

Chapter 13

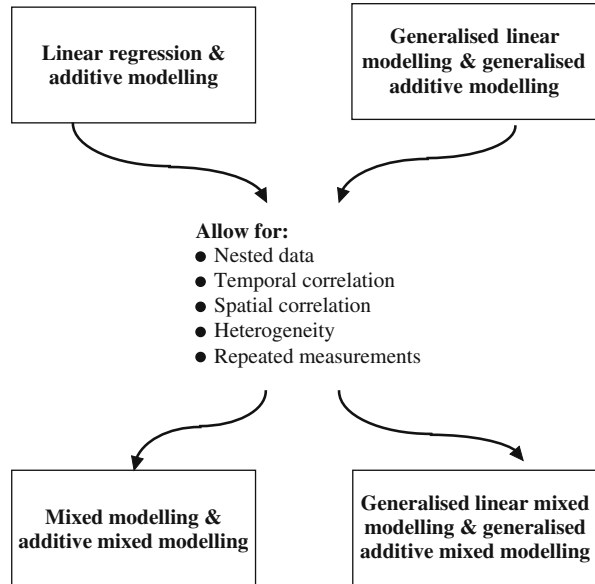
GLMM and GAMM

In Chapters 2 and 3, we reviewed linear regression and additive modelling techniques. In Chapters 4–7, we showed how to extend these methods to allow for heterogeneity, nested data, and temporal or spatial correlation structures. The resulting methods were called linear mixed modelling and additive mixed modelling (see the left hand pathway of Fig. 13.1). In Chapter 9, we introduced generalised linear modelling (GLM) and generalised additive modelling (GAM), and applied them to absence–presence data, proportional data, and count data. We used the Bernoulli and binomial distributions for 0–1 data (the 0 stands for absence and the 1 for presence), and proportional data (Y successes out of n independent trials), and we used the Poisson distribution for count data. However, one of the underlying assumptions of these approaches (GLM and GAM) is that the data are independent, which is not always the case. In this chapter, we take this into account and extend the GLM and GAM models to allow for correlation between the observations, and nested data structures. It should come as no surprise that these methods are called generalised linear mixed modelling (GLMM) and generalised additive mixed modelling (GAMM); see the right hand pathway of Fig. 13.1.

The good news is that these extensions follow similar steps we used in mixed modelling. For example, the inclusion of a random intercept in a GLM is imposing the compound symmetrical correlation structure, just as it did in the linear mixed model. In fact, just as the linear regression model is a GLM with a Gaussian distribution, so is the linear mixed model a GLMM with a Gaussian distribution.

When there is good news, there is often some bad news. And the bad news is that GLMM and GAMM are on the frontier of statistical research. This means that available documentation is rather technical, and there are only a few, if any, textbooks aimed at ecologists. There are multiple approaches for obtaining estimated parameters, and there are at least four packages in R that can be used for GLMM. Sometimes these give the same results, but sometimes they give different results. Some of these methods produce a deviance and AIC; others do not. This makes the model selection in GLMM more of an art than a science. The main message is that when applying GLMM or GAMM, try to keep the models simple or you may get numerical estimation problems.

Fig. 13.1 Relationship between linear regression, additive modelling, mixed modelling, additive modelling, GLM, GAM, GLMM, and GAMM. The Generalised Estimation Equations is an alternative technique for the *lower right box*



The literature that we consulted for writing this chapter were almost exclusively written for medical, economical, and social science. We strongly recommend Snijders and Bosker (1999), Raudenbush and Bryk (2002), Goldstein (2003), Fitzmaurice et al. (2004), Brown and Prescott (2006), and for the GAMM Rupert et al. (2003) and Wood (2006). With some effort, you should be able to work your way through these books after reading this chapter. Luke (2004) is reasonably non-technical and can be read as an introduction. If you have good mathematical skills, we recommend McCulloch and Searle (2001) or Jiang (2007). The good news is that publications using GLMM or GAMM are now appearing more frequently in the ecological literature, e.g. Vicente et al. (2006) and Pierce et al. (2007) among others.

13.1 Setting the Scene for Binomial GLMM

In Chapter 12, we used data from Vicente et al. (2005), who looked at the distribution and faecal shedding patterns of the first-stage larvae (L1) of *Elaphostrongylus cervi* in red deer across Spain. In this chapter, we focus on the relationship between the presence and absence of *E. cervi* L1 in deer and the explanatory variables length and sex of the animal and farm identity. Because the response variable is of the form 0–1, we are immediately in the world of a binomial GLM. The following model is applied on these data:

$$\text{logit}(p_{ij}) = \alpha + \beta_1 \times \text{Length}_{ij} + \beta_2 \times \text{Sex}_{ij} + \beta_3 \times \text{Length}_{ij} \times \text{Sex}_{ij} + \beta_4 \times \text{Farm}_i$$

The notation logit stands for the logistic link (Chapter 10), p_{ij} is the probability that animal j on farm i has the parasite, Length_{ij} is the length of the deer, Sex_{ij} tells us whether it is male or female, and Farm_i identifies the farm. Because of the large number of farms, we did not include an interaction term involving the variable farm.

The following code accesses the data, defines the nominal variables as nominal, and centres length. In Chapter 12, we gave a justification for centring length.

```
> library(AED); data(DeerEcervi)
> DeerEcervi$Ecervi.01 <- DeerEcervi$Ecervi
> DeerEcervi$Ecervi.01[DeerEcervi$Ecervi>0] <-1
> DeerEcervi$fSex <- factor(DeerEcervi$Sex)
> DeerEcervi$CLength <- DeerEcervi$Length -
                        mean(DeerEcervi$Length)
> DeerEcervi$fFarm <- factor(DeerEcervi$Farm)
```

The code below applies a GLM on the data, drops each allowable term in turn from the model, and applies a likelihood ratio test that is Chi-square distributed. Note that because the interaction between length and sex is included, we cannot drop the main terms `CLength` and `fSex`. The `drop1` function compares the deviance of the specified model with that of nested models. The difference between these two deviances is Chi-square distributed. The GLM model includes a farm effect, a length effect, a sex effect, and an interaction between length and sex.

```
> DE.glm<-glm(Ecervi.01 ~ CLength * fSex+fFarm,
              data = DeerEcervi, family = binomial)
> drop1(DE.glm, test = "Chi")
```

Single term deletions.

