Chapter 4
Dealing with Heterogeneity

This chapter, and the following three chapters, discuss solutions to the problems introduced in Chapters 2 and 3: heterogeneity, nested data, temporal correlation, and spatial correlation. We use both the linear regression model and the additive model as starting points. Figure 4.1 shows an overview of the methods we discuss in Chapters 4, 5, 6, and 7. In all these chapters, the model consists of a fixed term and a random term. The fixed term describes the response variable Y as a function of the explanatory variables via $\alpha + \beta_1 \times X_1 + \ldots + \beta_q \times X_q$ in linear regression or $\alpha + f_1(X_1) + \ldots + f_q(X_q)$ in additive modelling. This part of the model is described in Appendix A and Chapter 3. The random part contains components that allow for heterogeneity, nested data (random effects), temporal correlation, spatial correlation, and a real random term. It is also possible to have a combination of these components.

If the random part only contains the real random term, we are back to linear regression or additive modelling. If it allows for nested data, the resulting model is called a mixed effects model. If it only allows for heterogeneity, we call it a generalised least squares (GLS) model. This is essentially a weighted linear regression. GLS is the subject of this chapter. It is tempting to call the whole equation in Fig. 4.1 mixed effects modelling (or just mixed modelling), even if it only contains the heterogeneity bit, but strictly speaking this is wrong. However, as software routines for GLS, auto-correlation and nested data can all use the same R package, and sometimes the same routines, then it is easy to get confused about names.

We closely follow Chapter 5 in Pinheiro and Bates (2000), and the first 5 chapters of Verbeke and Molenberghs (2000). We also made extensive use of Diggle et al. (2002).We strongly recommend these books, as they provide a good technical explanation and a more unified overview of mixed modelling techniques than we have provided, albeit at a much higher mathematical level. Another good ecological source for the linear mixed model is Schabenberg and Pierce (2002), but it does not contain R code.

For the additive mixed modelling, Ruppert et al. (2003) and Wood (2006) are some of the few available books. But again, these are rather technical.

If you are willing to read non-ecological textbooks, we strongly recommend West et al. (2006), as it contains a series of case studies. However, a basic familiarity

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_4,

$$Y = \underbrace{\alpha+\beta_1 X_1+\cdots+\beta_q X_q \quad \alpha+f_1(X_1)+\cdots+f_q(X_q)}_{\text{fixed part}} + \underbrace{\begin{array}{l}\textbf{Heterogeneity}\\ \text{Nested data (random effects)}\\ \text{Temporal correlation}\\ \text{Spatial correlation}\\ \text{Random noise}\end{array}}_{\text{random part}}$$

Fig. 4.1 Outline of the different methodologies discussed in Chapters 4, 5, 6, and 7. The fixed part consists of the explanatory variables as we know from linear regression or additive modelling. The random part consists of a real random term and terms that allow for heterogeneity, nested data (random effects), temporal correlation, or spatial correlation. The subject of this chapter is heterogeneity

with linear mixed modelling is recommended as their first chapter summarises the underlying theory rather quickly. Other useful books, but mainly focussed on economics and social science are Goldstein (2003), Raudenbush and Bryk (2002), Snijders and Bosker (1999), and at a higher mathematical level, Jiang (2007).

The confusing aspects of most of these books are the wide range of different names and underlying mathematical notation. Mixed modelling, multilevel analysis, hierarchical linear models, and repeated measurements are just a few of the names that all refer to the same set of models.

4.1 Dealing with Heterogeneity

4.1.1 Linear Regression Applied on Squid

Several examples in Chapters 2 and 3 showed residual spread varying per stratum (level) of a nominal variable, or increasing or decreasing along an explanatory variable. For example, the spread in pelagic bioluminescent data (Chapter 2) decreased at deeper depths, and both the *Hediste diversicolor* and wedge clam data sets (Chapter 2) showed different residual spread per stratum for some of the variables (month, biomass, nutrient). This violates the homogeneity of variance assumption, one of the most important assumptions of linear regression and additive modelling. Ignoring this problem may result in regression parameters with incorrect standard errors, and an F statistic no longer F distributed and the t statistic not following a t distribution. This invalidates the statistics used in Chapters 2 and 3 for assessing statistical significance (Wooldridge, 2006). In this section, we provide several solutions to heterogeneity. The easiest solution is a data transformation, but we try to avoid this for as long as possible. In our view, heterogeneity is interesting ecological information that you should not throw away, just because it is statistically inconvenient. With a 'little' bit of extra mathematical effort, heterogeneity can be incorporated into the models and can provide extra biological information.

To illustrate the methods, we use data published by Smith et al. (2005), who looked at seasonal patterns in reproductive and somatic tissues in the squid *Loligo*

forbesi. They used several variables on female and male squid, but in this chapter, we only use the dorsal mantle length (in mm) and testis weight from 768 male squid. The aim is to model the testis weight as a function of the dorsal mantle length (DML) and the month recorded. The idea behind the original analysis was to investigate the role of endogenous and exogenous factors affecting sexual maturation, more specifically to determine the extent to which maturation is size-related and seasonal. Further biological information can be found in Smith et al. (2005). Our starting point is a linear regression model of the form (in words):

$$\begin{aligned} \text{Testisweight}_i &= \text{intercept} + \text{DML}_i + \text{Month}_i + \text{DML}_i{:}\,\text{Month}_i \\ &\quad + \text{residuals}_i \end{aligned} \tag{4.1}$$

Month is used as a nominal variable (with 12 levels) and is DML fitted as a continuous variable. The notation ':' is used for the interaction between DML and Month. Previous work on the related species *Loligo vulgaris* showed graphically that maturity was a function of both size and season, and that size-at-maturity differed between seasons (Raya et al., 1999). The index i runs from 1 to 768. The crucial assumption in Equation (4.1) is that the residuals are normally distributed with a mean of 0 and the variance is σ^2. In mathematical notation

$$\varepsilon_i \sim N(0, \sigma^2)$$

where ε_i are the residuals. The important thing is that $\text{var}(\varepsilon_i) = \sigma^2$. The following R code loads the data, applies linear regression, and produces the validation graphs in Fig. 4.2. Note that there is a clear violation of homogeneity.

```
> library(AED); data(Squid)
> Squid$fMONTH <- factor(Squid$MONTH)
> M1 <- lm(Testisweight ~ DML * fMONTH, data = Squid)
> op <- par(mfrow = c(2, 2), mar = c(4, 4, 2, 2))
> plot(M1, which = c(1), col = 1, add.smooth = FALSE,
       caption = "")
> plot(Squid$fMONTH, resid(M1), xlab = "Month",
       ylab = "Residuals")
> plot(Squid$DML, resid(M1), xlab = "DML",
       ylab = "Residuals")
> par(op)
```

The `DML * fMONTH` fits the main terms `DML` and `MONTH` (as a factor) and the interaction between these two variables ('$*$' replaces the ':' from the word equation to denote interaction). Alternatively, code that does the same is `DML + fMONTH + DML:fMONTH`. This keeps the notation similar to the one we used in Equation (4.1). By default, the `plot` command produces four graphs (see Chapter 2), but the `which = c (1)` ensures that only the residuals versus fitted values are plotted. We decided not to add a smoothing curve `(add.smooth = FALSE)`

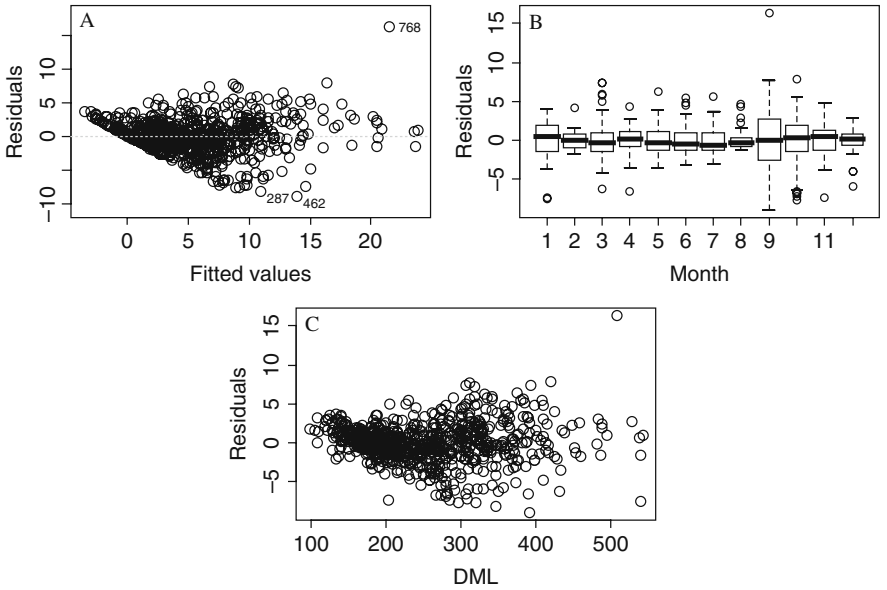

Fig. 4.2 **A**: Residuals versus fitted values. **B**: Residuals versus month. Because month is a nominal variable, boxplots are produced. **C**: Residuals versus DML. Panel **A** shows that there is clear violation of heterogeneity. Panels **B** and **C** were made to detect why there is heterogeneity

and omit the caption `(caption = "")`. All other commands are discussed in Chapters 2 and 3.

