Chapter 11
Zero-Truncated and Zero-Inflated Models for Count Data

11.1 Introduction

In this chapter, we discuss models for zero-truncated and zero-inflated count data. Zero truncated means the response variable cannot have a value of 0. A typical example from the medical literature is the duration patients are in hospital. For ecological data, think of response variables like the time a whale is at the surface before re-submerging, counts of fin rays on fish (e.g. used for stock identification), dolphin group size, age of an animal in years or months, or the number of days that carcasses of road-killed animals (amphibians, owls, birds, snakes, carnivores, small mammals, etc.) remain on the road. These are all examples for which the response variable cannot take a value of 0.

On their own, zero-truncated data are not necessarily a problem. It is the underlying assumption of Poisson and negative binomial distributions that may cause a problem as these distributions allow zeros within their range of possible values. If the mean is small, and the response variable does not contain zeros, then the estimated parameters and standard errors obtained by GLM may be biased. In Section 11.2, we introduce zero-truncated Poisson and zero-truncated negative binomial models as a solution for this problem. If the mean of the response variable is relatively large, ignoring the truncation problem, then applying a Poisson or negative binomial (NB) generalised linear model (GLM), is unlikely to cause a problem. In such cases, the estimated parameters and standard errors obtained by Poisson GLM and truncated Poisson GLM tend to be similar (the same holds for the negative binomial models).

In ecological research, you need to search very hard to find zero-truncated data. Most count data are zero *inflated*. This means that the response variable contains more zeros than expected, based on the Poisson or negative binomial distribution. A simple histogram or frequency plot with a large spike at zero gives and early warning of possible zero inflation. This is illustrated by the graph in Fig. 11.1, which shows the numbers of parasites for the cod dataset that was used in Chapter 10 to illustrate logistic regression. In addition to presence and absence of parasites in cod, Hemmingsen et al. (2005) also counted the number of parasites, expressed as intensity.

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_11,

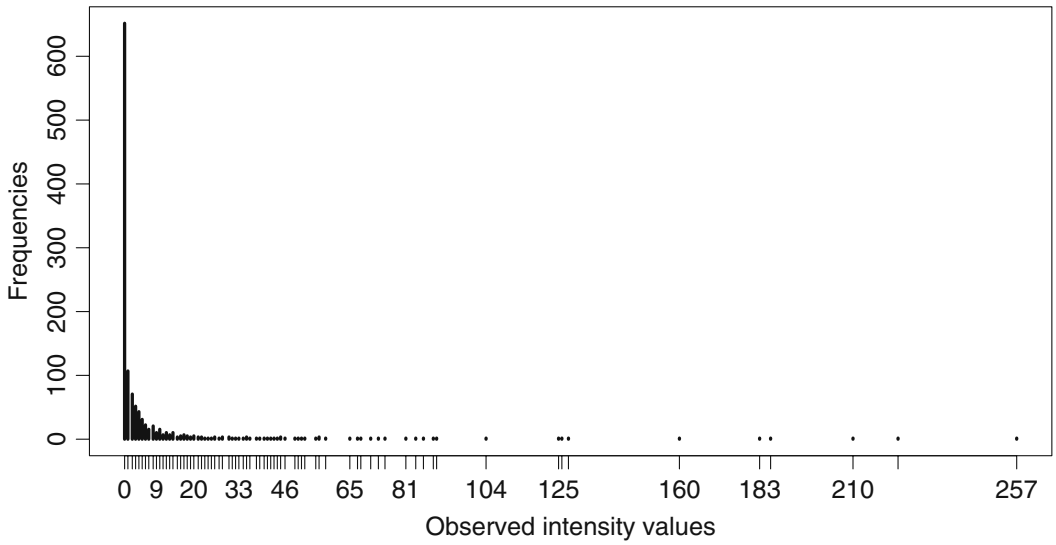

Fig. 11.1 Plot of the frequencies for the response variable `Intensity` from cod parasite data. There are 654 zeros, 108 ones, 71 twos, 52 threes, 44 fours, 31 fives, etc. Note the large numbers of zeros indicating zero inflation. R code to make this graph is presented in Section 11.4

In this chapter, four models are discussed that can deal with the excessive number of zeros; zero-inflated Poisson (ZIP), zero-inflated negative binomial (ZINB) models, zero-altered Poisson (ZAP), and zero-altered negative binomial (ZANB) models. There are two main distinctions in these abbreviations; ZI versus ZA, and P versus NB. The latter pair of Poisson versus negative binomial should be familiar territory with the negative binomial models (ZINB and ZANB) coping with a certain degree of overdispersion. Furthermore, because a Poisson GLM is nested in a NB GLM, the ZIP is nested in a ZINB, and a ZAP is nested in a ZANB. The difference between ZI and ZA is slightly more complicated and is related to the nature of the zeros. We discuss this further in Sections 11.3 and 11.4. What we call ZI is also called mixture models in the literature, and our ZA is normally known as two-part models.

In the past, software for mixture and two-part models used to be in obscure functions, and different software packages gave different results. It is only recently that these methods have become more popular and a growing number of people are using the software. This means that most of the bugs have now been filtered out, and publications with mixture and two-part models applied on ecological data are appearing more frequently (Welsh et al., 1996; Agarwal et al., 2002; Barry and Welsh, 2002; Kuhnert et al., 2005; Minamia et al., 2007; and Potts and Elith, 2006 among several others). There are also many applications outside ecology; see, for example, Lambert (1992), Ridout et al. (1998), Xie et al. (2001), and Carrivick et al. (2003) among many others in the fields of social science, traffic accident research, econometrics, psychology, etc. A nice overview and comparison of Poisson, NB, and zero-inflated models in R is given in Zeileis et al. (2008). This paper also gives a couple of useful references to publications using mixture and two-part models.

If you start digging into zero-inflated models, you have to rely mainly on papers as few statistical textbooks cover this topic in any detail. A few exceptions are

Cameron and Trivedi (1998), Hardin and Hilbe (2007), or Hilbe (2007), but only a small number of pages are dedicated to mixture and two-part models. As papers tend to present things in a compact and condensed format, we decided to use this chapter to explain these methods in more detail. We assume that you are fully familiar with the methods discussed in Chapters 8, 9, and 10.

A detailed explanation of the underlying principle of mixture and two-part models is given in Sections 11.2–11.5, and in Section 11.6, we compare the different models and discuss how to choose between them.

11.2 Zero-Truncated Data

In this section, we discuss models that can be used when the response variable is a count and cannot obtain the value of zero. In this case, we refer to the variable as being zero truncated. In Section 11.2.1, we discuss the underlying mathematics for zero-truncated Poisson models and the negative binomial equivalent. In Section 11.2.2, we give an example and discuss software. If you are not interested in the underlying mathematics, you can skip Section 11.2.1 (but you should still try and read the summary at the end of that section) and go straight to the example.

Knowledge of the material discussed in this section is required for ZAP and ZANB models discussed in Section 11.5.

11.2.1 The Underlying Mathematics for Truncated Models

11.2.1.1 Mathematics for the Zero-Truncated Poisson Model

Let Y_i be the response variable for observation i. We assume it is Poisson distributed with mean μ_i. We have already discussed in Chapter 8, how the Poisson probability function can be adjusted to exclude zeros, and we briefly revisit it here. The starting point was the Poisson probability function:

$$f(y_i; \mu_i | y_i \geq 0) = \frac{\mu^{y_i} \times e^{-\mu_i}}{y_i!} \tag{11.1}$$

Recall that y_i is a possible outcome of Y_i. The function gives the probability for each integer value of y_i that is equal or larger than 0 for a given mean μ_i. For example, the probability that $y_i = 0$ is

$$f(0; \mu_i) = \frac{\mu^0 \times e^{-\mu_i}}{0!} = e^{-\mu_i}$$

Recall from Chapter 8 that we can exclude the probability that $y_i = 0$ from the Poisson distribution by dividing its probability function in Equation (11.1) by 1 minus the probability that $y_i = 0$, resulting in

$$f(y_i;\mu_i|y_i>0)=\frac{\mu^{y_i}\times e^{-\mu_i}}{(1-e^{-\mu_i})\times y_i!} \tag{11.2}$$

From this point onwards, truncated Poisson GLM follows ordinary Poisson GLM. We use the same mean and variance relationships, the same systematic component, and the same link function. Hence, the mean value μ_i is modelled as an exponential function of the predictor function:

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

To find the regression parameters, we need to specify a likelihood criterion. The only difference with Poisson GLM is that we use the probability function in Equation (11.2) instead of the one in Equation (11.1), and this gives

$$L=\prod_i f(y_i;\mu_i|y_i>0)=\prod_i \frac{\mu^{y_i}\times e^{-\mu_i}}{(1-e^{-\mu_i})\times y_i!} \tag{11.3}$$

In Chapter 9, we explained that this expression is based on the probability rule that Pr(A and B) = Pr(A) × Pr(B) if A and B are independent. The *f*s in Equation (11.3) are the probabilities. The principle of maximum likelihood states that for the given data, we need to maximise L as a function of the regression parameters. To aid the numerical optimisation routines, we use the log-likelihood so that we can work with a sum instead of a product:

$$\log(L)=\sum_i \log(f(y_i;\mu_i|y_i>0))=\sum_i \log\left(\frac{\mu_i^{y_i}\times e^{-\mu_i}}{(1-e^{-\mu_i})\times y_i!}\right) \tag{11.4}$$

Using matrix notation, we replace the $\beta_1\times X_{1i}+\ldots+\beta_q\times X_{qi}$ by $\boldsymbol{X}_i\times\boldsymbol{\beta}$, where $\boldsymbol{\beta}=(\beta_1,\ldots,\beta_q)$, and $\mathbf{X}_i$ contains all explanatory variables for observation i. A bit of high school mathematics gives

$$\begin{aligned}\log(L)=&-\sum_i e^{\mathbf{X}_i\times\boldsymbol{\beta}}+\sum_i y_i\times\mathbf{X}_i\times\boldsymbol{\beta}-\sum_i\log(1-e^{\mathbf{X}_i\times\boldsymbol{\beta}})\\&-\sum_i\log(\Gamma(y_i+1))\end{aligned} \tag{11.5}$$

Just as for the Poisson GLM, we end up with a maximum likelihood criterion that needs to be maximised as a function of the regression parameters. The algorithm needs first-order and second-order derivatives (which can easily be determined and we leave this as an exercise for the reader), and then it is purely a matter of numerical optimisation, though we end up with a slightly different algorithm compared to Poisson GLM. Details can be found in Barry and Welsh (2002) or Hilbe (2007).

