Chapter 9
GLM and GAM for Count Data

9.1 Introduction

A generalised linear model (GLM) or a generalised additive model (GAM) consists of three steps: (i) the distribution of the response variable, (ii) the specification of the systematic component in terms of explanatory variables, and (iii) the link between the mean of the response variable and the systematic part. In Chapter 8, we discussed several different distributions for the response variable: Normal, Poisson, negative binomial, geometric, gamma, Bernoulli, and binomial distributions. One of these distributions can be used for the first step mentioned above. In fact, later in Chapter 11, we see how you can also use a mixture of two distributions for the response variable; but in this chapter, we only work with one distribution at a time.

We spent a lot of time looking at distributions in Chapter 8 because our experience teaching environmental scientists show that in general they are less familiar with some of these distributions, especially the negative binomial. Before reading this chapter, you should ensure that you are familiar with the material described in Chapter 8.

In this chapter, we focus on count data and use the Poisson and negative binomial distributions. In the next chapter we concentrate on logistic regression using the binomial distribution. We also revisit count data in Chapter 11, where we look at data sets with lots of zeros or no zeros. Models for these types of data use a mixture of techniques discussed in this and the next chapter.

Good references on GLM include McCullagh and Nelder (1998), Dobson (2002), and Agresti (2002). It is possible to dedicate an entire book to Poisson or logistic regression (see for examples: Hosmer and Lemeshow, 2000; Collet, 2003). Fox (2002), Ruppert et al. (2003), Wood (2006), and Keele (2008) are excellent GAM references.

We start this chapter showing that the linear regression model is also a GLM. This is merely a pedagogical choice as it allows us to start with something familiar, and after all, the Gaussian linear regression can also be used for count data, even though it is not the best option. In Section 9.3, Poisson GLM is introduced using an artificial data set that we know the regression parameters for. It allows us to demonstrate what

A.F. Zuur et al., *Mixed Effects Models and Extensions in Ecology with R*,
Statistics for Biology and Health, DOI 10.1007/978-0-387-87458-6_9,

the model is actually doing. In Section 9.4, we give the likelihood criterion and show how parameters can be estimated. In Sections 9.5, 9.6, 9.7, 9.8, and 9.9, we discuss Poisson GLM using a real data set and focus on overdispersion, model selection, and model validation. In Section 9.10, we present the negative binomial distribution and show how it can be used if there is overdispersion. Finally we look at GAM.

9.2 Gaussian Linear Regression as a GLM

A GLM consists of three steps:

1. An assumption on the distribution of the response variable Y_i. This also defines the mean and variance of Y_i.
2. Specification of the systematic part. This is a function of the explanatory variables.
3. The relationship between the mean value of Y_i and the systematic part. This is also called the link between the mean and the systematic part.

We discuss these three steps for the Gaussian linear regression model.

Step 1: In a Gaussian linear regression, we assume that the response variable Y_i is normally distributed with mean μ_i and variance σ^2. The index i refers to a case or observation.

Step 2: In the second step, we specify the systematic part of the model. This means that we need to select the explanatory variables. Define the predictor function $\eta(X_{i1}, \ldots, X_{iq})$ by:

$$\eta(X_{i1}, \ldots, X_{iq}) = \alpha + \beta_1 \times X_{i1} + \cdots + \beta_q \times X_{iq} \tag{9.1}$$

The systematic part is given by the predictor function $\eta(X_{i1}, \ldots, X_{iq})$.

Step 3: In the third step, we need to specify the link between the expected value of Y_i (which is μ_i) and the predictor function $\eta(X_{i1}, \ldots, X_{iq})$. We use the identity link, which means that $\mu_i = \eta(X_{i1}, \ldots, X_{iq})$.

These three steps give the following GLM:

$$\begin{aligned} &Y_i \sim N(\mu_i, \sigma^2) \\ &E(Y_i) = \mu_i \quad \text{and} \quad \text{var}(Y_i) = \sigma^2 \\ &\mu_i = \eta(X_{i1}, \cdots, X_{iq}) \end{aligned} \tag{9.2}$$

This model is also called a GLM with Gaussian distribution and identity link. Combining some of the elements in Equation (9.2) gives

$$E(Y_i) = \eta(X_{i1}, \cdots, X_{iq}) = \alpha + \beta_1 \times X_{i1} + \cdots + \beta_q \times X_{iq}$$

which is our familiar linear regression model from Chapter 2 and Appendix A. We can also write it as:

$$Y_i = \alpha + \beta_1 \times X_{i1} + \cdots + \beta_q \times X_{iq} + \varepsilon_i$$

where ε_i is normally and independently distributed with mean 0 and variance σ^2. Examples and further details of the Gaussian GLM with identity link function are given in Appendix A. In principle, you can use the Gaussian distribution to analyse count data, but the residuals often show heterogeneity. Options to solve this are a data transformation or using generalised least squares as discussed in Chapter 4.

The formulation of a generalised additive model with a Gaussian distribution is similar to the linear regression model, except that in step 2 we use smoothers in the predictor function:

$$\eta(X_{i1}, \ldots, X_{iq}) = \alpha + f_1(X_{i1}) + \cdots + f_q(X_{iq})$$

Obviously, we can also have a predictor function with smoothers and parametric or nominal variables.

9.3 Introducing Poisson GLM with an Artificial Example

In this section, we show the model formulation for a Poisson GLM, and we use an artificial example to demonstrate what the model is doing. We need the following three steps for a Poisson GLM:

1. Y_i is Poisson distributed with mean μ_i. By definition of this distribution, the variance of Y_i is also equal to μ_i.
2. The systematic part is given by $\eta(X_{i1}, \ldots, X_{iq}) = \alpha + \beta_1 \times X_{i1} + \cdots + \beta_q \times X_{iq}$.
3. There is a logarithmic link between the mean of Y_i and the predictor function $\eta(X_{i1}, \ldots, X_{iq})$. The logarithmic link (also called a log link) ensures that the fitted values are always non-negative.

As a result of these three steps, we get

$$\begin{aligned} &Y_i \sim P(\mu_i) \\ &E(Y_i) = \mu_i \quad \text{and} \quad \text{var}(Y_i) = \mu_i \\ &\log(\mu_i) = \eta(X_{i1}, \cdots, X_{iq}) \quad \text{or} \quad \mu_i = e^{\eta(X_{i1}, \cdots, X_{iq})} \end{aligned} \tag{9.3}$$

The Poisson GLM is particularly useful for count data as these tend to be heterogeneous and are always non-negative; both aspects are dealt with by the Poisson GLM.

In the remaining part of this section, we use an artificial data set to explain what a Poisson GLM model is doing. Creating artificial data is simple; choose some

arbitrary values for an intercept and slope, then choose arbitrary values for a covariate, and calculate some fitted values. We will start with the covariate X_i, which takes the values 0, 1, 2, 3, 4, 5, . . ., 100. We arbitrarily choose an intercept of 0.01 with a slope of 0.03 and calculate the fitted values μ_i using the equation:

$$\mu_i = \exp(0.01 + 0.03 \times X_i)$$

The problem is that in reality, we never measure a count of exp(0.01) or exp(0.03 + 0.01 × 1), because a count is an integer. We therefore sampled one value from a Poisson distribution with mean μ_i and the resulting value is Y_i. This process is repeated for each $i = 1, \ldots, 101$. A scatterplot of X_i and Y_i is given in Fig. 9.1. We fitted a Poisson GLM on these data (on the X_i and Y_i), which gave an estimated intercept and slope, and these allowed us to draw the fitted line in Fig. 9.1. Note the line shows an exponential relationship. The scatter of points around the line in Fig. 9.1 gives an idea of how much variation to expect from a Poisson distribution with values between 0 and 30 (the range of the vertical axis).

The same exponential line is shown in Fig. 9.2, except that the third axis now shows the probability of other realisation. At several values along the covariate, where $X = 2$, 15, 30, 50, and 75, we calculated the fitted values (the Y values in Fig. 9.2), which are the means μ_i of the Poisson distributions in Fig. 9.2. Note how the shape of the Poisson density curves change from small skewed curves to wide symmetric curves.

In this section, we pretended that we knew the intercept α and slope β, which allowed us to calculate the fitted values μ_i used to generate the count data Y_i. Obviously, in real life, the situation is the opposite way around. In real life, we measure Y_i and X_i, and do not know α and β (and therefore also μ_i). Hence, we need a mechanism that estimates the values of α and β, and this is discussed in the next section.

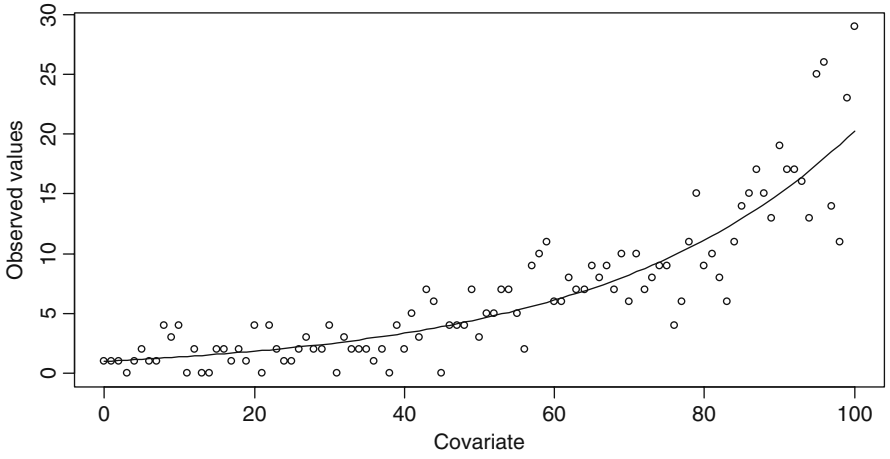

Fig. 9.1 Artificial data with a GLM Poisson model fitted. The fitted line is obtained from the GLM model, and X is the covariate with values from 0 to 100

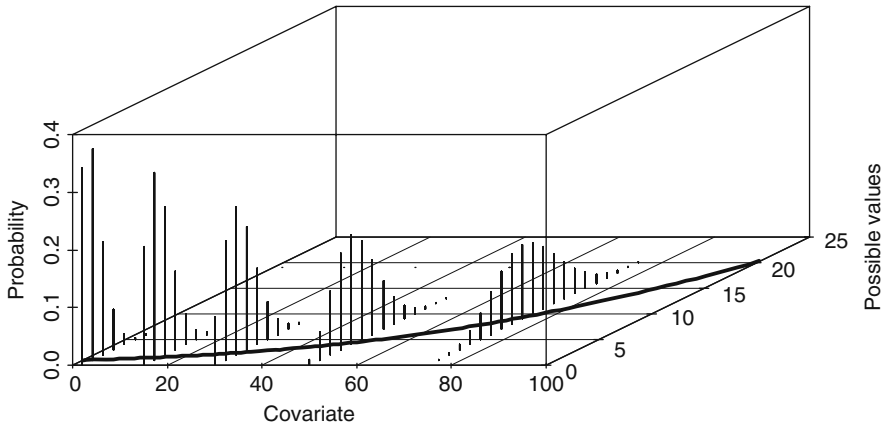

Fig. 9.2 Example of a Poisson GLM. The plane in the x–y axes shows the same exponential curve as in Fig. 9.1. The vertical lines along the third axis show Poisson probability curves at different values of the covariate: $X = 2$, 15, 30, 50, and 75. The widths of the probability curves show the spread of the data. This is the same graph as Fig. 2.5, except that we use a Poisson GLM here

9.4 Likelihood Criterion

The Poisson distribution was discussed in Chapter 8. Recall that it is given by

$$f(y_i; \mu_i) = \frac{\mu_i^{y_i} \times e^{-\mu_i}}{y_i!} \qquad y_i \geq 0, \ y_i \text{ integer}$$

It gives the probability that a particular y_i value is observed for a given mean μ_i. Within the context of a GLM, we add an index i to μ, and μ_i is a function of the covariates:

$$\mu_i = e^{\alpha + \beta_1 X_{1i} + \cdots + \beta_q X_{iq}}$$

The unknown parameters that we need to estimate are the intercept and slopes. In linear regression, we used ordinary least squares to minimise the residual sum of squares. Here, we use maximum likelihood estimation.

