12

The Mixed Model

12.1 Introduction

The exploratory analysis of Data Set 3, the Uruguayan rice fields, in Chapters 7 and 9 indicated that irrigation effectiveness plays an important role in distinguishing high-yielding from low-yielding fields. Irrigation effectiveness is represented by the ordinal variable *Irrig*, which is the expert agronomist's rating of irrigation effectiveness at each measured site. Can the association between *Irrig* and *Yield* be scaled down? That is, is irrigation effectiveness consistently associated with yield at the single-field scale, or, alternatively, is this an example of the ecological fallacy discussed in Section 11.5.2, in which an observation that holds at the scale of many fields does not "scale down" to the individual field? Figure 12.1 shows a trellis plot, made with the `lattice` package (Sarkar, 2008) function `xyplot()`, of yield versus irrigation effectiveness for each field. Yield generally seems to increase with irrigation effectiveness, although there is a lot of variability, and some fields actually seem to show a *decreasing* yield with increasing effectiveness. The individual plots generally look as though they would be appropriate for linear regression. The variable *Irrig*, however, is an ordinal scale variable, and as such the operations of multiplication and addition, which are used to compute the regression coefficients, are not meaningful when applied to its values. In principle, therefore, it is not appropriate to interpret a regression as indicating that yield increases by a certain percent for each unit increase in irrigation effectiveness. For the present application, however, our interest is not in predicting a value of yield based on irrigation effectiveness but rather on computing the regression coefficient in order to test the null hypothesis that this coefficient is equal to zero. We will consider this an acceptable breach of the rules.

We can use the function `lmList()` of the `nlme` package (Pinhiero et al., 2011) to examine the coefficients of the fits of a simple linear regression to each of the fields. The data frame `data.Set3` is loaded using the code in Appendix B.3.

```
> library(nlme)
> data.lis <- lmList(Yield ~ Irrig | Field, data = data.Set3)
> print(coef(data.lis), digits = 3)
  (Intercept)    Irrig
1        5680   642.50
2       11456      NA
3        4645   976.43
4        3664  1277.83
5        2042  1404.97
6        5919    -3.93
7        4361   562.78
```

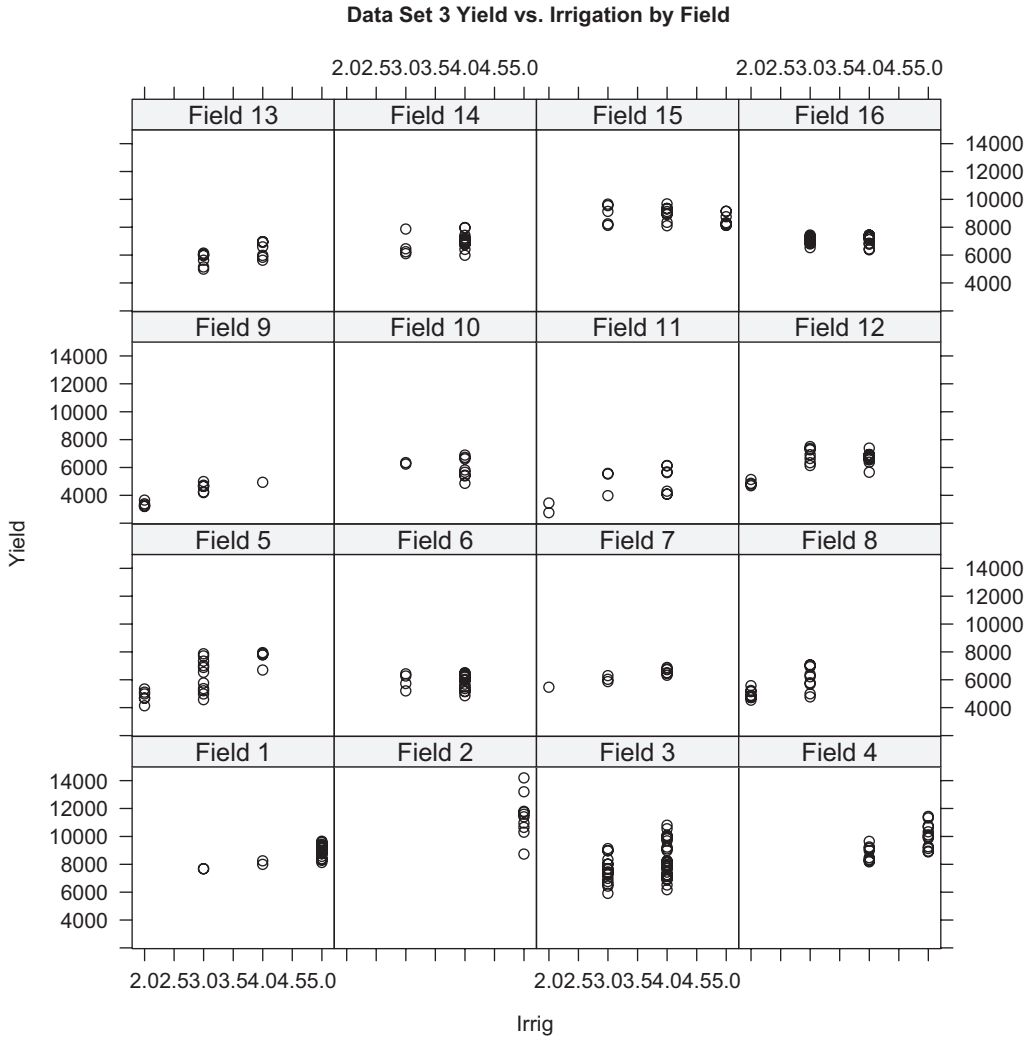

FIGURE 12.1
Yield versus irrigation effectiveness for each of the fields in Data Set 3.

```
8          2681 1138.03
9          1491  975.14
10         7414 -373.62
11         1828  861.34
12         4108  687.39
13         3568  710.00
14         5648  339.05
15         9846 -246.03
16         7042    6.51
```

This code is an example of the use of *grouped data*. Grouped data, which we have seen before in working with trellis graphics, are data in which there exists a *grouping factor*, that is, an index that divides the data into meaningful groups (Pinhiero and Bates, 2000, p. 99).

In the current example, the grouping factor is the field identification number. The vertical line in the formula `Yield ~ Irrig | Field` indicates that the variable *Field* is the grouping factor, and the function `lmList()` calculates a separate linear regression for each group, indexed by this factor. Grouped data are used extensively in the mixed-model analyses that are the subject of this chapter.

