Chapter 9

Generalized Linear Mixed Models

Chapter Outline

9.1 BINOMIAL MIXED MODEL

9.1.1 Background

To illustrate the binomial mixed model we have adapted a data set used by Grüebler et al. (2010) on barn swallow *Hirundo rustica* nestling survival (we have selected a nonrandom sample to be able to fit a simple model; hence, the results do not add unbiased knowledge about the swallow biology!). For 63 swallow broods, we know the clutch size and the number of the nestlings that fledged. The broods came from 51 farms, thus some of them had more than one brood. There are three predictors measured at the level of the farm: colony size (the number of swallow broods on that farm), cow (whether there are cows on the farm or not), and dung heap (the number of dung heaps, piles of cow dung, within 500 m of the farm).

The interest was to measure how swallows profit from insects that are attracted by livestock on the farm and by dung heaps. Broods from the same farm are not independent of each other. Also, the predictor variables were measured at the level of the farm, thus they are the same for all broods from a

Bayesian Data Analysis in Ecology Using Linear Models with R, BUGS, and Stan
http://dx.doi.org/10.1016/B978-0-12-801370-0.00009-5.

farm. We have to account for that when building the model by including farm as a random factor. The outcome variable consists of two values for each observation, as seen with the binomial model without random factors (Section 8.2.2): number of successes (fledge) and number of failures (chicks that died = clutch size minus number that fledged).

The random factor "farm" adds, to the intercept in the linear predictor, a farm-specific deviation b_g. These deviations are modeled as normally distributed with mean 0 and standard deviation σ_g.

$$y_i \sim Binom(p_i, n_i)$$
$$\text{logit}(p_i) = \beta_0 + b_{g[i]} + \beta_1 \text{colonysize}_i + \beta_2 \text{I}(\text{cow}_i = 1) + \beta_3 \text{dungheap}_i$$
$$b_g \sim Norm(0, \sigma_g)$$

9.1.2 Fitting a Binomial Mixed Model in R

To fit the model in R we use the function `glmer`, which uses the Laplace approximation. The Laplace approximation is an analytic method to solve integrals, which is often used in Bayesian statistics to obtain the posterior distribution of parameters. We z-transform the covariates (subtraction of the mean and division by the standard deviation so that the transformed variable has a mean of 0 and a standard deviation of 1), which often facilitates convergence of the model. The notation for the random factor with only a random intercept is `(1|farm.f)`.

```
data(swallowfarms); dat <- swallowfarms
dat$colsize.z <- scale(dat$colsize)
dat$dung.z    <- scale(dat$dung)
dat$die       <- dat$clutch - dat$fledge
dat$farm.f    <- factor(dat$farm)
mod <- glmer(cbind(fledge, die) ~ colsize.z + cow + dung.z +
           (1|farm.f), data=dat, family=binomial)
mod
Generalized linear mixed model fit by maximum likelihood ['glmerMod']
Family: binomial ( logit )
Formula: cbind(fledge, die) ~ colsize.z + cow + dung.z + (1 | farm.f)
  Data: dat
     AIC      BIC   logLik  deviance  df.resid
282.5240 293.2397 -136.2620 272.5240        58
Random effects:
Groups Name        Std.Dev.
farm.f  (Intercept) 0.4536
Number of obs: 63, groups: farm.f, 51
Fixed Effects:
(Intercept)    colsize.z          cow      dung.z
   -0.09533      0.05087    0.39370    -0.14236
```

9.1.3 Assessing Model Assumptions

As always, we first look at the model fit before drawing inferences.

```
par(mfrow=c(2,2)) # divide the graphic window in 4 subregions

qqnorm(resid(mod)) # qq-plot of residuals
qqline(resid(mod))

qqnorm(ranef(mod)$farm.f[,1])   # qq-plot of the random effects
qqline(ranef(mod)$farm.f[,1])

plot(fitted(mod), resid(mod))  # residuals vs fitted values
abline(h=0)

dat$fitted <- fitted(mod)    # fitted vs observed values
plot(dat$fitted, jitter(dat$fledge/dat$clutch,0.05))
abline(0,1)
```

The residual plots look fine for a binomial mixed model (Figure 9-1). The two bottom plots do show a distinct pattern, but that is alright with mixed models: we tend to have negative residuals with small fitted values and positive residuals with large fitted values. That is an effect of the shrinkage, because fitted values for farms with high survival compared to the others are shrunk toward the overall mean, especially from farms with few data points. Similarly, farms with small observed survival will have large negative residuals.

In addition to the residual plots shown in Figure 9-1 we also plotted the residuals against each predictor variable and could not detect indications of serious violations of the assumptions.

We have experienced, sometimes, that the function `glmer` produces wrong results without giving any warning or error. Such a failure may be diagnosed by checking the mean of each random effect. These means (only one in our case) should be close to 0:

```
mean(ranef(mod)$farm.f[,1])
−0.001690303
```

That seems acceptable. Because the farm effects are added to the overall mean to obtain the farm-specific fitted values, the estimate for the overall mean (on the logit-scale) will be 0.0017 too high. This translates to an overestimation of the overall mean survival of about 0.09% at the intercept, which is negligible in our study.

```
# mean survival estimate by model
t.should <- plogis(fixef(mod)["(Intercept)"])
```