The principle of maximum likelihood estimation is that we specify a joint likelihood criterion L for all observed data y_1 to y_n, and we maximise this likelihood criterion as a function of the unknown regression parameters. Formulated differently, what are the values of the regression parameters such that the probability L of the observed data is the highest? The starting point is

$$L = \text{Probability}(Y_1 = y_1 \text{ and } Y_2 = y_2 \text{ and} \ldots \text{and } Y_n = y_n)$$

Because we assume independence of the observations, we can use the basic probability rule $P(A \text{ and } B) = P(A) \times P(B)$. As a result the likelihood function, L can be written as

$$L = \prod_i \frac{\mu^{y_i} \times e^{-\mu_i}}{y_i!}$$

The roman pillar symbol stands for multiplication, and the Poisson distribution function was used for the probability that Y_i is y_i. From this point onwards, it is merely a matter of mathematics; how can we maximise L as a function of the regression parameters? To simplify the maximisation process, we make the likelihood criterion L additive by working with the logarithm of the likelihood:

$$\begin{aligned}
\log(L) &= \sum\nolimits_i \left(\log(\mu^{y_i} \times e^{-\mu_i}) - \log(y_i!)\right) \\
&= \sum\nolimits_i \left(\log(\mu^{y_i}) + \log(e^{-\mu_i}) - \log(y_i!)\right) \\
&= \sum\nolimits_i (y_i \times \log(\mu_i) - \mu_i - \log(y_i!)) \\
&= \sum\nolimits_i \left(y_i \times \mathbf{X}_i \times \boldsymbol{\beta} - e^{\mathbf{X}_i \times \boldsymbol{\beta}} - \log(y_i!)\right)
\end{aligned} \tag{9.4}$$

To speed up the numerical optimisation routines, we could drop the $\log(y_i!)$ term as it does not contain any regression parameters. You may remember from high school mathematics that to optimise a function, we need to obtain first-order derivatives, set them to 0 and solve the equations. The first-order derivatives are given by

$$\frac{\partial \log(L)}{\partial \boldsymbol{\beta}} = \sum\nolimits_i (y_i \times \mathbf{X}_i - \mathbf{X}_i \times e^{\mathbf{X}_i \times \boldsymbol{\beta}}) = \sum\nolimits_i \mathbf{X}_i \times (y_i - \mu_i)$$

Setting these to 0 gives

$$\sum_i \mathbf{X}_i \times (y_i - \mu_i) = \mathbf{0} \tag{9.5}$$

For the Gaussian linear regression model with an identity link, this gives a closed form solution. This means we get nice expressions for the unknown parameters that can easily be calculated. However, for most of the other distributions and link functions, this is not the case. Instead, we get a set of equations that have to be solved iteratively. A so-called iteratively reweighted least squares (IRWLS) algorithm is applied, and the numerical output of the GLM function in R has a sentence telling you how many iterations were carried out. To obtain standard errors for the parameters, we also need second-order derivatives of the log likelihood function, but we do not present them here.

If you open a book on GLM, it will be hard to find the likelihood equations for a Poisson GLM, as most books present these equations in terms of the general notation we used in Chapter 8. The advantage of this general notation is that, provided we use a canonical link (e.g. the log for a Poisson, or identity link for the Gaussian distribution), the internal mathematics of all GLMs can be written in the same way and with the same variable names that we used in Chapter 8. This makes it easy to program. However,

from a pedagogical point of view, we decided to focus first on the Poisson GLM, and then to mention the possibility of rewriting it in abstract, and general, mathematical notation. We refer the interested reader to McCullagh and Nelder (1989).

9.5 Introducing the Poisson GLM with a Real Example

9.5.1 Introduction

In Section 9.3, we arbitrarily chose a set of regression parameters and created artificial count data. It allowed us to explain the underlying concept of Poisson GLM and give an impression of how much variation can be expected in the data if they are from a Poisson distribution. In Section 9.4, we formulated the maximum likelihood criterion and presented the first-order derivatives. Luckily, other people have written software code that uses the log likelihood criterion and the equations for first-order derivatives to obtain parameter estimates. In this section, we show how to use the software and present a detailed example. Because we are now going to use a real example, all the misery will come at the same time.

The data used here (and in various other sections in this chapter) are fully analysed in Chapter 16 as a case study. It should be noted that a Poisson GLM is not the best tool to analyse these data, but it serves as a convenient example of how to progress through all steps of a GLM for count data.

The data set consists of roadkills of amphibian species at 52 sites along a road in Portugal. A scatterplot of the response variable roadkills against a possible explanatory variable 'distance to the natural park', denoted by D.PARK, is given in Fig. 9.3. The biological interpretation of 'distance to the park' is given in Chapter 16.

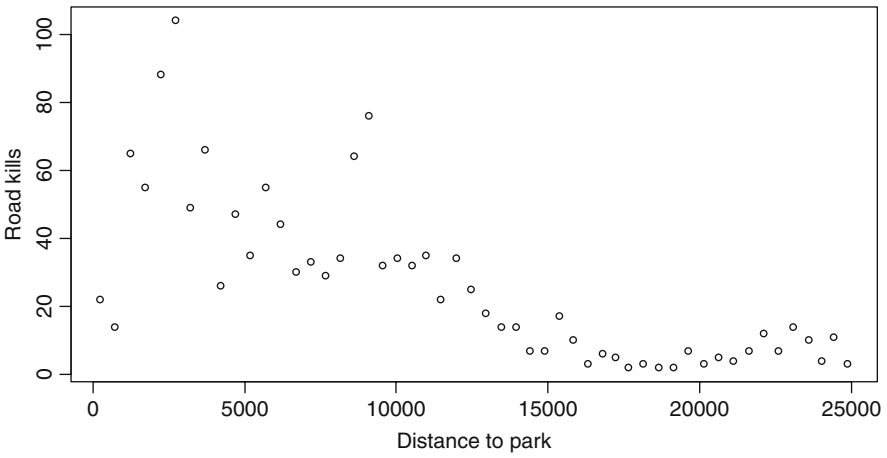

Fig. 9.3 Scatterplot of amphibian road kills versus distance (in metres) to a nearby Natural Park

The data are counts, and there seems to be a non-linear, perhaps exponential, relationship between roadkills and D.PARK. Also note that the variation is larger for larger values of roadkills. Taken together, this gives us all the ingredients for a Poisson GLM. Starting with only D.PARK as an explanatory variable, and ignoring the other 10 explanatory variables, is a pedagogical choice for presenting Poisson GLM in a textbook and is not a general recommendation for analysing these data. The following Poisson GLM was applied.

1. Y_i, the number of killed animals at site i, is Poisson distributed with mean μ_i.
2. The systematic part is given by $\eta(D.PARK_i) = \alpha + \beta \times D.PARK_i$.
3. There is a logarithm link between the mean of Y_i and the predictor function $\eta(D.PARK_i)$.

As a result of these three steps, we have

$$\begin{aligned} &Y_i \sim p(\mu_i) \\ &E(Y_i) = \mu_i \quad \text{and} \quad \text{var}(Y_i) = \mu_i \\ &\log(\mu_i) = \alpha + \beta \times D.PARK_i \quad \text{or} \quad \mu_i = e^{\alpha+\beta\times D.PARK_i} \end{aligned} \tag{9.6}$$

We now discuss how to fit this model in R.

9.5.2 R Code and Results

The following R code accesses the data, produces Fig. 9.3, applies the GLM, and presents the results.

```
> library(AED); data(RoadKills)
> RK <- RoadKills #Saves some space in the code
> plot(RK$D.PARK, RK$TOT.N, xlab = "Distance to park",
       ylab = "Road kills")
> M1 <- glm(TOT.N ~ D.PARK, family = poisson, data = RK)
> summary(M1)
```

The only new code here compared to linear regression (see Chapter 2 and Appendix A) is using the `glm` command instead of the `lm` command and the option `family = poisson`. Using `family = gaussian` applies linear regression, but we will not do that here (in fact, it is easier just to use the function `lm` for linear regression). The output of the `summary` command is slightly different from the `summary` output of an `lm` command and is given by:

```
Call:
glm(formula = TOT.N ~ D.PARK, family = poisson)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-8.1100  -1.6950  -0.4708   1.4206   7.3337

Coefficients:
              Estimate Std. Error z value  Pr(>|z|)
(Intercept)   4.316e+00  4.322e-02   99.87   <2e-16
D.PARK       -1.059e-04  4.387e-06  -24.13   <2e-16

(Dispersion parameter for poisson family taken to be 1)
    Null deviance: 1071.4  on 51  degrees of freedom
Residual deviance:  390.9  on 50  degrees of freedom

AIC: 634.29
Number of Fisher Scoring iterations: 4
```

The first two lines tell us which model has been fitted, which is handy if you save the output into a word processor document. Basic numerical information on the residuals is also provided, although in Section 9.8 we present more useful graphical tools that can be used for the model validation process. The estimated intercept and slope are 4.31 and –0.000106, respectively. Keep in mind that distance to the park is expressed in metres. To avoid parameter estimates with lots of zeros, you could (and perhaps should) express it in kilometres, as it will save some ink when presenting the estimated slope on paper. We also get a z-statistic and corresponding p-value for testing the null hypothesis that the slope (and intercept) is equal to 0 and an AIC, which can be used for model selection. The z-statistic is used because we know the variance. In a Gaussian model, the variance is estimated as well, and therefore, a t-statistic is used.