The numerical output (not shown here) shows that all regression parameters are significantly different from 0 at the 5% level. The problem is that we cannot trust these results as we are clearly violating the homogeneity assumption (note the cone shape pattern of the residuals in Fig. 4.2A). This means that the assumption that the residuals are normally distributed with mean 0 and variance σ^2 is wrong. However, in this case, the homogeneity clearly has an identifiable structure; the larger the length (DML), the larger the variation (Fig. 4.2C). So, instead of assuming that the residuals have variance $\text{var}(\varepsilon_i) = \sigma^2$, it might make more sense to assume that $\text{var}(\varepsilon_i)$ increases when DML_i increases. We can implement this in various mathematical parameterisations, and we discuss these next.

4.1.2 The Fixed Variance Structure

The first option is called the *fixed variance*, it assumes that $\text{var}(\varepsilon_i) = \sigma^2 \times \text{DML}_i$, and as a result we have

$$\varepsilon_i \sim N(0, \sigma^2 \times DML_i) \quad i = 1, \ldots, 768 \tag{4.2}$$

Such a variance structure allows for larger residual spread if DML increases. And the good news is that there are no extra parameters involved! Technically, this model is fitted using the generalised least squares (GLS) method, and the technical aspects of this method are discussed later in this chapter. To fit a GLS in R, the function `gls` from the `nlme` package can be used. The variance structure (and any of the others we discuss later) can be selected by specifying the `weights` arguments in the `gls` function. In fact, running the gls code without a `weights` option, gives you the same linear regression model already seen in Equation (4.1). The following R code applies the linear regression model in (4.1) and also the GLS with the fixed variance structure in Equation (4.2). The reason we refitted the linear regression model in Equation (4.1) with the `gls` function was to avoid a warning message in the anova comparison.

```
> library(nlme)
> M.lm <- gls(Testisweight ~ DML * fMONTH, data=Squid)
> vf1Fixed <- varFixed(~DML)
> M.gls1 <- gls(Testisweight ~ DML * fMONTH,
                weights = vf1Fixed, data = Squid)
> anova(M.lm, M.gls1)
```

The command `varFixed (~DML)` ensures a variance that is proportional to DML, and it needs to be specified via the `weights` argument in the `gls` function. Finally, the `anova` command gives

```
       Model df      AIC      BIC    logLik
M.lm       1 25 3752.084 3867.385 -1851.042
M.gls1     2 25 3620.898 3736.199 -1785.449
```

The models are not nested; so no log-likelihood ratio test statistic is given, but the AIC clearly favours the model with the fixed variance in Equation (4.2). Note that both models have the same number of parameters! You can also use the command `AIC(M.lm, M.gls1)`.

4.1.3 The VarIdent Variance Structure

Now, just for a moment, we will forget about the residual spread increasing for larger DML values. So instead of recognising from Fig. 4.2C that the spread increases for larger DML values, we now realise from Fig. 4.2B that the spread also differs per month. To incorporate this pattern into the model, it is better to slightly change the indices used in the model notation:

$$\text{Testisweight}_{ij} = \text{intercept} + DML_{ij} + \text{Month}_j + \text{DML}_{ij}{:}\text{Month}_j + \text{residuals}_{ij} \quad (4.3)$$

Testisweight_{ij} is the testis weight of the ith observation in month j. This is exactly the same model as in Equation (4.1); we have only changed notation of the indices.

However, the new notation makes it easier to formulate the variance structure with different spread per stratum:

$$\varepsilon_{ij} \sim N(0, \sigma_j^2) \quad j = 1, \ldots, 12 \tag{4.4}$$

So, we now have $\text{var}(\varepsilon_{ij}) = {\sigma_j}^2$, and each month is allowed to have a different variance. The following code implements different variances per stratum for month and applies the anova comparison.

```
> vf2 <- varIdent(form= ~ 1 | fMONTH)
> M.gls2 <- gls(Testisweight ~ DML*fMONTH, data =Squid,
                weights = vf2)
> anova(M.lm, M.gls1, M.gls2)
```

The output of the `anova` command is given by:

```
       Model df      AIC      BIC    logLik   Test  L.Ratio p-value
M.lm       1 25 3752.084 3867.385 -1851.042
M.gls1     2 25 3620.898 3736.199 -1785.449
M.gls2     3 36 3614.436 3780.469 -1771.218 2 vs 3 28.46161  0.0027
```

We have decreased the font size of the numerical output to ensure it fits the page. The first two lines in the output are the same as above. The AIC of the model using the different variances per month is lower. You can also use the command `AIC(M.lm, M.gls1, M.gls2)`.

Notice that due to the variance structure in Equation (4.4), we now have to estimate 11 more parameters. We discuss below why it is not 12. We also get a log likelihood ratio comparing the variance structures in Equations (4.2) and (4.4). However, as these models are not nested, it is better not to use the log likelihood ratio. However, comparing models (4.1) and (4.4) does make sense as they are both nested. The null-hypothesis is

$$H_0 : \sigma_1^2 = \sigma_2^2 = \sigma_3^2 = \ldots = \sigma_{12}^2$$

with the alternative that they are not equal to each other. The R code to carry out this test and the resulting output is given below.

```
> anova(M.lm, M.gls2)
```

```
       Model df      AIC      BIC    logLik   Test  L.Ratio p-value
M.lm       1 25 3752.084 3867.385 -1851.042
M.gls2     2 36 3614.436 3780.469 -1771.218 1 vs 2 159.6479  <.0001
```

You can see the log likelihood ratio test indicates that the model with different variances per month is better, allowing us to reject the null hypothesis that all variances are the same. The `summary(M.gls2)` command gives the different variances (along with lots of other information).

```
> summary(M.gls2)
...
Variance function:
Structure: Different standard deviations per stratum
Formula: ~1 | fMONTH
Parameter estimates:
2    9    12   11   8    10   5    7    6    4
1.00 2.99 1.27 1.50 0.98 2.21 1.63 1.37 1.64 1.42
1    3
1.95 1.97
...
Residual standard error: 1.27
```

The numbers under the months (2, 9, 12, etc.) are multiplication factors. They show the ratio with the estimated residual standard error (1.27), the estimator for σ. Let us call this estimator s; hence, $s = 1.27$. One multiplication factor is set to 1 (in this case month 2). In month 9, the variance is $2.99 \times s$, in month 12 it is $1.27 \times s$, etc. You can also change the nominal variable fMONTH and set January to the baseline. Note that months 9 and 10, and 3 have the highest ratios indicating that in these months there is more residual variation.

If you have two nominal explanatory variables, say month and location, and the spread differs for all stratum, then you can use `varIdent(form= ~ 1|fMONTH * factor(LOCATION))`. But we don't have location information for the squid data.

So, which option is better: different spread per month or different spread along DML? If in Fig. 4.2A, the smaller fitted values are from months with less spread and the larger fitted values are from months with higher spread, then using different variances per month makes more sense. The following code produces a graph like Fig. 4.2A and colours observations of the same month:

```
> plot(M.lm,which = c(1), col = Squid$MONTH,
       add.smooth = FALSE, caption = "")
```

The `col = Squid$MONTH` part ensures that observations of the same month have the same colour. This approach works here because MONTH is coded with values 1–12. If you coded it as 'January', 'February', etc. then you would need to make a new vector with values 1, 2, 3, etc.; see, for example, Dalgaard (2002) on how to do this. Although not presented here, the graph does not show any clear grouping.

Let us try to understand what is really going on. The R code below makes a coplot (explained in Chapter 2) of the residuals versus DML, conditional on month for the linear regression model in Equation (4.1). The resulting coplot is given in Fig. 4.3. The residual variation differs per month, but in some months (e.g. 3, 9, and 10) the residual spread also increases for larger DML values. So, both are influential: residual spread is influenced by both month and length!