11.2.1.2 Mathematics for the Negative Binomial Truncated Model

The NB truncated model follows the same steps. The starting point is the probability function for y larger or equal to 0 (Chapter 9):

$$f(y_i; k, \mu_i | y_i \geq 0) = \frac{\Gamma(y_i + k)}{\Gamma(k) \times \Gamma(y_i + 1)} \times \left(\frac{k}{\mu_i + k}\right)^k \times \left(1 - \frac{k}{\mu_i + k}\right)^{y_i} \quad (11.6)$$

The probability that $y_i = 0$ is given by

$$f(0; k, \mu_i) = \frac{\Gamma(0 + k)}{\Gamma(k) \times \Gamma(0 + 1)} \times \left(\frac{k}{\mu_i + k}\right)^k \times \left(1 - \frac{k}{\mu_i + k}\right)^0 = \left(\frac{k}{\mu_i + k}\right)^k$$

To exclude the probability that $y_i = 0$, we divide the probability function in Equation (11.6) by 1 minus the probability that $y_i = 0$, resulting in

$$f(y_i; \mu_i | y_i > 0) = \frac{\Gamma(y_i+k)}{\Gamma(k)\times\Gamma(y_i+1)} \times \left(\frac{k}{\mu_i+k}\right)^k \times \left(1 - \frac{k}{\mu_i+k}\right)^{y_i} \Bigg/ \left(1 - (\frac{k}{\mu_i+k})^k\right) \quad (11.7)$$

We can follow the same steps as in Equations (11.3) and (11.4) and also use the logarithmic link function. The end result is as follows:

$$\log(L) = \log(L_{NB}) - \log\left(1 - \left(\frac{k}{\mu_i + k}\right)^k\right) \quad (11.8)$$

where $\log(L_{\text{NB}})$ is the log likelihood from the NB GLM (see Chapter 9). Note that the notation in Hardin and Hilbe (2007) and Hilbe (2007) uses a slightly different parameterisation of $k = 1/\alpha$.

11.2.1.3 Summary

For those of you who skipped all the mathematical text in this subsection, here is a short summary. We adjusted the probability functions for the Poisson and negative binomial (NB) distributions to exclude the probability of a zero observation. We then specified the log likelihood criterion for the zero-truncated Poisson and NB models. First-order and second-order derivatives can easily be derived. It is now only a matter of numerical optimisation to find the regression parameters. Software code exists to fit these models in R, and an example is given in the next section.

11.2.2 Illustration of Poisson and NB Truncated Models

In this section, we illustrate zero-truncated models. The data are unpublished (at the time of writing) and were donated by António Mira (University of Évora, Portugal). The response variable is the number of days that carcasses of road-killed

animals remain on the road. For illustrative purposes, we only use snakes (*Coronella girondica*, *Coluber hippocrepis*, *Elaphe scalaris*, and *Macroprotodon cucullatus*). We removed some observations because of the unbalanced design (different sample sizes), and the remaining data set contains 130 observations. There are also potential issues with spatial and temporal correlation, but in this subsection, we only focus on the zero truncation.

Figure 11.2 shows a frequency plot of the number of days that snake carcasses remain on a road. The value of 1 does not represent 24 hours exactly, rather it is just that we start counting with 1 because each carcass is on the road for at least a couple of hours. The number of days will never be zero. Except for the lucky snakes that made it to the other side of the road. They will have a value of zero, but of course, are not (yet) part of this dataset.

The following R code accesses the data and produces the frequency plot in Fig. 11.2. The code is self explanatory.

```
> library(AED); data(Snakes)
> plot(table(Snakes$N_days))
```

Ignoring the zero truncation problem and analysing these data with a Poisson GLM is already a major challenge! The explanatory variables are `Size_cm` (mean size of adults of each species), `PDayRain` (proportion of days with rain), `Tot_Rain` (total rainfall in mm), `Temp_avg` (average daily mean temperature), `Road` (identity of the road representing traffic intensity; EN114 has high traffic, EN4 has medium traffic, and EN370_EN114_4 has low traffic), `Road_Loc` (location on the road; L = paved lane and V = paved verge), `Season`, and `Species`.

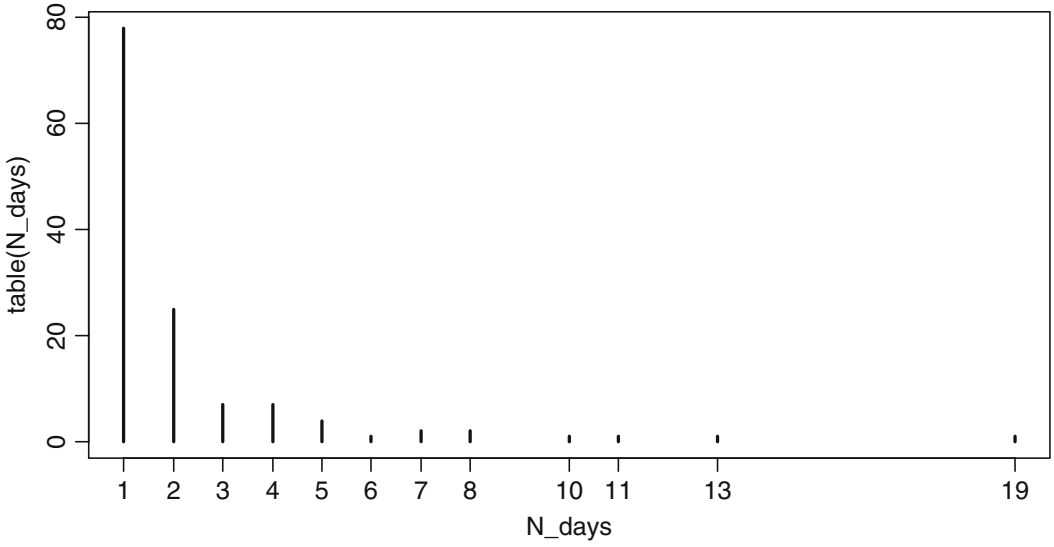

Fig. 11.2 Frequency plot of the response variable `N_days`, the number of days snake carcasses remain on the road. Note that a value of 0 cannot occur

The variables Size_cm, PDayRain, Tot_Rain, and Temp_avg are continuous; all others are nominal.

Exploring these data using pairplots and correlation coefficients for the continuous variables, and boxplots of each continuous explanatory variable conditional on each nominal explanatory variable, showed that Season is collinear with both Temp_avg and Tot_Rain, and there is also collinearity between PDayRain and Temp_avg. We therefore omitted Season and Temp_avg. All observations from the same species had the same size, and therefore, the covariate Species was also dropped. From a biological point of view, it may be argued that Species is a more useful covariate than size; however, the degrees of freedom rapidly increase if various two-way interactions with species are included in the model.

Using common sense, it can be argued that there may be interactions; perhaps, carcasses of bigger animals at sites with less rain stay longer on the road? Not all 2-way interactions can be fitted due to the experimental design. We started our data analysis with a Poisson GLM and quickly noticed overdispersion. Therefore, a quasi-Poisson model was applied. The results of this model are not presented here, but there is an overdispersion of 1.5 and various terms are not significant. The aim of this section is to show the difference between a GLM and a zero-truncated GLM, and because there is no such thing as a zero-truncated quasi-Poisson model, we switch to a negative binomial model as NB models allow for a more flexible approach to overdispersion. R code for the NB GLM, ignoring the zero truncation, is given by

```
> library(MASS)
> M1 <- glm.nb(N_days ~ Size_cm + PDayRain + Tot_Rain +
             Road + Road_Loc + Size_cm:PDayRain +
             Size_cm:Tot_Rain + Size_cm:Road +
             Size_cm:Road_Loc + PDayRain:Tot_Rain +
             PDayRain:Road + PDayRain:Road_Loc +
             Tot_Rain:Road, data = Snakes)
```

Similar code was used in Chapter 9. The results of the summary(M1) command are not presented here, but show that various terms are not significant at the 5% level. The optimal model was found using step(M1), and further fine tuning was done with the drop1(M1, test = "Chi") command. The optimal model is given by

```
> M2A <- glm.nb(N_days ~ PDayRain + Tot_Rain +
            Road_Loc + PDayRain:Tot_Rain, data = Snakes)
```

The two-way interaction PDayRain:Tot_Rain and the main term Road_Loc were significant at the 5% level. The explained deviance of this model is 40%. The parameter k (theta in the R output) in the variance function $\mu_i + \mu_i^2/k$ is equal to 6.72. Interestingly, the model selection process for the quasi-Poisson GLM gave the same results.

So far, we have used the `glm.nb` function from the MASS package for negative binomial GLM; but it can also be done in other packages, for example, in the VGAM (Vector Generalized Additive Models) package with the code:

```
> library(VGAM)
> M2B <- vglm(N_days ~ PDayRain + Tot_Rain + Road_Loc +
              PDayRain:Tot_Rain, family = negbinomial,
              data = Snakes)
> summary(M2B)
```

The VGAM package does not come with the base installation of R; so you will need to download and install it. Actually, this package is rather interesting as it contains many statistical techniques closely related to those we use in this book. For example, it has tools for multivariate (multiple response variables) GLMs and GAMs (Yee and Wild, 1996), and it is one of the few packages that can do zero-truncated models! It is certainly worthwhile having a look at the package description at www.stat.auckland.ac.nz/~yee/VGAM. The zero-truncated NB model is run with the following R code.

```
> M3A <- vglm(N_days ~ PDayRain + Tot_Rain + Road_Loc +
             PDayRain:Tot_Rain, family = posnegbinomial,
             control = vglm.control(maxit = 100),
             data = Snakes)
```

The `family = posnegbinomial` argument ensures that a zero-truncated NB model is applied. The `summary` command can be used to obtain estimated parameters and standard errors, but the `anova` and `drop1` functions have not yet been implemented in the VGAM package.

The option `family = pospoisson` runs a zero-truncated Poisson GLM, and if `vglm` is replaced by `vgam`, we obtain a zero-truncated GAM. To run an ordinary Poisson GLM, use `family = poissonff`; the extra ff is due to VGAM's incompatibility with the ordinary `family` option in R and is specific to this package. Another 'problem' with VGAM is that it overwrites existing functions. You can overcome this by using, for example, `stats::resid` after you have typed the `library(VGAM)` command. The `stats::` ensures that you use the `resid` function from the stats package (which is the one used in all chapters so far) and not VGAM's `resid` function, which is not compatible with `glm` and `lm` objects.

It is interesting to compare the parameters and standard errors estimated using NB GLM and truncated NB GLM. The following code looks intimidating, but only collates the corresponding estimated regression parameters in a table:

```
> Z <- cbind(coef(M2A), coef(M3A)[-2])
> ZSE <- cbind(sqrt(diag(vcov(M2A))),
               sqrt(diag(vcov(M3A))[-1]))
```

```
> Comp <- cbind(Z[,1], Z[,2], ZSE[,1], ZSE[,2])
> Comb <- round(Comp, digits = 3)
> colnames(Comb) <-
            c("NB", "Trunc.NB", "SE NB", "SE Trunc.NB")
> Comb
```

The `coef` command extracts the estimated parameters and the `vcov` the covariance matrix of the estimated parameters. The diagonal elements of this matrix are the estimated variances; hence, the square root of these gives the standard errors. [-2] ensures that only regression parameters are extracted and not the parameter k. The `cbind` command prints the columns next to each other, and the `colnames` command adds labels. The output is as follows:

```
                      NB Trunc.NB   SE NB   SE Trunc.NB
(Intercept)        0.365   -2.035   0.112         0.267
PDayRain          -0.001    0.114   0.193         0.449
Tot_Rain           0.120    0.254   0.020         0.065
Road_LocV          0.449    1.077   0.148         0.368
PDayRain:Tot_Rain -0.109   -0.234   0.022         0.070
```

The first two columns are the estimated parameters obtained by NB GLM and truncated NB GLM. As you can see, the estimated parameters obtained using these two methods are rather different! The same holds for the standard errors in the third and fourth columns. Also note that the standard errors of the truncated NB are all larger.