9.5.3 Deviance

The null and residual deviances are new phrases, and these are sort of maximum likelihood equivalents of the total sum of squares and the residual sum of squares, respectively. For the Poisson GLM, the residual deviance is defined as twice the difference between the log likelihood of a model that provides a perfect fit (also called the saturated model) for the model under study:

$$\text{Residual deviance} = 2\log(L(\mathbf{y};\mathbf{y})) - 2\log(L(\mathbf{y};\boldsymbol{\mu})) = 2\sum_i (y_i \log\frac{y_i}{\mu_i} - (y_i - \mu_i))$$

The notation $\mathbf{y}$ refers to a vector of all observations y_1 to y_n, and the same holds for the mean $\boldsymbol{\mu}$. The null deviance is the residual deviance in the model that only contains an intercept. Hence, the null deviance corresponds to the worst possible model (only an intercept), the residual deviance of the model under study, and the deviance of the saturated model from the best possible fit.

We do not have an R^2 in GLM models, but the closest we can get is the explained deviance, which is calculated as

$$100 \times \frac{\text{null deviance} - \text{residual deviance}}{\text{null deviance}} = 100 \times \frac{1071.4 - 390.9}{1071.4} = 63.51\%$$

So the explanatory variable distance to the park explains 63.51% of the variation in road kills. Dobson (2002) called this proportional increase in explained deviance the pseudo R^2.

The smaller the residual deviance, the better is the model. Some statistics programs also quote a p-value as it is supposedly Chi-square distributed with $n - p$ degrees of freedom, where p is the number of regression parameters in the model and n the number of observations. However, using the residual deviance as a goodness-of-fit measure is not without controversy; see McCullagh and Nelder (pg. 118–119, 1989). They argue that (at least for the binomial GLM) a large value of the residual deviance cannot always be seen as evidence of a poor fit.

The residual deviance is also sometimes called the deviance.

9.5.4 Sketching the Fitted Values

Before discussing how to assess the numerical output presented in Section 9.5.2, we will outline what the model is doing. But first we need to calculate the predicted values from the model and add these as a line in Fig. 9.3.

The function `predict` produces either predicted values on the scale of the predictor function or on the scale of the response variable. In the first case, we use the values $\eta(D.PARK_i) = 4.13 - 0.0000106 \times$ `D.PARK`$_i$, and in the second case, $\exp(\eta\ (D.PARK_i)) = \exp(4.13 - 0.0000106 \times$ `D.PARK`$_i$). If we want to show how good (or bad) the model fits the observed data, we should use the predicted values on the scale of the response variable (after taking the exponential).

Drawing the line is now simply a matter of sticking in a couple of values for `D.PARK` and calculating the fitted values. Instead of doing this manually, we can do it with a few commands in R. The code uses the `plot` command for Fig. 9.3 and the `glm` command we have already run.

```
> MyData <- data.frame(D.PARK = seq(from = 0,
                        to = 25000, by = 1000))
```

```
> G <- predict(M1, newdata = MyData, type = "link",
               se = TRUE)
> F <- exp(G$fit)
> FSEUP <- exp(G$fit + 1.96 * G$se.fit)
> FSELOW <- exp(G$fit - 1.96 * G$se.fit)
> lines(MyData$D.PARK, F, lty = 1)
> lines(MyData$D.PARK, FSEUP, lty = 2)
> lines(MyData$D.PARK, FSELOW, lty = 2)
```

You will find similar (and more extensive code) in the so-called white book on the S language (on which R is based), written by Chambers and Hastie (1992). We first create a new data frame `MyData`. The variables inside this data frame must have exactly the same names as the explanatory variables in the `glm` command; in this case there is only `D.PARK`. In the data frame, you can specify new values for the explanatory variables. The `predict` command takes as arguments the object from the `glm` function `(M1)`, the data frame with the new values of the explanatory variables, an argument `type` that tells the `predict` function at which level to predict (either the scale of the predictor function, or the response variables, and whether you want to have confidence intervals around the predicted line. We predicted at the level of the predictor function; so we get confidence bands that do not contain 0 and are asymmetric. Obviously, we have to do some basic maths ourselves, and the results are given in Fig. 9.4. Note the exponential shape of the curve and the increase in the width of the confidence bands for larger fitted values.

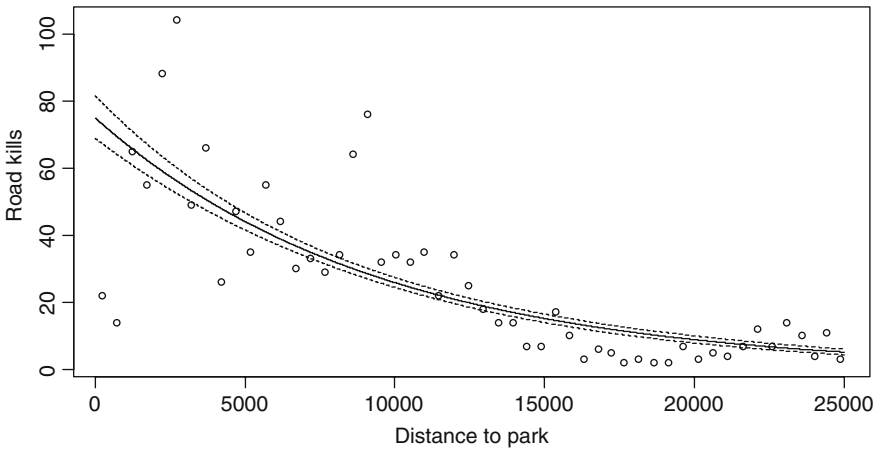

Fig. 9.4 Observed roadkills with a fitted Poisson GLM curve (*solid line*) and 95% confidence bands (*dotted lines*). Note the clear exponential shape of the curve. For smaller fitted values, there are groups of residuals above and below the fitted line. This is not good, and we need to deal with this in the model validation!

9.6 Model Selection in a GLM

9.6.1 Introduction

So far, we have only discussed the interpretation of the model in terms of an exponential curve fitted through a set of points; we now concentrate on things like model selection, hypothesis testing, and model validation. However, applying a model selection with only one explanatory variable is a bit unrealistic, so we now add a few more explanatory variables. The amphibian roadkills data set contains 17 explanatory variables. A list of these variables and abbreviations is given in Table 16.1. Some of the explanatory variables were square root transformed because of large values. Using variance inflation factors (Appendix A), a sub-selection of nine variables is made in Chapter 16 and we use the same sub-selection here. Note, this is still a relatively high number of explanatory variables for a data set with only 52 observations! A Poisson GLM for the roadkills data with nine variables is specified in a very similar way as in Equation (9.4), except that the systematic part now contains all nine explanatory variables (we have no biological reasons to believe there are interactions).

9.6.2 R Code and Output

The following R code implements the Poisson GLM with nine explanatory variables.

```
> RK$SQ.POLIC    <- sqrt(RK$POLIC)
> RK$SQ.WATRES   <- sqrt(RK$WAT.RES)
> RK$SQ.URBAN    <- sqrt(RK$URBAN)
> RK$SQ.OLIVE    <- sqrt(RK$OLIVE)
> RK$SQ.LPROAD   <- sqrt(RK$L.P.ROAD)
> RK$SQ.SHRUB    <- sqrt(RK$SHRUB)
> RK$SQ.DWATCOUR <- sqrt(RK$D.WAT.COUR)
> M2 <- glm(TOT.N ~ OPEN.L + MONT.S + SQ.POLIC +
            D.PARK + SQ.SHRUB + SQ.WATRES + L.WAT.C +
            SQ.LPROAD + SQ.DWATCOUR, family = poisson,
            data = RK)
> summary(M2)
```

The code is self-explanatory, and the relevant output of the `summary` command is given by

```
Coefficients:
              Estimate Std. Error z value  Pr(>|z|)
(Intercept)  3.749e+00  1.567e-01  23.935  < 2e-16
OPEN.L      -3.025e-03  1.580e-03  -1.915 0.055531
```

```
MONT.S        8.697e-02  1.359e-02   6.398 1.57e-10
SQ.POLIC     -1.787e-01  4.676e-02  -3.822 0.000133
SQ.SHRUB     -6.112e-01  1.176e-01  -5.197 2.02e-07
SQ.WATRES     2.243e-01  7.050e-02   3.181 0.001468
L.WAT.C       3.355e-01  4.127e-02   8.128 4.36e-16
SQ.LPROAD     4.517e-01  1.348e-01   3.351 0.000804
SQ.DWATCOUR   7.355e-03  4.879e-03   1.508 0.131629
D.PARK       -1.301e-04  5.936e-06 -21.923  < 2e-16

Dispersion parameter for poisson family taken to be 1
    Null deviance: 1071.44  on 51  degrees of freedom
Residual deviance:  270.23  on 42  degrees of freedom
AIC: 529.62
```

9.6.3 Options for Finding the Optimal Model

We want to know which explanatory variables are important, and because some terms are not significant, it is time for a model selection. The process is similar to the one used for linear regression (Appendix A). We can use either a selection criterion like the AIC or use a hypothesis testing approach.

Automatic forward, backward, and forward and backward selection can be applied with the command `step(M2)`. Results are not presented here, but a backward selection indicates that no term should be dropped.

For the hypothesis testing approach, we have three options:

1. Test the null hypothesis H_0: $\beta_i = 0$ using the z-statistic. This is the equivalent of the t-statistic in linear regression. This approach suggests to drop first `SQ.DWATCOUR` as it is the least significant term and then to refit the model and see whether there are still non-significant terms in the model.
2. Use the `drop1(M2,test = "Chi")` command, which drops one explanatory variable, in turn, and each time applies an analysis of deviance test. We explain this process below.
3. Use the `anova(M2)` command, which applies a series of analysis of deviance tests by removing each term sequential. We explain at the end of Subsection 9.6.5 how this process works.

Steps 2 and 3 are similar to the `anova` and `drop1` functions in linear regression, except that in linear regression we used an F test based on residual sum of squares of a full and a nested model. A nested model is defined as a model that is obtained from the full model by setting certain parameters equal to 0. We do not have residual sum of squares in Poisson GLM. Well, actually we do, but they are not used in these tests (residuals are discussed in Section 9.8). Instead, we use the residual deviance of two nested models.

9.6.4 The Drop1 Command

Suppose we have two models: model M_1 contains all nine explanatory variables, and in model M_2 we dropped the explanatory variable `OPEN.L`. So now the number of parameters for M_1 is $p_1 = 9$ and for M_2 is $p_2 = 8$. Obviously, the deviance of M_1 will always be equal or lower than the deviance of M_2, simply because it has one extra parameter. The null hypothesis is that the regression parameter β for `OPEN.L` equals 0. Under the null hypothesis, both deviances are equal, and therefore, a large difference between the deviances is evidence against the null hypothesis.