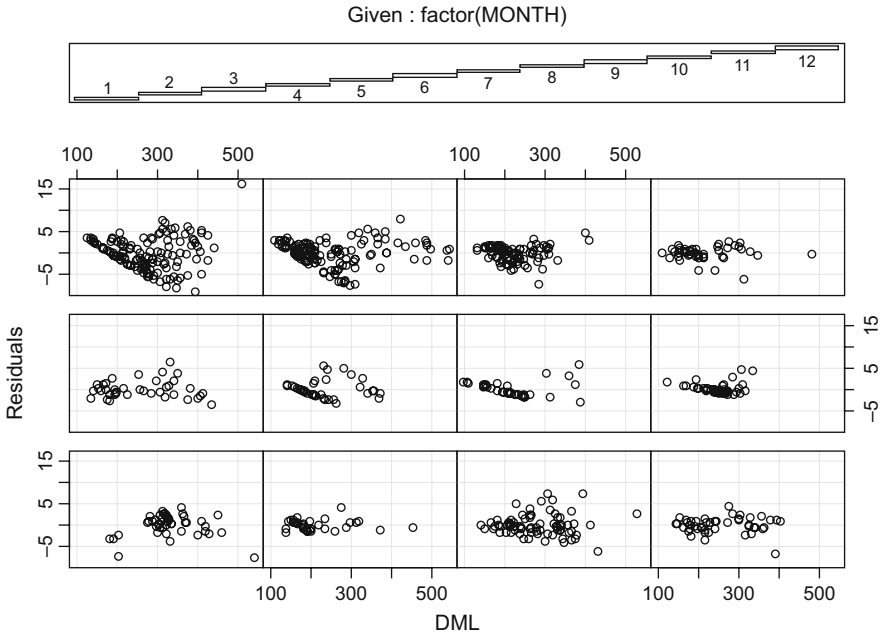

Fig. 4.3 Coplot of residuals obtained by the linear regression model in Equation (4.1) versus DML conditional on month. The lower left panel corresponds to month 1, the lower right to month 4, and the upper right to month 12. Note that some months show clear heterogeneity, and others do not. Sample size may also be an issue here!

```
> E <- resid(M.lm)
> coplot(E ~ DML | fMONTH, data = Squid)
```

Before discussing how to combine both types of variation (variation linked with DML and variation linked with Month), we introduce a few more variance structures. In all these structures, the variance of the residuals is not necessarily equal to σ^2, but is a function of DML and/or month.

An explanatory variable that is used in the variance of the residuals is called a *variance covariate*. The trick is to find the appropriate structure for the variance of ε_{ij}. The easiest approach to choosing the best variance structure is to apply the various available structures in R and compare them using the AIC or to use biological knowledge combined with some informative graphs like the coplot. Some of the variance functions are nested, and a likelihood ratio test can be applied to judge which one performs better for your data.

4.1.4 The varPower Variance Structure

So far, we have looked at the `varFixed` and `varIdent` variance structures. Next we look at the 'power of the covariate' variance structure. It uses the R

function varPower. For the squid data, a potential power of the covariate variance structure is

$$\varepsilon_{ij} \sim N(0, \sigma^2 \times |DML_{ij}|^{2\delta}) \tag{4.5}$$

Hence, $\text{var}(\varepsilon_{ij}) = \sigma^2 \times |DML_{ij}|^{2\delta}$. The variance of the residuals is modelled as σ^2, multiplied with the power of the absolute value of the variance covariate DML. The parameter δ is unknown and needs to be estimated. If $\delta = 0$, we obtain the linear regression model in Equation (4.1), meaning (4.1) and (4.5) are nested, and therefore the likelihood ratio test can be applied to judge which one is better. For $\delta = 0.5$ and a variance covariate with positive values, we get the same variance structure as specified in Equation (4.2). But if the variance covariate has values equal to 0, the variance of the residuals is 0 as well. This causes problems in the numerical estimation process, and if the variance covariate has values equal to zero, the varPower should not be used. For the squid data, all DML values are larger than 0 (DML is length); so it is not a problem with this example. The following R code implements the varPower function.

```
> vf3 <- varPower(form =~ DML)
> M.gls3 <- gls(Testisweight ~ DML * fMONTH,
                weights = vf3, data = Squid)
```

The AIC of this model is 3473.019, which is the lowest value so far (the lower the AIC the better the model). The summary command gives the value of $\delta = 1.75$. It is also possible to allow multiple variables in the form argument. This extension makes it possible to model an increase in spread for larger DML values, but only in certain months! The structure for the residuals is now

$$\varepsilon_{ij} \sim N(0, \sigma^2 \times |DML_{ij}|^{2\delta}) \tag{4.6}$$

Hence, $\text{var}(\varepsilon_{ij}) = \sigma^2 \times |\text{DML}_{ij}|^{2\delta j}$. The following R code implements this variance structure.

```
> vf4 <- varPower(form =~ DML | fMONTH)
> M.gls4 <- gls(Testisweight ~ DML * fMONTH,
                data = Squid, weights = vf4)
```

The anova command gives an AIC of 3407.51, now making it the best model so far. The parameters δ_j can be obtained using the summary command, and are

```
Variance function:
Structure: Power of variance covariate, dif-ferent strata
 Formula: ~DML | factor(MONTH)
 Parameter estimates:
2     9     12    11    8     10    5     7     6
1.73  1.79  1.73  1.75  1.62  1.79  1.75  1.67  1.75
4     1     3
1.71  1.70  1.72
```

So, instead of having one δ, we now have twelve of them (δ_j, $j = 1, \ldots, 12$). There is little variation between the estimated values of δ_j, but keep in mind they are multiplied by two, before being used to take the power. It is also possible to set the δ_j for some months equal to an a priori chosen value and keep it fixed. This is handy if you know or want to test whether the spread along DML in some months is constant (e.g. in month 4, as suggested by the coplot in Fig. 4.3). This can be done with the `fixed` option in `varPower` (see page 210 in Pinheiro and Bates (2000) and the help file of `varPower`). The AIC can be used to judge whether fixing or not fixing is better.

4.1.5 The varExp Variance Structure

If the variance covariate can take the value of zero, the exponential variance structure is a better option. It uses the `varExp` function in R, and for the squid data, a possible exponential variance structure is

$$\text{var}(\varepsilon_{ij}) = \sigma^2 \times e^{2\delta \times DML_i} \tag{4.7}$$

This structure models the variance of the residuals as σ^2 multiplied by an exponential function of the variance covariate DML and an unknown parameter δ. If $\delta = 0$, this gives the variance structure of model (4.1). There are no restrictions on δ or DML. This structure also allows a decrease of spread for DML values if δ is negative. As before, we can allow for different δ per month. The R code to implement the exponential variance structure is

```
> vf5 <- varExp(form =~ DML)
> M.gls5 <- gls(Testisweight ~ DML * fMONTH,
                weights = vf5, data = Squid)
```

The AIC of this model is 3478.15, which is slightly higher than for model `M.gls3`. Using `varExp(form =~ DML | fMONTH)` does the same trick as for model `M.gls4`, and allows the spread in DML to differ per month. Again, it is possible to fix some of the δ_js.

4.1.6 The varConstPower Variance Structure

Another variance structure is the *constant plus power of the variance covariate* function, and it is implemented in the function `varConstPower`. It is defined by

$$\text{var}(\varepsilon_{ij}) = \sigma^2 \times (\delta_1 + |DML_{ij}|^{\delta_2})^2 \tag{4.8}$$

This function looks rather complicated. If $\delta_1 = 1$ and $\delta_2 = 0$, we are back to the linear regression model in Equation (4.1). If not, then the variance is proportional to a constant plus the power of the variance covariate DML. According to Pinheiro and Bates (2000), this variance structure works better than the `varExp` if the variance covariate has values close to zero. To use this variance structure in R, use

```
> vf6 <- varConstPower(form =~ DML)
> M.gls6 <- gls(Testisweight ~ DML * fMONTH,
                weights = vf6, data = Squid)
```

Its AIC is 3475.02. Again, we can allow for different δ_1s and δ_2s per stratum of a nominal variable (e.g. MONTH). Such a model is fitted in R by

```
> vf7 <- varConstPower(form =~ DML | fMONTH)
> M.gls7 <- gls(Testisweight ~ DML * fMONTH,
                weights = vf7, data = Squid)
```

The AIC of this model is 3431.51. The associated variance structure is given by

$$\text{var}(\varepsilon_{ij}) = \sigma^2 \times (\delta_{1j} + |DML_{ij}|^{\delta_{2j}})^2 \tag{4.9}$$

The only difference with the variance in Equation (4.8) is the index j ($j = 1, \ldots,$ 12) from δ_1 and δ_2. Again, it is possible to set the δ_1s and δ_2s to a preset value for particular months and keep it fixed during the estimation process.