Model:		Df	Deviance	AIC	LRT	Pr(Chi)
<none>			745.50	799.50		
fFarm		23	1003.72	1011.72	258.22	< 2.2e-16
CLength:fSex		1	755.48	807.48	9.98	0.001579

The first line shows the deviance for the model in which no term is dropped. Its AIC is 799.50. By dropping the nominal variable farm from this model, the deviance increases to 1011.72, a change of 258.22. The change in deviance for a binomial GLM is Chi-square distributed with 23 degrees of freedom and has a p -value that is smaller than 0.001, which means that the term is highly significant. By dropping only the interaction term, the change in deviance is 9.98, which is also significant.

To obtain insight in what the model is doing, we want to visualise the predicted values. Because the model contains a length effect, sex effect, farm effect, and an interaction between length and sex, visualisation is not trivial. The easiest option is to choose a particular farm and sex and then plot the predicted probabilities versus length. We arbitrary decided to choose $\text{sex} = 1$ (female). The R code below first

creates a graph in which the observed presence and absence values of the parasite *E. cervi* L1 in deer are present. This is the `plot` command. The remaining code predicts the probability of the presence of the parasite for a range of length values at particular farms. This is done with a loop; each iteration represents a farm. The `order` command is used to avoid spaghetti plots.

```
> plot(DeerEcervi$CLength, DeerEcervi$Ecervi.01,
      xlab = "Length", ylab = "Probability of \
      presence of E. cervi L1", main = "Male data")
> I1 <- order(DeerEcervi$CLength)
> AllFarms <- unique(DeerEcervi$Farm)
> for (j in AllFarms){
  mydata <- data.frame(
    CLength = DeerEcervi$CLength,
    fSex = "1",
    fFarm = AllFarms[j])
  n <- dim(mydata)[1]
  if (n > 10){
    P.DE2 <- predict(DE.glm, mydata,
                     type = "response")
    lines(mydata$CLength[I1], P.DE2[I1])}}
```

The predicted values for the female data are presented in Fig. 13.2. To create this graph, we chose a particular farm and sex (see code above) and then calculated the probabilities as a (logistic) function of different length values and the chosen farm and sex. Doing this for different farms produces the multiple lines in the figure.

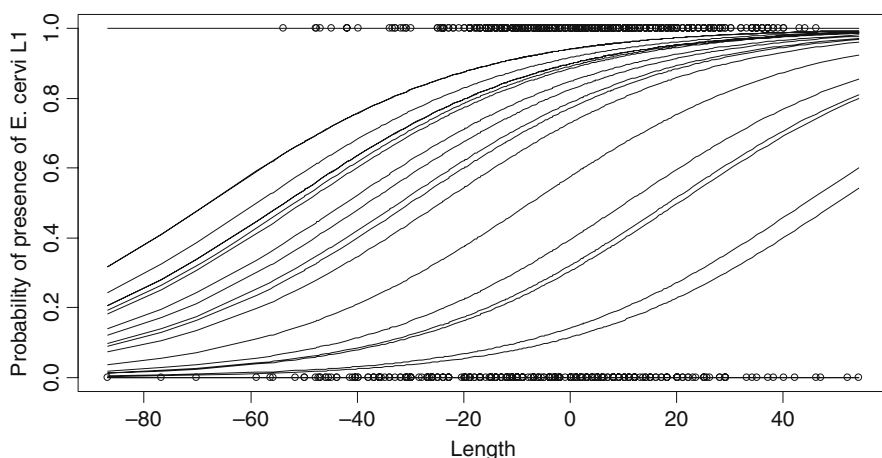


Fig. 13.2 Predicted probabilities of parasitic infection along (*centred*) deer length for females at all farms. Each line represents a farm

It is easy to do the same for the male data, and the graph looks similar (and is not presented here). Instead of using the `for` loop, it is also possible to do the prediction for all data at once (not farm by farm) and use the function `matlines` to do the plotting of individual curves; see its help file for examples.

The problem with this model is that the explanatory variable `farm` consumes 23 degrees of freedom and we are not even interested in knowing that there is a farm effect. We cannot drop it neither as it is highly significant. It is also possible that there is a length – farm interaction, costing another 23 parameters. The other problem is how we predict from this model. We can choose a value for `length` in Fig. 13.2, and then read of the probability for the presence of the parasite at a certain farm. But we can only do this exercise for our 24 farms. The model does not allow us to make a statement for farms in general.

This discussion should sound familiar to you, as it is identical to the discussion we had with the beach effect for the RIKZ data in Chapter 5. There, we had nine beaches, and on each beach, we had five observations. We replaced the nine parameters from beach by one random intercept and called the model a random intercept model. Now, we have 24 farms and multiple observations per farm. We can do exactly the same in a GLM, and this is discussed in the next section.

13.2 GLMM and GAMM for Binomial and Poisson Data

In this section, we apply GLMM and GAMM on two data sets; these were also used in Chapter 12. We start with the deer data we used above, followed by owl data (counts).