Let D_1 and D_2 be the deviances of models M_1 and M_2, respectively. The difference between D_2 and D_1 is asymptotically Chi-square distributed with $p_1 - p_2$ degrees of freedom. In formula

$$D_2 - D_1 \sim X^2_{p_1 - p_2} \tag{9.7}$$

The `drop1(M2,test = "Chi")` command drops each explanatory variable in turn, and each time it calculates the difference in Equation (9.7) and compares the difference to a Chi-square distribution; see the following output.

```
Single term deletions
Model: TOT.N ~ OPEN.L + MONT.S + SQ.POLIC + SQ.SHRUB +
        SQ.WATRES + L.WAT.C + SQ.LPROAD + SQ.DWATCOUR +
        D.PARK
             Df Deviance     AIC     LRT   Pr(Chi)
<none>            270.23  529.62
OPEN.L        1   273.93  531.32    3.69 0.0546474
MONT.S        1   306.89  564.28   36.66 1.410e-09
SQ.POLIC      1   285.53  542.92   15.30 9.181e-05
SQ.SHRUB      1   298.31  555.70   28.08 1.167e-07
SQ.WATRES     1   280.02  537.41    9.79 0.0017539
L.WAT.C       1   335.47  592.86   65.23 6.648e-16
SQ.LPROAD     1   281.25  538.64   11.02 0.0009009
SQ.DWATCOUR   1   272.50  529.89    2.27 0.1319862
D.PARK        1   838.09 1095.48  567.85 < 2.2e-16
```

The model containing all explanatory variables has a deviance of 270.3. If we drop `OPEN.L`, the deviance is 273.93: a difference of 3.69. The statistic $X^2 = 3.69$ follows (approximately) a Chi-square distribution with 1 degree of freedom, which gives a p-value of 0.054. This can be double checked with the R command: `1-pchisq(3.69,1)`.

Note that the analysis of deviance does not give exactly the same p-value as the z-statistic. This is because both tests are approximate. If in doubt, use the analysis of deviance test. The advantage of using the analysis of deviance test is that it also gives a p-value for a nominal variable.

9.6.5 Two Ways of Using the Anova Command

The same *p*-value for OPEN.L can be obtained by fitting a model with all explanatory variables (which is M2), a model without OPEN.L, and then use the anova command to compare the two models with an analysis of deviance. This is done with the following R code:

```
> M3 <- glm(TOT.N ~ MONT.S + SQ.POLIC + D.PARK +
            SQ.SHRUB + SQ.WATRES + L.WAT.C + SQ.LPROAD +
            SQ.DWATCOUR, family = poisson, data = RK)
> anova(M2, M3, test = "Chi")
```

The output is given by

```
Analysis of Deviance Table
  Resid. Df Resid. Dev Df Deviance  P(>|Chi|)
1        42    270.232
2        43    273.925 -1   -3.693     0.055
```

If you use this output in a paper or report, then you should write that the difference in deviance is 3.69 and approximately follows a Chi-square distribution with 1 degree of freedom. We have seen papers where a Chi-square distribution with 43 degrees of freedom was quoted from the output above, which is clearly wrong!

Be careful when using the command anova(M2); it applies an analysis of deviance test, but now the terms are removed sequentially and the order depends on the order they were typed. This is useful if all explanatory variables are independent or if the last term is an interaction.

9.6.6 Results

Using the drop1 function, we decided to remove the variable SQ.DWATCOUR. Refitting the model resulted in all explanatory variables being significant at the 5% level. This suggests that we are finished with the model selection process, and can proceed to the model validation process. However, things are never that easy. The results of the summary command presented above had a small sentence that said: 'overdispersion parameter for Poisson family taken to be 1'. This does not mean that the overdispersion really is 1; it just says it was taken as 1.We promised more misery, and overdispersion is the next stage.

In the next section, we show that all the results presented in this section can be put in the bin, because of overdispersion. If you analyse your own data, you should always first check for overdispersion, before doing any model selection or interpretation of the results. The reason why we did not start by looking at overdispersion was because we wanted to make sure you could read the output and judge whether there is overdispersion. For your own data, you should always start by checking for overdispersion and act accordingly. This is discussed in the next section.

9.7 Overdispersion

9.7.1 Introduction

Overdispersion means the variance is larger than the mean. How do you know your model is overdispersed? There are two options. The first is based on the X^2 approximation of the residual deviance. If there is overdispersion, then D/ϕ is Chi-square distributed with $n - p$ degrees of freedom, and this leads to the following estimator for ϕ:

$$\hat{\phi} = \frac{D}{n - p} \tag{9.8}$$

In this case, it is $270.23/42 = 6.43$. If this ratio is about 1, then you can safely assume there is no overdispersion and proceed to the model validation process. In this case the ratio is larger than 1 and provides evidence for overdispersion. Note this only identifies overdispersion. The model (and software) does not take into account of the overdispersion and we therefore cannot present the results as they are. Also note that the use of the estimator in Equation (9.8) is not without criticism.

The second option is to use a different estimator based on the so-called Pearson residuals and let the software make the corrections required for overdispersion (i.e. correct the standard errors and tell us the magnitude of the overdispersion based on the estimator using the Pearson residuals). But we have not yet discussed residuals for Poisson GLMs yet. This will be done in Section 9.8.

9.7.2 Causes and Solutions for Overdispersion

Hilbe (2007) discriminates between apparent and real overdispersion. Apparent overdispersion is due to missing covariates or interactions, outliers in the response variable, non-linear effects of covariates entered as linear terms in the systematic part of the model, and choice of the wrong link function. These are mainly model misspecifications. There are a couple of interesting examples in Hilbe (pg. 52–61, 2007). For example, he simulates a Poisson variable using five explanatory variables X_1 to X_5, applies a Poisson model using only explanatory variables X_2 to X_4, and shows how this causes overdispersion. Similar examples are given for the effects of outliers and using the wrong link function.

Real overdispersion exists when we cannot identify any of the previous mentioned causes. This can be because the variation in the data really is larger than the mean. Or there may be many zeros (which may, or may not, cause overdispersion), clustering of observations, or correlation between observations.

If adding covariates and interactions does not help, there is a quick-fix that can be tried before considering more complicated methods like the negative binomial GLM.

9.7.3 Quick Fix: Dealing with Overdispersion in a Poisson GLM

We can deal with overdispersion in the GLM by using a quasi-Poisson GLM, which consists of the following steps:

1. The mean and variance of Y_i are given by $\mathrm{E}(Y_i) = \mu_i$ and $\mathrm{var}(Y_i) = \phi \times \mu_i$.
2. The systematic part is given by $\eta(X_{i1}, \ldots, X_{iq}) = \alpha + \beta_1 \times X_{i1} + \ldots + \beta_q \times X_{iq}$.
3. There is a logarithmic link between the mean of Y_i and the predictor function $\eta(X_{i1}, \ldots, X_{iq})$.

The difference between the Poisson GLM and the Poisson GLM with overdispersion is that we no longer explicitly specify a Poisson distribution, but only a relationship between the mean and variance of Y_i.

Although we do not specify a Poisson distribution, we still use the same type of model structure in terms of the link function and predictor function. If the dispersion parameter $\phi = 1$, we get the same results (in terms of estimated parameters and standard errors) as the Poisson GLM.

If $\phi > 1$, we talk about overdispersion, and if $\phi < 1$, we have underdispersion. The latter means that the variance of the response variable is smaller than you would expect from a Poisson distribution. Reasons for underdispersion are the model is fitting a couple of outliers rather too well or there are too many explanatory variables or interactions in the model (overfitting). If this is not the case, then the consensus is not to correct for underdispersion. Models that take underdispersion into account are discussed in Chapter 7 of Hilbe (2007).

If $\phi > 1$, we need to correct for the overdispersion, which basically means refitting the model, estimating the parameter ϕ, and 'making some corrections'. Before addressing these corrections, we look at the following questions first:

1. How do we estimate the dispersion parameter ϕ?
2. How much larger than 1 should it be before we need to make a correction?
3. What is the effect of introducing a dispersion parameter ϕ?
4. At which point do we decide to do take an alternative approach?

The first question can only be answered in detail towards the end of Section 9.8 because the estimation of ϕ is based on residuals and we have not yet defined residuals for a GLM. The second question can only be answered in light of the third question. The price we pay for introducing a dispersion parameter ϕ, is that the standard errors of the parameters are multiplied with the square root of ϕ. For example, if ϕ is equal to 9, then all standard errors are multiplied by 3, and the parameters become less significant. If the parameters of a Poisson GLM are highly significant, then a small correction of the standard errors due to overdispersion, say $\phi = 1.5$, is not going to make any differences in the biological conclusions. But if you have a parameter with a p-value of 0.03, then multiplying the standard error with the square root of 1.5 may change the p-value in something that is no longer significant at the 5% level. So, it all depends: In general a ϕ larger than 1.5 means

that some action needs to be taken to correct it. Various tests for overdispersion are discussed in Hilbe (2007). For the fourth question, if ϕ is larger than 15 or 20, then you also need to consider other methods (e.g. the negative binomial GLM or zero-inflated models), see the negative binomial model in Section 9.10 and the models for zero-inflated data in Chapter 11.

9.7.4 R Code and Numerical Output

In R, the following command is required for this quick fix approach to correct for overdispersion.

```
> M4 <- glm(TOT.N ~ OPEN.L + MONT.S + SQ.POLIC +
            SQ.SHRUB + SQ.WATRES + L.WAT.C + SQ.LPROAD +
            SQ.DWATCOUR + D.PARK,
            family = quasipoisson, data = RK)
```

You can see the only difference is specifying the `family` option as `quasipoisson` instead of `poisson`. This gives the impression that there is a quasi-Poisson distribution, but there is no such thing! All we do here is specify the mean and variance relationship and an exponential link between the expected values and explanatory variables. It is a software issue to call this 'quasipoisson'. Do not write in your report or paper that you used a quasi-Poisson distribution. Just say that you did a Poisson GLM, detected overdispersion, and corrected the standard errors using a quasi-GLM model where the variance is given by $\phi \times \mu$, where μ is the mean and ϕ the dispersion parameter. To get the numerical output for this model, use `summary(M4)`, which gives

```
Coefficients:
              Estimate Std. Error t value  Pr(>|t|)
(Intercept)  3.749e+00  3.814e-01   9.830 1.86e-12
OPEN.L      -3.025e-03  3.847e-03  -0.786  0.43604
MONT.S       8.697e-02  3.309e-02   2.628  0.01194
SQ.POLIC    -1.787e-01  1.139e-01  -1.570  0.12400
SQ.SHRUB    -6.112e-01  2.863e-01  -2.135  0.03867
SQ.WATRES    2.243e-01  1.717e-01   1.306  0.19851
L.WAT.C      3.355e-01  1.005e-01   3.338  0.00177
SQ.LPROAD    4.517e-01  3.282e-01   1.376  0.17597
SQ.DWATCOUR  7.355e-03  1.188e-02   0.619  0.53910
D.PARK      -1.301e-04  1.445e-05  -9.004 2.33e-11

Dispersion parameter for quasipoisson family taken to
be 5.928003
    Null deviance: 1071.44  on 51  degrees of freedom
Residual deviance:  270.23  on 42  degrees of freedom
AIC: NA
```

Note that the ratio of the residual deviance and the degrees of freedom is still larger than 1, but that is no longer a problem as we now allow for overdispersion. The dispersion parameter ϕ is estimated as 5.93. This means that all standard errors have been multiplied by 2.43 (the square root of 5.93), and as a result, most parameters are no longer significant! We can move onto model selection.

