests on the appeal of the mixed population interpretation versus a single population interpretation.

17.7.10 Logistic regression with paired data (Example 17.8)

Analyzing paired outcomes when data are dichotomous can be accomplished via several routes. One of the simplest is conditional logistic regression. A very basic illustration uses case-control data where two groups of participants (cases and controls) are compared on some risk factor (coded zero for risk factor not present and one for risk factor present). What distinguishes this from an independent groups analysis is that the cases and controls are matched in pairs (e.g., for age, gender and so forth).

The first step is to load in the case-control pairs into a data frame with two columns (one for cases and one for controls).

```
cc.dat <- read.csv('case_control.csv')
```

There also needs to be a vector of 30 constants and a separate vector of 30 risk factor differences (one difference for each pair):

```
const <- rep(1,30)
diff <- cc.dat$case - cc.dat$control
```

To fit the conditional logistic regression, just run a logistic regression with no intercept with the constant as outcome and the differences as a predictor:

```
cc.fit <- glm(const ~ 0 + diff, family=binomial)
```

The log odds, odds ratio and probability of having the risk factor for the cases relative to the controls are:

```
cc.fit$coefficients
exp(cc.fit$coefficients)
exp(cc.fit$coefficients)/(1 + exp(cc.fit$coefficients))
```

The CI for the odds ratio is:

```
exp(confint(cc.fit))
```

To fit the same model using conditional logistic regression commands requires a different data structure, with separate variables for an identifier (the case number for case-control pairs or the participant number for repeated measures), case and predictor such as the risk factor in the example above. This format differs because it uses the repeated measures long form rather than the broad form. The file `cond_lg.csv` has data structured in this way.

```
clg.dat <- read.csv('cond_lg.csv')
```

To run the conditional logistic regression, use case as outcome, the identifier to define the repeated measures 'strata' and add the predictor in the usual way. The `clogit()` function in the `survival` package will fit this model:

```
library(survival)
clg.fit <- clogit(case ~ prf + strata(id), data = clg.dat)
```

However, this method doesn't provide profile likelihood CIs (and the Wald CI it does report is quite a bit wider here).

```
summary(clg.fit)
```

17.7.11 R packages

Bolker, B. M. (2009) *emdbook*: Ecological Models and Data (Book Support). R package version 1.2.

Bolker, B. M., and R Development Core Team (2011) *bbmle*: Tools for General Maximum Likelihood Estimation. R package version 0.9.7

Harrell, F. E. Jr. (2011) *rms*: Regression Modeling Strategies. R package version 3.3–1.

R-core members, DebRoy, S., Bivand, R., *et al.* (2011) *foreign*: Read Data Stored by Minitab, S, SAS, SPSS, Stata, Systat, dBase. R package version 0.8–42.
Therneau T., and Lumley, T. (2009) *survival*: Survival Analysis, including Penalised Likelihood. R package version 2.35–7.
Venables, W. N., and Ripley, B. D. (2002) *MASS*: Modern Applied Statistics with S. (4th edn) Springer: New York.
Yee, T. W. (2009) *VGAM*: Vector Generalized Linear and Additive Models. R package version 0.7–9.
Zeileis, A., Kleiber, C., and Jackman, S. (2008) Regression Models for Count Data in R. *Journal of Statistical Software*, 27(8).

17.8 Notes on SPSS syntax for Chapter 17

17.8.1 Generalized linear models

SPSS has both specialized commands (e.g., for binary logistic regression, ordinal logistic regression and loglinear models) and a very powerful generalized linear model command. For a basic logistic regression the `LOGISTIC REGRESSION` command can be used. This example uses the dream data, but input requires a dichotomous outcome for each participant (rather than as a contingency table):

SPSS data file: `dream.sav`

```
LOGISTIC REGRESSION VARIABLES scary
  /METHOD=ENTER group
  /SAVE=PRED
  /PRINT=CI(95).
```

The odds ratios are reported as `EXP(B)` where `B` is the coefficient on the log odds (logit) scale. Here the `/PRINT` subcommand requests a CI for the odds ratio and `/SAVE` requests predicted probabilities saved to the spreadsheet. Additional predictors can be entered by listing them after `group`.

As a rule these SPSS commands are quite easy to run via menus, but less flexible. In contrast, the generalized linear model command `GENLIN` is incredibly powerful, but can be rather fiddly to run using menus. For most models you can reply on the SPSS defaults to set up the model correctly with some basic syntax. Here is the same logistic regression via `GENLIN`.

```
GENLIN scary BY group
  /MODEL group DISTRIBUTION=BINOMIAL LINK=LOGIT.
```

The point of using `GENLIN` is that it can do a lot more than the simpler command and (even for a basic model such as this) automatically provides AIC, BIC and, most useful of all, AIC_C (as well as other information criteria and fit indices). It also provides profile likelihood CIs if requested:

```
GENLIN scary BY group
  /MODEL group DISTRIBUTION=BINOMIAL LINK=LOGIT
/CRITERIA CILEVEL=95 CITYPE=PROFILE(.0001).
```