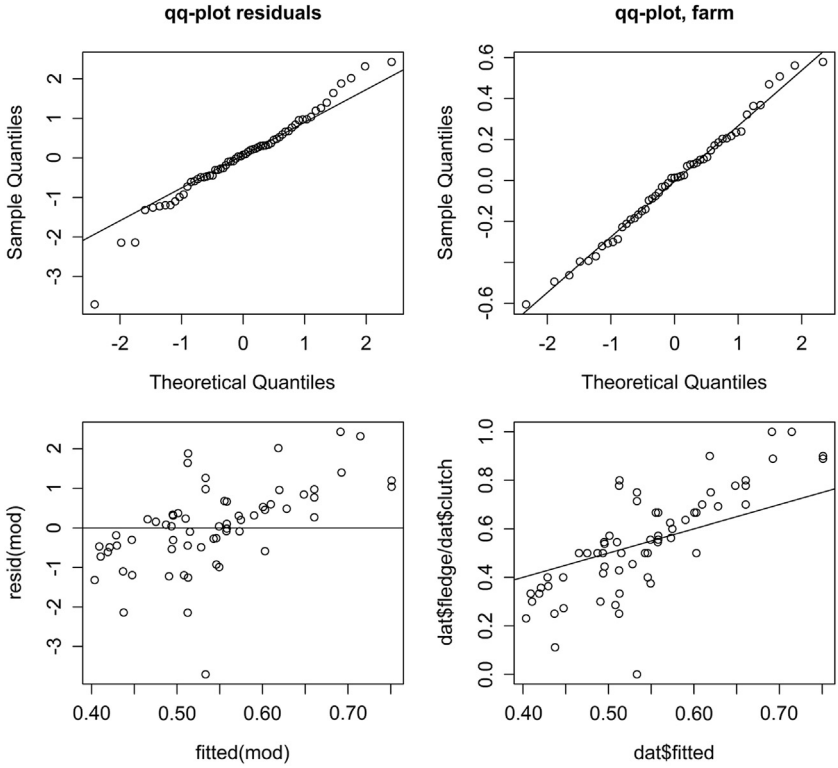

FIGURE 9-1 Diagnostic plots for the swallow nestling survival model. *Upper left*: QQ plot of the residuals; *upper right*: QQ plot of the random effects; *lower left*: residuals versus fitted values; *lower right*: observed versus fitted values.

```
# mean survival corrected for fitting failure
t.is    <- plogis(fixef(mod)["(Intercept)"] - 0.001690303)
(t.should - t.is) / t.should
0.0008853673
```

Note: a mean of the estimates of a random factor of, for example, 0.1 would translate to quite a substantial error of 5% (given the logit link function).

This problem of a nonzero mean random effect seems to appear more often when the model structure does not fit well to the data. For example, it appeared when we fitted a model to simulated data but added a random factor that was not related to the data. Thus, if the mean of the estimates of a random factor is far from 0, we should try to find a more realistic model.

To check for overdispersion, we can use a function that we downloaded from the R helplist (`dispersion_glmer` in blmeco; see its help page for the reference).

```
dispersion_glmer(mod)
[1]  1.192931
```

We get a value of 1.19, which suggests some, but tolerable, overdispersion (values over 1.4 would suggest more serious overdispersion). However, a value not indicating overdispersion does not guarantee that the model fits well. Sometimes, a combination of zero-inflation and overdispersion can lead to scale parameters close to 1, even though there is a clear lack of fit. Posterior predictive model checking can reveal such structures (Chapter 10).

9.1.4 Drawing Conclusions

The function `sim` draws random samples from the posterior distribution of the model parameters. We can use these simulations to obtain 95% CrI for the model parameters. For some versions of the package arm (e.g., version 1.6–10), the fixef slot of the object generated by sim (we usually name it bsim) does not contain the parameter names. In this case, we can add column names manually, for example, as shown in the third line that follows.

```
nsim <- 2000
bsim <- sim(mod, n.sim=nsim)
colnames(bsim@fixef) <- names(fixef(mod)) # for arm 1.6-10
fixef(mod)
(Intercept)   colsize.z         cow       dung.z
-0.09532525  0.05087098  0.39370024  -0.14236419
apply(bsim@fixef, 2, quantile, prob=c(0.025,0.975))
      (Intercept)   colsize.z         cow       dung.z
2.5%  -0.4665512  -0.1818510  -0.03813838  -0.34584466
97.5%  0.2819780   0.2757242   0.82054376   0.05972346
```

The credible intervals indicate large uncertainty for all parameters given their effect sizes. We can draw effect plots to better interpret the result

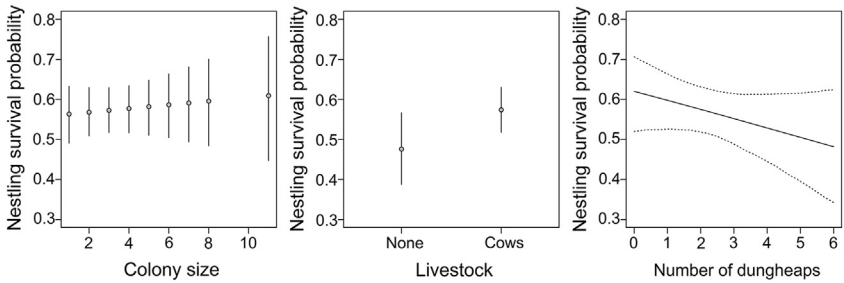

FIGURE 9-2 Nestling survival probability in relation to colony size, livestock presence, and the number of dung heaps around the farm. Given are the fitted values with 95% credible intervals. The effect of colony size is shown conditional on livestock being present and the number of dung heaps being 2. For showing the effect of livestock, colony size was set to 3.3, and the number of dung heaps to 2, and the effect of dung heaps is shown with colony size set to 3.3 and livestock present.