13.2.1 Deer Data

In Section 13.1, we applied a GLM on the deer data. We encountered two problems: The explanatory variable `farm` is using up a large number of degrees of freedom, and we can only make predictions for the current set of farms. We now use the same extension as we did for linear regression and random intercepts (Chapter 5) and work towards the GLM equivalent of a mixed model. Instead of using `farm` as a fixed effect with 24 levels, we use it as a random effect and the model becomes

$$Y_{ij} \sim \text{Bin}(1, p_{ij})$$

$$\text{logit}(p_{ij}) = \alpha + \beta_1 \times \text{Length}_{ij} + \beta_2 \times \text{Sex}_{ij} + \beta_3 \times \text{Length}_{ij} \times \text{Sex}_{ij} + a_i$$

$$a_i \sim N(0, \sigma_a^2)$$

Y_{ij} is 1 if animal j on farm i has *E. cervi* L1 and 0 otherwise. The random intercept a_i is assumed to be normally distributed with mean 0 and variance σ_a^2 . If this variance is small, then the contribution from a_i is also rather small and all farms will have a similar logistic curve. On the other hand, if σ_a^2 is relatively large, then

each farm will have very different intercepts. This approach reduces the number of parameters considerably compared to using Farm as a fixed effect.

Using farm as a random intercept has another major advantage. Just as in linear mixed modelling, a random intercept model is implicitly introducing the compound symmetrical correlation structure. This implies that the probability of a deer carrying the parasite is correlated to other deer on the same farm.

There are various functions in R that can be used for GLMM; the main ones are `glmmPQL` from the MASS package, `lmer` from the lme4 package, and `glmmML` from the glmmML package. Later in this section, we compare the output from all these models, but first we concentrate on the `glmmPQL` method. The following R code applies the GLMM model described above.

```
> library(MASS)
> DE.PQL <- glmmPQL(Ecervi.01 ~ CLength * fSex,
  random = ~ 1 | fFarm, family = binomial,
  data = DeerEcervi)
> summary(DE.PQL)
```

We used the object name `DE.PQL` because it reminds us of `DEer` and which tool was used (PQL, which will be discussed later in this chapter). The function `glmmPQL` is in the MASS package from Venables and Ripley (2002), and we first need to load this package. The random effect is specified in a similar way as we did for linear mixed models in Chapter 5. In fact, the only new code is the `family = binomial` option. The probability of presence of the parasite is modelled as a function of length, sex, and their interaction. The random effect farm is adding a random term to the intercept. The results of the `summary` command are given below.

Linear mixed-effects model fit by maximum likelihood

Data: DeerEcervi

AIC BIC logLik

NA NA NA

Random effects:

Formula: ~1 | fFarm

(Intercept) Residual

StdDev: 1.462108 0.9620576

Variance function:

Structure: fixed weights

Formula: ~invwt

Fixed effects: Ecervi.01 ~ CLength * fSex

	Value	Std.Error	DF	t-value	p-value
(Intercept)	0.8883697	0.3373283	799	2.633547	0.0086
CLength	0.0378608	0.0065269	799	5.800768	0.0000
fSex2	0.6104570	0.2137293	799	2.856216	0.0044
CLength:fSex2	0.0350666	0.0108558	799	3.230228	0.0013

Number of Observations: 826

Number of Groups: 24

The random intercept a_i has a standard error of 1.462, and the residual standard error is 0.962. The residual standard error is for the working residuals, which are used internally and are less useful than, for example, Pearson residuals. The AIC and BIC are not defined, and we explain later why not. The interaction term is significant at the 5% level, and this means that we have to include the main terms as well. We now discuss how to interpret this output. For a female deer (fSex = '1'), the probability that a deer has the parasite *E. cervi* L1 is given by

$$\text{logit}(p_{ij}) = 0.888 + 0.037 \times \text{Length}_{ij} + a_i \quad a_i \sim N(0, 1.462^2)$$

The first level of the variable Sex is used as baseline; hence, the contribution from the Sex and the interaction are 0. For a male deer (Sex = 2), the formula is given by

$$\text{logit}(p_{ij}) = 1.498 + 0.072 \times \text{Length}_{ij} + a_i \quad a_i \sim N(0, 1.462^2)$$

The value of 1.498 is obtained by adding the contribution from the main term fSex to the intercept and 0.072 is the correction for the intercept for the male species ($= 0.037 + 0.035$). Just as before, we will only visualise the results for the female deer.

The random intercept a_i is assumed to be normally distributed with mean 0 and variance 1.462^2 . This means that the majority of the values (95% to be more exact) of a_i are between -1.96×1.462 and 1.96×1.462 . Figure 13.3 shows three lines.

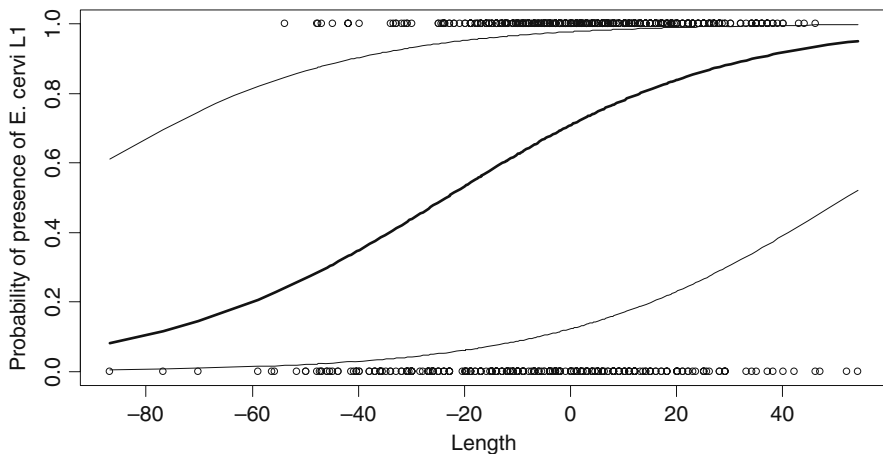


Fig. 13.3 GLMM predicted probabilities of parasitic infection along (*centred*) deer length for females at all farms. The *thick line* in the *middle* represents the predicted values for the ‘population of farms’, and the other two lines are obtained by adding and subtracting 1.95×1.462 for the random intercept to the predictor function. The space between these *two curves* shows the variation between the predicted values per farm

The thick line in the middle shows the estimated probabilities for a range of length values for the female data. These are predicted probabilities for a typical farm. Typical means that in this case $a_i = 0$. The other two lines are obtained by adding 1.96×1.462 to the predictor function and subtracting 1.96×1.462 from the predictor function. Hence, 95% of the farms have logistic curves between these two extremes. The interpretation of the graph is as follows. Go to a typical farm and sample a deer of average length ($\text{Length} = 0$). It has a probability of approximately 0.7 of having the parasite (this value is taken from the curve for the population). However, depending on which particular farm we visit, for the majority of farms this probability can be anything between 0.1 and 0.9! So, there is considerable between-farm variation. At this stage, it should be emphasised that the model can still be improved.