The PROFILE(.0001) argument sets the required accuracy of the CI (which is obtained by iterative fitting). It is also trivial to change the reference category for the outcome:

```
GENLIN scary (REFERENCE=LAST) BY group
  /MODEL group DISTRIBUTION=BINOMIAL LINK=LOGIT
/CRITERIA CILEVEL=95 CITYPE=PROFILE(.0001).
```

The syntax is similar for all GENLIN models. Thus, for a Poisson regression the syntax takes the form:

```
GENLIN
outcome BY factor1 WITH covariate1
/MODEL factor1 covariate1 INTERCEPT=YES
DISTRIBUTION=POISSON LINK=LOG.
```

Diagnostics, predictions and so forth can be obtained from /SAVE and profile intervals from a /CRITERIA subcommand as above. SPSS command syntax reference (from the <Help> menu) will list all the options for different subcommands. The following example fits a main effects Poisson regression model for the surgical checklist data for counts arranged in a contingency table. The default is to include intercept and use a canonical link function so this syntax should work fine. The number of operations is first rescaled (as in Example 17.4) using the COMPUTE command:

SPSS data file: mortality.sav

```
COMPUTE op_k=operations/1000.
EXECUTE.

GENLIN mortality BY income time WITH op_k
  /MODEL income time op_k DISTRIBUTION=POISSON
  /CRITERIA CILEVEL=95 CITYPE=PROFILE(.0001).
```

In practice the number of operations should be an offset, in which case the following syntax can be used:

```
COMPUTE log_op_k = LN(op_k).
EXECUTE.

GENLIN mortality BY income time
  /MODEL income time DISTRIBUTION=POISSON OFFSET=log_op_k
  /CRITERIA CILEVEL=95 CITYPE=PROFILE(.0001).
```

It is also possible to change the distribution to negative binomial by altering the DISTRIBUTION statement. However, the dispersion is fixed equal to one by default. To fit estimate the dispersion add DISTRIBUTION=NEGBIN(MLE) to the MODEL subcommand:

```
GENLIN mortality BY income time
  /MODEL income time DISTRIBUTION=NEGBIN(MLE) OFFSET=log_op_k
  /CRITERIA CILEVEL=95 CITYPE=PROFILE(.0001).
```

This produces output similar to (but not quite the same as) that from `glm.nb()` in R.

Ordered logistic regression uses similar syntax within the `PLUM` command:

```
PLUM outcome BY factor1 WITH covariate1
  /CRITERIA=CIN(95)
  /LINK=LOGIT
  /PRINT TPARALLEL.
```

The `TPARALLEL` statement requests a test of parallel lines (in effect a significance test of the proportional odds assumption – though as already noted this NHST tends to reject the assumption too often in larger samples). The GENLIN command does a similar analysis but supports profile likelihood CIs:

```
GENLIN outcome (ORDER=ASCENDING) BY factor1 WITH covariate1
  /MODEL factor1 covariate1
DISTRIBUTION=MULTINOMIAL LINK=CUMLOGIT
  /CRITERIA CILEVEL=95 CITYPE=PROFILE(.0001).
```

17.8.2 Conditional logistic regression

SPSS won't accept a constant as an outcome for logistic regression, so regular logistic regression commands won't work for the case-control data in Example 17.8. However, the survival model approach will work. As with the example using R, the data needs to be in long form (as in the file `cond_lg.sav`). First, use `COMPUTE` to get a *time* variable that codes time as one when case = 1 and two when case = 0 (i.e., for controls), then run the survival using the following syntax:

SPSS data file: `cond_lg.sav`

```
COMPUTE time=1+(case=0).
EXECUTE.

COXREG time WITH prf
  /STATUS=case(1)
  /STRATA=id
  /PRINT=CI(95).
```

Other predictors can be added as covariates or factors in the normal way using `WITH` or `BY`. Note that the measures don't really need to be separated in time (the correlation between case and control or other matched observations just needs to be treated as if it is a repeated measure).

17.9 Bibliography and further reading

Agresti, A. (1996) *An Introduction to Categorical Data Analysis*. New York: Wiley.

Atkins, D. C., and Gallop, R. J. (2007) Rethinking How Family Researchers Model Infrequent Outcomes: A Tutorial on Count Regression and Zero-inflated Models. *Journal of Family Psychology*, 21, 726–35.

Hilbe, J. M. (2007) *Negative Binomial Regression*. Cambridge: Cambridge University Press.

17.10 Online supplement 4: Pseudo-R^2 and related measures

For pseudo-R^2 statistics as well as alternatives such as predictive power (Zheng & Agresti, 2000) and percentage correct classification, go to online supplement 4 at www.palgrave.com/psychology/baguley.

17.11 Online supplement 5: Loglinear models

For a detailed introduction to loglinear models go to online supplement 5 at www.palgrave.com/psychology/baguley. These models are closely related to both logistic regression and Poisson regression and are widely employed for analysis of contingency table data. Examples of how to fit loglinear models in R and SPSS are provided.