The regression on Field 2 above returns a value of `NA` because all of the data records associated with this field have a value of *Irrig* equal to 5 (Figure 12.1). The field is uniformly well irrigated (and has the highest mean yield). The fact that all of the data records in Field 2 have a value of *Irrig* equal to 5 can be verified by applying the functions `tapply()` and the argument `FUN` set to `unique` to apply the function `unique()`.

```
> tapply(data.Set3$Irrig, data.Set3$Field, unique)
$`1`
[1] 5 3 4

$`2`
[1] 5

$`3`
[1] 3 4
   *   *   *   DELETED   *   *   *
```

This field is therefore extracted from the data set.

```
> data.Set3a <- data.Set3[-which(data.Set3$Field == 2),]
```

If in fact the regression relationship between yield and irrigation effectiveness were consistent across fields, then one could use the analysis of covariance (ANCOVA) (Kutner et al., 2005, p. 314; Searle, 1971, p. 348; Sokal and Rohlf, 1981, p. 509) to construct and analyze a model of the relationship between field, irrigation effectiveness, and yield. The appropriate ANCOVA model is

$$Y_{ij} = \mu + \alpha_i + \beta X_{ij} + \varepsilon_{ij},\ j = 1,\ldots,n_i,\ i = 1,\ldots,m, \tag{12.1}$$