The code to produce the graph is as follows.

```
> g <- 0.8883697 + 0.0378608 * DeerEcervi$CLength
> p.averageFarm1 <- exp(g) / (1 + exp(g))
> I1 <- order(DeerEcervi$CLength) #Avoid spaghetti plot
> plot(DeerEcervi$CLength, DeerEcervi$Ecervi.01,
       ylab = "Probability of presence of E. cervi L1",
       xlab = "Length")
> lines(DeerEcervi$CLength[I1], p.averageFarm1[I1], lwd=3)
> p.Upp <- exp(g + 1.96 * 1.462108) / (1 + exp(g + 1.96 * 1.462108))
> p.Low <- exp(g - 1.96 * 1.462108) / (1 + exp(g - 1.96 * 1.462108))
> lines(DeerEcervi$CLength[I1], p.Upp[I1])
> lines(DeerEcervi$CLength[I1], p.Low[I1])
```

The first two lines calculate the predicted probabilities for the curve in the middle. Instead of using some complex programming, we calculated these manually. The `order` command is used to avoid a spaghetti plot. The rest of the code calculates the probabilities for the other two curves and superimposes the lines.

We mentioned earlier in this section that the GLMM can be run in at least two other libraries, and we now briefly discuss the code and the output. The mathematical details and the reason why we have different functions are discussed in Section 13.4.

The second function you can use for GLMM is the `lmer` function from the package `lme4`. The following code runs exactly the same model as before.

```
> library(lme4)
> DE.lme4 <- lmer(Ecervi.01 ~ CLength * fSex +
                  (1 | fFarm), family = binomial,
                  data = DeerEcervi)
> summary(DE.lme4)
```

The random effect is now specified by $(1 | \text{fFarm})$. We only present the results and compare it with the `glmPQL` results towards the end of this section.


```
Generalized linear mixed model fit using Laplace
Formula: Ecervi.01 ~ CLength * fSex + (1 | fFarm)
Data: DeerEcervi. Family: binomial(logit link)
```

AIC	BIC	logLik	deviance
832.6	856.1	-411.3	822.6

Random effects:

Groups	Name	Variance	Std.Dev.
fFarm	(Intercept)	2.3859	1.5446

number of obs: 826, groups: fFarm, 24

Estimated scale (compare to 1) 0.9684129

Fixed effects:

	Estimate	Std. Error	z value	Pr(> z)
(Intercept)	0.941504	0.354827	2.653	0.00797
CLength	0.038975	0.006815	5.719	1.07e-08
fSex2	0.624665	0.222848	2.803	0.00506
CLength:fSex2	0.035866	0.011348	3.161	0.00157

The standard error of the random intercepts a_i is now 1.54. The main difference between `lmer` and `glmmPQL` is that the `lmer` gives an AIC, BIC, log likelihood value, and a deviance. This makes model comparison with `lmer` easier. Standard errors, z-values, and *p*-values obtained by both methods are similar.

The last option we discuss is the `glmmML` function in the package with the same name. This package is extensively used in Chapter 21, where the presence and absence of koalas are analysed using a binomial GLMM. The following R code can be used.

```
> library(glmmML)
> DE.glmmML <- glmmML(Ecervi.01 ~ CLength * fSex,
                      cluster = fFarm, family = binomial,
                      data = DeerEcervi)
> summary(DE.glmmML)
```

In this function, the random intercept is specified with the option `cluster = fFarm`. Its output is given below. Again, we get an AIC and estimated values are similar to the other two functions, except for the residual standard error.

```
Call: glmmML(formula = Ecervi.01 ~ CLength * fSex,
family = binomial, data = DeerEcervi, cluster = fFarm)
```

	coef	se(coef)	z	Pr(> z)
(Intercept)	0.93968	0.357915	2.625	8.65e-03
CLength	0.03898	0.006956	5.604	2.10e-08
fSex2	0.62451	0.224251	2.785	5.35e-03
CLength:fSex2	0.03586	0.011437	3.135	1.72e-03

Standard deviation in mixing distribution: 1.547
Std. Error: 0.2975

Residual deviance: 822.6 on 821 degrees of freedom
AIC: 832.6

13.2.1.1 Comparison of Results

Let us now compare the results from the functions `glmmPQL`, `lmer`, and `glmmML`. For convenience, we have reproduced all estimated regression parameters and standard errors in Table 13.1. We have also added the binomial GLM and GEE results.

Note that the `lmer` and `glmmML` results are nearly the same. The `glmmPQL` method also gives very similar results. As can be expected, the GLM obtained without any correlation structure gives slightly different results; note the different sex estimate. Except for the intercept, the GEE results are also similar to the GLMM results. Further comments comparing GEEs with GLMMs can be found on p. 300 of Venables and Ripley (2002). They also mentioned the package `glme`, which apparently can do a GLMM and fix the overdispersion to a pre-set value (`glmmPQL` automatically estimates overdispersion, also if you do not want this).

Finally, we comment on the different interpretation of the parameters in a GLMM and GEE. In the GLMM in Fig. 13.3, the thick line is the length effect of a *typical* farm. Hence, the regression parameters in the GLMM are with respect to an individual farm due to the random intercept a_i . For the GEE, the regression parameters represent the effect of the population.