(Figure 9-2). Remember that such an effect plot shows the effect of one predictor while holding constant the other predictors on specific values (e.g., their means). Therefore, the effect plot shows the effect, after controlling for the effects of all other predictors. This is relevant when the predictors are correlated (i.e., in unbalanced or collinear data). In such cases, a plot of the raw data against the predictor often looks more or less different than the effect plot, because raw data are not corrected for all the other influencing factors. A look at the pairs plot that shows the correlation between all predictors can be helpful (see Section 4.2.7).

With our example, we observe the expected effect of livestock on the farm: with cows, the nestling survival was about 57%, but only 48% on farms with no livestock. Nestling survival seems to decrease by around 10% from 0 to 6 dung heaps within 500 m of the nest. (Note that in the original, unmanipulated data this effect was positive.) The effect of colony size is difficult to judge due to the large degree of uncertainty.

9.2 POISSON MIXED MODEL

9.2.1 Background

The differences between the normal linear mixed model presented in Section 7.2 and the Poisson mixed model is that the distribution of the observations around the expected value λ is now a Poisson distribution instead of a normal distribution, and that the linear predictor is (usually) related to the logarithm of the expected value rather than to the expected value itself.

$$y_i \sim Pois(\lambda_i)$$
$$\log(\lambda_i) = \beta_0 + b_{g[i]} + \beta_1 \text{ predictor } A_i + \beta_2 \text{ predictor } B_i + \ldots$$
$$b_g \sim Norm\left(0, \sigma_b^2\right)$$

We added a random group effect to the intercept, thus a group-specific intercept β_0 is modeled. Of course, any other parameter in the fixed effects part of the model can be modeled depending on group, if the data allow (or require).

9.2.2 Fitting a Poisson Mixed Model in R

To illustrate the Poisson mixed model, we go back to the whitethroat data introduced in Section 8.4.5. There, we selected one census year of the study and focused on one species, the common whitethroat. Now, we use all years and we model the number of species found in each wildflower field, including, among others, stonechat *Saxicola rubicola*, yellowhammer *Emberiza citrinella*, and skylark *Alauda arvensis*.

Here, we do not use field size as an offset to account for unequal field sizes because species number is not directly linked to the area (unlike the

density of breeding pairs as used in Section 8.4.5). Rather, the increase of species number with field size is expected to gradually level off as the field size increases, a connection known as the species-area relationship. We do have to take field size into account, but we include it as a linear and quadratic term, and not as an offset; depending on the data, it might be advisable to add more polynomial terms to model the species-area curve or use a nonlinear model.

To familiarize us with the data set again, we look at it using the function `str`:

```
data(wildflowerfields)
dat <- wildflowerfields
str(dat)
'data.frame':  136 obs.  of  8 variables:
$ field       : int 1 1...        # ID of the wildflower field
$ year        : int 2006 2007...  # year of bird count
$ age         : int 8 9...        # age of the wildflower field in years
$ bp          : int 1 1...        # number of whitethroat breeding pairs
$ X           : int 526025 526025... # X-coordinate of the field
$ Y           : int 166425 166425... # Y-coordinate of the field
$ size        : int 148 148...        # size of the wildflower field [are]
$ Nspec       : int 7 6 2 4 5...      # number of species in the field
```

The aim is to estimate the optimal age of the wildflower field regarding the number of species that use the field. We provide more exploratory data analyses in Section 9.2.5—for example, regarding correlations between explanatory variables. For now, we model the species number using age and size as fixed factors and year and field as random factors. Both age and size are expected to have a nonlinear relationship with the number of species, thus we include polynomials of each. However, using the simple polynomials, up to the third degree of these variables did not work well: the model fitting algorithm did not converge.

This reminds us of an intricacy often encountered, especially with generalized linear (mixed) models: collinear predictors can make model convergence difficult, and age.z^2 and age.z^3 (age.z = centered and scaled age) are heavily correlated with each other, and the same holds for size.z^2 and size.z^3. Using orthogonal polynomials (see Section 4.2.9) solves this problem. For the effect plot we want to draw afterwards, we need to store the poly object, and we add the orthogonal polynomials as new variables to the data frame:

```
t.poly.age <- poly(dat$age,3)
dat$age.l  <- t.poly.age[,1]
dat$age.q  <- t.poly.age[,2]
dat$age.c  <- t.poly.age[,3]
```

Similarly, orthogonal linear and quadratic effects were calculated for size (not shown here), and the following model can be fit using the function `glmer` with the family argument set to poisson:

```
mod <- glmer(Nspec ~ age.l + age.q + age.c + size.l + size.q +
           (1|year.f) + (1|field.f), data=dat, family=poisson)

mod
Generalized linear mixed model fit by maximum likelihood (Laplace
Approximation) ['glmerMod']
 Family: poisson ( log )
Formula: Nspec ~ age.l + age.q + age.c + size.l + size.q +
(1 | year.f) + (1|field.f)
   Data: dat
     AIC       BIC    logLik  deviance  df.resid
485.9526  509.2538 -234.9763  469.9526       128
Random effects:
 Groups Name         Std.Dev.
 field.f    (Intercept) 0.2849
 year.f     (Intercept) 0.1335
Number of obs: 136, groups: field, 67; year, 8
Fixed Effects:
(Intercept)     age.l     age.q     age.c    size.l    size.q
     0.7435    4.0966   -4.7101    2.8929    2.6328   -0.8874
```