4.1.7 The varComb Variance Structure

The last variance structure we discuss is the *combination of variance structures* using the `varComb` function. With this variance structure, we can allow for both an increase in residual spread for larger DML values as well as a different spread per month. This variance structure is of the form:

$$\text{var}(\varepsilon_{ij}) = \sigma_j^2 \times e^{2\delta \times DML_{ij}} \tag{4.10}$$

Note that σ has an index j running from 1 to 12, allowing for different spreads per month. Additionally, the variance increases for larger DML values. This is a combination of `varIdent` and `varExp`. The following R code applies this variance structure and gives the AIC of all models applied so far.

```
> vf8 <- varComb(varIdent(form =~ 1 | fMONTH) ,
                 varExp(form =~ DML) )
> M.gls8 <- gls(Testisweight ~ DML * fMONTH,
                weights = vf8, data = Squid)
```

```
> anova(M.lm, M.gls1, M.gls2, M.gls3, M.gls4,
        M.gls5, M.gls6, M.gls7, M.gls8)

       Model df      AIC      BIC    logLik   Test   L.Ratio p-value
M.lm       1 25 3752.084 3867.385 -1851.042
M.gls1     2 25 3620.898 3736.199 -1785.449
M.gls2     3 36 3614.436 3780.469 -1771.218 2 vs 3    28.461  0.0027
M.gls3     4 26 3473.019 3592.932 -1710.509 3 vs 4   121.417  <.0001
M.gls4     5 37 3407.511 3578.156 -1666.755 4 vs 5    87.507  <.0001
M.gls5     6 26 3478.152 3598.066 -1713.076 5 vs 6    92.641  <.0001
M.gls6     7 27 3475.019 3599.544 -1710.509 6 vs 7     5.133  0.0235
M.gls7     8 49 3431.511 3657.501 -1666.755 7 vs 8    87.507  <.0001
M.gls8     9 37 3414.817 3585.463 -1670.409 8 vs 9     7.306  0.8367
```

The model allowing for an increase in spread for larger DML values (which is allowed to differ per month), `M.gls4`, has the lowest AIC and is therefore selected as the optimal model. Note that the tests above depend on the order in the `anova` command. If you are only after the AIC, you better use the command:

```
> AIC(M.lm, M.gls1, M.gls2, M.gls3, M.gls4,
      M.gls5, M.gls6, M.gls7, M.gls8)
```

This command only gives the AICs of the models. The `anova (M.gls4)` command shows that the interaction is highly significant. Testing fixed terms in the model is further discussed in Section 4.2.

4.1.8 Overview of All Variance Structures

Table 4.1 shows all the applied variance structures and their names. As well as these functions, you can also specify your own variance structure; see pg. 214 in Pinheiro and Bates (2000). Instead of using a covariate in the variance structure, we can use the fitted values of the model, which allows the spread in residuals to increase (or decrease) for larger fitted values.

Table 4.1 Various variance structures used in this section. The table follows Pinheiro and Bates (2000)

Name of the function in R	What does it do?
VarFixed	Fixed variance
VarIdent	Different variances per stratum
VarPower	Power of the variance covariate
VarExp	Exponential of the variance covariate
VarConstPower	Constant plus power of the variance covariate
VarComb	A combination of variance functions

If the variance covariate has large values (e.g. larger than 100), numerical instabilities may occur; *exp*(100) is rather large! In such cases, it is better to rescale the variance covariate before using it in any of the variance structures. For example, we could have used DML/max(DML) or express it in meters instead of millimetres in the variance functions. The unscaled DML can still be used in the fixed part of the model.

Remember from Appendix A, that two models are called nested if one model can be obtained from the other model by setting specific parameters equal to zero. The same definition also applies to variance structures. For example, the variance structure of the linear regression model in Equation (4.1) is nested within most of the other models. However, one of the exceptions is the linear regression model and the `varFixed` structure.

In this case, we cannot obtain the homogeneous residual variance from the linear regression model by setting a specific parameter in the `varFixed` model equal to zero. To see this, compare the following two variance structures:

$$\varepsilon_i \sim N(0, \sigma^2) \qquad \varepsilon_i \sim N(0, \sigma^2 \times DML_i)$$

The first variance structure is from the linear regression model and the second one from the `varFixed`. We cannot obtain the variance structure on the left from the right one, unless DML is equal to 1 for all observations. Compare this with the linear regression model and the `varPower` structure:

$$\varepsilon_i \sim N(0, \sigma^2) \qquad \varepsilon_{ij} \sim N(0, \sigma^2 \times |\text{DML}_{ij}|^{2\delta_j})$$

By setting all δ_js equal to zero in the right variance structure, we obtain the left variance structure; hence, these are nested variance structures. Note that the `varIdent` is nested in the `varPower` structure! And nested models mean that we can apply the likelihood ratio test.

To test certain types of heterogeneity, we can apply the log likelihood ratio test. For example, for model (4.1) and the optimal variance structure in (4.6), we can type `anova (M.lm, M.gls4)`, which gives:

```
       Model df      AIC      BIC    logLik   Test  L.Ratio p-value
M.lm       1 25 3752.084 3867.385 -1851.042
M.gls4     2 37 3407.511 3578.156 -1666.755 1 vs 2 368.5728  <.0001
```

The log likelihood ratio statistic is 368.57, indicating that the variance structure in (4.6) is considerably better than the constant variance in the linear regression model (4.1). Hence, the `varPower` option provides a significantly better variance structure than the one used for the linear regression model in (4.1). In a paper, you would write this as $L = 368.57$ ($df = 12$, $p < 0.001$). This model comparison provides a better testing procedure for homogeneity than those presented in Chapter 2.

4.1.8.1 Which One to Choose?

So, which variance structure should you choose, and how do you decide which one is best? If the variance covariate is a nominal variable, the choice is simple; use `varIdent`. In our example, it allowed modelling different residual variation for the testis weight per month.

The underlying variance structure imposed by `varIdent` is relatively easy to understand, but the difference between the variance structured modelled by the `varFixed`, `varPower`, `varExp`, and `varConstPower` functions are more difficult to explain. All four variance structures allow for an increase (or decrease) in residual variation for the testis weight data along a continuous variance covariate like DML (an explanatory variable in this case).

But, the `varFixed` is rather limited, as it assumes that the variance of the residuals is linearly related to a variance covariate. This causes problems if the variance covariate takes non-positive values or where the linear relationship requirements between variation and the variance covariate is too stringent.

In practise, it may be better to use the `varPower`, `varExp`, or `varConstPower` functions, which allow for more flexibility than the `varFixed`. So, how to choose between these three? The difference between them is the mathematical parameterisation of the variance function. The `varPower` should not be used if the variance covariate takes the value of zero. In this case, this is not an issue as DML (length) is always larger than zero. But it may be an issue with variance covariates like temperature or height compared to a baseline, etc.

However, finding the right variance structure for a variance covariate like DML, which is always non-zero, is more a matter of trial and error, and the best choice is judged through using tools like the AIC. Another important aspect is biological knowledge. If you know a priori that there is a certain type of heterogeneity in your data, then you can greatly speed up the selection process by including this information!

4.1.9 Graphical Validation of the Optimal Model

For graphical model validation, we can use two types of residuals: (i) residuals calculated as observed minus fitted values (also called ordinary residuals) and (ii) normalised residuals. We start with the first one. The following R code extracts the residuals and plots them in a coplot (Fig. 4.4). Note that these residuals still show heterogeneity, but this is now allowed (because the residual variation differs depending on the chosen variance structure and values of the variance covariate). Hence, these residuals are less useful for the model validation process.

```
> E1 <- resid(M.gls4)
> coplot(E1 ~ DML | fMONTH,
         ylab = "Ordinary residuals", data = Squid)
```

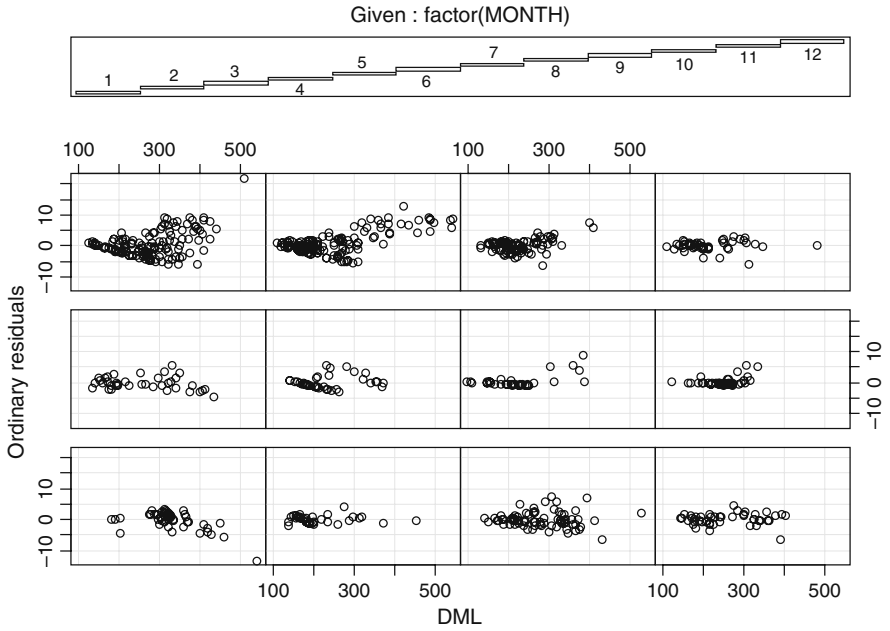

Fig. 4.4 Ordinary residuals (observed minus fitted values) versus DML conditional on month for the optimal model. These residuals are allowed to have a cone effect

You should use standardised residuals instead of the ordinary residuals for the model validation. These are obtained by calculating the observed minus the fitted values and then dividing by the square root of the variance. These residuals are therefore obtained from

$$\varepsilon_{ij} = \frac{Testisweight_{ij} - Fitted\ values_{ij}}{\sqrt{\sigma^2 \times |DML_{ij}|^{2\delta_j}}} \tag{4.11}$$

Plotting these residuals should not show any heterogeneity. If there is any heterogeneity, then further model improvement is required. Luckily, we don't have to program Equation (4.11) as the standardised residuals can be obtained using an R function.