Table 13.1 Estimated regression parameters and standard errors obtained by `glm`, `glmmPQL`, `lmer`, `glmmML`, and GEE. Note that further differences can be obtained by changing the estimation methods within a function

	Estimates	SE		Estimates	SE
Glm			lmer		
Intercept	0.652	0.109	Intercept	0.941	0.354
Length	0.025	0.005	Length	0.038	0.006
Sex	0.163	0.174	Sex	0.624	0.222
Length × Sex	0.020	0.009	Length × Sex	0.035	0.011
glmmPQL			glmmML		
Intercept	0.888	0.337	Intercept	0.939	0.357
Length	0.037	0.006	Length	0.038	0.006
Sex	0.610	0.213	Sex	0.624	0.224
Length × Sex	0.035	0.010	Length × Sex	0.035	0.011
GEE					
Intercept	0.773	0.280			
Length	0.030	0.006			
Sex	0.476	0.217			
Length × Sex	0.027	0.014			

13.2.2 The Owl Data Revisited

In Chapters 5, 6, and 12, we used a data set from Roulin and Bersier (2007), who analysed the begging behaviour of nestling barn owls. In Chapters 5 and 6, we analysed the response variable sibling negotiation, which is defined as the number of calls just before arrival of a parent at a nest divided by the number of siblings per nest. The data were log-transformed and a Gaussian linear mixed effects model was applied, and also an additive mixed effects model with arrival time as smoother. In Chapter 5, we used nest as random effect, and in Chapter 6 an auto-regressive correlation structure was implemented. In Chapter 12, we analysed the number of calls using a GLM with a Poisson distribution (number of calls is a count) and the log-transformed number of siblings per nest was used as an offset variable in the linear predictor function. Two GEE models were applied: a GEE with the compound correlation structure between all observations from the same nest and one GEE with an auto-regressive correlation between sequential observations from the same nest per night. Here, we will analyse these data in yet another way, namely, with a GLMM using the Poisson distribution (number of calls is a count) and also with a GAMM.

The Poisson GLMM for these data is given by the following:

$$\begin{aligned}
 \text{NCalls}_{is} &\sim \text{Poisson}(\mu_{is}) \Rightarrow E(\text{NCalls}_{is}) \sim \mu_{is} \\
 \eta_{is} &= \text{offset}(\text{LBroodSize}_{is}) + \beta_1 \times \text{SexParent}_{is} + \beta_2 \times \text{FoodTreatment}_{is} \\
 &\quad + \beta_3 \times \text{ArrivalTime}_{is} + \beta_4 \times \text{SexParent}_{is} \times \text{FoodTreatment}_{is} \\
 &\quad + \beta_5 \times \text{SexParent}_{is} \times \text{ArrivalTime}_{is} + a_i \\
 a_i &\sim N(0, \sigma_a^2) \\
 \log(\mu_{is}) &= \eta_{is}
 \end{aligned}$$

The first line states that the number of calls for observation s at nest i , NCalls_{is} , is Poisson distributed with mean μ_{is} . The linear predictor function looks similar to that of an ordinary Poisson GLM, except that we use the log transformed broodsize as an offset (Chapter 9), and there is the a_i bit at the end. Its purpose is exactly the same as the random intercept for farm in Section 13.2.1; it allows for a different intercept for each nest. We assume that it is normally distributed with mean 0 and variance σ_a^2 . We use `lmer` to fit the model. The same model in terms of explanatory variables is used as in Chapters 5, 6, and 12. The following code was used.

```

> library(AED) ; data(Owls)
> library(nlme)
> Owls$NCalls <- Owls$SiblingNegotiation
> Owls$LBroodSize <- log(Owls$BroodSize)
> Owls$fNest <- factor(Owls$Nest)
> O1.lmer <- lmer(NCalls ~ offset(LBroodSize) +
  SexParent * FoodTreatment+

```

```

      SexParent * ArrivalTime + (1 | fNest),
      data = Owls, family = poisson)
> summary(O1.lmer)

Generalized linear mixed model fit using Laplace
Formula: NCalls ~ offset(LBroodSize) + SexParent *
      FoodTreatment + SexParent * ArrivalTime + (1 | fNest)
Data: Owls
Family: poisson(log link)
      AIC   BIC logLik deviance
3329 3359  -1657     3315

Random effects:
Groups Name      Variance Std.Dev.
fNest  (Intercept) 0.20980  0.45803
number of obs: 599, groups: fNest, 27

Estimated scale (compare to 1)  2.332117

Fixed effects:
              Estimate Std. Error z value Pr(>|z|)
(Intercept)      3.58145    0.36262   9.877  <2e-16
SexParentMale      0.38785    0.44861   0.865  0.3873
FoodTreatmentSatiated -0.66680    0.05610 -11.886  <2e-16
ArrivalTime      -0.11948    0.01440  -8.298  <2e-16
SexParentMale:FoodTreatment.Sat  0.13239    0.07044   1.880  0.0602
SexParentMale:ArrivalTime -0.01647    0.01836  -0.897  0.3697

Correlation of Fixed Effects:
              (Intr) SxPrnM FdTrtS ArrvlT SPM:FT
SexParentMl -0.739
FdTrtmntStt -0.077  0.062
ArrivalTime -0.964  0.759  0.017
SxPrntM:FTS  0.055 -0.073 -0.767 -0.010
SxPrntMl:AT  0.737 -0.995 -0.012 -0.765  0.012

```

The correlation between the intercept and the slope for arrival time is rather large (-0.964). This is because arrival time was not centred. In case of numerical problems, centring continuous variables may help. The model can be further simplified because the interaction between sex of the parent and arrival time is not significant. You can reach the same conclusion by dropping this interaction, refitting the model, and comparing the change in likelihood.

```

> O2.lmer <- lmer(NCalls ~ offset(LBroodSize) +
      SexParent * FoodTreatment +
      ArrivalTime + (1 | fNest), data = Owls,
      family = poisson)
> anova(O1.lmer, O2.lmer)