Differences between NB GLM and truncated NB GLM will become smaller if the observed values are further away from zero. But in this case, with 93% of the observations smaller than 5, it makes a substantial difference!

11.3 Too Many Zeros

Zero inflation means that we have far more zeros than what would be expected for a Poisson or NB distribution. Let us have another look at Fig. 11.1, but only at the frequencies between 0 and 10 (see Fig. 11.3). If the data followed a Poisson distribution, you would not expect 651 zeros! It depends a bit on the value of the mean of the Poisson distribution, but 100 zeros would be more likely (see also the shapes of the Poisson probability functions in Chapter 8).

Ignoring zero inflation can have two consequences; firstly, the estimated parameters and standard errors may be biased, and secondly, the excessive number of zeros can cause overdispersion. Before discussing two techniques that can cope with all these zeros, we need to ask the question: Why do we have all these zeros?

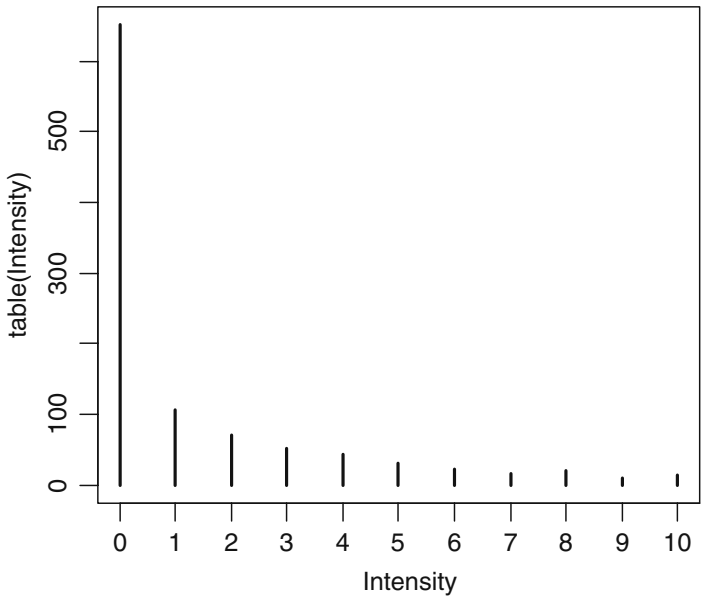

Fig. 11.3 Intensity of parasites in cod. This is the same graph as Fig. 11.1, except that only frequencies between 0 and 10 are shown

11.3.1 Sources of Zeros

If we assume a Poisson distribution for the data presented in Fig. 11.3, then we would expect approximately 100–150 zeros. These are at the lower part of the vertical line at intensity $= 0$. All the other zeros are excess zeros and more than we expect. Some authors try to make a distinction between these two groups of zeros. For example, Kuhnert et al. (2005) and Martin et al. (2005) discriminate between various types of errors that may be causing the zeros in the context of bird abundances in forest patches.

1. First of all, there are structural errors. This means that a bird is not present because the habitat is not suitable.
2. The second is design error, where poor experimental design or sampling practises are thought to be the reason. As an example, imagine counting the number of puffins on the cliffs in the winter. It is highly likely that all samples will be 0 as it is the wrong season and they are all at sea. Another design error is sampling for too short a time period or sampling too small an area.
3. The third cause for zeros is observer error. Some bird species look similar, or are difficult to detect. The less experienced the observer, the more likely he/she will end up with zero counts for bird species that are difficult to identify. Alternatively, the observer may be highly experienced, but it is extremely difficult to detect a tiny dark bird in a dark field on a dark day.
4. The 'bird' error. This means that the habitat is suitable, but the site is not used.

There is even a fifth type of zero, the so-called naughty naughts (Austin and Meyers, 1996). For non-native English readers, this can be translated as the bad

zeros. These are zeros due to sampling outside the habitat range that an animal lives in; for example, sampling for elephants in the sea. Any such zeros should be removed.

The zeros due to design, survey, and observer errors are also called false zeros or false negatives. In a perfect world, we should not have them. The structural zeros are called positive zeros, true zeros, or true negatives. It should be noted that these definitions of true and false zeros are open to discussion. In some studies, a false zero may actually be a true zero; see also Martin et al. (2005) for a discussion.

11.3.2 Sources of Zeros for the Cod Parasite Data

Hemmingsen et al. (2005) looked at the effect of introducing the red king crab *Paralithodes camtschaticus* in the Barents Sea. This species is a host for the leech *Johanssonia arctica*, which in turn is a vector for a trypanosome blood parasite of marine fish, including cod. The data set contains a large number of zeros. Let us discuss what type of zeros we have.

First of all, there are fish that have not been exposed to the parasite, either because they were caught at a place where there are no red king crabs or they had migrated long distances and arrived when Hemmingsen and colleagues turned up to catch them. These zeros can probably be labelled as zeros due to 'poor' experimental design; however, we put quotation marks around poor as there is not much the biologists can do about it. None the less they are still false zeros that we need to deal with. We also have zeros because of observer errors. Apparently, it is not always easy to detect trypanosomes in fish with light infections, even for experienced parasitologists (Ken MacKenzie, personal communication). So these are also false zeros. The other type of zeros, the true zeros or the true negatives, come from fish that may have been in contact with red king crabs; but for some reason, they have zero parasites. There may be many reasons for this, including habitat, immunity, and environmental conditions.

11.3.3 Two-Part Models Versus Mixture Models, and Hippos

In the next section, four models are used to analyse the zero-inflated data: ZIP, ZINB, ZAP, and ZANB (see also Table 11.1). We have already discussed the difference between the P and the NB. That is Poisson versus negative binomial, where the negative binomial allows for extra overdispersion in the positive (non-zero) part of the data. The difference between the mixture and two-part models is how they deal with the different types of zeros. The two-part models (ZAP and ZNAB) are probably easier to explain; they consist of two parts:

Table 11.1 Overview of ZIP, ZAP, ZINB and ZANB models. All models can cope with overdispersion due to excessive numbers of zeros. The negative binomial models can also cope with overdispersion due to extra variation in the count data. The ZIP and ZINB are mixture models in the sense that they consist of two distributions. The ZAP and ZANB are also called hurdle models, conditional models, or compatible models

Model	Full name	Type of model	Overdispersion
ZIP	Zero-inflated Poisson	Mixture	Zeros
ZINB	Zero-inflated negative binomial	Mixture	Zeros and counts
ZAP	Zero-altered Poisson	Two-part	Zeros
ZANB	Zero-altered negative binomial	Two-part	Zeros and counts

1. In first instance, the data are considered as zeros versus non-zeros and a binomial model is used to model the probability that a zero value is observed. It is possible to use covariates in this model, but an intercept-only model is also an option.
2. In the second step, the non-zero observations are modelled with a truncated Poisson (ZAP) or truncated negative binomial (ZANB) model, and a (potentially different) set of covariates can be used. Because the distributions are zero truncated, they cannot produce zeros.

You can use specific software for ZAPs and ZANBs, but it is also possible to carry out these two steps manually with a binomial GLM and a Poisson/NB GLM; both give the same results in terms of estimated parameters and standard errors. The advantage of using specialised ZAP or ZANB software is that it gives one AIC for both models (this can also be calculated manually from the two separate models), and it is more flexible for hypothesis testing for the combined model. Figure 11.4 shows a graphical presentation of the two-part, or hurdle, models for the hippo example. The name hurdle comes from the idea that whatever mechanism is causing the presence of hippos, it has to cross a hurdle before values become non-zero. The important point is that the model does not discriminate between the four different types of zeros.

The ZIP and ZINB models work rather differently. They are also called mixture models because the zeros are modelled as coming from two different processes: the binomial process and the count process. As with the hurdle models, a binomial GLM is used to model the probability of measuring a zero and covariates can be used in this model. The count process is modelled by a Poisson (ZIP) or negative binomial (ZINB) GLM. The fundamental difference with the hurdle models is that the count process can produce zeros (the distribution is not zero truncated).

The underlying process of the mixture model is sketched in Fig. 11.5. In the count process, the data are modelled with, for example, a Poisson GLM, and under certain covariate conditions, we count zero hippos. These are true zeros. But there is also a process that generates only false zeros, and these are modelled with a binomial model. Hence, the binomial GLM models the probability of measuring a false positive versus all other types of data (counts and true zeros).

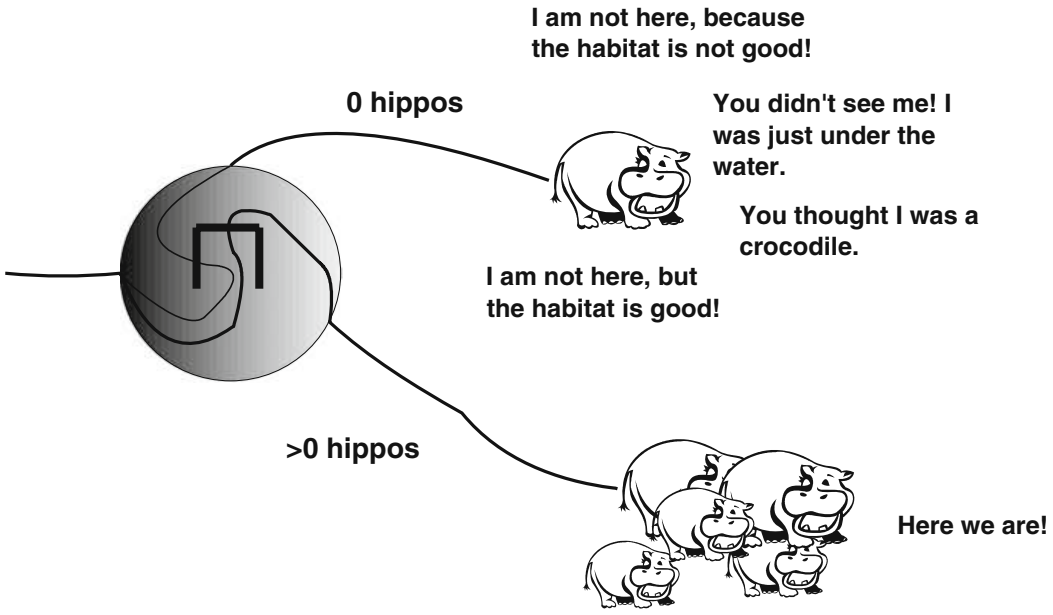

Fig. 11.4 Sketch of a two-part, or hurdle model. There are two processes; one is causing zeros versus non-zeros, the other process is explaining the non-zero counts. This is expressed with the hurdle in the *circle*; you have to cross it to get non-zero counts. The model does not make a distinction between the different types of zeros