The following R code extracts the standardised residuals, and makes a coplot, (Fig. 4.5) where there is no clear evidence of heterogeneity.

```
> E2 <- resid(M.gls4, type = "normalized")
> coplot(E2 ~ DML | fMONTH, data = Squid,
         ylab = "Normalised residuals")
```

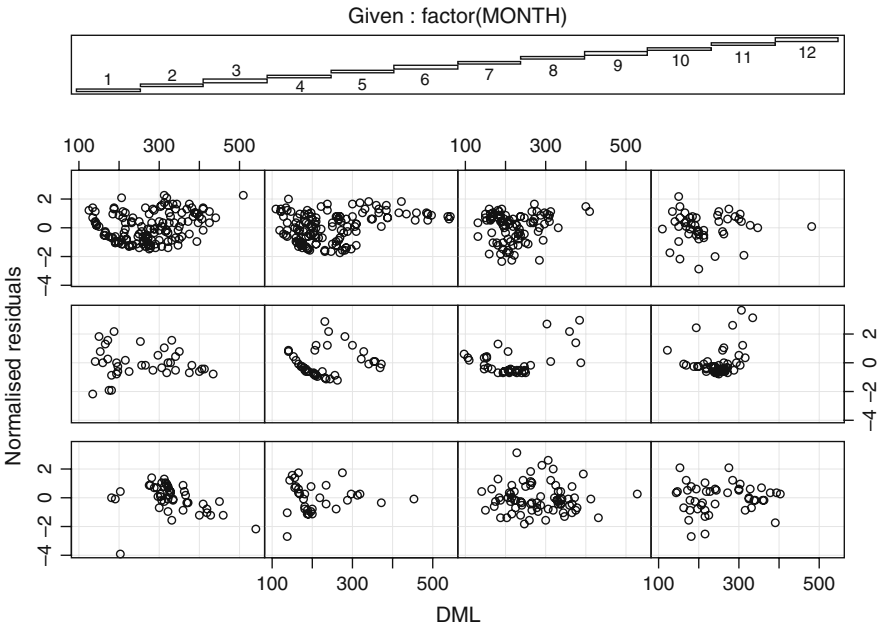

Fig. 4.5 Coplot of standardised residuals versus DML conditional on month for the optimal model. There is no evidence of heterogeneity

The option `type = "normalized"` ensures that `E2` contains the standardised residuals.

4.2 Benthic Biodiversity Experiment

4.2.1 Linear Regression Applied on the Benthic Biodiversity Data

In this section, we provide another example of a linear regression model for statistically heterogeneous data. Based on experimental protocols developed in Emmerson and Raffaelli (2000), Emmerson et al. (2001), Solan and Ford (2003), and Ieno et al. (2006), among others, replicate mesocosm experiments (using plastic ice containers) were carried out. Benthic macrofaunal single and/or multiple species (biodiversity) were manipulated in a multi-patch environment, and the release of ammonium (NH_4-N), nitrate (NO_x-N) and phosphate (PO_4-P) concentrates were recorded from the sediment (ecosystem processes).

The data used for the specific example shown below relies on both published data (Ieno et al., 2006) and unpublished data (Oceanlab, University of Aberdeen). The experiment examines the effect of macrofauna density (*Hediste diversicolor,* Polychaeta), and habitat heterogeneity on sediment nutrient release. Figure 4.6 shows the experimental set up.

Fig. 4.6 Photograph showing the experimental set up. One nutrient is measured per container

At the start of the experiment, each container was filled with homogenised sediment from mudflats on the Ythan estuary (Scotland, UK). The macrofaunal biomass (*H. diversicolor*) was fixed across the following levels (0, 0.5, 1, 1.5, and 2 g), and replicated within each biomass level ($n = 3$). The response variable is the concentration of a particular nutrient.

To study the effect of habitat heterogeneity, the previous procedure was repeated for algae-enriched sediment. This gave 36 observations per nutrient, 18 enriched, and 18 non-enriched. Because there are three nutrients, the data set contains 108 samples (containers).

We can either analyse the data for each nutrient separately or combine all the data and analyse it all at the same time. The latter option is applied here as it allows us to test for interactions between nutrients and treatment levels (note that the nutrients were not measured in the same container; so there are no pseudo-replication problems).

To analyse the concentration data from all three nutrients, we need to concatenate the 36 observations from each nutrient, resulting in a response variable of length 108 (36×3), one continuous explanatory variable (biomass), and two nominal explanatory variables: enrichment (with or without algae), and a variable identifying the nutrient with the levels NH_4-N, NO_3-N, and PO_3-P.

There is, however, a major problem with the statistical analysis of the combined data. Due to the nature of the variables, we expect massive differences in variation in concentrations per nutrient and enrichment combination. This is illustrated in Fig. 4.7, which shows a boxplot for each nutrient–enrichment combination. Note

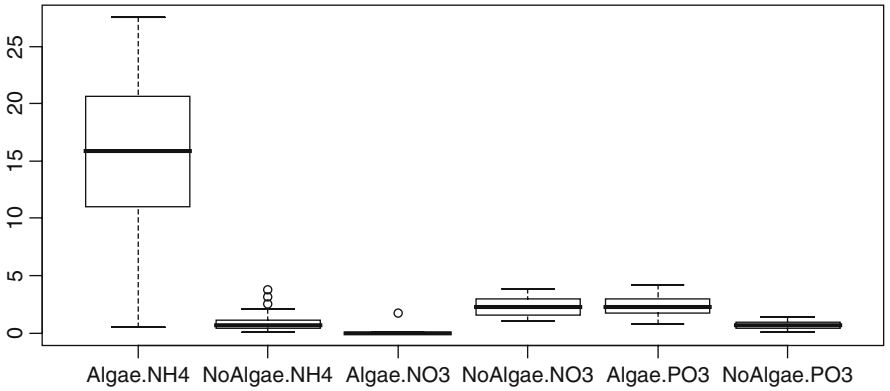

Fig. 4.7 Boxplot of concentrations for NH_4, NO_3 and PO_3. The first two boxplots on the left hand side are for enriched and non-enriched NH_4 concentrations

that the samples enriched with algae and with NH_4, have higher concentrations and show more variation.

An initial linear regression analysis, using biomass, enrichment and nutrient, with all the two-way interactions, and the three-way interaction as explanatory variables clearly showed serious violation of homogeneity, as can be seen from Fig. 4.8.

A $\log_{10}$(Concentration + 0.5) transformation was applied, but the enrichment $\times$ NO_3 combination still had lower variation than the other combinations. We, therefore, cannot easily obtain homogeneity with a data transformation. And, as we mentioned in Chapter 2, we want to avoid data transformations whenever possible. So, instead of transforming the data, we will allow for different variances by using GLS.

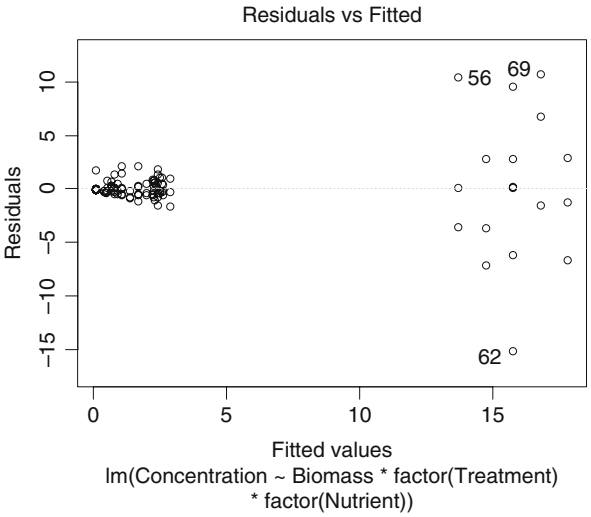

Fig. 4.8 Residuals versus fitted values for the linear regression model. Note the difference in spread

The following R code was used to make Figs. 4.7 and 4.8.

```
> library(AED); data(Biodiversity);
> Biodiv <- Biodiversity             #Saves some space
> Biodiv$fTreatment <- factor(Biodiv$Treatment)
> Biodiv$fNutrient  <- factor(Biodiv$Nutrient)
> boxplot(Concentration ~
       fTreatment * fNutrient, data = Biodiv)
> M0 <- lm(Concentration ~
           Biomass * fTreatment * fNutrient,
           data = Biodiv)
> plot(M0, which = c(1), add.smooth = FALSE)
```

The `library` and `data` commands are used to load the data. The variables Treatment and Nutrient are converted into factors, and the rest is basic code for a boxplot (Chapter 2) and linear regression (Appendix A).