```

Models:

```

O2.lmer: NCalls ~ offset(LBroodSize) + SexParent * FoodTreatment +
      SexParent + ArrivalTime + (1 | fNest)

```

```
O2.lmer: NCalls ~ offset(LBroodSize) + SexParent * FoodTreatment +
              SexParent * ArrivalTime + (1 | fNest)
```

	Df	AIC	BIC	logLik	Chisq	Chi	Df	Pr(>Chisq)
O2.lmer	6	3327.4	3353.7	-1657.7				
O1.lmer	7	3328.6	3359.3	-1657.3	0.8029		1	0.3702

You can repeat this process and drop the second two-way interaction as it is not significant neither and the same holds for the main term sex of the parent. This means that we end up with a GLMM that only contains the two main terms arrival time and food treatment, and nest as random effect. The code and relevant numerical output is given below.

```
> O3.lmer <- lmer(NCalls ~ offset(LBroodSize) +
                  FoodTreatment + ArrivalTime + (1 | fNest),
                  data = Owls, family = poisson)
> summary(O3.lmer)
```

Generalized linear mixed model fit using Laplace

```
Formula: NCalls ~ offset(LBroodSize) + FoodTreatment + ArrivalTime +
        (1 | fNest)
```

Data: Owls

Family: poisson(log link)

	AIC	BIC	logLik	deviance
	3328	3346	-1660	3320

Random effects:

Groups	Name	Variance	Std.Dev.
fNest	(Intercept)	0.20854	0.45666

number of obs: 599, groups: fNest, 27

Estimated scale (compare to 1) 2.331403

Fixed effects:

	Estimate	Std. Error	z value	Pr(> z)
(Intercept)	3.802867	0.243846	15.60	<2e-16
FoodTreatmentSatiated	-0.589608	0.035941	-16.41	<2e-16
ArrivalTime	-0.128840	0.009258	-13.92	<2e-16

You can continue the analysis by trying to add a random slope for arrival time or even a generalised additive mixed model in which arrival time is fitted as a smoother. The latter model is specified by

$$NCalls_{is} \sim \text{Poisson}(\mu_{is}) \Rightarrow E(NCalls_{is}) \sim \mu_{is}$$

$$\eta_{is} = \text{offset}(LBroodSize_{is}) + \beta_1 \times \text{FoodTreatment}_{is} + s(\text{ArrivalTime}_{is}) + a_i$$

$$\log(\mu_{is}) = \eta_{is}$$

$$a_i \sim N(0, \sigma_a^2)$$

Arrival time is now fitted with a smoother. To implement this model in R, we need the gamm function from the mgcv package.

```
> library(mgcv)
> O4.gamm <- gamm(NCalls ~ offset(LBroodSize) +
  FoodTreatment + s(ArrivalTime),
  random = list(fNest =~ 1), data = Owls,
  family = poisson)
```

The object `O4.gamm` has two items, a `$gam` and a `$lme` bit. Using the words from the `gamm` help files, some of the output in the `$lme` looks rather bizarre. Let us start easy with the `$gam` part. We can use the following commands:

```
> summary(O4.gamm$gam, cor = FALSE)
> anova(O4.gamm$gam)
> plot(O4.gamm$gam)
```

We only present the output of the first command as the second one shows merely a condensed version of it (it is useful if you have nominal variables with more than two levels).

```
Family: poisson. Link function: log
Formula: NCalls ~ offset(LBroodSize) + FoodTreatment + s(ArrivalTime)
```

Parametric coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	0.60731	0.07716	7.870	1.70e-14
FoodTreatmentSatiated	-0.57624	0.07949	-7.249	1.32e-12

Approximate significance of smooth terms:

	edf	Est.rank	F	p-value
s(ArrivalTime)	6.781	9	9.724	6.23e-14

```
R-sq.(adj) = 0.211 Scale est. = 5.1031 n = 599
```

The scale estimator is the variance of the working residuals inside the algorithm. The information on the parametric coefficients tells us that the food treatment is significantly different from 0 at the 5% level. To be more specific, observations that received the satiated treatment had an intercept that is 0.57 lower than for food-deprived nests. The arrival time smoother had 6.7 degrees of freedom, and is significant. The `plot` command presents this smoother, see Fig. 13.4. Note that the shape of the smoother is very similar to the one in Fig. 5.8! In order to get the fitted values for a *typical* observation, we need to add the intercept (0.607), the food treatment effect (−0.576 for satiated observations), and the offset.

Finally, let us focus on the `$lme` part of the output; it is a little intimidating though! This reason for this is that `gamm` is repeatedly calling `glmPQL` if a non-Gaussian distribution, or non-identity link function, is used. For Gaussian distributions with the identity link, it calls `lme`. Now here is the confusing bit: It's possible to show that the smooth terms of the GAMM can be presented in the mixed-effects form (Wood, 2006, p. 317), namely, $\mathbf{X}_F \times \boldsymbol{\beta}_F + \mathbf{Z} \times \mathbf{b}$, where \mathbf{X}_F is a matrix