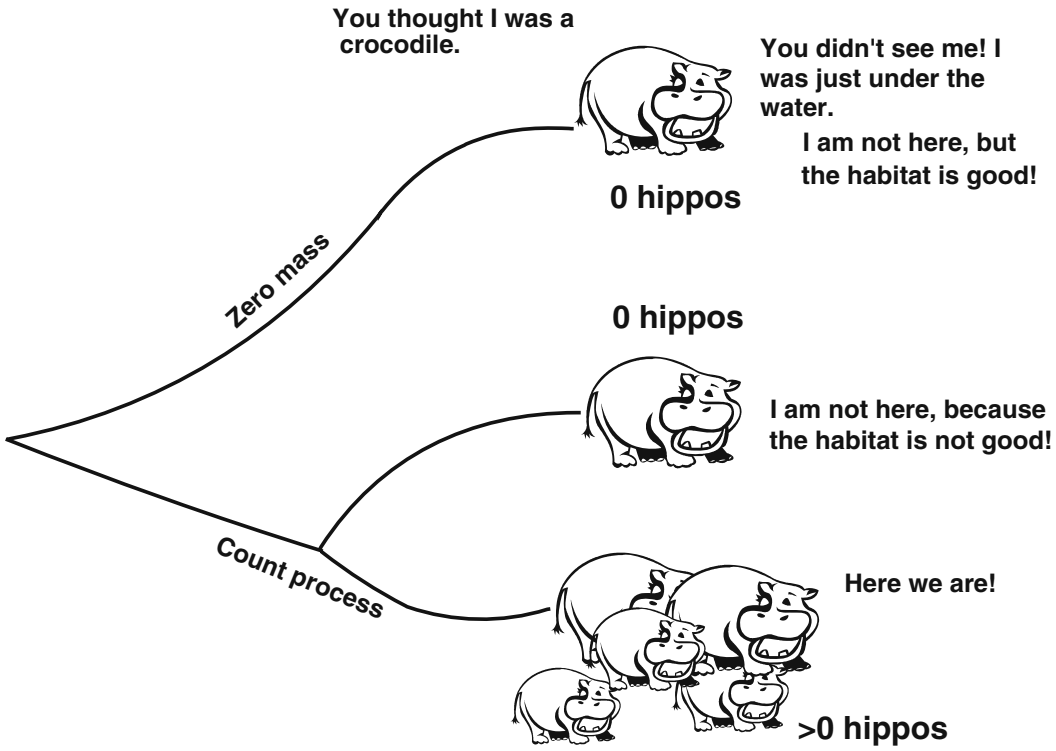

Fig. 11.5 Sketch of the underlying principle of mixture models (ZIP and ZINB). In counting hippos at sites, one can measure a zero because the habitat is not good (the hippos don't like the covariates), or due to poor experimental design and inexperienced observers (or experienced observers but difficult to observe species)

Summarising, the fundamental difference between mixture and two-part models is how the zeros are modelled. Or formulated differently, how do you want to label the zeros in the data? There are many papers where selection criteria (for example, Akaike Information Criterion (AIC), Bayesian Information Criterion (BIC), and estimated parameters) are obtained from Poisson, quasi-Poisson, NB, ZIP, ZINB, ZAP, and ZANB GLMs, and the model with the lowest value is deemed as 'the best' model. We do this later in this chapter, but it is perhaps better to choose between the latter four models based on biological knowledge.

It should be noted that labelling the different types of zeros and classifying them into two groups, false and true zeros, is useful for the ecological interpretation, but the bottom line is that in a mixture model, some of the zeros are modelled with the covariates that are also used for the positive count data, and all extra zeros are part of the zeros in the binomial model. For this process to work, it is unnecessary to split the data into true zeros and false zeros.

11.4 ZIP and ZINB Models

We follow the same approach as in Section 11.2; first we discuss the maximum likelihood for the ZIP and ZINB models in Section 11.4.1 and provide an example and R code in Section 11.4.2. If you are not interested in the underlying mathematics, just read the summary at the end of Section 11.4.1, and continue with the example.

11.4.1 Mathematics of the ZIP and ZINB

Let us return to the hippo example in Fig. 11.5 and focus on the question: What is the probability that you have zero counts? Let $\Pr(Y_i)$ be the probability that at site i, we measure a hippo. The answer to the question is

$$\begin{aligned}\Pr(Y_i = 0) = {} & \Pr(\text{False zeros}) + (1 - \Pr(\text{False zeros})) \\ & \times \Pr(\text{Count process gives a zero})\end{aligned} \tag{11.9}$$

The component Pr(False zeros) is the upper part of the graph in Fig. 11.5. The second component comes from the probability that it is not a false zero multiplied by the probability that it is a true zero. Basically, we divide the data in two imaginary groups; the first group contains only zeros (the false zeros). This group is also called the observations with zero mass. The second group is the count data, which may produce zeros (true zeros) and as well as values larger than zero. Note that we are not actively splitting the data in two groups; it is just an *assumption* that we have these two groups. We do not know which of the observations with zeros belong to a specific group. All that we know is that the non-zeros (the counts) are in group 2.

Things like 'probability of false zero', and 1 minus this probability indicates a binomial distribution, and indeed, this is what we will do. We assume that the

probability that Y_i is a false zero is binomially distributed with probability π_i, and therefore, we automatically have the probability that Y_i is not a false zero is equal to $1 - \pi_i$. Using this assumption, we can rewrite Equation (11.9):

$$\Pr(Y_i = 0) = \pi_i + (1 - \pi_i) \times \Pr(\text{Count process at site } i \text{ gives a zero}) \quad (11.10)$$

So, what do we do with the term Pr(Count process gives a zero)? We assume that the counts follow a Poisson, negative binomial, or geometric distribution. And this is the difference between zero-inflated *Poisson* and zero-inflated *negative binomial* models. Because the geometric distribution is a special case of the NB, it does not have a special name like ZIP or ZINB.

Let us assume for simplicity that the count Y_i follows a Poisson distribution with expectation μ_i. We have already seen its probability function a couple of times, but just to remind you

$$f(y_i; \mu_i | y_i \geq 0) = \frac{\mu^{y_i} \times e^{-\mu_i}}{y_i!} \quad (11.11)$$

In Section 11.2, we showed that for a Poisson distribution, the term Pr(Count process gives a zero) is given by

$$f(y_i = 0; \mu_i | y_i \geq 0) = \frac{\mu^0 \times e^{-\mu_i}}{0!} = e^{-\mu_i} \quad (11.12)$$

Hence, Equation (11.10) can now be written as

$$\Pr(y_i = 0) = \pi_i + (1 - \pi_i) \times e^{-\mu_i} \quad (11.13)$$

The probability that we measure a 0 is equal to the probability of a false zero, plus the probability that it is not a false zero multiplied with the probability that we measure a true zero.

This was the probability that $Y_i = 0$. Let us now discuss the probability that Y_i is a non-zero count. This is given by

$$\Pr(Y_i = y_i) = (1 - \Pr(\text{False zero})) \times \Pr(\text{Count process}) \quad (11.14)$$

Because we assumed a binomial distribution for the binary part of the data (false zeros versus all other types of data) and a Poisson distribution for the count data, we can write Equation (11.14) as follows:

$$\Pr(Y_i = y_i | y_i > 0) = (1 - \pi_i) \times \frac{\mu^{y_i} \times e^{-\mu_i}}{y_i!} \quad (11.15)$$

Hence, we have the following probability distribution for a ZIP model.

$$\begin{aligned} \Pr(y_i = 0) &= \pi_i + (1 - \pi_i) \times e^{-\mu_i} \\ \Pr(Y_i = y_i | y_i > 0) &= (1 - \pi_i) \times \frac{\mu^{y_i} \times e^{-\mu_i}}{y_i!} \end{aligned} \tag{11.16}$$

The notation Pr() stands for probability; it is probably better to use the notation in terms of probability functions f:

$$\begin{aligned} f(y_i = 0) &= \pi_i + (1 - \pi_i) \times e^{-\mu_i} \\ f(y_i | y_i > 0) &= (1 - \pi_i) \times \frac{\mu^{y_i} \times e^{-\mu_i}}{y_i!} \end{aligned} \tag{11.17}$$

The last step we need is to introduce covariates. Just as in Poisson GLM, we model the mean μ_i of the positive count data as

$$\mu_i = e^{\alpha + \beta_1 \times X_{i1} + \cdots + \beta_q \times X_{iq}} \tag{11.18}$$

Hence, covariates are used to model the positive counts. What about the probability of having a false zero, π_i? The easiest approach is to use a logistic regression with an intercept:

$$\pi_i = \frac{e^{\nu}}{1 + e^{\nu}} \tag{11.19}$$

where ν is an intercept. But, what if the process generating false zeros depends on covariates? Nothing stops us from including covariates that model the probability of false zeros:

$$\pi_i = \frac{e^{\nu + \gamma_1 \times Z_{i1} + \cdots \gamma_q \times Z_{iq}}}{1 + e^{\nu + \gamma_1 \times Z_{i1} + \cdots \gamma_q \times Z_{iq}}} \tag{11.20}$$

We used the symbol Z for the covariates as these may be different to the covariates that influence the positive counts. γs are regression coefficients.

We are now back on familiar territory; we have a probability function in Equation (11.17), and we have unknown regression parameters α, β_1, …, β_q, ν, γ_1, …, γ_q. It is now a matter of formulating the likelihood equation based on the probability functions in Equation (11.17); take the logarithm, get derivatives, set them to zero, and use a very good optimisation routine to get parameter estimates and standard errors. We do not present all the mathematics here, instead see p. 126 in Cameron and Trivedi (1998) or p. 174 in Hilbe (2007).

The only difference between a ZIP and ZINB is that the Poisson distribution for the count data is replaced by the negative binomial distribution. This allows for overdispersion from the non-zero counts. The probability functions of a ZINB are simple modifications of the ones from the ZIP:

$$\begin{aligned} f(y_i = 0) &= \pi_i + (1 - \pi_i) \times \left(\frac{k}{\mu_i + k} \right)^k \\ f(y_i | y_i > 0) &= (1 - \pi_i) \times f_{NB}(y) \end{aligned} \tag{11.21}$$

The function $f_{NB}(y)$ is given in Equation (11.6).

11.4.1.1 The Mean and the Variance in ZIP and ZINB Models

Before giving an example, we need to discuss what the expected mean and variance of a ZIP and ZINB model are. In a Poisson GLM, we have $\mathrm{E}(Y_i) = \mu_i$ and $\mathrm{var}(Y_i) = \mu_i$, whereas in an NB GLM we have $\mathrm{E}(Y_i) = \mu_i$ and $\mathrm{var}(Y_i) = \mu_i + \mu_i{}^2/k$. In ZIP and ZINB, this is slightly different due to the definition of the probability functions in Equations (11.17) and (11.21). To derive these means and variances, we need a couple of basic rules:

1. $\mathrm{E}(Y) = \Sigma\ y \times f(y)$. The summation is over $y = 0, 1, 2, 3$, etc. The function f is either the Poisson probability function in Equation (11.11) or the NB from Equation (11.6).
2. $\mathrm{var}(Y) = \mathrm{E}(Y^2) - \mathrm{E}(Y)^2$.
3. $\Gamma(y + 1) = y\,\Gamma(y)$.