4.2.2 GLS Applied on the Benthic Biodiversity Data

As with the squid data, we have to investigate why there is heterogeneity in these benthos data. Biological knowledge suggests that treatment and nutrient levels, possibly both, may be driving the heterogeneity. A scatterplot of biomass versus concentration did not show any clear increase or decrease in spread. This indicates that the potential variance covariates are nutrient and/or enrichment. The following R code assumes you have already loaded the data. It first applies the linear regression model again with the `gls` command, and then the three GLS models with different variance covariates are fitted.

```
> library(nlme)
> f1 <- formula(Concentration ~ Biomass * fTreatment *
                 fNutrient)
> M0 <- gls(f1, data = Biodiv)
> M1A <-gls(f1, data = Biodiv, weights = varIdent(
              form =~ 1 | fTreatment * fNutrient))
> M1B <-gls(f1, data = Biodiv,
            weights = varIdent(form =~ 1 | fNutrient))
> M1C <-gls(f1, data = Biodiv,
            weights = varIdent(form =~ 1 | fTreatment))
```

The first model M0 is the linear regression model without any variance covariates. The second model M1A uses one variance term per nutrient–enrichment combination. And the third and fourth models use as variance covariates, nutrient and enrichment, respectively. The models have all main terms, two-way interactions, and the three-way interaction term as a fixed component. The anova command can be used to compare the models.

```
> anova(M0, M1A, M1B, M1C)

    Model df      AIC      BIC    logLik   Test   L.Ratio p-value
M0      1 13 534.5203 567.8569 -254.2602
M1A     2 18 330.1298 376.2881 -147.0649 1 vs 2 214.39054  <.0001
M1B     3 15 380.0830 418.5482 -175.0415 2 vs 3  55.95320  <.0001
M1C     4 14 439.7639 475.6647 -205.8819 3 vs 4  61.68087  <.0001
```

The AIC of the model with both nutrient and enrichment as variance covariates (`M1A`) is by far the best model, as judged by the AIC and BIC. Note that not all the likelihood ratio tests make sense (not all comparisons are from nested models). The `plot(M1A, col = 1)` command plots the standardised residuals versus fitted values. The graph is not shown here, but there is no sign of heterogeneity.

The commands `anova(M1A)` and `summary(M1A)` give information of the significance of the fixed explanatory variables (the three-way interaction, etc.). Results are not given here, but both functions show that the three-way interaction is not significant.

4.2.3 A Protocol

The problem is that we have still not discussed all aspects of model selection. This requires knowledge of things like maximum likelihood (ML) estimation and restricted maximum likelihood estimation (REML), and we discuss these in more detail in the next chapter. For the moment, we present them in a rather abstract manner and justify them later in Chapter 5. So to fully understand the differences between ML and REML, you need to read Chapter 5. In Chapter 5, the protocol for model selection in mixed modelling is explained (and justified) in detail, but the same protocol applies for GLS and is introduced in less detail below.

1. Start with a linear regression model that contains as many explanatory variables and their interactions as possible. The residuals of this model are assumed to be normally distributed with mean 0 and variance σ^2. Investigate whether the homogeneity assumptions are valid by plotting the standardised residuals versus fitted values and by plotting the standardised residuals versus each individual explanatory variable. Any sign of variation in residual patterns is an indication of heterogeneity and means you have to go on to step 2. If you do not see any clear violation of homogeneity, there is no need to continue to step 2; just continue with a model selection on the explanatory variables (Appendix A). It should be noted that the graphical assessment of heterogeneity is difficult for small data sets.
2. For formal model comparison, repeat step 1 using the `gls` function from the nlme package. Do not specify any special variance structure yet and ensure that REML estimation is used (the default estimation method). You will get exactly the same estimated values, *t*-values and *p*-values as in step 1. The reason for

this step is that the `anova` command cannot compare objects obtained by the functions `lm` and `gls`. A call to the `gls` function without any extra options is a linear regression.

3. Depending on the graphical model validation in step 1, choose an appropriate variance structure. It helps to plot residuals versus fitted values and use different colours and symbols for different nutrients and/or enrichment levels (for our particular example). In the previous section, a wide range of residual variance structures was introduced.
4. Fit a new `gls` model with the selected variance covariance structure selected in step 3. Ensure that REML estimation is used, which is done with `gls(..., method = "REML")`, and that you use the same selection of explanatory variables. This is now called the fixed part of the model, and the residuals are called the random part. We will first try to find the optimal random structure using as many explanatory variables in the fixed part as possible.
5. Compare the new GLS model with the earlier results using the AIC, BIC, or likelihood ratio test. If the new model is better, extract the normalised residuals, and inspect these for homogeneity (using the same tools as in step 1). If the homogeneity assumption is not valid for the normalised residual of the model obtained in step 4, then go to step 6. If it is valid, then go to step 7.
6. If the residuals still show heterogeneity, go to step 4, and choose another residual variance structure. If you keep iterating between steps 4, 5, and 6, either try improving the fixed component (using for example additive modelling), try a different distribution (e.g. Poisson or negative binomial), consider a transformation on the response variable as a last resort, or conclude that your residual spread is not related to any of the measured covariates.
7. You are now half way. You have found the optimal residual variance structure using REML estimation. Now it is time to find the optimal fixed component. Or stated differently, which explanatory variables are significant, and which are not. You have three tools to find the optimal fixed component: the *t*-statistic, the *F*-statistic, and the likelihood ratio test. The *t*-statistics are obtained with the `summary` command, and the *F*-statistic with the `anova` command. Both functions are applied on one model, e.g. by typing `summary(M1A)` or `anova(M1A)`. Ensure that REML estimation is used in the `gls` command. Remember the `anova` command is doing sequential testing. This is useful for testing the significance of the highest interaction term, but not for the other terms in the model. It is also of less use if you only have main terms as the order of the variables is of importance in sequential testing. The problem with the *t*-statistic is that it should not be used to assess the significance of a nominal variable with more than two levels (e.g. nutrient). The third option is the likelihood ratio test. You need to specify a full model and a nested model (Appendix A). Both models need ML estimation (and the same random structure, but you already selected these in step 5). This approach is conceptually probably the easiest to work with, but it can be time consuming.

8. Apply any of the model selection tools described in step 7, and stop once all terms are significant.
9. Reapply the model that was found in step 8, and refit it with REML estimation. Apply a graphical model validation, checking for homogeneity (see step 1), normality, and independence. If no problems are highlighted, go to step 10. If problems are identified, return to step 8, and consider adding non-significant terms to see if this improves the model validation graphs.
10. Present the results in a table and try to understand what it all means in terms of ecology.

We demonstrated steps 1–7 for the benthic biodiversity data earlier in this chapter and now continue with this example for the remaining steps in the protocol just described.

4.2.4 Application of the Protocol on the Benthic Biodiversity Data

The `anova(M1A)` command gives the following output.

```
Denom. DF: 96
                                numDF   F-value p-value
(Intercept)                         1 205.73781  <.0001
Biomass                             1   1.22179  0.2718
fTreatment                          1  14.62895  0.0002
fNutrient                           2   1.57754  0.2118
Biomass:fTreatment                  1   0.26657  0.6068
Biomass:fNutrient                   2   4.17802  0.0182
fTreatment:fNutrient                2 121.57149  <.0001
Biomass:fTreatment:fNutrient        2   1.09043  0.3402
```

An explanation of the nominator and denominator degrees of freedom is delayed until Chapter 5. Here, we focus on the value of the F-statistic and its p-value. The `anova` function applies sequential testing. This means that the p-values will change if you change the order of the main terms or the order of the two-way interactions. In this example, it is only the last term that is of real interest as it shows the significance of the three-way interaction term (you can't change the order of this term). In this case, it is not significant at the 5% level. This means that we can drop the three-way term and refit the model.

Refitting the model with the main terms and all three two-way terms gives exactly the same anova table as above, except for the last line. The problem is that we cannot assess the significance of the Biomass × Treatment term, and the biomass × Nutrient term, due to the order how they were put in. Obviously, we could apply three models, ensure each time that a different two-way term is the last, and deselect the least significant two-way interaction.

The second model selection approach (using hypothesis testing) is based on the *t*-statistic, but we do not want to use this option as nutrient has three levels. One level will be used as baseline, and the *p*-values from the *t*-statistic will only tell us whether the second and third nutrients are different from the baseline nutrient.