9.7.5 Model Selection in Quasi-Poisson

The model selection process in quasi-Poisson GLMs is similar to Poisson GLMs; however, there are small, but important differences. First of all, in quasi-Poisson models the AIC is not defined. Hence, there is no automatic backward or forward selection with the `step` function! The hypothesis testing approach is also slightly different. The analysis of deviance approach to compare two nested models M_1 (full model) and M_2 (nested model) uses a different test statistic:

$$\frac{D_2 - D_1}{\phi(p_1 - p_2)} \sim F_{p_1-p_2, n-p_1} \tag{9.9}$$

where ϕ is the overdispersion parameter, and $p_1 + 1$ and $p_2 + 1$ are the number of regression parameters in models M_1 and M_2, respectively. The '+1' is for the intercept. Under the null-hypothesis, the regression parameters of the omitted explanatory variables are equal to zero, and the F-ratio follows an F-distribution with $p_1 - p_2$ and $n - p_1$ degrees of freedom (n is the number of observations).

Using the command `drop1(M4,test = "F")` gives us the equivalent of the `drop1` function for the Poisson GLM; one term is dropped in turn. The output is as follows.

```
Single term deletions
Model: TOT.N ~ OPEN.L + MONT.S + SQ.POLIC + SQ.SHRUB +
 SQ.WATRES + L.WAT.C + SQ.LPROAD + SQ.DWATCOUR + D.PARK

            Df Deviance F value    Pr(F)
<none>           270.23
OPEN.L       1   273.93  0.5739 0.452926
MONT.S       1   306.89  5.6970 0.021574
SQ.POLIC     1   285.53  2.3776 0.130585
SQ.SHRUB     1   298.31  4.3635 0.042814
SQ.WATRES    1   280.02  1.5217 0.224221
L.WAT.C      1   335.47 10.1389 0.002735
SQ.LPROAD    1   281.25  1.7129 0.197727
SQ.DWATCOUR  1   272.50  0.3526 0.555802
D.PARK       1   838.09 88.2569    7e-12
```

These results suggest dropping `SQ.DWATCOUR` from the model and then refitting the model with the remaining terms to see if there are still any non-significant

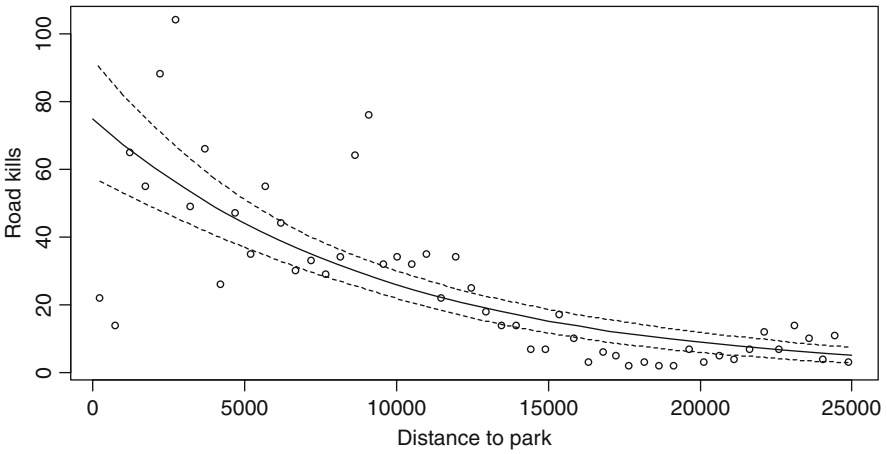

Fig. 9.5 Fitted line of the optimal quasi-Poisson model using only `D.PARK` as the explanatory variables. R code to make this graph is given on the book's website

terms. After doing this, some terms are still non-significant so the process has to be repeated. The variables were dropped in the following order: `OPEN.L`, `SQ.WATRES`, `SQ.LPROAD`, `SQ.SHRUB`, `SQ.POLIC`, `MONT.S`, and `L.WAT.C`. Finally, we ended up with a model that only contained `D.PARK`. So, ignoring overdispersion can result in a completely different biological conclusion!

We finally present the numerical output of the quasi-Poisson model that uses only `D.PARK`. Its estimated parameters, standard errors, etc. are given below and the fitted line is presented in Fig. 9.5. Note that the confidence intervals around the line are now larger than before due to the overdispersion correction.

```
              Estimate Std. Error t value  Pr(>|t|)
(Intercept)   4.316e+00  1.194e-01  36.156  < 2e-16
D.PARK       -1.058e-04  1.212e-05  -8.735 1.24e-11

Dispersion parameter for quasipoisson family taken to
be 7.630148
    Null deviance: 1071.4  on 51  degrees of freedom
Residual deviance:  390.9  on 50  degrees of freedom
```

9.8 Model Validation in a Poisson GLM

Just as in linear regression, we have to apply a model validation after we have decided on the optimal GLM, and the residuals are an important tool for this. Earlier in linear regression and additive modelling, these were defined as

$$\text{Linear regression :} \quad \hat{\varepsilon}_i = y_i - \hat{\mu}_i = y_i - \hat{\alpha} - \hat{\beta}_1 \times X_{i1} - \cdots - \hat{\beta}_q \times X_{iq}$$

$$\text{Additive modelling :} \quad \hat{\varepsilon}_i = y_i - \hat{\mu}_i = y_i - \hat{\alpha} - \hat{f}_1(X_{i1}) - \cdots - \hat{f}_q(X_{iq})$$

We used the notation ˆ to indicate that we are working with estimated values, parameters, or smoothing functions. To save space, we focus on the GLM, but the approach is identical for the GAM.

The question is as follows: What are residuals in a GLM? An obvious starting point would be to define residuals in exactly the same way as we do for linear regression using $y_i - \mu_i$, which is the vertical distance between an observation and the solid line in Fig. 9.5. The next question is whether a large residual at `D.PARK` = 1000 m is any worse than a large residual at `D.PARK` = 20000 m? The answer is not as easy as it may look, and we discuss this next!

9.8.1 Pearson Residuals

As for larger fitted values (left part of the fitted line) with Poisson distributions, we can allow for more variation around the line than with other distributions. Therefore, while we still want to see small residuals $y_i - \mu_i$ for small values of μ_i, residuals are allowed to be larger for larger μ_i. That makes a plot of the residuals $y_i - \mu_i$ versus fitted values μ_i, one of our prime graphs in Chapters 2 and 4, not particularly useful here.

In Chapter 4, we had a similar problem and our solution was to divide the residuals $y_i - \mu_i$ by the square root of the variance of Y_i, also called the normalised residuals. Here, we can do the same and call them the Pearson residuals.

$$\hat{\varepsilon}_i^P = \frac{y_i - \hat{\mu}_i}{\sqrt{\text{var}(Y_i)}} = \frac{y_i - \hat{\mu}_i}{\sqrt{\hat{\mu}_i}} \tag{9.10}$$

For this, each residual is divided by the square root of the variance. The name 'Pearson' (for a Poisson GLM) is because squaring and summing all the Pearson residuals gives you the familiar Pearson Chi-square goodness of fit criteria.

When we use an overdispersion parameter ϕ, the variance is adjusted with this parameter, and we divide the residuals $y_i - \mu_i$ by the square root of $\phi\mu_{\text{i}}$.

It is also possible to define standardised Pearson residuals by dividing the Pearson residuals by the square root of 1 – h_i, where h_i is the leverage of observation i; see also Appendix A.

9.8.2 Deviance Residuals

Recall that the residual deviance is the GLM equivalent of the residual sum of squares; the smaller the better. It would be nice to know the contribution of each observation (case) to the residual deviance. Perhaps some observations are not fitted well by the model, and this can be detected by looking at the deviance residuals. They are defined by

$$\hat{\varepsilon}_i^D = \text{sign}(y_i - \mu_i)\sqrt{d_i} \tag{9.11}$$

The notation 'sign' stands for sign and has the value 1 if y_i is larger than μ_i, and -1 if y_i is smaller than μ_i. The quantity d_i is the contribution of the ith observation to the deviance. The d_i was formulated in Section 9.5.3. The sum of squares of the deviance residuals d_i equals the residual deviance D.

9.8.3 Which One to Use?

So, we have three types of residuals in a GLM: (i) the ordinary residuals $y_i - \mu_i$, also called the response residuals, (ii) the Pearson residuals, and (iii) deviance residuals. In fact, there are more types of residuals (e.g. working residuals and Anscombe residuals, see McCullagh and Nelder (1989)), but these are the most popular ones for the purpose of model validation. Which one should we use?

By default, R uses the deviance residuals, and for most data sets used in this book, there is not much difference between using Pearson or deviance residuals for a Poisson GLM. This may not, however, be the case for data sets with lots of zeros (small variance) or for Binomial GLMs. McCullagh and Nelder (p. 398, 1989) recommend using the deviance residuals for model checking as these have distributional properties that are closer to the residuals from a Gaussian linear regression model than the alternatives; use Pierce and Schafer (1986) for a justification.

However, it should be noted that we are not looking for normality from the Pearson or deviance residuals. It is all about lack of fit and looking for patterns in the deviance or Pearson residuals.

9.8.4 What to Plot?

We need to take the residuals of choice (e.g. deviance) and plot them against (i) the fitted values, (ii) each explanatory variable in the model, (iii) each explanatory variable not in the model (the ones not used in the model, or the ones dropped during the model selection procedure), (iv) against time, and (v) against spatial coordinates, if relevant. We do not want to see any patterns in these graphs. If we do, then there is something wrong, and we need to work out what it is.

If there are patterns in the graph with residuals against omitted explanatory variables, then the solution is simple; include them in the model. If there are patterns in the graph showing residuals against each explanatory variable used in the model, then either include quadratic terms, use GAM, or conclude that there is violation of independence. If you plot the residuals against time or spatial coordinates, and there are patterns, conclude you are violating the assumption of independence. Patterns in spread (detected by plotting residuals against fitted values) may indicate overdispersion or use of the wrong mean-variance relationship (e.g. wrong choice of distribution).

Violation of independence nearly always means that an important covariate was excluded from the model. If you did not measure it, then if possible, go back into the field and measure it now. That is assuming you have any idea of what the

missing covariate might be! If this is not an available solution, then curse yourself for a poor experimental design and hope that applying a generalised linear mixed model or generalised estimation equation (GEE) will bale you out. See Chapters 12 and 13.