Using these rules and a bit of basic mathematics (and a lot of paper), we obtain the following expressions for the mean and variance of a ZIP.

$$\begin{aligned} E(Y_i) &= \mu_i \times (1 - \pi_i) \\ \mathrm{var}(Y_i) &= (1 - \pi_i) \times (\mu_i + \pi_i \times \mu_i^2) \end{aligned} \tag{11.22}$$

You can find these also on p. 126 in Cameron and Trivedi (1998). If the probability of false zeros is zero, that is $\pi_i = 0$, we obtain the mean and variance equations from the Poisson GLM. If $\pi_i > 0$, then the variance is larger than the mean; hence, excessive number of (false) zeros causes overdispersion!

The equations for the ZINB follow the same steps (and are a bit more tedious to obtain) and are as follows.

$$\begin{aligned} E(Y_i) &= \mu_i \times (1 - \pi_i) \\ \mathrm{var}(Y_i) &= (1 - \pi_i) \times (\mu_i + \frac{\mu_i^2}{k}) + \mu_i^2 \times (\pi_i^2 + \pi_i) \end{aligned} \tag{11.23}$$

If the probability of false zeros is 0, we obtain the mean and variance of the NB GLM. Now that we have expressions for the mean and variances of ZIP and ZINB models, we can calculate Pearson residuals:

$$\text{Pearson residual}_i = \frac{Y_i - (1 - \pi_i) \times \mu_i}{\sqrt{\mathrm{var}(Y_i)}}$$

Depending whether a ZIP or ZINB is used, substitute the appropriate variance. μ_i and π_i are given by Equations (11.18) and (11.20), respectively.

11.4.1.2 Summary

If you skipped the mathematics above, here is a short summary. We started asking ourselves how you can measure zero hippos. This is because we can measure either false zeros or true zeros. We then defined π_i as the probability that we measure a

false zero at site i, and for the count data we assumed a Poisson distribution with mean μ_i. This leads to a statement of the form: The probability that we measure 0 hippos is given by the probability that we measure a false zero plus the probability that we do not measure a false zero multiplied with the probability that we measure a true zero. In the same way we can specify the probability that we measure a non-zero count: The probability that we do not measure a false zero multiplied with the probability of the count. Now fill in the distributions, and we get Equation (11.17). The mean values μ_i and π_i can be modelled in terms of covariates. For example, the average number of hippos at site i may depend on the availability of food, and the probability of counting a false zero (false zero) may be because the observer needs better glasses (use observer experience as covariate to model π_i). The rest is a matter of formulating and optimising a maximum likelihood equation, which follows the type of equations we saw in earlier sections and chapters.

It is important to realise that our count process, as modelled by a Poisson process can produce zeros.

11.4.2 Example of ZIP and ZINB Models

We now show an application of ZIP and ZINB models using the cod parasite data. Recall that the choice between a ZIP and ZINB depends whether there is overdispersion in the count data. So, if you apply a ZIP, and there is still overdispersion, just apply the ZINB. We use the `pscl` package (Zeileis et al., 2008) for inflated models.

In Chapter 10, we applied a binomial model for the cod parasite data. However, the numbers of parasites were also measured, and this is a count. The following code loads the data, defines the nominal variables, and removes the missing values (which are present in the response variable). Removing missing values is not really necessary, but it makes the R code for model validation easier, especially when plotting residuals versus the original explanatory variables.

```
> library(AED); data(ParasiteCod)
> ParasiteCod$fArea <- factor(ParasiteCod$Area)
> ParasiteCod$fYear <- factor(ParasiteCod$Year)
> I1 <- is.na(ParasiteCod$Intensity) |
        is.na(ParasiteCod$fArea) |
        is.na(ParasiteCod$fYear) |
        is.na(ParasiteCod$Length)
> ParasiteCod2 <- ParasiteCod[!I1, ]
> plot(table(ParasiteCod2$Intensity),
      ylab = "Frequencies",
      xlab = "Observed intensity values") #Fig. 11.1
```

The `pscl` package is reasonably new, and we are using version 0.92. The function `zeroinfl` applies a zero-inflated model, and the required R code is as follows.

```
> library(pscl)
> f1 <- formula(Intensity ~ fArea*fYear +
                Length | fArea * fYear + Length)
> Zip1 <- zeroinfl(f1, dist = "poisson",
                link = "logit", data = ParasiteCod2)
```

We could also have typed `zeroinfl(f1)` as we used default settings for the `dist` and `link` options. The `dist` option specifies the distribution for the count data, and the available choices are Poisson, negative binomial, and geometric. The `link = logit` option specifies the logistic link for the false zeros versus the non-false zeros (the true zeros plus the positive counts). But the distribution will always be a binomial. Now let us focus on the more difficult bit, the formula `f1`. The function `zeroinfl` allows the following formulae specifications.

1. $Y \sim X_1 + X_2$. This is equivalent to: Y $\sim X_1 + X_2 \mid 1$.
2. $Y \sim X_1 + X_2 \mid X_1 + X_2$
3. $Y \sim X_1 + X_2 \mid Z_1 + Z_2$

The first option specifies the following link functions for the count data and the binomial data:

$$\mu_i = e^{\alpha+\beta_1\times X_{i1}+\beta_2\times X_{i2}} \quad \text{and} \quad \pi_i = \frac{e^{\nu}}{1+e^{\nu}}$$

The mean μ_i for the Poisson count data is modelled in terms of the covariates X_1 and X_2 and the probability π_i for the binomial distribution with a constant. If you think, purely based on biology, that the probability of false zeros is also a function of X_1 and X_2, then use the second option:

$$\mu_i = e^{\alpha+\beta_1\times X_{i1}+\beta_2\times X_{i2}} \quad \text{and} \quad \pi_i = \frac{e^{\nu+\gamma_1\times X_{i1}+\gamma_2\times X_2}}{1+e^{\nu+\gamma_1\times X_{i1}+\gamma_2\times X_{i2}}}$$

If you want to model the probability of false zeros with a different set of covariates, say Z_1 and Z_2, then go for option 3, and use

$$\mu_i = e^{\alpha+\beta_1\times X_{i1}+\beta_2\times X_{i2}} \quad \text{and} \quad \pi_i = \frac{e^{\nu+\gamma_1\times Z_{i1}+\gamma_q\times Z_{i2}}}{1+e^{\nu+\gamma_1\times Z_{i1}+\gamma_q\times Z_{i2}}}$$

In this model, the count process is modelled with a different set of covariates compared to the process generating the false zeros. In the theory section, we explained this in terms of measuring no hippos because you forgot your glasses (Z describes the quality of the observer) and X for the count process can be habitat variables.

We went for option 2, but we show in a moment that the model in option 1 is nested in the model in option 2, which means that we can compare them with a likelihood ratio test. Let us return to our R code for the formula.

```
f1 <- formula(Intensity ~ fArea * fYear +
              Length | fArea*fYear + Length)
```

This means that the following link functions (in words) are applied.

$$\mu_i = e^{\text{Area+Year+Area}\times\text{Year+Length}} \quad \text{and} \quad \pi_i = \frac{e^{\text{Area+Year+Area}\times\text{Year+Length}}}{1 + e^{\text{Area+Year+Area}\times\text{Year+Length}}}$$

You could also copy the code inside the `formula` command directly into the `zeroinfl` command, but the code becomes rather intimidating. The `summary(Zip1)` command gives the estimated parameters, standard errors, z-values, and p-values, but these values are not presented here. The interaction term for the log-link function is significant, and the same can be said for the logistic link function. Hence, the Area $\times$ Year term seems to be important for the counts, but also for the probability of measuring false zeros. Length has no effect on the false zeros.

However, the ZIP model uses a Poisson distribution for the counts, and the ordinary Poisson GLM applied on these data already showed overdispersion. Before continuing with the model selection and validation, we need to look whether we have dealt properly with the overdispersion. Remember that the ZIP model only deals with zero inflation, not directly with overdispersion in the non-zero count data. If the overdispersion in a Poisson GLM is caused by the excessive number of zeros, then the ZIP will take care of the overdispersion, and we are finished. But if the overdispersion is not caused by the zeros, then the ZIP is not the appropriate model either! The best way to judge whether the ZIP is acceptable is to compare it with a ZINB as these models are nested.

The following code applies a ZINB, and applies a likelihood ratio test, and the output is given as well. The package `lmtest` is not part of the base installation, and you will need to download and install it.

```
> Nb1 <- zeroinfl(f1, dist = "negbin", link = "logit",
                  data = ParasiteCod2
> library(lmtest)
> lrtest(Zip1,Nb1)

Likelihood ratio test
Model 1: Intensity ~ fArea  * fYear + Length | fArea *
                     fYear  + Length
Model 2: Intensity ~ fArea  * fYear + Length | fArea *
                     fYear  + Length

  #Df  LogLik Df  Chisq Pr(>Chisq)
1  26 -6817.6
2  27 -2450.4  1 8734.2  < 2.2e-16
```

Recall from Chapter 9 that with the likelihood ratio test, we are testing whether the variance structure of the Poisson, $\text{var}(Y_i) = \mu_i$, is the same as the

variance structure of the NB, $\text{var}(Y_i) = \mu_i + \mu_i^2 / k$. For the purpose of this test, it is probably easier to use the notation $\text{var}(Y_i) = \mu_i + \alpha \times \mu_i^2$ for the NB, where $\alpha = 1/k$, because the null hypothesis (the Poisson variance equals the NB variance) can then be written as H_0: $\alpha = 0$ (note that we are testing on the boundary, but the `lrtest` function corrects for this). The results of this test provide overwhelming evidence to go for a ZINB, instead of a ZIP. The numerical output of the ZINB is obtained with the command `summary(Nb1)` and is as follows.

```
> summary(Nb1)
Call:
zeroinfl(formula = f1, data = ParasiteCod2,
         dist = "negbin", link = "logit")

Count model coefficients (negbin with log link):
                  Estimate Std. Error z value Pr(>|z|)
(Intercept)       3.724165   0.344488  10.811  < 2e-16
fArea2            0.197832   0.329187   0.601  0.54786
fArea3           -0.646241   0.277843  -2.326  0.02002
fArea4            0.709638   0.252319   2.812  0.00492
fYear2000         0.063212   0.295670   0.214  0.83071
fYear2001        -0.940197   0.605908  -1.552  0.12073
Length           -0.036246   0.005109  -7.094  1.3e-12
fArea2:fYear2000 -0.653255   0.535476  -1.220  0.22248
fArea3:fYear2000  1.024753   0.429612   2.385  0.01707
fArea4:fYear2000  0.534372   0.415038   1.288  0.19791
fArea2:fYear2001  0.967809   0.718086   1.348  0.17773
fArea3:fYear2001  1.003671   0.677373   1.482  0.13842
fArea4:fYear2001  0.855233   0.654296   1.307  0.19118
Log(theta)       -0.967198   0.096375 -10.036  < 2e-16

Zero-inflation model coefficients (binomial with logit
link):
                  Estimate Std. Error z value Pr(>|z|)
(Intercept)        0.19106    0.78312   0.244 0.807249
fArea2             2.01576    0.57396   3.512 0.000445
fArea3             1.90753    0.55093   3.462 0.000535
fArea4            -0.73641    0.86427  -0.852 0.394182
fYear2000         -1.07479    2.01183  -0.534 0.593180
fYear2001          3.29534    0.71139   4.632 3.62e-06
Length            -0.03889    0.01206  -3.226 0.001254
fArea2:fYear2000   0.46817    2.09007   0.224 0.822759
fArea3:fYear2000  -0.79393    2.16925  -0.366 0.714369
fArea4:fYear2000 -12.93002  988.60803  -0.013 0.989565
fArea2:fYear2001  -3.20920    0.83696  -3.834 0.000126
fArea3:fYear2001  -3.50640    0.83097  -4.220 2.45e-05
fArea4:fYear2001  -2.91175    1.10650  -2.631 0.008501
```

```
Theta = 0.3801
Number of iterations in BFGS optimization: 52
Log-likelihood: -2450 on 27 Df
```

The z- and p-values of the parameters for the count model (upper part of the output) are rather different, compared to the ZIP! You would expect this as there is overdispersion. The sentence with the BFGS phrase refers to the number of iterations in the optimisation routines.