where Y_{ij} is value of the response variable (*Yield*) at the *jth* location of the *ith* group (*Field*), μ is the population grand mean, α_i is the effect of Field i (initially assumed fixed), the quantity βX_{ij} is the effect explained by the variate X_{ij} (*Irrig*), and the $\varepsilon_{ij} \sim N(0,\sigma^2)$ are independent and identically distributed random variables for all i and j. As a first step, we carry out a traditional fixed effects analysis of covariance. This is accomplished by leaving `Irrig` as a `numeric` object rather than converting it to a factor in the arguments to the function `aov()`.

```
> Yld.aov <- aov(Yield ~ factor(Field)+ Irrig , data = data.Set3a)
> summary(Yld.aov)
               Df    Sum Sq  Mean Sq F value     Pr(>F)
factor(Field)  14 630893068 45063791  74.899 < 2.2e-16 ***
Irrig           1  42761520 42761520  71.072  1.51e-15 ***
Residuals     297 178693447   601661
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The result, not surprisingly, is a highly significant effect of both field and irrigation effectiveness.

There are several very serious problems with this analysis. The first is that by assuming the regression coefficient β in Equation 12.1 to be fixed, the analysis implicitly assumes an affirmative answer to one of the questions being asked, namely, whether the response of individual fields to irrigation effectiveness can be considered consistent across fields. As such, the result does not really provide assistance in answering the other question, whether there is a significant field scale yield response to irrigation effectiveness. These are not the only problems, however. The field data, because they are spatially autocorrelated, may violate the assumptions of the model, which include independence of errors. In addition, the assumption that the α_i are fixed means that the results of the analysis should be interpreted only for these specific 16 fields. We are not, however, interested in just these fields, but rather in using these fields as a representative of a larger population of rice fields in the region.

These defects in the model can be addressed by analyzing the model of Equation 12.1 as a *mixed effects model*, which is generally called simply a *mixed model* in the modern literature. The traditional interpretation of a linear model such as the ANCOVA model of Equation 12.1 treats the effects (field effect α_i and irrigation effect β) as *fixed effects*, that is, as nonrandom, fixed quantities. The results of the analysis are then valid only for these values of the predictors, which means in the present example that the results are only valid for these specific 16 fields. An alternative is to consider some or all of the effects as *random effects*, that is, as random variables whose values are considered to be drawn from a hypothetical larger pool of effects. In our case, this is interpreted as meaning that the fields are randomly drawn from a population of fields, that the irrigation effectiveness of each field is itself a random variable drawn from a population of irrigation effectiveness values, or both. A mixed effects model is one in which some of the effects are random and some are fixed.

In a mixed-model analysis, both the objectives and the procedure are different from those of a fixed effects study. Suppose, for example, that in the ANCOVA model (Equation 12.1), the field effect α_i is random. The procedure is different in that instead of selecting certain specific fields, one hypothetically draws the fields at random from the entire population of fields. These fields represent the variability of the total population. The study forgoes the opportunity to meaningfully compare the effect on yield response of the specific fields, and instead focuses on the question of variability of the population of fields.

Of course, as is common with field studies, reality and theory do not match very well. The fields in Data set 3 were not chosen at random; they were selected haphazardly from a set of volunteers (cf. Section 5.1 and Cochran, 1977, p. 16). On the other hand, there is no specific interest in these fields, or in these particular farmers. The researchers really do intend these fields to be representative of rice fields throughout the region. The region itself is not precisely specified, but is assumed to consist of similarly managed rice fields throughout east-central Uruguay. The question that must be answered in deciding whether an effect is fixed or random is whether the effect level can reasonably be assumed to represent a probability distribution (Henderson, 1982; Littell et al., 2002, p. 92; Searle et al., 1992, p. 16). It is important to emphasize that it is not the *factor* but rather the *effect* that is assumed random. For example, suppose a measurement is made in successive years. The years themselves are certainly not randomly chosen, but if there is no trend in the data, and there is no interest in the specific years in which the measurements were made, then it may be that the year effects can be taken to be random (although they may be autocorrelated). This permits us to relax slightly the assumption that the fields are randomly drawn; we must

instead assume that, however the fields are drawn, their effects are a random sample of the population of field effects.

Mixed model analyses are based on a grouping factor. The grouping factor "groups" the data in the sense that data values within the same group are presumed to be more related than data records from different groups. In our example, the grouping factor is the field. This will be treated as a random effect. Although it is not always evident from the computer output, mixed model analysis is based on the accurate estimation of the variance of the grouping factor. It is necessary to have more than a few factor levels to get good variance estimates and, hence, results from the mixed-model analysis. For this reason, it is generally not recommended that a mixed-model analysis be carried out if there are fewer than about six to eight levels of the grouping factor (Littell et al., 2002, p. 168). If there are insufficient levels, the analysis should proceed using a fixed-effects model, with no pretense of applying the results to a wider range of cases.

The basic mixed model is discussed in Section 12.2, and this model is then applied to Data Set 3 in Section 12.3. The application of mixed models to spatially autocorrelated data is discussed in Section 12.4. The remaining two sections are devoted to a pair of other methods that are similar to the mixed model. Section 12.5 discusses generalized least squares, which is applied to data for which there is no grouping variable. Section 12.6 discusses the quasibinomial model, which is an iterative method used to deal with autocorrelated data.

12.2 Basic Properties of the Mixed Model

Consider again the ANCOVA model of Equation 12.1. The grand mean μ is a fixed effect, and we initially continue with the assumption that the irrigation effectiveness β is a fixed effect as well. We replace the fixed effect assumption on the field effects α_i with the assumption that the α_i are random variables with mean zero and variance σ_α^2. In addition, the α_i and the ε_{ij} are assumed to be mutually independent. The quantities σ^2 and σ_α^2 are called the *variance components* of the model (Searle, 1971, p. 376). As described in the previous section, the random variable α whose levels are the α_i may be considered as a grouping factor of the data.

One crucial difference between the fixed-effects model and the mixed-effects model concerns the covariance relationship between the response variables. In the case of the fixed-effects model, the only random variables are the independent errors, and therefore the response variables Y_{ij} are also independent with the same variance σ^2 as the error terms ε_{ij}. In the case of the mixed model, this is no longer true. The quantities α_i and ε_{ij} are random variables and both contribute to the variance and covariance of Y_{ij}. We would expect the Y_{ij} values from the same group (i.e., same Field i in our example) to be more related, and this is indeed the case. It turns out that the covariance between the Y_{ij} is (Stanish and Taylor, 1983)

$$\begin{aligned} \text{cov}\{Y_{ij}, Y_{i'j}\} &= \frac{\sigma_a^2}{\sigma_a^2 + \sigma^2}, \quad i = i' \\ \text{cov}\{Y_{ij}, Y_{i'j'}\} &= 0, \quad i \neq i'. \end{aligned} \tag{12.2}$$

To carry out a hypothesis test on the coefficient β of the model (Equation 12.1), we use the general linear test, which is described in Appendix A.3 and discussed in Section 8.3.

To recapitulate, if, for example, one wishes to test the null hypothesis $H_0 : \beta = 0$ against the alternative hypothesis $H_a : \beta \neq 0$, then one regards model (Equation 12.1) as the *full* model, denoted with the subscript *F*, and develops a model called the *restricted* model, denoted with the subscript *R*, in which the null hypothesis is satisfied. In our case, the restricted model is

$$Y_{ij} = \mu + \alpha_i + \varepsilon_{ijk} \tag{12.3}$$

The restricted model of Equation 12.3 is nested in the full model of Equation 12.1 because the model of Equation 12.3 can be derived from that of Equation 12.1 by restricting one or more of the parameters of the model (Equation 12.1) to certain values. In this particular example, the value of β is restricted to $\beta = 0$. If the null hypothesis is true, then a statistic involving the ratio of the sums of squares (Equation A.44 in Appendix A.3) has an *F* distribution, and thus one can compute a *p* value based on this distribution. However, the statistic *G* of Equation A.44 has an *F* distribution only if the Y_{ij} are independent (Kutner et al., 2005, p. 699). A consequence of the nonzero covariance structure of Equation 12.2 is, therefore, that the general linear test on the sums of squares can no longer be used to test H_0. The maximum likelihood method, however, remains valid when the response variables are not independent. This method is described in Appendix A.5.

We continue the discussion considering the case in which the full model is given by Equation 12.1, $Y_{ij} = \mu + \alpha_i + \beta X_{ij} + \varepsilon_{ij}$ and the restricted model is given by restricting $\beta = 0$ to obtain Equation 12.3, $Y_{ij} = \mu + \alpha_i + \varepsilon_{ij}$. According to the discussion in Appendix A.5, if there are k_F parameters in the full model and k_R in the restricted model, with $k_F > k_R$, and if L_F is the maximum likelihood estimate of the full model and L_r that of the restricted model, then asymptotically as *n* approaches infinity the quantity

$$G = -2\log(L_F / L_R) = 2(l_F - l_R) \tag{12.4}$$

has a chi-square distribution with $(k_F - k_R)$ degrees of freedom (Kutner et al., 2005, p. 580; Theil, 1971, p. 396). In our case, $l_F = l(\mu, \alpha, \beta, \sigma^2 \mid Y)$, $l_R = l(\mu, \alpha, \sigma^2 \mid Y)$ and $k_F - k_R = 1$.

The likelihood ratio test is implemented by computing the maxima of the log likelihood statistics l_F and l_R and subtracting them. Finding these maxima is a nonlinear optimization problem. There are two methods used to carry out this optimization, the *restricted maximum likelihood* (REML) method and the ordinary maximum likelihood (ML) method (Appendix A.5). The REML provides better variance estimates than the ML. However, it cannot be used in a likelihood ratio test of a fixed-effect term. Therefore, we cannot use the REML to test the null hypothesis $\beta = 0$ and must instead use the full maximum likelihood method.

There is one further advantage to using the maximum likelihood method, or the restricted maximum likelihood method if it were available, in our ANCOVA problem of testing the effect of irrigation effectiveness on yield. An examination of the data set reveals that different numbers of measurements were made in different fields. This is expressed in Equation 12.1, which describes the model, in the condition $j = 1, \ldots, n_i$, $i = 1, \ldots, m$, that is, in the fact that the value of n_i, the number of data records in Field *i*, depends on *i*. Such an experimental design is said to be *unbalanced*. The maximum likelihood method can be adapted to deal with unbalanced data in simple ANCOVA models (Henderson, 1982). McCulloch et al. (2008, p. 95) provide the mathematics for the adaptation for unbalanced data of the model of Equation 12.1 in which the term α_i is random.

12.3 Application to Data Set 3

We now return to the problem of determining the field scale relationship between irrigation effectiveness and yield. To summarize, the initial full model is given by Equation 12.1, $Y_{ij} = \mu + \alpha_i + \beta X_{ij} + \varepsilon_{ij}$, $j = 1,...,n_i$, $i = 1,...,m$. The ultimate goal is to determine whether irrigation effectiveness has a consistent, positive effect on yield at the field scale. We will carry out the analysis in two stages. First, we will let the field effect α be a random variable, and then we will include a random component in the irrigation effect. With β as a fixed effect, we test the null hypothesis $H_0 : \beta = 0$ against the alternative $H_a : \beta \neq 0$. Since β is a fixed effect, we must use the ML rather than REML method. The restricted model, obtained by setting $\beta = 0$ in Equation 12.1 is given by Equation 12.3.

We will test this and other hypotheses involving mixed models using functions from the `nlme` package. The use of the functions in this package is extensively described by Pinhiero and Bates (2000). The package has the major (for us) advantage that it includes the capacity to model spatially autocorrelated data. The fundamental mixed linear modeling function in this package is `lme()`. The primary arguments of this function are (1) the *fixed* part of the model, (2) the data source, (3) the *random* part of the model, and (4) the optional method specification. The function `anova()`, when applied to an object created by `lme()`, carries out a likelihood ratio test as described in the previous section.

The first call to `lme()` generates the restricted model.