9.9 Illustration of Model Validation in Quasi-Poisson GLM

To explain model validation, we use the optimal quasi-Poisson GLM for the amphibian roadkills data. Recall from Section 9.7.5 that there was an overdispersion of 7.63 and that the only significant explanatory variable was `D.PARK`. Figure 9.6 shows the standard output from a `plot` command, and Fig. 9.7 contains the response residuals, Pearson residuals, scaled Pearson residuals (we divided the Pearson residuals by the square root of the overdispersion parameter), and the deviance residuals. Both figures indicate that there is a clear pattern in the residuals. Note that it is hard to detect any differences between Pearson and deviance residuals. Some additional exploration into the residuals against other explanatory variables and spatial locations is done in Chapter 16.

As in linear regression, we can also use leverage and the Cook distance statistic. There are no influential observations.

The following R code was used to produce Figs. 9.6 and 9.7.

```
> M5 <- glm(TOT.N ~ D.PARK, family = quasipoisson, data = RK)
> EP <- resid(M5, type = "pearson")
> ED <- resid(M5, type = "deviance")
> mu <- predict(M5, type = "response")
> E <- RK$TOT.N - mu
> EP2 <- E / sqrt(7.630148 * mu)
> op <- par(mfrow = c(2, 2))
> plot(x = mu, y = E, main = "Response residuals")
> plot(x = mu, y = EP, main = "Pearson residuals")
> plot(x = mu, y = EP2,
       main = "Pearson residuals scaled")
> plot(x = mu, y = ED, main = "Deviance residuals")
> par(op)
```

The first line re-applies the quasi-Poisson model, even though we could have omitted it as we had already applied it in the previous subsection. `EP` and `ED` are the Pearson and deviance residuals, respectively. Unfortunately, the function `resid` ignores the overdispersion; so we need to manually divide the Pearson residuals by the square root of 7.63 or calculate these residuals from scratch (as we did here). The rest of the code plots the residuals and should be self explanatory.

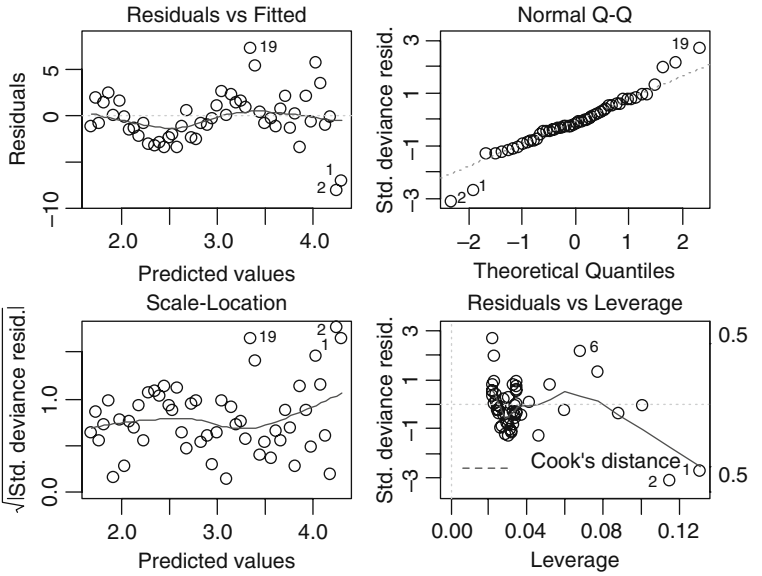

Fig. 9.6 Standard output from a GLM function applied on the amphibian roadkills data obtained by the `plot` command

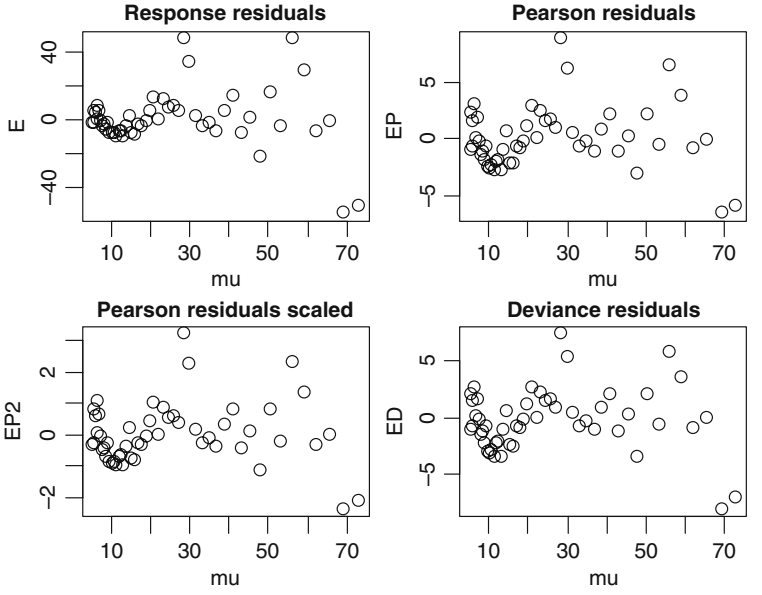

Fig. 9.7 Response residuals (observed minus fitted values, also called ordinary residuals), Pearson residuals, scaled Pearson residuals (the overdispersion is taken into account) and the deviance residuals for the optimal quasi-Poisson model applied on the amphibian roadkills data

The last thing we explain is how the overdispersion parameter ϕ in a Poisson GLM is estimated by R. It takes the Pearson residuals, squares them, adds them all up, and then divides the sum by $n - p$, where n is the number of observations and p the number of regression parameters (slopes) in the model. Check it with the R command `sum(EP^2)/(52-1)`.

9.10 Negative Binomial GLM

9.10.1 Introduction

In the previous sections of this chapter, we applied Poisson GLM on the amphibian roadkills data set and found that there is an overdispersion of 7.63. Consequently, all standard errors were corrected by multiplying them with the square root of 7.63 when we applied the quasi-Poisson model. An alternative approach is to apply the negative binomial model. In Chapter 16, a negative binomial GAM is applied on the amphibian roadkills data, but for illustration purposes we apply the negative binomial GLM here.

Books that contain a chapter on the negative binomial GLM are for example Venables and Ripley (2002), Agresti (2002), or Gelman and Hill (2007). A book dedicated to negative binomial regression is Hilbe (2007). If you are going to apply the negative binomial GLM, then this book is a 'must read'. It even discusses negative binomial GLMM models. Stata, rather than R, is used for this book, but this does not dominate the text.

Just as for Gaussian and Poisson GLMs, we specify the model with three steps. The NB GLM is given by

1. Y_i is negative binomial distributed with mean μ_i and parameter k (see also Chapter 8). By definition, the variance of Y_i is also equal to μ_i and its variance is $\mu_i + {\mu_i}^2 / k$.
2. The systematic part is given by $\eta(X_{i1}, \ldots, X_{iq}) = \alpha + \beta_1 \times X_{i1} + \ldots + \beta_q \times X_{iq}$.
3. There is a logarithm link between the mean of Y_i and the predictor function $\eta(X_{i1},\ldots, X_{iq})$. The logarithmic link (also called log link) ensures that the fitted values are always non-negative.

As a result of these three steps, we have

$$\begin{aligned} & Y_i \sim NB(\mu_i, k) \\ & E(Y_i) = \mu_i \quad \text{and} \quad \text{var}(Y_i) = \mu_i + \frac{\mu_i^2}{k} \\ & \log(\mu_i) = \eta(X_{i1}, \cdots, X_{iq}) \quad \text{or} \quad \mu_i = e^{\eta(X_{i1}, \cdots, X_{iq})} \end{aligned} \tag{9.12}$$

To estimate the regression parameters, we need to specify the likelihood criterion, and obtain the first-order and second-order derivatives. The process is the same as

for the Poisson GLM in Section 9.4. To avoid repetition, we only show how the log likelihood criterion is derived.

Recall from Chapter 8 that the negative binomial probability function is given by

$$f(y_i; k, \mu_i) = \frac{\Gamma(y_i + k)}{\Gamma(k) \times \Gamma(y_i + 1)} \times \left(\frac{k}{\mu_i + k}\right)^k \times \left(1 - \frac{k}{\mu_i + k}\right)^y \tag{9.13}$$

These probability functions are then used in the log likelihood criterion:

$$\log(L) = \sum\nolimits_i \log(f(y_i; k, \mu_i)) \tag{9.14}$$

It is now a matter of substituting Equation (9.13) into the log likelihood function in (9.14), and using high school mathematics to simplify things. There is some contradiction in the literature regarding how much you should simplify this equation. For example, Equation (5.30) in Hilbe (2007) looks very different from the one we have here, but it is exactly the same thing, just written down differently. If you start inspecting these equations, do not panic if you find differences; some textbooks have small mistakes! Keeping it simple gives us

$$\begin{aligned} \log(L) &= \sum\nolimits_i \log(f(y_i; k, \mu_i)) \\ &= \sum\nolimits_i (k \times \log\left(\frac{k}{\mu_i + k}\right) + y_i \times \log\left(\frac{\mu_i}{\mu_i + k}\right) + \log(\Gamma(y_i + k)) \\ &\quad - \log(\Gamma(k)) - \log(\Gamma(y_i + 1))) \end{aligned} \tag{9.15}$$

This can be further simplified. It is also possible to express the NB probability function in Equation (9.13) as an exponential function. The advantage of this is that the whole model can be written in the same notation as the other GLMs; see also Section 13.2.2 in Hardin and Hilbe (2007).

The function `glm.nb` from the `MASS` package can be used to apply the negative binomial GLM in R. We start with all 11 explanatory variables again.

```
> library(MASS)
> M6 <- glm.nb(TOT.N ~ OPEN.L + MONT.S + SQ.POLIC +
         SQ.SHRUB + SQ.WATRES + L.WAT.C + SQ.LPROAD +
         SQ.DWATCOUR + D.PARK, link = "log", data = RK)
```

You can choose from the logarithmic, identity, and square root link function, and an example with the identity link can be found in Agresti (2002). Here, we use the logarithmic link (which is also the default link in the function `glm.nb`, but not the canonical link function); so we can compare the results with those from the Poisson GLM. The command `summary(M6, cor = FALSE)` gives the relevant numerical output.

```
              Estimate Std. Error z value Pr(>|z|)
(Intercept)  3.951e+00  4.145e-01   9.532   <2e-16
OPEN.L      -9.419e-03  3.245e-03  -2.903   0.0037
MONT.S       5.846e-02  3.481e-02   1.679   0.0931
SQ.POLIC    -4.618e-02  1.298e-01  -0.356   0.7221
SQ.SHRUB    -3.881e-01  2.883e-01  -1.346   0.1784
SQ.WATRES    1.631e-01  1.675e-01   0.974   0.3301
L.WAT.C      2.076e-01  9.636e-02   2.154   0.0312
SQ.LPROAD    5.944e-01  3.214e-01   1.850   0.0644
SQ.DWATCOUR -1.489e-05  1.139e-02  -0.001   0.9990
D.PARK      -1.235e-04  1.292e-05  -9.557   <2e-16

Dispersion parameter for Negative Binomial(5.5178)
family taken to be 1

Null deviance: 213.674 on 51 degrees of freedom
Residual deviance: 51.803 on 42 degrees of freedom
AIC: 390.11
Theta: 5.52
Std. Err.: 1.41
2 x log-likelihood: -368.107
```

The output is similar to the Poisson GLM output, except we also get a parameter theta, which is the k in the negative binomial variance function. We also get its standard error, but care is needed with its use as the interval is not symmetric and we are testing on the boundary. Note that as half of the regression parameters are not significant at the 5% level, a model selection is required.