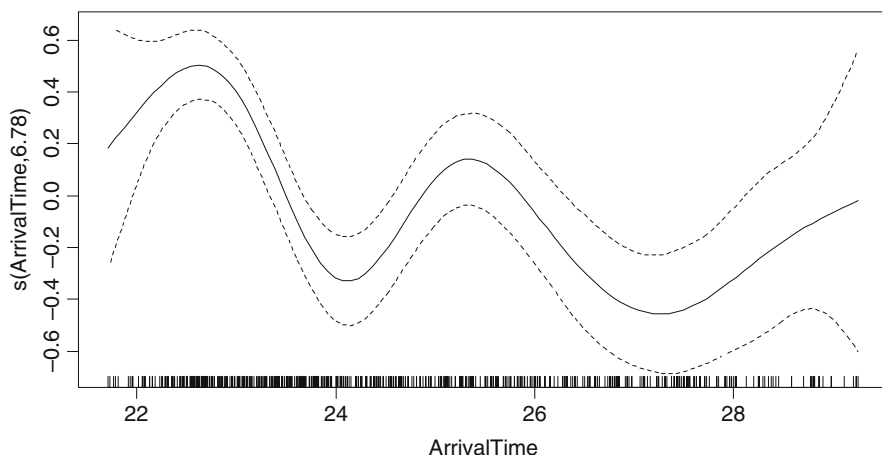


Fig. 13.4 Estimated smoother for the GAMM. Note that the smoother is centred around zero. To get fitted values, you need to add the intercept, food treatment effect, the offset, and the contribution from the random effect for a nest. The smoother shows two bumps: one at 22.30 and one at about 01.30 (in real time). An explanation can be sought in the biology, but before you do this, you need to exclude the possibility that there is still somehow a nest effect in here. Perhaps the bumps are due to activity at only a group of nests during parts of the night. The random intercept will take care of changes in mean values of the number of calls per nest, but not of changes *in the relationship* between arrival time and calls at different nests. Make boxplots of nest activity during the night (are owls active during the entire night or only part of the night), and inspect the residuals from a random intercept and slope GLMM for any patterns

containing the smoother basis; see Chapter 3. \mathbf{Z} is a matrix containing the random effects (Chapter 5) derived from the smoother basis and penalty matrix (presenting the penalty as a quadratic form) and \mathbf{b} are the random effects, which are assumed to be normally distributed with mean 0 and variance \mathbf{I}/λ . Hence, the GAMM is written in parametric terms and the penalty λ , also called the wiggly component in Wood (2006), is used in the random component. This makes the `lme` summary part rather bizarre; see below.

```
> summary(O4.gamm$lme)

Linear mixed-effects model fit by maximum likelihood
Data: strip.offset(mf)
    AIC BIC logLik
    NA  NA    NA

Random effects:
Formula: ~Xr.1 - 1 | g.1
Structure: pdIdnot
           Xr.11  Xr.12  Xr.13  Xr.14  Xr.15  Xr.16
Xr.17  Xr.18
StdDev: 19.57691 19.57691 19.57691 19.57691 19.57691 19.57691
19.57691 19.57691
```

```

Formula: ~1 | fNest %in% g.1
          (Intercept) Residual
StdDev:    0.2935566 2.259006

Variance function:
  Structure: fixed weights
  Formula: ~invwt

Fixed effects: y ~ X - 1 + offset(LBroodSize)
              Value Std.Error DF   t-value p-value
X(Intercept)    0.6073122 0.0771576 570   7.871062  0.0000
XFoodTreatmentSatiated -0.5762376 0.0795368 570  -7.244919  0.0000
Xs(ArrivalTime)Fx1      0.6378219 0.5925502 570   1.076401  0.2822

Correlation:
              X(Int) XFdTrS
XFoodTreatmentSatiated -0.365
Xs(ArrivalTime)Fx1      -0.050  0.058

Standardized Within-Group Residuals:
      Min      Q1      Med      Q3      Max
-1.5701868 -0.7615271 -0.2231992  0.5589554  4.9173689

Number of Observations: 599
Number of Groups:
      g.1 fNest %in% g.1
      1          27

```

The interesting bit from this output is the variance of the random intercept for nests; it is equal to 0.293^2 . The residual standard deviation (of the working residuals) was also presented earlier using `summary(O4.gamm$gam)`, except that it was presented as a variance. Because the `glmmPQL` routine is used, no AIC is given. The random effects part gives information on \mathbf{I}/λ . It is probably easier to obtain this via

```
> intervals(O4.gamm$lme, which = "var-cov")
```

Approximate 95% confidence intervals

```

Random Effects:
  Level: g.1
              lower      est.      upper
sd(Xr.1 - 1)  98.3855 383.2554 1465.705
  Level: fNest
              lower      est.      upper
sd((Intercept)) 0.1911345 0.2935566 0.4508631

Within-group standard error:
  lower      est.      upper
2.130732 2.259006 2.395004

```


The 383.255 is the square of 19.57^2 , which we already met in the `lme` summary output. To be precise, 383.255 is equal to σ^2/λ , where σ^2 is the variance of the (working) residuals. This gives $\lambda = 2.259^2/383.255 = 0.013$. However, we already know the amount of smoothing from the `anova (O4.gamm$gam)` command; hence, this is probably not worthwhile to mention in a report or paper, unless you want to focus on the *approximate* confidence intervals.

The information in the summary `lme` output on the fixed effects bit is not interesting neither; just use the `anova (O4.gamm$gam)` command for clearer information on the significance on individual terms. Further details can be found in Sections 6.5–6.7 in Wood (2006). He also presented residual plots, where the residuals were obtained from the `$lme` bit. For our model, these are obtained via

```
> E4 <- resid(O4.gamm$lme, type = "normalized")
```

These take into account the random effects. You can plot these residuals versus arrival time, and see whether there is any auto-correlation structure left in the data. If there is, try adding an auto-regressive correlation within a specific nest and day combination; see also Chapter 6. Other interesting validation tools are to plot square-root-transformed fitted values versus square-root-transformed observed values (should be a straight line), Pearson residuals versus square-root-transformed fitted values (should form a band with no patterns), and raw residuals (observed versus fitted values) versus square-root-transformed fitted values (should show a clear cone); see also Fig. 6.11 in Wood (2006) and associated code.

13.2.3 A Word of Warning

Although the analyses presented in the previous two subsections look relatively simple, you should not be too enthusiastic with all the p -values, AICs, and nested model comparisons. All these values are rather approximate! Furthermore, at the time of writing, the `lme4` package was under development with no support for the `correlation` argument in Poisson GLMMs, and the `resid` function did not give residuals for a Poisson GLMM. Type: `resid (O3.lmer)`; it gives: Error: 'resid' is not implemented yet. This does not mean that a package that does give residuals for a Poisson GLMM is any better; it is just that it is not trivial to calculate them.