```
> Yld.lmeR1 <- lme(Yield ~ 1, data = data.Set3a,
+    random = ~ 1 | Field, method = "ML")
```

The function call and the model are interpreted together as follows:

$$Y_{ij} = \mu \,\|\, + \alpha_i + \varepsilon_{ij} \tag{12.5}$$

`lme(Yield ~ 1,…,‖ random = ~ 1 | Field)`

The vertical double line in the model (which is included for clarification and is not part of the R code) equation separates the fixed part on the left from the random part on the right. The corresponding symbol in the line of R code represents the same separation of fixed and random effects. The only fixed component is the mean μ, which is represented by the number 1. As usual, the random error term ε_{ij} is implicitly included. The random part is the field effect α_i. This is represented by the statement `random = ~ 1 | Field`. The quantity on the right of the vertical bar is the grouping factor, which in this case is the field. The quantity on the left of the bar is called the *primary covariate* and represents the factor (if any) that is grouped by the grouping factor. In this case, the only fixed effect is the grand mean, and this is indicated by the 1 to the left of the bar, so there is no primary covariate. The second call to `lme()` generates the full model.

```
> Yld.lmeF1 <- lme(Yield ~ Irrig, data = data.Set3a,
+    random = ~ 1 | Field, method = "ML")
```

The function call and the model are interpreted together as follows:

$$Y_{ij} = \mu + \beta X_{ij} \,\|\, + \alpha_i + \varepsilon_{ij} \tag{12.6}$$

`lme(Yield ~ Irrig,‖ random = ~ 1 | Field)`

Since there is another fixed effect besides the grand mean, the mean itself can be made implicit. The variable representing irrigation effectiveness is incorporated into the fixed effects, and the random effect is unchanged. Both of these models could be examined using the function `summary()`. We will put this off for now. The likelihood ratio test is carried out using the function `anova()`.

```
> anova(Yld.lmeR1, Yld.lmeF1)
          Model df      AIC      BIC    logLik   Test  L.Ratio p-value
Yld.lmeR1     1  3 5186.099 5197.337 -2590.049
Yld.lmeF1     2  4 5117.571 5132.556 -2554.785 1 vs 2 70.52764  <.0001
```

The effect of irrigation effectiveness is highly significant. There is a complication involved in interpreting the results of this hypothesis test of a fixed effect that will be discussed below.

The next step in the analysis is to determine whether the effect of irrigation effectiveness on yield can be considered as constant over fields. We treat irrigation effectiveness as the sum of a fixed effect β that represents the mean irrigation effectiveness effect and a random effect γ that represents the individual field's irrigation effectiveness effect (Searle, 1971, p. 355). The former full model of Equation 12.1 becomes the new restricted model and the new full model, along with its representation in `lme()`, is

$$Y_{ij} = \mu + \beta X_{ij} \,\|\, + \alpha_i + \gamma_i X_{ij} + \varepsilon_{ij} \tag{12.7}$$

```
lme(Yield ~ Irrig, || random = ~ Irrig | Field)
```

where γ_{ij} represents the random component of the total irrigation effect. We can update the model in R using the function `update()`. We can use the restricted maximum likelihood method in this case because our null hypothesis involves a random effect, so we drop the argument `method = "ML"`.

```
> Yld.lmeR2 <- lme(Yield ~ Irrig, data = data.Set3a,
+    random = ~ 1 | Field)
> Yld.lmeF2 <- update(Yld.lmeR2, random = ~ Irrig | Field)
> anova(Yld.lmeR2, Yld.lmeF2)
          Model df      AIC      BIC    logLik   Test L.Ratio p-value
Yld.lmeR2     1  4 5093.738 5108.697 -2542.869
Yld.lmeF2     2  6 5077.556 5099.994 -2532.778 1 vs 2 20.1826  <.0001
```

The results of the call to `anova()` indicate that we cannot drop the random component of the irrigation effectiveness from the model, that is, that the effect of irrigation cannot be considered constant across fields. Instead, it is the sum of a fixed effect, which is the mean irrigation effect over all the fields, and a random effect, which is the effect of that individual field.

Returning to the test carried out of the null hypothesis that the fixed effect β equals zero, the function `anova()` applied to the comparison of the models (Equation 12.5) and (Equation 12.6) implements a likelihood ratio test and gives a p value less than 0.0001,

which we would like to interpret as indicating the fixed effect of irrigation effectiveness is significant in the model. As was mentioned earlier, there is a complication that must be addressed.

Recall that the log likelihood ratio asymptotically follows a χ^2_k distribution, where k is the difference in the number of degrees of freedom between the full and restricted model. The function `logLik()`, when applied to a model, returns the log likelihood and the number of degrees of freedom, so this can be used to determine k. Pinhiero and Bates (2000, p. 89) indicate that the likelihood ratio test applied to the fixed effects of a model may be "anticonservative," that is, may have an inflated Type I error rate. Since we have rejected the null hypothesis in this case at a p value less than 0.0001, we should not be too concerned about making a Type I error. However, we can double check the result using the function `simulate.lme()` from the `nlme` package. This function generates a Monte Carlo simulation of the model to produce an empirical distribution of p values. These can be plotted, and if the curve of nominal p values falls below the 45° line, then this indicates "anticonservative" behavior. Here is the code to carry out this test.

```
> logLik(Yld.lmeR1)
'log Lik.' -2590.049 (df=3)
> logLik(Yld.lmeF1)
'log Lik.' -2554.785 (df=4)
> sim.lme <- simulate.lme(Yld.lmeR1,Yld.lmeF1,
+   nsim = 1000, seed = 123)> plot(sim.lme, df = 1)
```

Figure 12.2 shows a plot of the values for the models `Yld.lmeR1` and `Yld.lmeF1`. For $df = 2$, the p values lie along the 45° line, which indicates that there should be little problem with accepting the failure to reject the null hypothesis in this case. For more details, see Pinhiero and Bates (2000, p. 89).

In the test of the model of Equation 12.7, we are not actually testing the hypothesis $\gamma_i = 0$. The γ_i are random variables with mean zero, and the null hypothesis is $H_0 : \sigma^2_\gamma = 0$ against the alternative $H_a : \sigma^2_\gamma > 0$, where σ^2_γ is the variance of γ. Of course, if γ has mean zero and variance zero, then it is identically zero, so in effect the test is the same. There is, however, a complication with the p values of this test. Pinhiero and Bates (2000, p. 86), citing the results of Stram and Lee (1994), point out that the value of σ^2_γ that satisfies the null hypothesis is "on the boundary of the parameter space," (i.e., the boundary of the range of possible values of σ^2_γ). This may make the likelihood ratio test conservative, that is, it may cause an inflated Type II error rate, failing to reject the null hypothesis when it is false. Pinhiero and Bates (2000, p. 86) suggest simulating the test at one degree of freedom less than the nominal difference between the restricted and full as well as computing a range of p values at a mixture of these two degrees of freedom values. Here is the code to carry out this simulation.