The available tools for a model selection are similar to those we have seen in the previous section: hypothesis testing and using a model selection tool like the AIC. For hypothesis testing, we can use

1. The z-statistic (table above).
2. Analysis of deviance tables obtained by the `anova(M6, test = "Chi")` command (this is doing sequential testing).
3. Drop each term in turn and compare the full model with a nested model using the `drop1(M6, test = "Chi")` command.
4. Manually specifying a nested model, call it for example `M7`, and use the command `anova(M6, M7, test = "Chi")`.

An automatic backward (or forward) selection procedure based on the AIC can be applied by the command `step(M6)` or `stepAIC(M6)`. The latter option is the main advantage over quasi-Poisson, where we do not have a likelihood function and therefore cannot use AIC and automatic selection procedures.

A negative binomial model can also be overdispersed, and the approach described earlier of using the ratio of the residual deviance and the degrees of freedom can be used. In this case, there is a small amount of overdispersion. A quasi-negative binomial option does not exist.

Hilbe (2007) discusses a large range of extensions that can be applied (see his Table 5.1). It is even possible to model the parameter k as a function of covariates, but you may have to program your own model in R. Another exotic cousin of the negative binomial model is the NB-P model, which has as variance $\mu_i + \mu_i^p/k$. If $p = 2$, we end up with the ordinary NB GLM again. These are all useful options if there is overdispersion in the NB GLM, but appropriate R software is scarce.

9.10.2 Results

The intermediate results of the model selection (using first the AIC and then some fine tuning using hypothesis testing) is not given here, but the final model contains the explanatory variables `OPEN.L` and `D.PARK`. You could also decide to use `L.WAT.C` as well because its p-value in a model with `OPEN.L` and `D.PARK` is 0.02. We decided to drop it, because these p-values are approximate, and it is so close to the magic 5% level.

Our optimal model and its numerical and graphical output are obtained by the following R code.

```
> M8 <- glm.nb(TOT.N ~ OPEN.L + D.PARK, link = "log",
               data = RK)
> summary(M8)
> drop1(M8, test = "Chi")
> op <- par(mfrow = c(2, 2))
> plot(M8)
> par(op)
```

The output from the `drop1` function is given below. Both explanatory variables are significant at the 5% level.

```
Single term deletions
Model: TOT.N ~ OPEN.L + D.PARK

       Df Deviance    AIC    LRT   Pr(Chi)
<none>       51.84 385.43
OPEN.L  1    59.73 391.32   7.89  0.004967
D.PARK  1   154.60 486.19 102.76 < 2.2e-16
```

The `summary` command gives

```
             Estimate Std. Error z value  Pr(>|z|)
(Intercept)  4.6717034  0.1641768  28.455   <2e-16
OPEN.L      -0.0093591  0.0031952  -2.929   0.0034
D.PARK      -0.0001119  0.0000113  -9.901   <2e-16
```

```
Dispersion parameter for Negative Binomial(4.1328)
family taken to be 1

Null deviance: 170.661  on 51  degrees of freedom
Residual deviance:  51.839  on 49  degrees of freedom
AIC: 387.43
Theta:  4.133
Std. Err.:  0.980
2 x log-likelihood:  -379.432
```

Theta is the parameter k from the variance function. Note that the analysis of deviance results gives slightly different p-values compared to the z-statistics, but the biological conclusions will be similar. The graphical validation plots are presented in Fig. 9.8 and do not show any problems.

The model seems to suggest that the further away you are from the park, the fewer roadkills. Open land cover also has a negative effect of roadkill numbers.

So, which model is better, the quasi-Poisson or the negative binomial GLM? The answer is simple: the quasi-Poisson model has patterns in the residuals and the

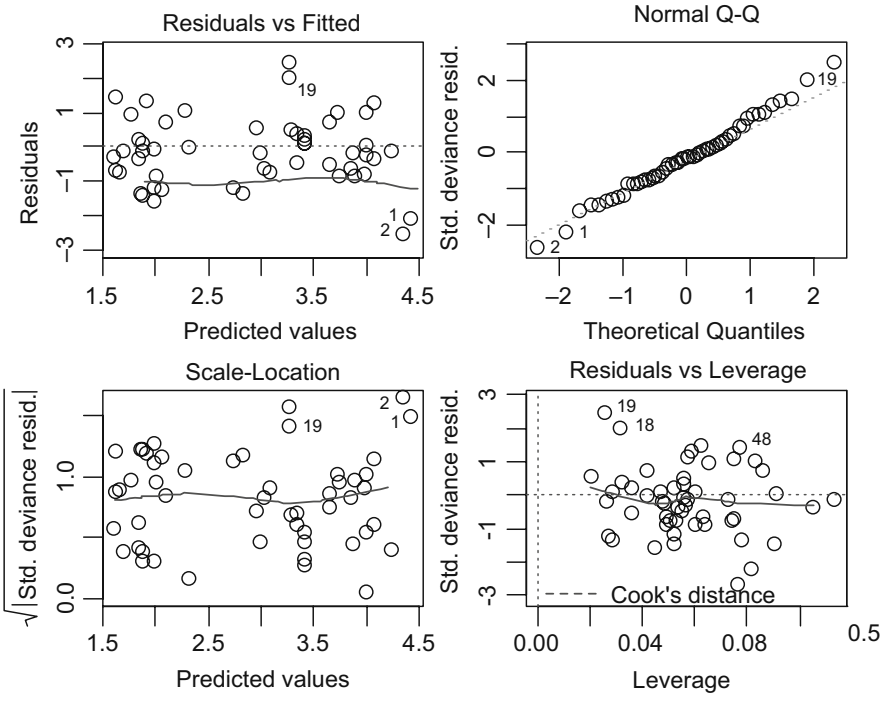

Fig. 9.8 Graphical validation tools for the negative binomial GLM. The graphs do no indicate any problems. We also plotted Pearson residuals versus the fitted values (not shown here), and this graph did not show any problems neither

negative binomial has no patterns, so this is the preferred model. Adding OPEN.L as an explanatory variable to the quasi-Poisson model does not remove the pattern. A bonus of the negative binomial GLM is that the AIC is defined, which allows us to do automatic selection procedures.

If the residual graphs do not show a clear winner, then you can also apply a test to compare the NB and Poisson GLMs; they are nested. The variance of the Poisson is: $\text{var}(Y_i) = \mu_i$, and for the NB we have $\text{var}(Y_i) = \mu_i + \mu_i^2/k$. We can also write the variance of the NB model as $\text{var}(Y_i) = \mu_i + \alpha \times \mu_i^2$. The models will give the same variance if $\alpha = 0$; so we can use a likelihood ratio test and the null hypothesis is H_0: $\alpha = 0$. However, we are testing on the boundary again (the alternative is H_1: $\alpha > 0$. We saw a similar problem when we tested the significance of a random effect in Chapter 5, and the same solution of dividing the p-value by 2 can be applied. The Poisson model with OPEN.L and D.PARK is fitted with

```
> M9 <- glm(TOT.N ~ OPEN.L + D.PARK, family = poisson,
            data = RK)
```

The log likelihood test is obtained by

```
> llhNB = logLik(M8)
> llhPoisson  = logLik(M9)
> d <- 2 * (llhNB - llhPoisson)
> pval <- 0.5 * pchisq(as.numeric(d), df = 1,
                 lower.tail = FALSE)
```

The statistic is equal to 244.66, and the p-value is $p < 0.001$. Note that we divided the p-value by 2. Hence, there is strong support for the negative binomial model. The same result can be obtained with the command `odTest(M8)` from the `pscl` package, which is not part of the base installation.

The amphibian roadkills data set is further analysed in Chapter 16. A comparison of the Poisson, quasi-Poisson, negative binomial, and three alternative models in case there are lots of zeros (the hurdle model, zero-inflated Poisson, and zero-inflated negative binomial models) is presented in Chapter 11.

9.11 GAM

Having explained Gaussian additive models in detail in Chapter 3 and the Poisson and negative binomial GLM in detail in earlier sections in this chapter, it is rather simple to explain Poisson or negative binomial GAM. A Poisson GAM has these assumptions:

1. Y_i is Poisson distributed with mean μ_i. By definition the variance of Y_i is also equal to μ_i.

2. The systematic part is given by $\eta(X_{i1}, \ldots, X_{iq}) = \alpha + f_1(X_{i1}) + \ldots + f_q(X_{iq})$, where the f_js are smoothing functions.
3. There is a logarithm link between the mean of Y_i and the predictor function $\eta(X_{i1}, \ldots, X_{iq})$. The logarithmic link ensures that the fitted values are always non-negative.

As a result of these three assumptions, we have

$$\begin{aligned} Y_i &\sim P(\mu_i) \\ E(Y_i) = \mu_i &\quad \text{and} \quad \text{var}(Y_i) = \mu_i \\ \log(\mu_i) = \eta(X_{i1}, \cdots, X_{iq}) &\quad \text{or} \quad \mu_i = e^{\eta(X_{i1}, \cdots, X_{iq})} \end{aligned} \tag{9.16}$$

For a negative binomial GAM, we only have to change step 1 from the Poisson distribution to a negative binomial distribution and the variance is then given by $\mu_i + {\mu_i}^2/k$. A detailed example of the negative binomial GAM is given in Chapter 16. Below, we present a short example of a GAM that also illustrates the use of the offset variable in Poisson and NB GLMs and GAMs.

9.11.1 Distribution of larval Sea Lice Around Scottish Fish Farms

The data used in this example are taken from Penston et al. (2008). Plankton tows were taken approximately weekly at two depths (0 and 5 m) at five stations for two years. In the original paper, numbers of *nauplii* and *copepodids* were analysed in two separate univariate analysis where production week (time expressed in weeks since March 2002, when the local farms stocked their cages with lice-free, juvenile fish), station, and depth were the covariates. There are five stations labelled as A, C, E, F, and G. Stations C and G are beside salmon farms, stations A and F are landward of these farms, and station E is seaward of the farms. Here, we only use copepodids. Further biological details can be found in Penston et al. (2008).