Summarising, you should be very careful with p -values close to magic 5% borderline in GLMMs and GAMMs, even more careful as in ordinary GLMs and GAMs.

13.3 The Underlying Mathematics in GLMM

This section may be skipped by readers not interested in the underlying mathematical details. In this section, we first explain the difference between conditional

and unconditional distributions, then present the likelihood function for the GLMM models, and finally discuss how it is calculated.

The difference between conditional and unconditional distributions is best explained within a Gaussian context. Recall from Chapter 5 that the linear mixed model is given by

$$Y_{ij} = \alpha + X_i \times \beta + Z_i \times b_i + a_i + \varepsilon_{ij}$$

Note that the random term b_i is assumed to be normally distributed with a mean of 0 and covariance matrix D (actually, in this case, D just contains one element, but it is easier to use matrix notation as it is more general). The same holds for the second random term ε_{ij} . Its covariance matrix is given by Σ_i . The mean value of Y_{ij} , where the mean is taken over all observations i , is given by

$$E(Y_{ij}) = X_i \times \beta.$$

We now introduce a new concept: the conditional mean of Y_{ij} . It is the mean value of Y_{ij} for given b_i . So, we pretend we know the value of b_i . The mathematical notation for this is $E(Y_{ij}|b_i)$. The vertical line followed by b_i means that it is ‘conditional on b_i ’. Its value is given by

$$E(Y_{ij}|b_i) = X_i \times \beta + Z_i \times b_i$$

We can do the same for the variance of Y_{ij} . The conditional variance of Y_{ij} is given by $\text{cov}(Y_{ij}|b_i) = \Sigma_i$, and the unconditional variance is

$$\text{cov}(Y_{ij}) = \mathbf{Z}_i \times \mathbf{D} \times \mathbf{Z}_i' + \Sigma$$

The principle of conditional mean and variances can be extended to distributions. Hence, we can specify a Poisson or Binomial distribution conditional on b_i . This allows us to define a Poisson GLMM with the following three steps.

1. Conditional on the random effects b_i , the counts Y_{ij} are assumed to be Poisson distributed with mean $\mu_{ij}|b_i$. As a consequence, we have the following relationship between the mean and the variance of Y_{ij} : $E(Y_{ij}|b_i) = \text{var}(Y_{ij}|b_i)$.
2. The relationship between the conditional mean and the explanatory variables is determined by the log link $\log(\mu|b_i) = \alpha + X_i \times \beta + Z_i \times b_i$.
3. The random effects b_i are assumed to be normally distributed with mean 0 and covariance matrix D .

The only difference with an ordinary Poisson GLM model is the specification of a conditional distribution, and the inclusion of the random term. The Binomial GLMM can be defined in a similar way, namely,

1. Conditional on the random effects b_i , the presence and absence data Y_{ij} are assumed to be binomial distributed with probability $p_{ij}|b_i$. As a consequence,

we have the following relationship between the mean and the variance of Y_{ij} :
 $E(Y_{ij}|b_i) = p_{ij}|b_i$ and $\text{var}(Y_{ij}|b_i) = p_{ij}|b_i \times (1 - p_{ij}|b_i)$

2. The relationship between the conditional mean and the explanatory variables is determined by the logistic link $\text{logit}(p_{ij}|b_i) = \alpha + X_i \times \beta + Z_i \times b_i$.
3. The random effects b_i are assumed to be normally distributed with mean 0 and covariance matrix D .

It is also possible to formulate the GLMM in an abstract formulation using the same formula as presented in Chapter 9. See, for example, Fitzmaurice et al. (2004). We do not do that here. So far, the mathematics is not that intimidating. However, we have reached the point where the problems begin as we now look at the formulation of the likelihood function. In ordinary GLM models, the maximum likelihood function is specified and derivatives with respect to parameters are calculated and set to 0. The parameters that maximise the likelihood are then found by solving these equations. As output, we get estimated parameters, standard errors, a deviance, and an AIC among other information. In GLMM, this is a considerably more complicated process, and the output may not contain a deviance and AIC. The reason for this is that the likelihood function for the GLMM has the following form

$$L(\beta, D) = \prod_i \int f(Y_{ij}|b_i) \times f(b_i) db_i$$

The symbol that looks like two vertical roman pillars with one horizontal pillar on top of it represents a multiplication operator. The second one is the integral. The terms $f(Y_{ij})$ and $f(b_i)$ are distribution functions. As explained above, in ordinary GLM models, we take the derivative of $L()$ with respect to the parameters, and after some basic algebra and an iterative algorithm, we end up with the parameters. This process does not work well for the GLMM and there are no simple solutions for the parameter estimates.

One option is to use numerical integration techniques and replace the integral by a summation. This is called Gaussian quadrature. But various choices have to be made in this process, and the higher the requested accuracy of the solutions, the higher the computational burden. For complicated models, it may not converge at all. Chapter 10 in McCulloch and Searle (2001) describes a series of methods for getting parameter estimates, for example, numerical quadrature (numerical integration of the integral) with various flavours like Markov chain Monte Carlo algorithms, stochastic approximation algorithms, simulated maximum likelihood, and penalised quasi-likelihood (PQL) methods. Key concepts in the last approach are Laplace's approximation and Taylor series expansions. To fully understand what these different methods are doing requires a high degree of mathematical knowledge.

The main message to take away from this is the difficulty in obtaining parameter estimates in GLMM. It depends on the package and method used. Some packages do not produce a deviance and AIC; hence, model selection is based on standard errors and Wald statistics. A few packages do produce a deviance and AIC, but interpretation should still be done with care.

A different approach to GLMM is given in Chapter 23, where we discuss Bayesian approaches.