```
> logLik(Yld.lmeR2)
'log Lik.' -2542.869 (df=4)
> logLik(Yld.lmeF2)
'log Lik.' -2532.778 (df=6)
> sim.lme <- simulate.lme(Yld.lmeR2,Yld.lmeF2, nsim = 1000, seed = 123)
> plot(sim.lme, df = c(1,2))
```

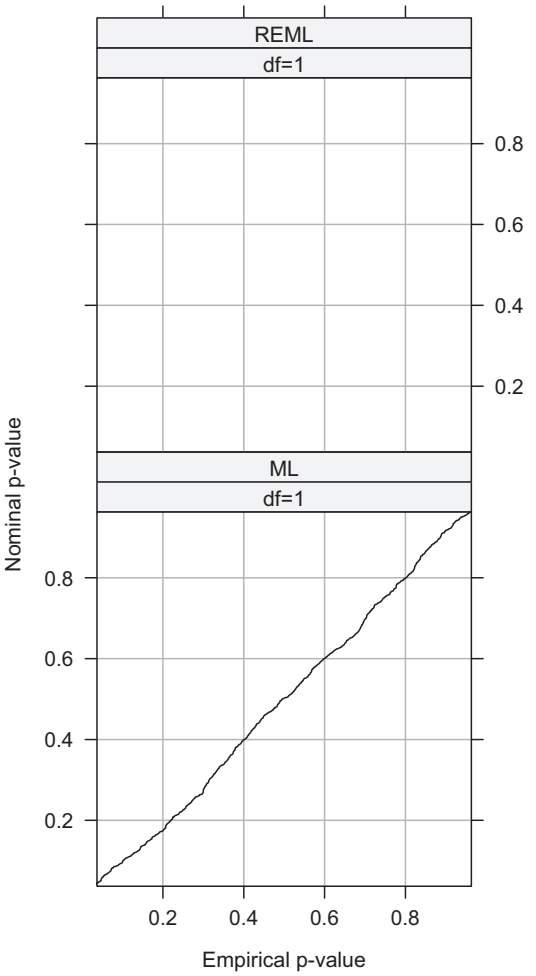

FIGURE 12.2
Plot of the nominal p value of the test of the hypothesis $\beta = 0$ against the empirical value generated by a Monte Carlo simulation. The approximately straight line of equal values for maximum likelihood indicates that the nominal p value can be accepted.

The result, shown in Figure 12.3, indicates that the nominal p values tend to be higher than the empirical ones; in other words, the test tends to be conservative. Since the test rejects the null hypothesis ($p < 0.0001$), we can conclude that the rejection of the null hypothesis is appropriate and that the effect of irrigation effectiveness is not constant across fields.

12.4 Incorporating Spatial Autocorrelation

Until now, the mixed model analysis has included the assumption that the residuals ε_{ij} are independent, normally distributed random variables with fixed variance σ^2. We now incorporate the effects of spatial autocorrelation into the model. One of the reasons for

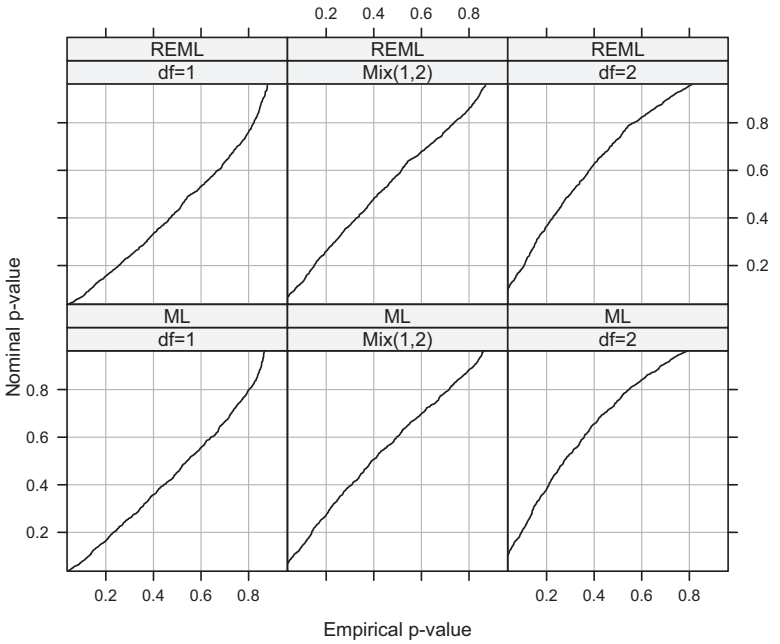

FIGURE 12.3
Plots of nominal versus empirical p values for the REML and ML methods for the test of H_0: $\sigma_\gamma^2 = 0$ against the alternative $H_a : \sigma_g^2 > 0$.

using mixed model analysis for the testing of hypotheses involving spatially autocorrelated data is that the maximum likelihood method does not require the assumption of independent errors in a model such as that of Equation 12.7. The function `lme()` has the ability to incorporate spatial autocorrelation by calculating a variogram and incorporating a variogram model (or other correlation models) into the correlation structure. The available variogram models include exponential, Gaussian, linear, quadratic, and spherical.

The `nlme` package contains the function `Variogram()` (capital *V*), which computes an empirical variogram (Section 4.6) for the residuals of the mixed model. This variogram can be plotted and inspected and, if appropriate, a variogram model can then be used to define the error correlation structure of the data. This correlation structure is then incorporated into the analysis. The function `Variogram()` takes two primary arguments. The first argument is an `lme` object that is used to obtain the regression residuals. Pinhiero and Bates (2000, p. 245) provide examples of the use of this function. Taking up from where we left off in Section 12.2, here is how we initially apply it to compute and plot the variogram for the residuals of the full model `Yld.lmeF2` for our problem.