The question that we should now focus on is which of the explanatory variables can be dropped from the model. The candidates are the Area × Year interaction term for the count model (most levels have high p-values) and the Area × Year interaction term for the logistic model (some levels are not significant). In fact, why don't we just drop each term in turn and select the optimal model using the likelihood ratio statistic or AIC. The options are

1. Drop length from the count model. Call this `Nb1A`.
2. Drop the Area × Year term from the count model. Call this `Nb1B`.
3. Drop length from the logistic model. Call this `Nb1C`.
4. Drop the Area × Year term from the logistic model. Call this `Nb1D`.

The models `Nb1` (without dropping anything), `Nb1A`, `Nb1B`, `Nb1C`, and `Nb1D` are given below.

$$\begin{aligned}
&\text{nb1}: && \mu_i = e^{\text{Area+Year+Area}\times\text{Year+Length}} && \pi_i = \frac{e^{\text{Area+Year+Area}\times\text{Year+Length}}}{1+e^{\text{Area+Year+Area}\times\text{Year+Length}}}\\
&\text{nb1A}: && \mu_i = e^{\text{Area+Year+Area}\times\text{Year}} && \pi_i = \frac{e^{\text{Area+Year+Area}\times\text{Year+Length}}}{1+e^{\text{Area+Year+Area}\times\text{Year+Length}}}\\
&\text{nb1B}: && \mu_i = e^{\text{Area+Year+Length}} && \pi_i = \frac{e^{\text{Area+Year+Area}\times\text{Year+Length}}}{1+e^{\text{Area+Year+Area}\times\text{Year+Length}}}\\
&\text{nb1C}: && \mu_i = e^{\text{Area+Year+Area}\times\text{Year+Length}} && \pi_i = \frac{e^{\text{Area+Year+Area}\times\text{Year}}}{1+e^{\text{Area+Year+Area}\times\text{Year}}}\\
&\text{nb1D}: && \mu_i = e^{\text{Area+Year+Area}\times\text{Year+Length}} && \pi_i = \frac{e^{\text{Area+Year+Length}}}{1+e^{\text{Area+Year+Length}}}
\end{aligned}$$

You can implement these models with the code

```
> #Drop Length from count model
> f1A <-formula(Intensity ~ fArea * fYear |
                fArea * fYear + Length)
> #Drop interaction from count model
> f1B <-formula(Intensity ~ fArea + fYear+
                Length | fArea * fYear + Length)
> #Drop Length from binomial model
> f1C<-formula(Intensity ~ fArea * fYear+
                Length | fArea * fYear)
> #Drop interaction from binomial model
> f1D<-formula(Intensity ~ fArea * fYear+
                Length | fArea + fYear + Length)
```

```
> Nb1A <- zeroinfl(f1A, dist = "negbin",
                   link = "logit", data = ParasiteCod2)
> Nb1B <- zeroinfl(f1B, dist = "negbin",
                   link = "logit", data = ParasiteCod2)
> Nb1C <- zeroinfl(f1C, dist = "negbin",
                   link = "logit", data = ParasiteCod2)
> Nb1D <- zeroinfl(f1D, dist = "negbin",
                   link = "logit", data = ParasiteCod2)
```

Just as we did in Chapters 4, 5, and 6, we use the likelihood ratio test to compare each nested model `Nb1A`, `Nb1B`, `Nb1C`, and `Nb1D` with the full model `Nb1`, and if a term is not significant, drop the least significant one. The required code is

```
> lrtest(Nb1,Nb1A); lrtest(Nb1,Nb1B)
> lrtest(Nb1,Nb1C); lrtest(Nb1,Nb1D)
```

Table 11.2 shows the results. The AIC values were obtained with the command `AIC(Nb1A,Nb1B,Nb1C,Nb1D)`. The model, in which the Area × Year interaction was dropped from the count data model gave the lowest AIC and an associated p-value of 0.026; so we might as well drop it. These tests are approximate, and therefore, $p = 0.026$ is not convincing. The AICs of the model with and without the Area × Year interaction are also similar.

This means that we continue with the model selection procedure and test whether Length, Area, or Year can be dropped from the count model and length and the Area × Year interaction from the logistic model. Results are not shown here, but no further terms could be dropped. This means that we can now say: 'Thank you for producing the numerical output from the first ZINB model, but it is not the information we need'. The parameters of the optimal model are given by

```
> summary(Nb1B)
Call:
zeroinfl(formula = f1B, data = ParasiteCod2,
         dist = "negbin", link = "logit")

Count model coefficients (negbin with log link):
             Estimate Std. Error z value Pr(>|z|)
(Intercept)  3.497280   0.326888  10.699  < 2e-16
fArea2       0.254160   0.229988   1.105  0.26912
fArea3      -0.200901   0.205542  -0.977  0.32836
fArea4       0.912450   0.195039   4.678 2.89e-06
fYear2000    0.462204   0.173067   2.671  0.00757
fYear2001   -0.128524   0.166784  -0.771  0.44094
Length      -0.034828   0.004963  -7.017 2.27e-12
Log(theta)  -0.985648   0.095385 -10.333  < 2e-16
```

Table 11.2 Results of the model selection in ZINB

Dropped term	df	AIC	Likelihood ratio test	
None	27	4954.897		
Length from μ_i	26	4994.993	$X^2 = 42.096$	($df = 1, p < 0.001$)
Area $\times$ Year from μ_i	21	4957.146	$X^2 = 14.249$	($df = 6, p = 0.026$)
Length from π_i	26	4965.019	$X^2 = 12.122$	($df = 1, p < 0.001$)
Area $\times$ Year from π_i	21	4961.751	$X^2 = 18.853$	($df = 6, p = 0.004$)

```
Zero-inflation model coefficients (binomial with logit
link):
                  Estimate Std. Error z value Pr(>|z|)
(Intercept)       -0.16057    0.85842  -0.187 0.851617
fArea2             2.18198    0.65106   3.351 0.000804
fArea3             2.23765    0.61803   3.621 0.000294
fArea4            -0.50954    0.90067  -0.566 0.571570
fYear2000         -0.60158    1.55344  -0.387 0.698564
fYear2001          3.71075    0.72278   5.134 2.84e-07
Length            -0.03588    0.01150  -3.121 0.001801
fArea2:fYear2000   0.40925    1.61583   0.253 0.800055
fArea3:fYear2000  -1.81000    1.83708  -0.985 0.324495
fArea4:fYear2000 -10.94642  285.39099  -0.038 0.969404
fArea2:fYear2001  -3.71145    0.84033  -4.417 1.00e-05
fArea3:fYear2001  -3.99409    0.81410  -4.906 9.29e-07
fArea4:fYear2001  -3.37317    1.09981  -3.067 0.002162

Theta = 0.3732
Number of iterations in BFGS optimization: 45
Log-likelihood: -2458 on 21 Df
```

For publication, you should also give one p-value for the Area and Year terms in the count model, and one p-value for the interaction term in the logistic model. Just drop these terms in turn, use the likelihood ratio test, and quote the Chi-square statistic, degrees of freedom and a p-value. If you are not 100% sure, here are our results for the count model: Length ($X^2 = 41.604$, df $= 1$, $p < 0.001$), Year ($X^2 = 12.553$, df $= 2$, $p = 0.002$), Area ($X^2 = 47.599$, df $= 3$, $p < 0.001$), and for the logistic model: length ($X^2 = 10.155$, df $= 1$, $p = 0.001$) and the Area $\times$ Year interaction ($X^2 = 47.286$, df $= 6$, $p < 0.001$).

This was the model selection. There are two more things we need to do; model validation and model interpretation of the optimal ZINB model.

11.4.2.1 Model Validation

The keyword is again residuals. You need to plot Pearson residuals against the fitted values and Pearson residuals against each explanatory variable and you should

not see any pattern. It is also useful to plot the original data versus the fitted data; hopefully, they form a straight line.

If you fit a ZIP model with the function `zeroinfl`, Pearson residuals for the count data can be obtained by the R command:

```
> EP <- residuals(Zip1, type = "pearson").
```

There are multiple packages for zero-inflated data, and it is not always clear how exactly residuals are calculated. Because we believe in 'know what you are doing', we show you how to get the Pearson residuals using the equations we derived in the previous subsection.

Let us extract the probabilities π_i, the probability of a false zero. They are obtained by

```
> EstPar <- coef(Nb1B,model = "zero")
> Z <- model.matrix(Nb1B,model = "zero")
> g <- Z %*% EstPar
> p <- exp(g) / (1 + exp(g))
```

The p in the code is π_i. The `coeff` command with the option `model = "zero"` gives the estimated parameters presented above (`Nb1B` is our optimal ZINB model). The μ_i from Equation (11.18) is obtained by

```
> EstPar2 <- coef(Nb1B, model = "count")
> X <- model.matrix(Nb1B, model = "count")
> g <- X %*% EstPar2
> mu1 <- exp(g)
```

Using Equation (11.23), the expected values of a ZINB model are given by

```
> mu <- (1 - p) * mu1
```

If you compare this result with the results of `fitted(Nb1B)` or `predict (Nb1B)`, you should get the same values. Finally, we show how to get the variance and Pearson residuals:

```
> k <- Nb1B$theta
> VarY <- ((mu^2) / k + mu)*(1 - p) +
            (mu^2)*(p^2 + p)
> EP <- (ParasiteCod2$Intensity - mu) / sqrt(VarY)
```

These should give the same results as the `residuals` command; but it is good to know that we can do it ourselves! The rest is a matter of plotting these residuals against everything we have and hope that there are no clear patterns. We do not show these graphs here.