9.2.3 Assessing Model Assumptions

Plotting the data together with the model is one important tool to assess model fit; however, it is not sufficient. We do the usual residual analysis; the plots, not shown here, do not indicate serious problems.

```
qqnorm(resid(mod)) # qq plot of residuals
qqline(resid(mod))

qqnorm(ranef(mod)$year.f[,1])          # qq plot of random effects
qqline(ranef(mod)$year.f[,1])
qqnorm(ranef(mod)$field.f[,1])
qqline(ranef(mod)$field.f[,1])

scatter.smooth(fitted(mod), resid(mod)) # fitted versus residuals

plot(fitted(mod),dat$Nspec)             # data versus fitted values
```

To check whether overdispersion is present we can again use the function `dispersion_glmer` (see Section 9.1.3); it yields a value of 0.99—that is, there is no indication of overdispersion. An alternative way to look at the issue is to fit a model with an observation level random factor added. We can then have

a look at the variance of this factor and judge from that whether we think overdispersion could be important; in our case, this seems not to be the case:

```
dat$obsid <- factor(1:nrow(dat))
modod <- glmer(Nspec ~ age.l + age.q + age.c + size.l + size.q +
               (1|year.f) + (1|field.f) + (1|obsid), data=dat,
               family=poisson)
modod
[...]
Random effects:
 Groups Name          Std.Dev.
 obsid    (Intercept) 0.0000
 field.f  (Intercept) 0.2849
 year.f   (Intercept) 0.1335
[...]
```

The additional variance is estimated to be 0. There is apparently negligible overdispersion.

9.2.4 Drawing Conclusions

The uncertainty measurements for the parameter estimates are obtained from their posterior distributions simulated by `sim`.

```
nsim <- 2000
bsim <- sim(mod, n.sim=nsim)
apply(bsim@fixef, 2, quantile, prob=c(0.025,0.5,0.975))
     (Intercept)   age.l    age.q   age.c  size.l   size.q
2.5%     0.55300  2.3891  -6.3780  1.2966  1.0409  -2.35111
50%      0.74840  4.0820  -4.7063  2.8711  2.6342  -0.91090
97.5%    0.94287  5.9022  -2.9683  4.4356  4.1916   0.61581
```

The fitted values and their uncertainties are also derived from the simulated values of the posterior distributions of the model parameters. Take care to use the right inverse link function to transform the linear predictor to the scale of the response. Here, we have to use `exp`, since we used the log-link. Also, any transformations that had been done to the predictors must be done in the same way if we want to predict on the original scale of a predictor. Because we have used orthogonal polynomials, we do the same transformation to the age-values for which we want to get fitted values, using the function `predict` (Section 4.2.9). The effect plot for age shows that fields around five years old are associated with the highest number of species (Figure 9-3).

```
newdat <- data.frame(
   year = mean(dat$year),
   age = seq(min(dat$age), max(dat$age), length.out=100),
   size = mean(dat$size))
```

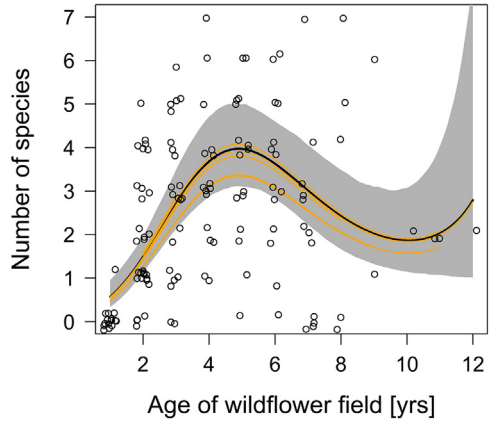

FIGURE 9-3 Species number in relation to the age of wildflower fields with average regression line over all fields and years (bold line) including the 95% credible interval (gray), and year-specific regression lines (orange lines). Circles are the raw data.

```
newdat$age.l  <- predict(t.poly.age, newdat$age)[,1]
newdat$age.q  <- predict(t.poly.age, newdat$age)[,2]
newdat$age.c  <- predict(t.poly.age, newdat$age)[,3]
newdat$size.l <- predict(t.poly.size, newdat$size)[,1]
newdat$size.q <- predict(t.poly.size, newdat$size)[,2]

Xmat <- model.matrix(~ age.l + age.q + age.c + size.l + size.q,
                     data=newdat)
newdat$fit <- exp(Xmat %*% fixef(mod)) # exp = inverse link function
```

Now, we calculate for each fitted value in newdat 2000 values that are random draws from their posterior distributions. The 2.5% and 97.5% quantiles of these values are used as lower and upper limits of the 95% credible interval.

```
fitmat <- matrix(nrow=nrow(newdat), ncol=nsim)
for(i in 1:nsim) fitmat[,i] <- exp(Xmat %*% bsim@fixef[i,])
newdat$lwr <- apply(fitmat, 1, quantile, prob=0.025)
newdat$upr <- apply(fitmat, 1, quantile, prob=0.975)
```