```
> var.lmeF <- Variogram(Yld.lmeF2, form = ~ Easting + Northing)
```

The variogram in Figure 12.4 was produced by a call to the function `plot()`. A word of caution is advisable here about this function. It is the basic workhorse plotting function that produces many of the graphs displayed throughout this book. These graphs are not, however, all produced by the same function, and not even all by the same type of function. This is an example of the use of polymorphism, described in Sections 2.5 and 2.6.2. In brief, the specific form of the function depends on the arguments. In the traditional graphics version of `plot()`, the first argument is a vector of x coordinates. In other applications of

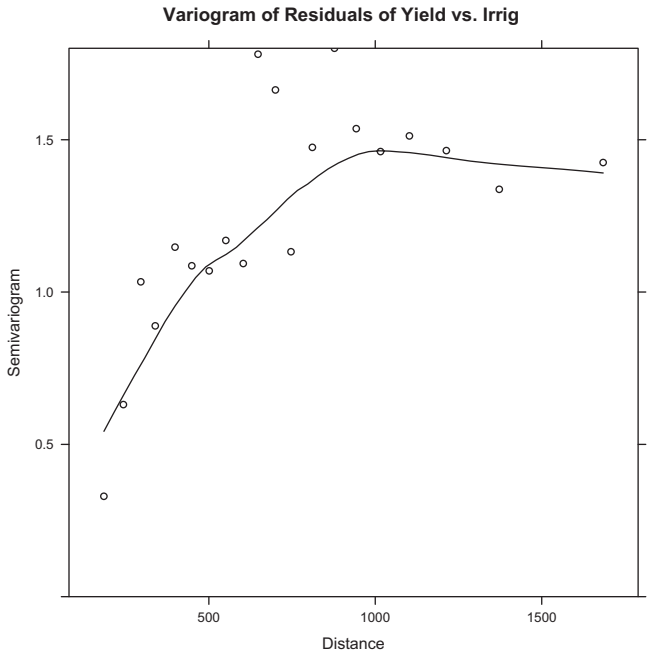

FIGURE 12.4
Variogram of the residuals of the full model of Equation 12.9 incorporating spatial autocorrelation.

`plot()`, the first argument is an R object that determines the form of the function `plot()` that will be used. In the creation of Figure 12.3, the first coordinate is a `simulate.lme` object. In the creation of Figure 12.4, the first argument is a `Variogram` object. Why does this matter? Because it crucially affects how we exercise special control over the graphics. The plot in Figure 12.4 looks like an ordinary traditional graphics plot. In its unmodified form, the plotted material is blue, but for best reproduction we would like it to be black. Following the normal procedure with the traditional graphics version of `plot()`, we would enter the following.

```
> plot(var.lmeF, col = "black") # This doesn't work
```

This doesn't work (try it!). Typing `?plot.Variogram` brings up the Help screen, which indicates that, although it creates what looks like a traditional graphics plot, `plot.Variogram()` is actually a trellis graphics function based on the `lattice` package. Therefore, in order to obtain a black-and-white graph, we must use a trellis graphics statement such as the following.

```
> trellis.device(color = FALSE)
```

Figure 12.4 shows that the variogram of the residuals of the model of Equation 12.7 has the shape associated with a spherical correlation structure of Equation 4.22 (Isaaks and Srivastava, p. 374; Pinhiero and Bates, 2000, p. 233), and so this correlation structure will be used in the model.

We are now almost ready to carry out a likelihood ratio test for the model with spatial autocorrelation against the model without it. The correlation structure is introduced

into the residuals through the use of an argument in `lme()` of the form `correlation = corSpher(form = ~ X + Y, nugget = TRUE)`. This describes the correlation structure of the data. The symbols `X` and `Y` represent the data fields that identify the x and y coordinates of the data record. In our example, these are `Easting` and `Northing`. Let's try plugging these in.

```
> # This produces an error
> Yld.lmeF3 <- update(Yld.lmeF2, correlation
    + = corSpher(form = ~ Easting + Northing, nugget = TRUE))
Error in getCovariate.corSpatial(object, data = data) :
  Cannot have zero distances in "corSpatial"
```

The error occurs because some of the data records contain data that are measured in successive years at the same location, and therefore the distance between some of the data records is zero. We can get around this by using the function `jitter()` to add a very small random quantity to every x and y coordinates, not enough to affect the result but enough to avoid the zero distance problem. As usual, we set a random number seed. It turns out that our usual seed value of 123 produces a particularly extreme result (try it), so we use a different one.

```
> set.seed(456)
> data.Set3a$EAST2 <- jitter(data.Set3a$Easting)
> data.Set3a$NORTH2 <- jitter(data.Set3a$Northing)
```

Now we revise the model to include the jittered coordinates.

```
> # Redo the model with the new data.Set3a
> Yld.lmeF2a <- update(Yld.lmeF2, data = data.Set3a)
> var.lmeF2a <- Variogram(Yld.lmeF2a, form = ~ EAST2 + NORTH2)
> # Check that the variogram hasn't changed visibly
> plot(var.lmeF2a)
```

Now we are ready to carry out the hypothesis test involving spatial autocorrelation. Formally, the null hypothesis is $H_0: cor\{\varepsilon_i, \varepsilon_j\} = 0$, $i \neq j$, and the alternative is that the correlation is not zero for some i and j. We must be sure that the restricted model is nested in the full model.

The full model is given by

$$\begin{aligned} Y_{ij} &= \mu + \beta X_{ij} + \alpha_i + \gamma_i X_{ij} + \varepsilon_{ij}, \\ (\alpha_i, \gamma_i)' &\sim N(0, \Psi),\ \varepsilon_i \sim N(0, \sigma^2 \Lambda_i), \end{aligned} \tag{12.8}$$

where Λ_i is a variance-covariance matrix whose off-diagonal terms are not necessarily zero. This is analogous to the general form of the mixed model given by Pinhiero and Bates (2000, p. 202), which is known as the *Laird-Ware* form. The expression $(\alpha_i, \gamma_i)' \sim N(0, \Psi)$ indicates that the random effects may be correlated. This correlation was also generally present in the analysis of the mixed model with independent errors, as indicated in Equation 12.2.

The expression $\varepsilon_i \sim N(0, \sigma^2 \Lambda_i)$ indicates that the magnitude of the variance of the error terms is represented by the parameter σ^2, and that error terms from the same grouping

factor (same i) may be correlated. One way (the way we will use) to represent the correlation is by using a correlogram (Section 4.2.6). We can write (Pinhiero and Bates, 2000, p. 205)

$$\Lambda_i = V_i R_i V_i, \tag{12.9}$$

where V_i is a diagonal matrix and R_i is a correlation matrix. Since we are modeling this correlation structure by fitting a spherical variogram model to the residuals, `lme()` uses the relationship $\rho(h) = (\gamma(h) - C(0))/C(0)$ obtained by combining Equations 4.25 and 4.26. It models the correlation structure in Equation 12.9 by incorporating the range of the variogram into the matrix R_i and the square root of the sill into the V_i.

The restricted model is given by Equation 12.7, which we repeat here, including explicitly the conditions on the random effects and errors:

$$\begin{aligned} &Y_{ij} = \mu + \beta X_{ij} + \alpha_i + \gamma_i X_{ij} + \varepsilon_{ij}, \\ &(\alpha_i, \gamma_i)' \sim N(0, \Psi),\ \varepsilon_i \sim N(0, \sigma^2 I). \end{aligned} \tag{12.10}$$

As just mentioned, the expression $\varepsilon_i \sim N(0, \sigma^2 I)$ indicates that the error terms are independent and with constant variance σ^2. Therefore, the restricted model is indeed a special case of the full model in which $\text{cov}\{\varepsilon_i, \varepsilon_j\} = 0$ for $i \neq j$, and so the restricted model is nested in the full model. Therefore, we are justified in using the likelihood ratio test.

Here is an implementation in R.