The third model selection approach (using hypothesis testing) is based on comparing nested models. Let us go back a step and test the significance of the three-way interaction term again. We compare the full model (with the three-way interaction term) with a model that does not contain the three-way interaction term using the likelihood ratio test. Both models need ML estimation. The R code for this is as follows.

```
> M2A1 <- gls(Concentration ~ Biomass + fTreatment +
             fNutrient +
             Biomass:fTreatment +
             Biomass:fNutrient +
             fTreatment:fNutrient +
             Biomass:fTreatment:fNutrient,
             weights = varIdent(form =~ 1 |
                                fTreatment * fNutrient),
             method = "ML", data = Biodiv)

> M2A2 <- gls(Concentration ~ Biomass + fTreatment +
             Nutrient +
             Biomass:fTreatment +
             Biomass:fNutrient +
             fTreatment:fNutrient,
             weights=varIdent(form =~ 1 |
                              fTreatment * fNutrient),
             method = "ML", data = Biodiv)
```

The output of the anova (M2A1, M2A2) command is given below.

```
     Model df      AIC      BIC    logLik   Test  L.Ratio p-value
M2A1     1 18 321.0648 369.3432 -142.5324
M2A2     2 16 319.4653 362.3794 -143.7327 1 vs 2 2.400507  0.3011
```

The `anova` command also indicates that the three-way interaction can be dropped. In the next step of the model selection, we have to find a *p*-value for each two-way interaction. This is done as follows. Use model `M2A2` as the starting point and drop each of the two-way interactions in turn, and use the `anova` command to obtain a *p*-value. Also consider whether any of the main terms can be dropped. The rule is that if an interaction term is included, then all the associated main terms should be included as well, and are not a candidate for dropping. However, if you have the main terms A, B, C, and the interaction A × B, then the two terms that can be potentially dropped are A × B and also C!

This whole process is rather time consuming and you will want to think twice before adding four-way interactions! It was our intention to put the code for this example online, but all our book reviewers asked us to include it in the text of the

book. Perhaps they are right, and you should see this at least once in your life. So, take a deep breath, and read on!

4.2.4.1 Round 1 of the Backwards Selection

The following code drops each two-way interaction and applies a likelihood ratio test.

```
> vfOptim <- varIdent(form =~ 1 | fTreatment*fNutrient)
> #Assess significance of all 3 2-way interactions
> #Full model
> M3.Full <- gls(Concentration ~
              Biomass + fTreatment + fNutrient +
              Biomass:fTreatment +
              Biomass:fNutrient +
              fTreatment:fNutrient,
              weights = vfOptim,
              method = "ML", data = Biodiv)
> #Drop Biomass:fTreatment
> M3.Drop1 <- gls(Concentration~
                Biomass + fTreatment + fNutrient +
                Biomass:fNutrient +
                fTreatment:fNutrient,
                weights = vfOptim,
                method = "ML", data = Biodiv)
> anova(M3.Full, M3.Drop1)

         Model df      AIC      BIC    logLik   Test  L.Ratio p-value
M3.Full      1 16 319.4653 362.3794 -143.7327
M3.Drop1     2 15 319.3730 359.6050 -144.6865 1 vs 2 1.907680  0.1672

>
> #Drop Biomass:fNutrient
> M3.Drop2 <- gls(Concentration ~
               Biomass + fTreatment + fNutrient +
               Biomass:fTreatment +
               fTreatment:fNutrient,
               weights = vfOptim,
               method = "ML", data = Biodiv)
> anova(M3.Full, M3.Drop2)

         Model df      AIC      BIC    logLik   Test  L.Ratio p-value
M3.Full      1 16 319.4653 362.3794 -143.7327
M3.Drop2     2 14 323.2165 360.7664 -147.6083 1 vs 2 7.751179  0.0207

>
> #Drop fTreatment:fNutrient
> M3.Drop3 <- gls(Concentration ~
               Biomass + fTreatment + fNutrient +
               Biomass:fTreatment +
               Biomass:fNutrient,
               weights = vfOptim,
               method = "ML", data = Biodiv)
```

```
> anova(M3.Full, M3.Drop3)
         Model df      AIC      BIC    logLik   Test  L.Ratio p-value
M3.Full      1 16 319.4653 362.3794 -143.7327
M3.Drop3     2 14 403.3288 440.8786 -187.6644 1 vs 2 87.86346  <.0001
```

So, we dropped each two-way interaction term in turn, applied the likelihood ratio test, and obtained *p*-values. Clearly, the two way interaction term `Biomass:fTreatment` is not significant at the 5% level and should be dropped. You can make the code above a bit friendlier using the `update` command. The following code produces exactly the same results.

```
> #Alternative coding with same results
> fFull <- formula(Concentration~
            Biomass + fTreatment + fNutrient +
            Biomass:fTreatment +
            Biomass:fNutrient + fTreatment:fNutrient)
> M3.Full <- gls(fFull, weights = vfOptim,
               method = "ML", data = Biodiv)

> #Drop Biomass:fTreatment
> M3.Drop1<-update(M3.Full, .~. - Biomass:fTreatment)
> anova(M3.Full, M3.Drop1)

         Model df      AIC      BIC    logLik   Test  L.Ratio p-value
M3.Full      1 16 319.4653 362.3794 -143.7327
M3.Drop1     2 15 319.3730 359.6050 -144.6865 1 vs 2 1.907680  0.1672

> #Drop Biomass:fNutrient
> M3.Drop2 <- update(M3.Full, .~. - Biomass:fNutrient)
> anova(M3.Full, M3.Drop2)

         Model df      AIC      BIC    logLik   Test  L.Ratio p-value
M3.Full      1 16 319.4653 362.3794 -143.7327
M3.Drop2     2 14 323.2165 360.7664 -147.6083 1 vs 2 7.751179  0.0207

> #Drop fTreatment:fNutrient
> M3.Drop3<-update(M3.Full, .~. - fTreatment:fNutrient)
> anova(M3.Full,M3.Drop3)

         Model df      AIC      BIC    logLik   Test  L.Ratio p-value
M3.Full      1 16 319.4653 362.3794 -143.7327
M3.Drop3     2 14 403.3288 440.8786 -187.6644 1 vs 2 87.86346  <.0001
```

As you can see, this gives the same results. The advantage of the `update` command is that the code is shorter, but you it also makes it easier to lose track what exactly you are fitting.

4.2.4.2 Round 2 of the Backwards Selection

Whichever coding you use, we need to drop the term `Biomass:fTreatment`. This means that the new full model is

```
> #New full model
> M4.Full <- gls(Concentration~
```

```
          Biomass + fTreatment + fNutrient +
          Biomass:fNutrient + fTreatment:fNutrient,
          weights = vfOptim,
          method = "ML", data = Biodiv)
```

From this model, you can drop two of the two-way interaction terms. No main terms can be dropped yet. We will use the `update` command again and try to avoid turning this chapter into something that looks like a telephone book.

```
>#Drop Biomass:fNutrient
> M4.Drop1 <- update(M4.Full, .~. -Biomass:fNutrient)
> anova(M4.Full, M4.Drop1)

         Model df      AIC      BIC    logLik   Test  L.Ratio p-value
M4.Full      1 15 319.3730 359.6050 -144.6865
M4.Drop1     2 13 321.7872 356.6549 -147.8936 1 vs 2 6.414148  0.0405

> #Drop fTreatment:fNutrient
> M4.Drop2<-update(M4.Full, .~. -fTreatment:fNutrient)
> anova(M4.Full, M4.Drop2)

         Model df      AIC      BIC    logLik   Test  L.Ratio p-value
M4.Full      1 15 319.3730 359.6050 -144.6865
M4.Drop2     2 13 404.8657 439.7335 -189.4329 1 vs 2 89.49272  <.0001
```

A p-value of 0.04 for the `Biomass:fNutrient` interaction is not impressive, especially not with a series of hypothesis tests. So, we decided to drop it as well and continue with the following full model.

4.2.4.3 Round 3 of the Backwards Selection

The new full model is

```
> #New full model
> M5.Full <- gls(Concentration ~
               Biomass + fTreatment + fNutrient +
               fTreatment:fNutrient,
               weights = vfOptim, method = "ML",
               data = Biodiv)
```

We can drop the `fTreatment:fNutrient` interaction term, but also the main term `Biomass`.

```
> #Drop fTreatment:fNutrient
> M5.Drop1 <-update(M5.Full, .~.-fTreatment:fNutrient)
> anova(M5.Full, M5.Drop1)

         Model df      AIC      BIC    logLik   Test  L.Ratio p-value
M5.Full      1 13 321.7872 356.6549 -147.8936
M5.Drop1     2 11 406.7950 436.2985 -192.3975 1 vs 2 89.00786  <.0001
> #Drop Biomass
```

```
> M5.Drop2 <- update(M5.Full, .~. -Biomass)
> anova(M5.Full, M5.Drop2)
         Model df      AIC      BIC    logLik   Test  L.Ratio p-value
M5.Full      1 13 321.7872 356.6549 -147.8936
M5.Drop2     2 12 321.2595 353.4450 -148.6297 1 vs 2 1.472279   0.225
```

The biomass term is not significant and can be dropped.

4.2.4.4 Round 4 of the Backwards Selection

The new full model is

```
> M6.Full<-gls(Concentration ~ fTreatment + fNutrient+
          fTreatment:fNutrient,
          weights = vfOptim, method = "ML",
          data = Biodiv)
```

The only term that can be dropped is the interaction term.

```
> M6.Drop1<-update(M6.Full, .~. -fTreatment:fNutrient)
> anova(M6.Full, M6.Drop2)
         Model df      AIC      BIC    logLik   Test  L.Ratio p-value
M6.Full      1 12 321.2595 353.4450 -148.6297
M6.Drop1     2 10 406.0323 432.8536 -193.0161 1 vs 2 88.77283  <.0001
```

The interaction term `fTreatment:fNutrient` is highly significant, so no further terms can be dropped.