Now, we can use newdat$age, newdat$fit, newdat$lwr, and lwedat$upr to produce an effect plot on the original scale of age:

```
plot(Nspec ~ age, data=dat)
polygon(c(newdat$age, rev(newdat$age)),
    c(newdat$lwr, rev(newdat$upr)),
    col=grey(0.7), border=NA)    # 95% CrI given as a shadow
```

```
lines(newdat$age, newdat$fit, lwd=2)  # population mean
# add separate lines for each year:
for(i in 1:nlevels(dat$year.f)) {
  t.year <- levels(dat$year.f)[i]
  x <- seq(min(dat$age[dat$year.f==t.year]),
           max(dat$age[dat$year.f==t.year]), length=100)
  x.l <- predict(t.poly.age, x)[,1]  # analogous transformation
  x.q <- predict(t.poly.age, x)[,2]  # to get orthogonal linear,
  x.c <- predict(t.poly.age, x)[,3]  # quadratic and cubic terms
  y <- exp(fixef(mod)[1] + ranef(mod)$year.f[i,1] +
           fixef(mod)[2]*x.l + fixef(mod)[3]*x.q + fixef(mod)[4]*x.c)
  lines(x,y, col="orange",lwd=1.2)
  }
```

9.2.5 Modeling Bird Densities by a Poisson Mixed Model Including an Offset

To present the Poisson mixed model with an offset, we again use the whitethroat data introduced in Section 8.4.5 and used in Sections 9.2.2 through 9.2.4, too. Now, we again focus on the common whitethroat, but unlike in Section 8.4.5 we use all years, which means that many of the wildflower fields were monitored repeatedly. To account for these repeated measurements, the wildflower field ID is used as a random factor.

The variables in the data file we use here are field (ID of the wildflower field), year (census year), age (age of the wildflower field in years), bp (number of whitethroat breeding pairs), X and Y (the coordinates of the field), and size (the size of the field in areas, i.e., 10 × 10 m).

First, we do some exploratory analyses to get an overview of the structure of the data set. Specifically, we are interested in how balanced the data are and whether the predictor variables year and age are correlated, or whether there are other structures we have to take into account when modeling whitethroat density.

```
data(wildflowerfields); dat <- wildflowerfields
table(table(dat$field)) # how many cases with n observations per field?
 1  2  3  4  5
26 27  6  2  6
```

From 26 of the 67 wildflower fields, only one bird census exists. Twenty-seven fields were monitored in 2 years, and 14 were monitored for 3 to 5 years. To see which wildflower field was monitored in which year we can type the next line:

```
table(dat$field, dat$year) # output not shown
```

The highest numbers of wildflower fields were monitored during the years 2006 and 2007.

```
table(dat$year)
2004  2005  2006 2007  2008  2009  2010  2011
   1     1    41   41    11    16    13    12
```

The main variable of interest in this study is the age of the wildflower field; remember that the question was at which age wildflower fields are optimal for birds. We, therefore, check whether age is confounded with other variables, such as year. We use boxplots, and we see that the median ages per year do not show a trend, despite the fact that the three oldest wildflower fields were all monitored during the last three years of the study. We also do not find a significant correlation between size and age of the wildflower fields (Figure 9-4). Thus, the data seem to be suited for analysis of age effects on whitethroat densities.

```
boxplot(age ~ year, dat, ylab="Age of wildflower field")
scatter.smooth(dat$size, dat$age)
```

Now, we can construct our model to analyze whitethroat densities in the wildflower fields in relation to age, year, and size of the wildflower field. The number of whitethroat territories y_i is directly related to the size of the wildflower field. We, therefore, include size as an offset in the model. We further include a linear trend of year as well as a random year effect. The linear trend accounts for systematic changes in whitethroat densities over the years whereas the random year effect accounts for random between-year variance, for example, due to different weather situations.

Similar to the analysis done with just one year (Section 8.4.5) we include polynomials of "age" up to the third degree because we expect an optimal age of the wildflower field for whitethroat density. Also, we again would like to see whether the size of a wildflower field affects whitethroat density, thus we include a linear trend of size (i.e., size is the offset but also a predictor).

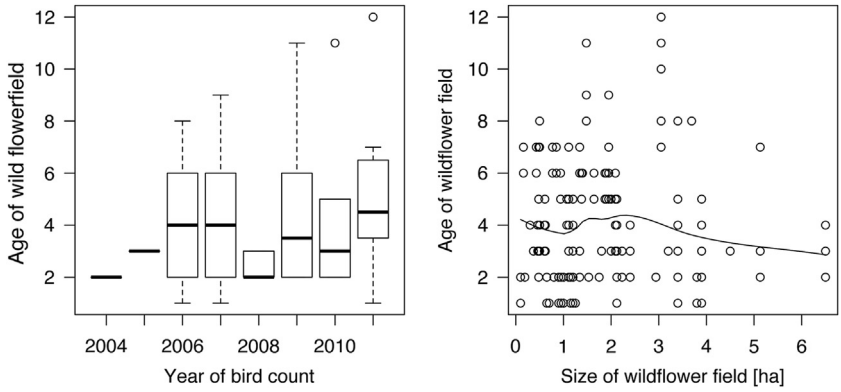

FIGURE 9-4 Age of the wildflower fields in relation to year and size of the field.