```
> Yld.lmeF3 <- update(Yld.lmeF2a, correlation
    + = corSpher(form = ~ EAST2 + NORTH2, nugget = TRUE))
> anova(Yld.lmeF2, Yld.lmeF3)
          Model df      AIC      BIC    logLik   Test  L.Ratio p-value
Yld.lmeF2     1  6 5077.556 5099.994 -2532.778
Yld.lmeF3     2  8 5078.361 5108.279 -2531.180 1 vs 2 3.195141  0.2024
```

Based on the results of the test, we can conclude that the simpler model without autocorrelation is adequate. We conclude that the most appropriate model is that of Equation 12.9, represented by the R object `Yld.lmeF2`, in which both the parameter representing the field and that representing irrigation effectiveness are random effects, and the errors are spatially uncorrelated.

Turning to the interpretation of the results, the function `summary()` as usual provides us with a lot of information.

```
> summary(Yld.lmeF2)
Linear mixed-effects model fit by REML
 Data: data.Set3a
       AIC      BIC    logLik
  5077.556 5099.994 -2532.778
Random effects:
 Formula: ~Irrig | Field
 Structure: General positive-definite, Log-Cholesky parametrization
            StdDev     Corr
(Intercept) 2092.8975  (Intr)
Irrig        451.3202  -0.83
```

```
Residual     730.4511       Fixed effects: Yield ~ Irrig
              Value Std.Error  DF  t-value p-value
(Intercept) 4410.58 615.9446 297 7.160676       0
Irrig        660.28 141.5584 297 4.664363       0
Correlation:
      (Intr)
Irrig -0.867
Standardized Within-Group Residuals:
        Min          Q1         Med          Q3         Max
-3.21519988 -0.60943706 0.03328884  0.52766754 3.11924207

Number of Observations: 313
Number of Groups: 15
```

The estimated variance components are given as standard deviations in the information about random effects. They are $\sigma = 730.4511$, $\sigma_\gamma = 451.3202$, $\sigma_\alpha = 2092.8975$. The value of β, the mean effect of irrigation effectiveness, is given in the fixed effects as 660.28. Since irrigation effectiveness is an ordinal variable, we cannot assign too much meaning to this beyond the statement that on average, irrigation effectiveness has a significantly positive effect on yield. The correlation between the intercept term a and the irrigation effect term γ is −0.867. The intercept term α represents the random effect of the field on yield and can be interpreted as the effect of everything about the field but irrigation effectiveness. The random effect γ represents the variation with field of the rate of change of yield with increasing irrigation effectiveness. The negative correlation indicates that the association between irrigation effectiveness is reduced as the intercept of the plot field's yield is increased. It is important to note, however, that the intercept is *not* the mean yield of the field.

Because irrigation effectiveness is an ordinal variable, we cannot with any sort of theoretical justification use these results to predict yield. In other cases, however, the concomitant variable may be interval or ratio scale, in which case such a prediction would be justified. To see what we could do under those circumstances, we will pretend for the moment that irrigation effectiveness is of an appropriate scale to justify this sort of prediction. To get an idea of the regression relationships of each field we can use the `nlme` function `augPred()` (for "augmented prediction"). This is most effectively used with the function `plot()` as follows.

```
> plot(augPred(Yld.lmeF2, as.formula("~Irrig"))) # Fig. 12.5
```

Figure 12.5 shows the results of this statement, which generates a regression line for each field. The second argument in the function `augPred()` specifies the primary covariate of the grouped data as defined above. These plots demonstrate that because the regression predictor variables X_i data are not *centered*, that is, because the model in Equation 12.9 is written as it is and not as $Y_{ij} = \mu + \alpha_i + (\beta + \gamma_i)(X_{ij} - \bar{X}_i) + \varepsilon_{ij}$, the intercept of the regression line is not the mean yield for the field but rather is related both to the slope and the mean yield. The figure indicates that the negative covariance between the intercept and the slope of the regression line is due partly (or maybe mostly) to the fact that regression lines with a steep slope tend to have a lower intercept. Inspection of the figure indicates that there is not much, if any, relationship between regression slope and mean yield.

Finally, we can compare the values of the coefficients generated by the individual models using `lmList()`with those generated using the model `Yld.lmeF2`. They are displayed side by side for ease of comparison.

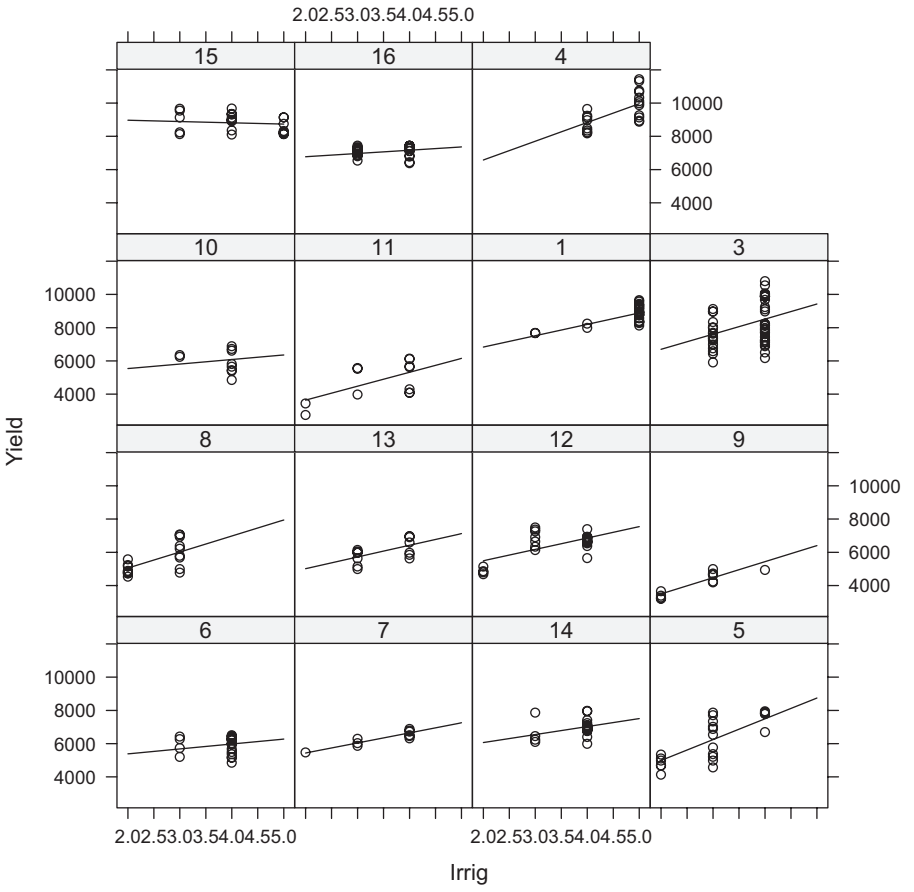

FIGURE 12.5
Regression of yield versus irrigation effectiveness by field, as generated by the model in Equation 12.9.