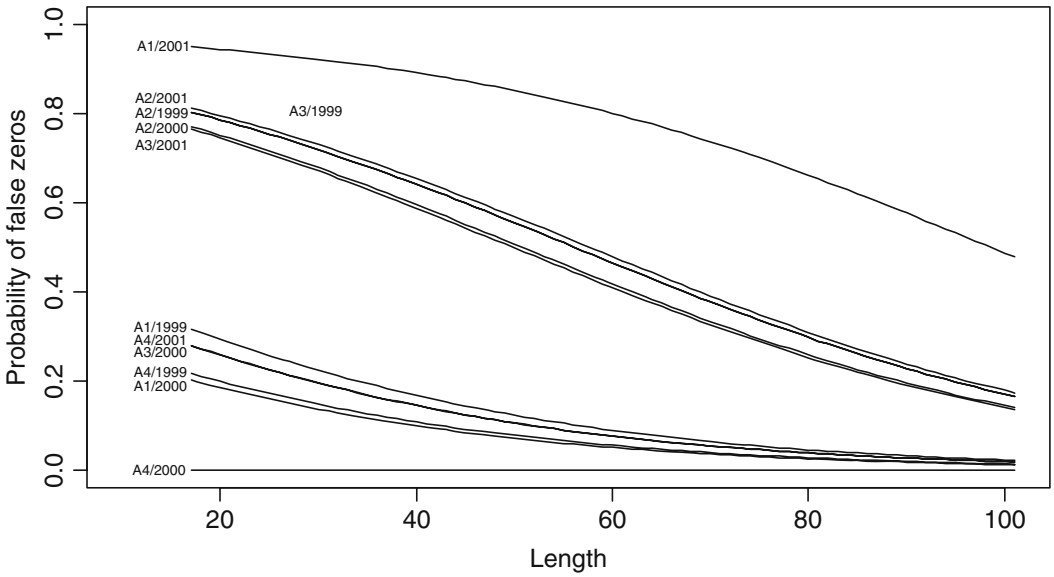

Fig. 11.6 *Fitted curves* for the logistic regression model. The vertical axis shows the probability of measuring a false zero, and the horizontal axis length of cod. Each line corresponds to an area and year combination

11.4.2.2 Model Interpretation

The question we now focus on is: What does it all mean? To answer this question, we sketch the results of the model. There are two components to plot; the logistic model for the false zeros versus all other data, and the count model versus all other data. We first focus on the logistic regression part. Fitted values can be obtained by the `predict` function, or you can do it manually (which is what we did). We took the estimated intercepts and slopes from the zero-inflated part of the optimal ZINB model (`Nb1B`), created length values from 17 to 100 cm, and calculated the fitted values for each area and year combination. This is a straightforward exercise and was explained in Chapter 10. The results are given in Fig. 11.6. It seems that the highest probabilities of false zeros are obtained for small fish in area 1 in 2001, in area 2 in all years, and in area 3 in 1999 and 2001. Explained differently, in these areas and these years, you are likely to catch small cod with zero parasites, but these zeros are false zeros.

A similar graph was drawn for the count data. In this case, fitted values are obtained from Equation (11.23). Regression coefficients were taken from the upper part of the `summary(Nb1B)` output. Area 4 in 1999 and 2000 has the highest values. This information can also be derived from the estimated regression parameters; so the need for a graph is limited.

11.5 ZAP and ZANB Models, Alias Hurdle Models

In the previous section, we assumed the zeros for the cod data consist of false zeros and true zeros. In this section, we do not discriminate between the four types of zeros; they are just zeros.

We follow the same approach as in Section 11.2; first we present the probability functions for the two-part models and give the maximum likelihood equations in Section 11.5.1, and an example plus the R code is presented in Section 11.5.2. If you are not interested in the underlying mathematics, just read the summary at the end of Section 11.5.1.

11.5.1 Mathematics of the ZAP and ZANB

In the hurdle model (ZAP and ZANB), we consider the data on a presence and absence level and analyse the presence data with a count model. Actually, if you apply two analyses, one binomial GLM and one Poisson (or NB) GLM, you will get the same estimated regression parameters.

A small difference is that with ZIP and ZINB, the binomial GLM models the probability of a false zero versus other types of data, whereas in ZAP and ZANB, the binomial GLM models the probability of presence versus absence. Hence, the estimated regression parameters obtained by ZAP and ZANB should have opposite signs compared to those obtained by ZIP and ZINB due to the definition of π.

The underlying idea for the hurdle model is that there are two ecological processes playing a role. In the context of the hippo example, one process is causing the absence of hippos, and at those sites where hippos are present, there is a second process influencing the number of hippos. The probability function for a hurdle model is build up accordingly. The binomial distribution is used to model the absence and presence of hippos, and a Poisson (or negative binomial or geometric) distribution for the counts. This leads to the following probability function for the Poisson ZAP:

$$f_{\text{ZAP}}(y;\beta,\gamma) = \begin{cases} f_{\text{binomial}}(y=0;\gamma) & y=0 \\ (1-f_{\text{binomial}}(y=0;\gamma)) \times \dfrac{f_{\text{Poisson}}(y;\beta)}{1-f_{\text{Poisson}}(y=0;\beta)} & y>0 \end{cases} \tag{11.24}$$

So, the probability of measuring zero hippos is modelled with a binomial distribution, where π_i is the probability that $y_i = 0$. Hence, $1 - \pi_i$ is the probability that we do not measure zero hippos. Just as for the ZIP model, π_i is modelled in terms of covariates Z and regression parameters γ; see also Equation (11.20). To measure a non-zero count, the ecosystem needs to cross a hurdle to produce a non-zero value *and* the Poisson count process has to exclude the probability of zero values, which we called a zero-truncated Poisson distribution in Section 11.2. So, the second part in the above equation says that the probability of measuring a non-zero value equals the probability that it is not a zero multiplied with the probability determined by a zero-truncated Poisson. The mean of the Poisson distribution, μ_i, is modelled in terms of covariates X and regression parameters β; see also Equation (11.18).

The next task is to find the optimal regression parameters γ and β. As with the ZIP, a likelihood criterion is formulated using the probability function in Equation (11.24). Finding the regression parameters that optimise the log-likelihood is a matter of numerical optimisation, and the required formulae can be found in

Section 4.7.1 in Cameron and Trivedi (1998). The function `hurdle` from the `pscl` package in R will do the hard work for you.

The difference between a ZAP and a ZANB is due to the assumption for the distribution of the count data. If we assume a Poisson distribution, we end up with a ZAP and if a negative binomial distribution is used, we get a ZANB. Justification for the ZANB is extra overdispersion in the count data.

In Equations (11.22) and (11.23), we formulated the mean and variance for the ZIP and ZINB. For the ZAP, these are as follows.

$$E_{\text{ZAP}}(Y_i;\pi_i,\mu_i) = \frac{1-\pi_i}{1-e^{-\mu_i}} \times \mu_i$$

$$\text{Var}_{\text{ZAP}}(Y_i;\pi_i,\mu_i) = \frac{1-\pi_i}{1-e^{-\mu_i}} \times (\mu_i + \mu_i^2) - \left(\frac{1-\pi_i}{1-e^{-\mu_i}} \times \mu_i\right)^2$$

And for the ZANB, we have

$$E_{\text{ZANB}}(Y_i;\pi_i,\mu_i,k) = \frac{1-\pi_i}{1-P_0} \times \mu_i \quad \text{where} P_0 = \left(\frac{k}{\mu_i + k}\right)^k$$

$$\text{Var}_{\text{ZANB}}(Y_i;\pi_i,\mu_i,k) = \frac{1-\pi_i}{1-P_0} \times \left(\mu_i^2 + \mu_i + \frac{\mu_i^2}{k}\right) - \left(\frac{1-\pi_i}{1-P_0} \times \mu_i\right)^2$$

The mean and variance can be used to calculate the Pearson residuals.

11.5.2 Example of ZAP and ZANB

The whole modelling process in two-part models is identical compared to mixture models. First you need to decide whether you need a ZAP or ZANB. The best option is to run them both and compare them with a likelihood ratio test. This can be done with the following R code.

```
> H1A <- hurdle(f1, dist = "poisson", link = "logit",
                data = ParasiteCod2)
> H1B <- hurdle(f1, dist = "negbin", link = "logit",
                data = ParasiteCod2)
```

The command `lrtest(H1A,H1B)` produces a Chi-square statistic of 8752.50 (which is overwhelming evidence in favour of the negative binomial model) and the command `AIC(H1A, H1B)`, gives an AIC of 13687.59 for the ZAP and 4939.08 for the ZANB, confirming the choice for the ZANB. The `summary(H1B)` gives the estimated parameters, but because the model has various nominal variables with multiple levels, it is better to compare the full model `H1B`, with models in which a particular term is dropped, and use the `lrtest` command to get one p-value for the interaction term in the count model and in the binomial model. R code for these analyses were provided in Section 11.4 and are not repeated here (the code

can also be found on the book's website). In the first round of model simplification, length was dropped from the binomial model, and in the second (and last) round, the Area × Year interaction term was dropped from the Poisson model. The code and estimated regression parameters for the optimal ZANB model are as follows.

```
> fFinal <- formula(Intensity ~ fArea + fYear +
                Length | fArea*fYear )
> HFinal <- hurdle(f1, dist = "negbin", link = "logit",
                   data = ParasiteCod2)
> summary(HFinal)

Call:
hurdle(formula = f1, data = ParasiteCod2,
       dist = "negbin", link = "logit")
Count model coefficients (truncated negbin with log
link):
                  Estimate Std. Error z value Pr(>|z|)
(Intercept)       3.366059   0.399420   8.427  < 2e-16
fArea2            0.379211   0.380945   0.995  0.31952
fArea3           -0.504376   0.312256  -1.615  0.10625
fArea4            0.893614   0.291517   3.065  0.00217
fYear2000        -0.040511   0.328434  -0.123  0.90183
fYear2001        -0.757718   0.688097  -1.101  0.27082
Length           -0.037309   0.005867  -6.359 2.03e-10
fArea2:fYear2000 -0.639059   0.616450  -1.037  0.29989
fArea3:fYear2000  1.193440   0.494530   2.413  0.01581
fArea4:fYear2000  0.510433   0.476990   1.070  0.28457
fArea2:fYear2001  0.707730   0.819333   0.864  0.38770
fArea3:fYear2001  0.912374   0.775776   1.176  0.23956
fArea4:fYear2001  0.601263   0.746292   0.806  0.42043
Log(theta)       -1.498146   0.239114  -6.265 3.72e-10

Zero hurdle model coefficients (binomial with logit
link):
                  Estimate Std. Error z value Pr(>|z|)
(Intercept)       0.085255   0.295071   0.289   0.7726
fArea2           -1.321373   0.285258  -4.632 3.62e-06
fArea3           -1.449183   0.243885  -5.942 2.81e-09
fArea4            0.300728   0.271105   1.109   0.2673
fYear2000         0.395069   0.343817   1.149   0.2505
fYear2001        -2.652010   0.433404  -6.119 9.42e-10
Length            0.006933   0.004655   1.489   0.1364
fArea2:fYear2000 -0.080344   0.507970  -0.158   0.8743
fArea3:fYear2000  0.870585   0.450277   1.933   0.0532
fArea4:fYear2000  0.864622   0.592387   1.460   0.1444
```

```
fArea2:fYear2001  2.737488   0.532903   5.137 2.79e-07
fArea3:fYear2001  2.718986   0.499487   5.444 5.22e-08
fArea4:fYear2001  2.541437   0.518245   4.904 9.39e-07
Theta: count = 0.2235
Number of iterations in BFGS optimization: 29
Log-likelihood: -2442 on 28 Df
```

The difference between the optimal ZINB and ZANB is that length is not significant in the binomial part of the ZANB. For the rest, both models are the same in terms of selected explanatory variables.