Finally, we include the field ID as a random factor because some fields have been monitored in more than one year, so that these data are not independent. The model formula looks like this:

$$
\begin{aligned}
y_i &\sim Pois(\lambda_i \text{size}_i) \\
\log(\lambda_i \text{size}_i) &= \beta_0 + \beta_1 \text{year}_i + \beta_2 \text{age}_i + \beta_3 \text{age}_i^2 + \beta_4 \text{age}_i^3 \\
&\quad + \beta_5 \text{size}_i + b_{\text{ID}[i]} + d_{\text{year}[i]} + \log(\text{size}_i) \\
b_{\text{ID}} &\sim \text{Norm}(0, \sigma_b) \\
d_{\text{year}} &\sim \text{Norm}(0, \sigma_d)
\end{aligned}
$$

Before fitting such a complicated model, it is recommended to transform the offset variable to a sensible scale, to standardize covariates, and, possibly, to use orthogonal polynomials (Section 4.2.9). Otherwise, the fitting algorithm often may not converge and estimates could become unreliable due to correlations between the model parameters or due to some values of the linear predictor taking on values beyond computer accuracy. For example, if we measure the size of the fields in square meters and use this variable as the offset, the fitted number of territories will be close to zero; similarly, if we include a polynomial of a covariate with large values, the polynomials may be too large for the computer.

A sensible scale for measuring whitethroat density is the number of breeding pairs per hectare. Therefore, we transform the size variable originally measured in ares (10 × 10 m) to hectares (100 × 100 m) to be used as the offset. For the linear size effect, we use the standardized size variable size.z. We also standardize year and use orthogonal polynomials of age.

```
dat$size.ha <- dat$size/100
dat$size.z <- as.numeric(scale(dat$size))
dat$year.z <- as.numeric(scale(dat$year))
t.poly.age <- poly(dat$age,3)
dat$age.l <- t.poly.age[,1]
dat$age.q <- t.poly.age[,2]
dat$age.c <- t.poly.age[,3]
dat$field.f <- factor(dat$field)
dat$year.f <- factor(dat$year)
```

Now, we are ready to fit the model:

```
mod <- glmer(bp ~ year.z + age.l + age.q + age.c + size.z +
             (1|field.f) + (1|year.f) + offset(log(size.ha)),
             family=poisson, data=dat)
mod
Generalized linear mixed model fit by maximum likelihood (Laplace
Approximation) ['glmerMod']
Family: poisson ( log )
```

```
Formula: bp ~ year.z + age.l + age.q + age.c + size.z + (1|field.f) +
  (1|year.f) + offset(log(size.ha))
   Data: dat
      AIC       BIC     logLik  deviance  df.resid
 308.7927  332.0940 -146.3964  292.7927       128
Random effects:
 Groups Name        Std.Dev.
 field.f  (Intercept) 0.425
 year.f  (Intercept) 0.000
Number of obs: 136, groups: field.f, 67; year.f, 8
Fixed Effects:
(Intercept) year.z   age.l    age.q   age.c   size.z
    -0.8529  0.2656  4.2290  -5.7332  3.0469  -0.2655
```

We see that the between-year variance in whitethroat density is negligible. Before drawing conclusions we look at the residuals.

```
par(mfrow=c(2,2))
scatter.smooth(fitted(mod), resid(mod))
scatter.smooth(dat$year.z, resid(mod))
scatter.smooth(dat$age.z, resid(mod))
scatter.smooth(dat$size.z, resid(mod))
```

From the plots in Figure 9-5 we cannot see that we have missed important structures in the data. Also, the QQ plots of the residuals and the two random factors do not look very bad (not shown). To check whether the data are overdispersed, we can include an observation level random factor:

```
dat$obsid <- factor(1:nrow(dat))
modod <- glmer(bp ~ year.z + age.l + age.q + age.c + size.z +
   (1|field.f) + (1|year.f) + (1|obsid) + offset(log(size.ha)),
   family=poisson, data=dat)
modod
Random effects:
Groups Name        Std.Dev.
obsid  (Intercept) 0.000
field.f (Intercept) 0.425
year.f (Intercept) 0.000
Number of obs: 136, groups: obsid, 136; field.f, 67; year.f, 8
```

We see that the extra variance between the observations is essentially zero. Thus, we do not need the observation level random factor.

Finally, we check whether spatial correlation is an issue in these data.

```
library(gstat); library(sp)
spdata <- data.frame(resid=resid(mod), x=dat$X, y=dat$Y)
coordinates(spdata) <- c("x", "y")
bubble(spdata, "resid", col=c("blue", "orange"), main="Residuals")
```

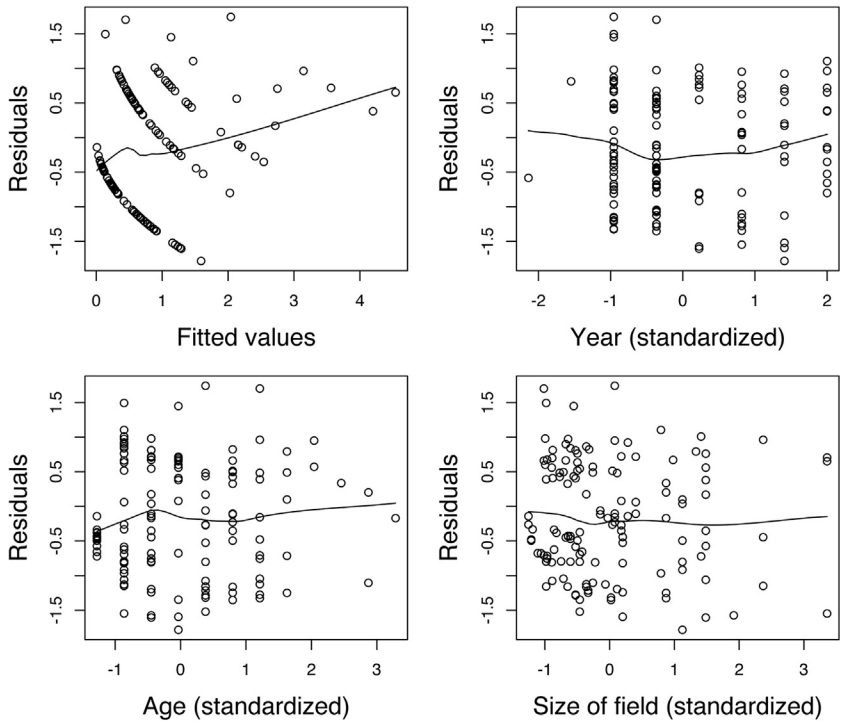

FIGURE 9-5 Residual plots of the whitethroat model: Residuals versus (1) fitted values, (2) standardized year, (3) standardized age, and (4) standardized size.