4.2.4.5 The Aftermath

We applied the process of comparing nested models several times, and ended up with a model containing Nutrient, Enrichment, and their interaction. The two-way interaction term was significant. We reapplied this model with REML estimation (step 9). Normality and homogeneity can safely be assumed (see Fig. 4.9). Figure 4.9 was created with the following R code.

```
> MFinal <- gls(Concentration ~ fTreatment * fNutrient,
                weights = vfOptim, method = "REML",
                data = Biodiv)
> E <- resid(MFinal, type = "normalized")
> Fit <- fitted(MFinal)
> op <- par(mfrow = c(1, 2))
> plot(x = Fit, y = E,
       xlab = "Fitted values", ylab = "Residuals",
       main = "Residuals versus fitted values")
> identify(Fit, E)
> hist(E, nclass = 15)
> par(op)
```

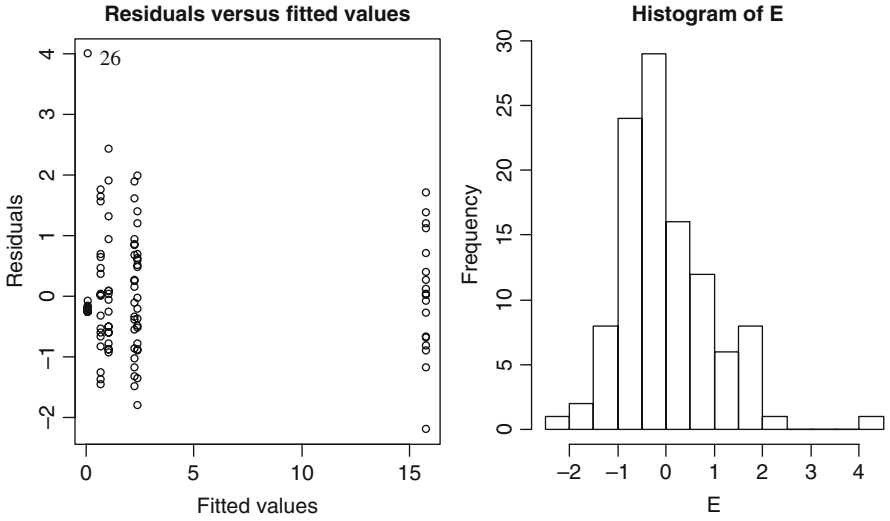

Fig. 4.9 Residuals versus fitted values and a histogram of the residuals (denoted by E) for the optimal GLS model that contains Nutrient, Enrichment, and their interaction

The `gls` command refits the model with REML, the `resid` command extracts the normalised residuals, the object `Fit` are the fitted values, the `plot` command plots the fitted values versus the residuals, and the `hist` command makes a histogram with 15 bars. The `identify` command allows us to identify the observation with the large residual (observation 26). We will return to this observation in a moment.

Assuming that everything is ok, we can now proceed to step 10 and present the relevant output of the final model using the `summary(MFinal)` command.

```
Generalized least squares fit by REML
Model: Concentration ~ fTreatment + fNutrient + fTreatment:fNutrient
  Data: Biodiv
       AIC      BIC    logLik
  327.9174 359.4171 -151.9587

Variance function:
 Structure: Different standard deviations per stratum
 Formula: ~1 | fTreatment * fNutrient
 Parameter estimates:
NoAlgae*NO3 Algae*NO3 NoAlgae*NH4   Algae*NH4  NoAlgae*PO3  Algae*PO3
    1.00000   0.50104     1.33233     8.43635      0.48606    1.10733
Coefficients:
                                 Value Std.Error   t-value p-value
(Intercept)                   15.78139  1.629670  9.683792       0
fTreatmentNoAlgae            -14.69763  1.649868 -8.908365       0
fNutrientNO3                 -15.66972  1.632542 -9.598358       0
fNutrientPO3                 -13.36137  1.643649 -8.129089       0
```

```
fTreatmentNoAlgae:fNutrientNO3  16.86929  1.663956 10.138067       0
fTreatmentNoAlgae:fNutrientPO3  12.95293  1.666324  7.773353       0

Residual standard error: 0.8195605
Degrees of freedom: 108 total; 102 residual
```

The AIC and BIC are model selection tools, and there is little to say about them at this point as we have passed the model selection stage. The information on the different standard deviations (multiplication factors of σ) is interesting, as it shows the different variances (or better: the ratio with the standard error) per treatment–nutrient combination. The estimated value for σ is 0.819. Note that the combination enrichment with algae and NH_4 has the largest variance, namely $(8.43 \times 0.819)^2$.

The estimated regression parameters, standard errors, *t*-values, *p*-values, and other relevant information are given as well. Note that all terms are significantly different from 0 at the 5% level. To understand what the model is trying to tell us, it can be helpful to consider a couple of scenarios and obtain the equations for the fitted values or just graph the fit of the model. The easiest way of doing this is

```
> boxplot(predict(MFinal) ~ fTreatment * fNutrient,
          data = Biodiv)
```

This only works because all the explanatory variables are nominal. The resulting graph is shown in Fig. 4.10 and clearly shows that the observations exposed to algae treatment *and* NH_4 enrichment have the highest values. This explains why the interaction term is significant. Unfortunately, at the time of writing, the `predict.gls` function (which is the one used to obtain the predicted values) does not give standard errors for predicted values. To obtain the 95% confidence bands around the fitted values, you need to use equations similar to those used for linear regression

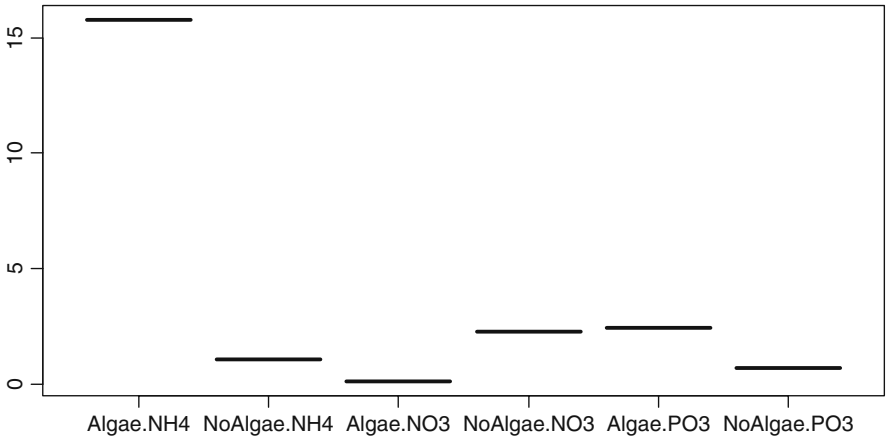

Fig. 4.10 Fitted values for the optimal model. Note the high values for the algae–NH_4 combination

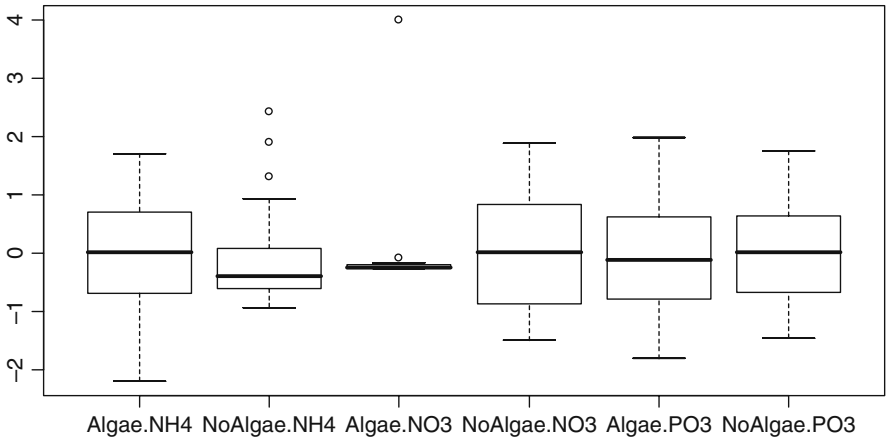

Fig. 4.11 Normalised residuals versus treatment–nutrient combination. Note the effect of the outlier for the algae–NO_3 combination. This is observation 26

(Appendix A), but this requires some ugly R programming. Alternatively, you can do some bootstrapping.

Before you happily write your paper using these results, there is one final point you should know. Figure 4.11 shows a boxplot of normalised residuals versus the treatment–nutrient combination. Note the effect of observation 26! We suggest that you repeat the entire analysis without this observation. If this was an email, we would now add a ☺ as this obviously means a lot of extra work!. You will need to remove row 26 from the data, or add `subset = -26` to each `gls` command. The first option is a bit clumsy, but avoids any potential error messages in the validation graphs (due to different data sizes).