There are three potential problems with the analysis of these data: we have longitudinal (over time) data at each station, there may be correlation between adjacent stations, and there is a large variation in the sampled water volume. As to the first two problems, we follow the same strategy as the paper by showing there is no temporal correlation *within* each of the residual time series, and that there are no strong Pearson correlations *between* the 5 residual time series. The third problem of different volumes per observation was discussed in Chapter 8. Define Y_i as the number of copepodids measured for observation i. We could have used a notation Y_{sk}, referring to observation s at station k, but we will keep the notation simple and stick to Y_i. As Y_i is a count, a Poisson, negative binomial or geometric distribution is appropriate. We start with the Poisson distribution. So far in this chapter, we assumed that Y_i is Poisson distributed with mean μ_i, which we wrote as $\text{P}(\mu_i)$ with its probability function as

$$f(y_i;\mu_i) = \frac{\mu_i^{y_i} e^{-\mu_i}}{y_i!} \qquad y_i \geq 0,\ y_i \text{ integer} \tag{9.17}$$

The problem with these data is that the water volumes differ per observation, see Fig. 9.9. We may measure a large number of copepodids simply because the water volume was large. The easiest solution is to work with densities, and analyse these with a Gaussian distribution. The disadvantage of this is that the fitted values may become negative, there may be heterogeneity, etc. It is also an option to use Volume as an explanatory variable, but then you would be modelling a functional relationship between Volume and numbers of copepodids. A neater approach is to use Volume as an offset; this process works as follows.

Assume that Y_i is Poisson distributed with mean $\mu_i \times V_i$. V_i is also called the exposure or intensity parameter of the Poisson process, and μ_i is the expected number of copepodids for a one unit volume. The expected value and variance are: $\mathrm{E}(Y_i) = \mu_i \times V_i$ and $\mathrm{var}(Y_i) = \mu_i \times V_i$. The following simple algebra leads to a GLM (or GAM) with an offset variable.

$$E(Y_i) = \mu_i \times V_i \quad \Rightarrow \quad \log(E(Y_i)) = \log(\mu_i) + \log(V_i) = \alpha + \beta \times X_{i1} + f(X_{2i}) + \log(V_i)$$

The term $\log(V_i)$, where log is the natural log, is the offset. Using basic mathematics, we have placed the V_i inside the predictor function, but note there is no regression parameter in front of this term. The other terms α and β are the regression parameters and $f()$ is a smoothing function. R will estimate the regression parameters and smoothers, and you can express the fitted values of the model either as μ_i or as $\mu_i \times V_i$.

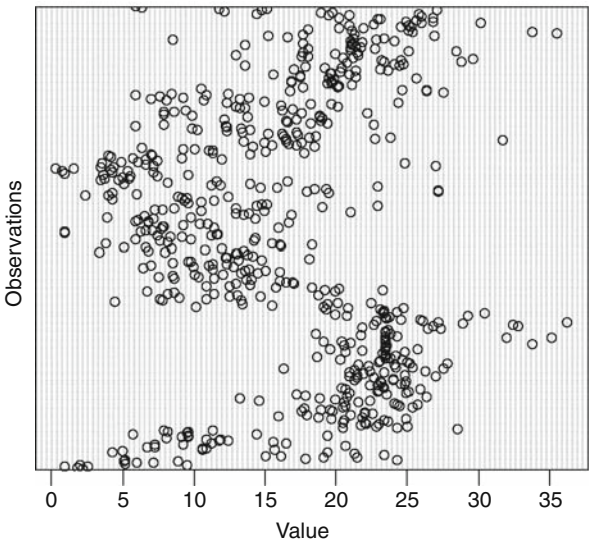

Fig. 9.9 Cleveland dotplot of the sampled volumes. Note that there are considerable differences in volumes! The graph was produced with the R command `dotchart (Volume, xlab = "Value" , ylab = "Observations")`

The offset can be used for a Poisson, negative binomial, and geometric distribution. The advantages of the offset approach compared to analysing densities are that the fitted values are always positive, the confidence intervals around the fitted values do not contain negative values, and we allow for heterogeneity within the context of a Poisson or NB distribution.

To use an offset variable in a GLM or GAM in R, use the following code.

```
> library(AED); data(Lice)
> Lice$LVol <- log(Lice$Volume)
> Lice$fStation <- factor(Lice$Station)
> L0 <- glm(Copepod ~ offset(LVol) + fStation,
        family = poisson, data = Lice)
```

The first two commands import the data. The variable `LVol` contains the natural log transformed volumes, and `offset(LVol)` ensures that the `glm` function is not putting a parameter in front of it. The only problem is that unfortunately, the model itself is rubbish; we have only shown it to illustrate how to use an offset in a GLM or GAM. So we will now move on and do it for real. There are three explanatory variables, `Station`, `Depth` (both are factors), and `Production_week`. Simple scatterplots indicate no clear relationships, and we therefore used a GAM. We start with a Poisson distribution. The most complicated model that we can apply contains a smoother for production week for each station and depth combination, the main terms station and depth, and the interaction between station and depth. This is the GAM equivalent of 3-way interaction. The problem is that such a model ended in an error message (numerical convergence problems), and we therefore switched to a negative binomial distribution. The following code was used.[1]

```
> library(mgcv)
> Lice$PW <- Lice$Production_week #saves some space
> Lice$fDepth <- factor(Lice$Depth)
> L1 <- gam(Copepod ~ offset(LVol)+
    s(PW, by=as.numeric(Depth=="0m" & Station=="A")) +
    s(PW, by=as.numeric(Depth=="0m" & Station=="C")) +
    s(PW, by=as.numeric(Depth=="0m" & Station=="E")) +
    s(PW, by=as.numeric(Depth=="0m" & Station=="F")) +
    s(PW, by=as.numeric(Depth=="0m" & Station=="G")) +
    s(PW, by=as.numeric(Depth=="5m" & Station=="A")) +
    s(PW, by=as.numeric(Depth=="5m" & Station=="C")) +
    s(PW, by=as.numeric(Depth=="5m" & Station=="E")) +
```

[1] We used R version 2.6.0. More recent R versions require slightly different code; see the book website for updated code.

```
    s(PW, by=as.numeric(Depth=="5m" & Station=="F")) +
    s(PW, by = as.numeric(Depth=="5m" & Station=="G")) +
    fDepth * fStation,
    family = negative.binomial(1), data = Lice)
```

This model also gave a warning message, but including the option `gamma = 1.4` allows the code to run. This option helps against overfitting by the smoothers (Wood, 2006); it puts a heavier penalty on each degrees of freedom in the GCV score (Chapter 3).

A backward selection resulted in various numerical problems, and therefore in the original paper, Penston et al. (2008) adopted a slightly different approach for the model selection process. They estimated the parameter k (used in the NB variance function) from one of the larger models, e.g. from `L3`, and kept it fixed during the backwards selection. This gave an optimal model, and the whole backward selection process was then repeated using the k from the first optimal model. Both selection rounds ended up in the same model, namely,

```
> L3 <- gam(Copepod ~ offset(LVol) +
            s(PW, by = as.numeric(Depth=="0m")) +
            s(PW, by = as.numeric(Depth=="5m")) +
            fDepth + fStation, data = Lice,
            family = negative.binomial(1), gamma = 1.4)
```

This model contains a smoother for production week for each depth together with depth and station as factors. We can compare this model with its Poisson equivalent using the likelihood ratio test:

```
> L4 <- gam(Copepod ~ offset(LVol) +
            s(PW,by = as.numeric(Depth=="0m")) +
            s(PW,by = as.numeric(Depth=="5m")) +
            fDepth + fStation, data = Lice,
            family = poisson, gamma = 1.4)
> llhNB <- logLik(L3); llhPoisson <- logLik(L4)
> d <- 2 * (llhNB - llhPoisson)
> pval <- 0.5 * pchisq(as.numeric(d), df = 1,
                       lower.tail = FALSE)
```

The likelihood ratio statistic is 2137.20, which is strong evidence to choose the NB GAM over the Poisson GAM. The numerical output of the NB GAM is obtained by the `summary(L3)` command:

```
Family: Negative Binomial(0.3569). Link function: log
Parametric coefficients:
                 Estimate Std. Error t value  Pr(>|t|)
(Intercept)       -1.7030     0.1956  -8.708  < 2e-16
factor(Depth)5m   -1.3921     0.2203  -6.319  5.2e-10
```

```
factor(Station)C  -0.3496     0.2513  -1.391  0.16470
factor(Station)E  -0.4661     0.2546  -1.830  0.06769
factor(Station)F  -0.8455     0.2656  -3.183  0.00153
factor(Station)G   0.1102     0.2524   0.437  0.66253

Approximate significance of smooth terms:
                                 edf     F  p-value
s(PW):as.numeric(Depth=="0m") 8.36 15.62  < 0.001
s(PW):as.numeric(Depth=="5m") 6.14  5.45  < 0.001

R-sq.(adj) =  0.212. Deviance explained = 72.6%
GCV score = 1.0644. Scale est. = 1. n = 608
```

The model explains 72.6% of the null deviance. The *p*-values for the levels of station only indicate which stations are significantly different from the baseline station A (Dalgaard, 2002). A post-hoc test can be applied to investigate which sites are different from each other. The fitted values are given in Fig. 9.10.

Further discussions on the results, model validation (there was no significant temporal auto-correlation within the four residual time series), biological interpretation, and analyses can be found in Penston et al. (2008). Besides the NB GAM, it may also be an option to apply a zero-inflated GAM. These models are discussed in Chapter 11.

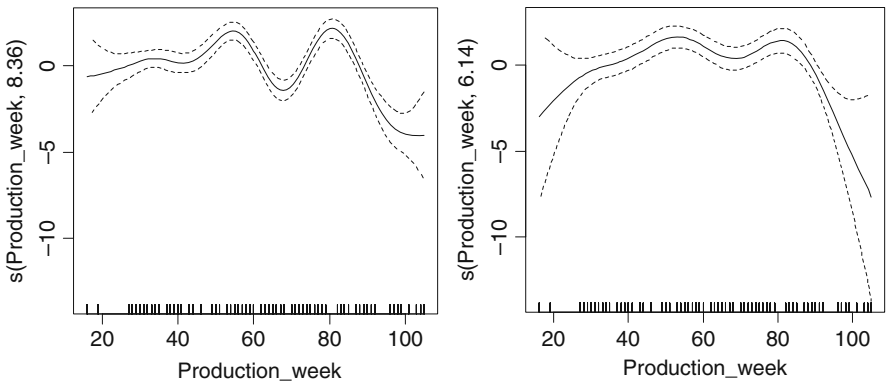

Fig. 9.10 Estimated smoothing curves for depth at the surface (*left*) and depth and 5 m (*right*). The *solid line* is the smoother and the dotted lines are 95% point-wise confidence bands