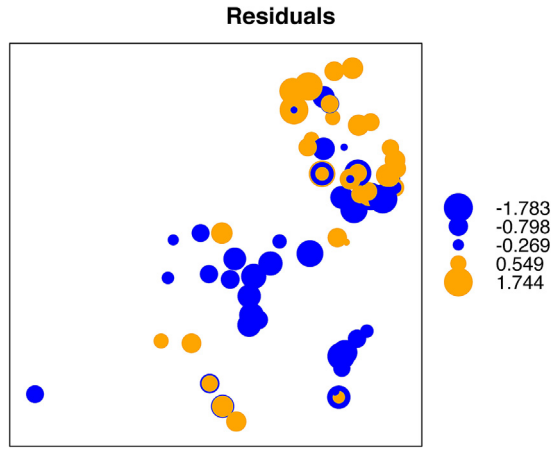

FIGURE 9-6 Spatial distribution of the residuals in the whitethroat data.

Indeed, we see that the residuals are spatially correlated (Figure 9-6). The positive residuals cluster in two patches (upper right corner and bottom center) whereas in the center of the study area negative residuals are overrepresented. Such a pattern means that the observations are not independent of each other. As a consequence, the uncertainty of the parameter estimates is underestimated (due to pseudoreplication). However, in the semivariogram of the residuals the spatial correlation is only weakly discernible. Therefore, we proceed here assuming no spatial correlation. But, in the real world, we would probably proceed in the following way: First, try to find a factor that explains the pattern such as a habitat or landscape characteristics and, second, include subregions within the study area as a random factor. If this does not help, use a method presented in Chapter 13.

For the moment, we trust the semivariogram suggesting only very weak spatial correlation and start drawing conclusions. Let's first have a look at the 95% credible intervals of the parameters.

```
nsim <- 2000
bsim <- sim(mod, n.sim=nsim)
apply(bsim@fixef, 2, quantile, prob=c(0.025, 0.5, 0.975))
      (Intercept)    year.z    age.l     age.q    age.c     size.z
2.5%    -1.11672  0.060595  1.1712   -8.5458  0.30477   -0.49780
50%     -0.84925  0.261393  4.1953   -5.7838  3.04104   -0.26055
97.5%   -0.59134  0.485135  7.2399   -2.8823  5.70323   -0.02071
```

We see a slight increase of the number of territories over the years. Because we have used a log-link function, the effects are not additive. The exponential of the coefficient (exp(0.263) = 1.304) is the multiplicative change in the outcome variable. Thus, when year.z increased by 1 (which corresponds to the standard deviation of the original variable year, `sd(dat$year)` = 1.7 years), whitethroat density increased by 30%. We also see a slight negative relationship between whitethroat density and the size of the wildflower field. With each increase of the field size by `sd(dat$area)` = 139 are = 1.4 ha, density decreases to exp(−0.261) = 0.77 = 77%.

The nonlinear age effect is more difficult to describe just from looking at the estimated model parameters. Therefore, we plot the effect of age. We calculate fitted whitethroat densities (expected number of breeding pairs per ha) for several different values of age while holding the other two predictors constant. To do so, we prepare a new data frame containing all age values for which we would like to predict. We further add the variable "year.z" and set it to 0. Because year.z is a standardized variable, the value 0 corresponds to the mean of the data. As we would like to predict whitethroat density for 1 ha, we insert for the variable "size.z" the value that corresponds to 1 ha, which is 100 minus the mean of the original variable "size" divided by its standard

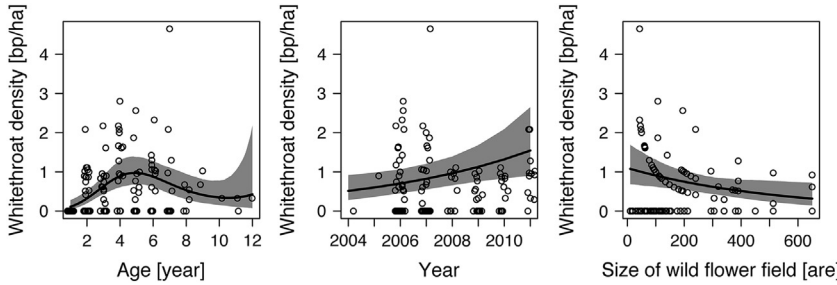

FIGURE 9-7 Whitethroat densities on sown wildflower fields with respect to age of the field, year, and size of the field. Circles = observed densities [bp/ha], solid line = fitted values with 95% credible interval (gray area).

deviation. Because we have used orthogonal age polynomials, we have to transform age in the new data frame as we have done it with the original variable, using `predict` (compare to Section 4.2.9).