It is also interesting to compare the estimated parameters of the optimal ZINB and ZANB models. For the count part of the model, the sign and magnitude of the significant parameters are very similar. Plotting the fitted values as in Fig. 11.7 gives a similar graph. Hence, the biological conclusions for the count part are similar. For the binomial part of the model, things look different, at least in the first instance. However, the p-values of corresponding terms in both tables give the same message. The magnitudes of the significant parameters are similar as well. It is only the sign of the regression parameters that are different. But this is due to the opposite definition of the πs in both methods!

In summary, for the cod parasite data, the ZINB and ZANB give similar parameter estimates. The difference is how they treat the zeros. The ZINB labels the excessive number of zeros (which occur at small fish and in certain areas in particular years) as false zeros, whereas the ZANB models the zeros versus the non-zeros (and identifies the area $\times$ year interaction as a driving factor for this), and the non-zeros with a truncated NB GLM jointly.

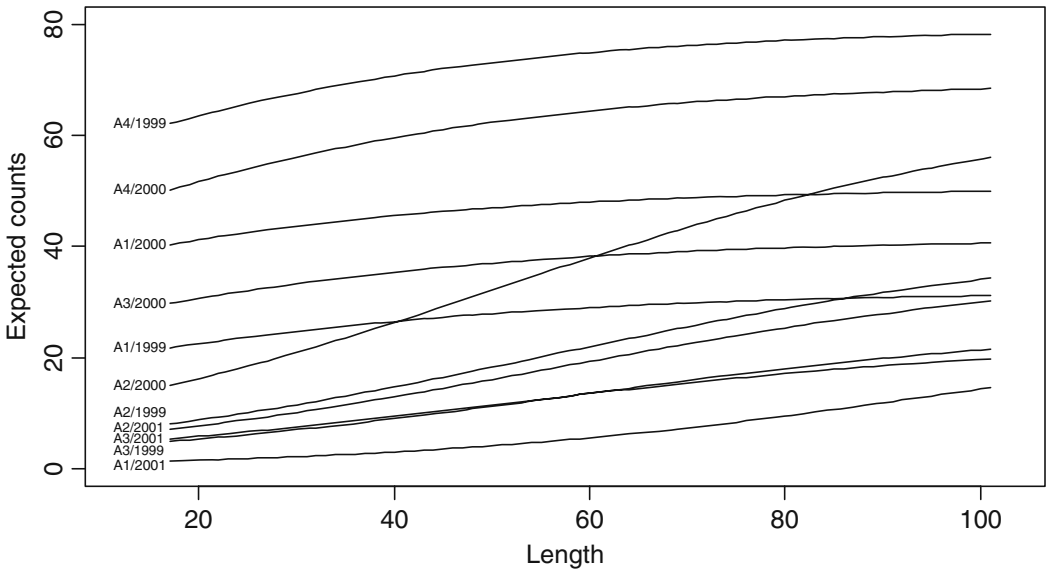

Fig. 11.7 *Fitted curves* for the count model. The vertical axis shows the expected counts (assuming a ZINB distribution) and the horizontal axis length of cod. Each line corresponds to an area and year combination

11.6 Comparing Poisson, Quasi-Poisson, NB, ZIP, ZINB, ZAP and ZANB GLMs

In the previous sections and chapters, we applied Poisson, quasi-Poisson, NB GLM, ZIP, ZINB, ZAP, and ZANB models on the cod parasite data. The question is now: What is the best model? There are many ways to answer this question.

Option 1: Common Sense

The first option is common sense. First, you should decide whether there is overdispersion. If there is no overdispersion, you are lucky and you can stick to the Poisson GLM. If there is overdispersion, ask yourself why you have overdispersion; outliers, missing covariates, or interactions are the first things you should consider. Small amounts of overdispersion can be modelled with quasi-Poisson. Let us assume that this is not the case. Do you have overdispersion due to excessive number of zeros or due more to variation in the count data? Make a frequency plot of the data and you will know whether it is zero inflation. If there is zero inflation, go the route of ZIP, ZAP, ZINB, and ZANB. If the overdispersion is not caused by excessive number of zeros, but due to more variation than expected by the Poisson distribution in the positive part of the count data, use the negative binomial distribution. In case of zero inflation *and* extra variation in the positive count data, use ZINB or ZANB. The choice between ZINB and ZANB (or ZIP and ZAP) should be based on *a priori* knowledge of the cause of your excessive number of zeros.

Option 2: Model Validation

A second option to help decide on the best model (if there is such a thing) is to plot the residuals of each model and see whether there are any residual patterns. Drop each model that shows patterns in the residuals.

Option 3: Information Criteria

Another option is to apply all methods and print all estimated parameters, standard errors and AICs, BICs, etc. in one big table. Compare them, and based on the AICs, judge which one is the best. You can find examples of this approach in most books discussing these statistical methods.

Option 4: Hypothesis Tests – Poisson Versus NB

Formal hypotheses tests can be used to choose between Poisson and negative binomial models as these are nested. This also holds for ZIP versus ZINB and ZAP versus ZANB.

Option 5: Compare Observed and Fitted Values

Potts and Elith (2006) compared the fitted and observed values of all the models. To assess how good each technique predicts the fitted values, they used various tools. For example, high values of the Pearson correlation coefficient and Spearman's rank correlation between fitted and observed values mean that these are similar.

It is also possible to apply a linear regression model of the form $\text{Observed}_i = \alpha + \beta \times \text{Fitted}_i + \varepsilon_i$, where Observed_i are the observed data and Fitted_i the fitted values of a particular method. An estimated intercept of 0 and slope of 1 indicates a perfect fit. Potts and Elith (2006) discuss the interpretation of non-significant slopes.

Other ways to quantify how similar the observed and fitted values are the root mean square errors and mean absolute error (where error is defined as the difference between the observed and fitted values).

All these statistics are discussed in Potts and Elith (2006) and require bootstrapping. We implemented their algorithm, and the results are presented in Table 11.3. Note that the Pearson correlation coefficients and the Spearman rank correlations of all five methods are nearly identical. The ZANB is the only model that gives an intercept of 0. The AIC of this model is also the lowest, and therefore based on these numerical tools, the ZANB can be selected as the best possible model.

Another approach to compare (and select) models is discussed in Ver Hoef and Boveng (2007), who plotted the variance versus the mean and the weights that are used inside the algorithm versus the mean.

Instead of sticking to one of these five methods, you may need multiple approaches to arrive at the best model. The hypothesis testing approach showed that an NB model is preferred above the Poisson GLM. A frequency plot indicated zero inflation; hence, we should apply a ZINB or ZANB. A discussion with one of the involved researchers revealed that we have both false and true zeros. We can either try to determine the contribution from each of these (with a ZINB) or just consider them as zeros and use the ZANB. So, the choice between the ZINB and ZANB depends on the underlying questions with regards to the zeros. If you close your eyes and compare the ZINB and ZANB, then the latter should be selected as judged by the AIC.

Table 11.3 Model comparison tools for the Poisson GLM, quasi-Poisson GLM, NB GLM, ZINB, and ZANB models. The Pearson correlation coefficient (r), Spearman rank correlation (p), intercept and slope (of a linear regression of observed versus fitted), RMSE, MAE (mean absolute error), AIC, log likelihood and degrees of freedom (df).

Model	r	p	Intercept	Slope	RMSE	MAE	AIC	Log lik	Df
Poisson	0.33	0.36	0.32	0.96	18.60	7.50	20377.86	–10175.93	13
Quasi-Poisson	0.33	0.36	0.32	0.96	18.63	7.50	NA	NA	13
NB GLM	0.34	0.37	–0.20	1.07	18.49	7.42	5030.67	–2501.33	14
ZINB	0.33	0.37	0.30	0.96	18.57	7.49	4954.89	–2450.44	27
ZANB	0.34	0.37	–0.06	1.04	18.47	7.47	4937.08	–2441.54	27

11.7 Flowchart and Where to Go from Here

In this chapter, we have discussed tools to analyse zero-inflated models, resulting in four extra models (ZIP, ZAP, ZINB and ZANB) in our toolbox for the analysis of count data. Mixture models and two-part models should be part of every ecologist's toolbox as zero inflation and overdispersion are commonly encountered in ecological data. If you are now confused with the large number of models to analyse count data, Fig. 11.8 will help you to visualise the difference between some of the models discussed in Chapters 9, 10, and 11.

So, where do we go from here? In Chapters 12 and 13, we concentrated on models that allow for correlation and random effects in Poisson and binomial GLMs and GAMs. These models are called generalised estimation equations (GEE), generalised linear mixed modelling (GLMM), and generalised additive mixed modelling (GAMM). At the time of writing this book, software for GEE, GLMM, and GAMM for zero-inflated data consists mainly of research or publication specific code. By this, we mean that papers using random effects or spatial and temporal correlations structures in combination with zero inflation are now being published (e.g. Ver Hoef and Jansen, 2007), but general software code is not yet available. So, a bit of challenging R programming awaits you, should you want to model zero-inflated GLMMs.

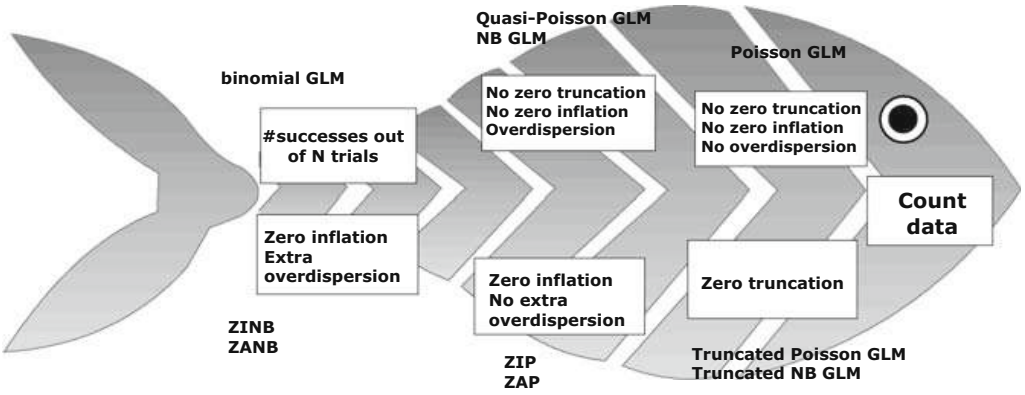

Fig. 11.8 GLMs for count data. Instead of GLM, you can also use GAM. Try sketching in the R functions for each box. If there is no zero truncation, no zero inflation and no overdispersion (*upper right box*), you can apply a Poisson GLM. If there is overdispersion (*upper middle box*), then consider quasi-Poisson or negative binomial GLM. The '#successes out of N trials' box refers to a logistic regression. The trials need to be independent and identical. For zero-truncated data (*lower right box*), you need to apply a zero-truncated Poisson GLM or a zero-truncated negative binomial GLM. If there is zero inflation, you are in the world of ZIP, ZAP, ZINB, and ZINB models. The difference between the P and NB is whether there is overdispersion in the non-zero data. It is a nice exercise to add the names of the corresponding R functions! You can also use the offset in the ZIP, ZAP, ZINB, and ZANB models