As we have done before, we construct the model matrix "Xmat" and obtain the predicted values for the new predictor variables in newdat. Then we do this for all nsim simulated model parameters to obtain a 95% credible interval of the fitted values. At last, we plot the data together with the fitted values. We do this in turn for showing the effects of age, year, and size of the wildflower field (Figure 9-7). The R code that follows produces, as an example, the effect of age.

```
newdat <- data.frame(age=seq(1,12, by=0.1), size=100)
newdat$year.z <- 0 # corresponds to the mean of dat$year
newdat$size.z <- (newdat$size-mean(dat$size))/sd(dat$size)
newdat$age.l <- predict(t.poly.age,newdat$age)[,1]
newdat$age.q <- predict(t.poly.age,newdat$age)[,2]
newdat$age.c <- predict(t.poly.age,newdat$age)[,3]

Xmat <- model.matrix( ~ year.z + age.l + age.q + age.c + size.z ,
                      data=newdat)
b <- fixef(mod)
newdat$fit <- exp(Xmat %*% b) # exp = inverse link function
fitmat <- matrix(ncol=nsim, nrow=nrow(newdat))
for(i in 1:nsim) fitmat[,i] <- exp(Xmat %*% bsim@fixef[i,])

newdat$lwr <- apply(fitmat, 1, quantile, prob=0.025)
newdat$upr <- apply(fitmat, 1, quantile, prob=0.975)

plot(dat$age, dat$bp/dat$size.ha, xlab="Age [year]",
     ylab="Whitethroat density [bp/ha]", las=1, cex.lab=1.4,
     cex.axis=1.2, type="n")
```

```
# draw 95% CrI as a shade:
polygon(c(newdat$age, rev(newdat$age)), c(newdat$lwr,
          rev(newdat$upr)),  border=NA, col=grey(0.5))
lines(newdat$age, newdat$fit, lwd=2)
points(jitter(dat$age), dat$bp/dat$size.ha)
```

The left panel in Figure 9-7 shows what the data tell us about the relationship between whitethroat density and the age of sown wildflower fields. Our original question was, at what age of the wildflower fields do we find the highest density of whitethroat? We can answer this question more directly by giving the age at which whitethroat density is maximized as a point estimate with 95% credible interval.

In Figure 9-7 we see that the maximum is between 4 and 5 years. To get the exact point of the maximum, we need to take the first derivative from the cubic function (for that, we refit the model using normal polynomials of the z-transformed age rather than orthogonal polynomials), set it to 0 and solve it. We do this for the 2000 simulated sets of model parameters (defining 2000 different cubic functions) to obtain 2000 simulated values from the posterior distribution of the optimal age. In such cases, where the calculation of the derived parameter involves several steps, we find it convenient to write an R function that does the whole calculation.

We will also find similar preprogrammed functions on the internet when we, for example, search within the R webpage (type "r-project.org: maximum of polynomial function" into Google). However, if it is not too difficult, we prefer to write our own functions, because then we know exactly what we have done. In the following function, we get the x- and y-values of the maximum of an exponentiated cubic curve. The exponentiated cubic curve corresponds to the regression line for age in our Poisson model.

```
maxofcubicfun <- function(Intercept, x){
  b <- x[1]
  c <- x[2]
  d <- x[3]
  D <- 4*(c^2 - 3*b*d)
  if(D<0){
    xmax <- ymax <- NA
    }
  if(D>0){
    xzero <- (-2*c + c(-1,1)*sqrt(D))/(6*d)
    yzero <- exp(Intercept + b*xzero + c*xzero^2 + d*xzero^3)
    index <- yzero==max(yzero)
    xmax <- xzero[index]
    ymax <- yzero[index]
    }
  return(c(xmax,ymax))
  }
```

We can apply this function to the cubic function in Figure 9-7.

```
maxofcubicfun(b[1],b[3:5])
[1]  0.3367687  0.8312031
```

The first number is the x-coordinate of the maximum and the second number is the corresponding maximal mean number of breeding pairs per ha. Of course, we have to back-transform the first number because we have z-transformed age before the model fit:

```
maxofcubicfun(b[1],b[3:5])[1]*sd(dat$age) + mean(dat$age)
[1]  4.892504
```

Now, we do this for all the 2000 sets of model coefficients in the bsimobject (we have to rerun `sim` on the model with the normal polynomials for age and call it "bsim2").

```
postoptage.z <- numeric(nsim)
for(i in 1:nsim) postoptage.z[i] <-
  maxofcubicfun(fixef(bsim2)[i,1],fixef(bsim2)[i,3:5])[1]
postoptage <- postoptage.z*sd(dat$age) + mean(dat$age)
```

And then we extract the 95% CrI. We also calculate the proportion of regression lines that did not have a maximum at all, which was only 0.15%, thus, we can be quite sure that there is an optimal age somewhere between 4 and 7 years.

```
quantile(postoptage, prob=c(0.025,0.975), na.rm=TRUE)
    2.5%     97.5%
4.262726  6.220102
sum(is.na(postoptage))/nsim  # proportion with no maximum
[1] 0.0015
```

FURTHER READING

In a review on generalized linear mixed models (GLMM) in ecology, Bolker et al. (2008) recommended using Bayesian methods to calculate uncertainty estimates.

The R package MCMCglmm (Hadfield, 2010) provides functions to fitting fairly complex GLMMs, such as models for data with correlation structure caused by genetic relationships (pedigree or phylogeny). GLMMs are the basis of many more complicated ecological models. Therefore, we find very good introductions in books like the spatial capture-recapture book by Royle et al. 2014.