```
> coef(data.lis)
   (Intercept)       Irrig
1     5679.500  642.500000
3     4644.995  976.433155
4     3664.417 1277.833333
5     2041.734 1404.966805
6     5918.529   -3.926471
7     4360.978  562.777778
8     2680.694 1138.027778
9     1490.786  975.142857
10    7414.375 -373.625000
11    1828.181  861.337349
12    4107.708  687.388740
13    3567.714  710.000000
14    5647.850  339.050000
15    9845.544 -246.025316
16    7041.615    6.512821
```

```
> coef(Yld.lmeF2)
   (Intercept)       Irrig
1     5467.691  683.22006
3     4881.098  908.86211
4     4321.939 1127.59501
5     2487.202 1252.04669
6     4795.256  295.92892
7     4224.195  606.28036
8     3122.792  964.31254
9     1564.360  967.92928
10    4994.641  272.53620
11    1966.939  837.58022
12    4118.344  685.14777
13    3609.932  703.03172
14    5105.962  480.40071
15    9121.889  -77.50652
16    6376.453  196.83426
```

Statistically, these two outputs of the function `coef()` represent two completely different things. The left column, which was generated by the function `lmList()`, represents a set of best linear unbiased estimates (BLUE) of the fixed effect β for a set of 15 independent regressions of yield on irrigation effectiveness. The column on the right represents a set of best linear unbiased predictors (BLUP) of the random effect γ_i added to the estimate of the fixed effect β for a single regression of yield on irrigation effectiveness. Each coefficient in the left column is generated independently with no influence from the data of other fields, while the coefficients in the right column are based on the data from all the fields. Generally the two sets of coefficients tend to be similar, but the extreme cases in the left-hand column are not as extreme in the right-hand column.

12.5 Generalized Least Squares

We now consider applying the mixed model concept to the data of Field 1 of Data Set 4. In Section 9.3, we constructed a multiple linear regression model for the field. We would like to be able to apply the mixed-model method to regression models such as this one to try to account for spatial autocorrelation of the errors, but there are no variables in these regression models that can reasonably be considered to be random effects. In other words, there is no grouping variable. In this case, Equation 12.8 becomes, in matrix notation,

$$\begin{aligned} Y &= \mu + \beta X + \varepsilon, \\ \varepsilon &\sim N(0, \sigma^2 \Lambda). \end{aligned} \tag{12.11}$$

Such a model, with no grouping variable, is referred to as a *generalized least squares* model (Pinhiero and Bates, 2000, p. 204), and can be solved using the same techniques as those described for the solution of Equation 12.8. Before discussing this, it might be worthwhile to review some of the other models with similar names. The term *generalized* has appeared several times, so let's make sure we have them all straight. The first use was the generalized linear model (GLM) in Section 8.4. This has a form such as $g(\pi_i) = \beta_0 + \beta_1 X_i$ and is used, for example, when the response variable is binomial. The next use was in Section 9.2 with the generalized additive model (GAM), which has the form $g(\pi_i) = \beta_0 + f(X_i)$ where $f(X)$ is a smooth function such as a spline. This is used to provide a better fit to the data. The generalized least squares model of Equation 12.11 is used as one form of solution of a linear regression model where the error structure can be modeled using a variogram.

The `nlme` package contains a function `gls()` that can be applied to generalized least squares regression in the same way that `lm()` is applied to ordinary linear regression. We will apply this function to the analysis of the data from Field 4.1 following an example of similar analysis carried out by Pinhiero and Bates (2000, p. 260) on data from wheat variety trials. We developed five candidate models, but we will continue to focus on `model.5`. as an example. Our objective is to see whether the incorporation of spatial autocorrelation in the errors affects the result.

```
> model.5lm <- lm(Yield ~ Clay + SoilP + I(Clay*SoilP) + Weeds,
+ data = data.Set4.1)
> summary(model.5lm)
                Estimate Std. Error t value Pr(>|t|)
(Intercept)     16083.290 1523.702 10.555 < 2e-16 ***
```

```
Clay              -324.716 37.852 -8.578 5.42e-13 ***
SoilP             -890.167 205.394 -4.334 4.17e-05 ***
I(Clay * SoilP)   24.051 5.277 4.558 1.81e-05 ***
Weeds             -390.776 71.283 -5.482 4.64e-07 ***
```

This forms the null model for the analysis in which the errors may be correlated. We can re-express it in terms of `gls()` as follows.

```
> library(nlme)
> model.5gls1 <- gls(Yield ~ Clay + SoilP + I(Clay*SoilP) +
+     Weeds, data = data.Set4.1)
```

The `summary()` function, shown below, indicates that the coefficients are all exactly the same. We now develop the variogram model that can be used to characterize the spatial structure of the residuals. To exclude lag groups with very few members, the variogram is calculated for lag values less than or equal 300 m.

```
> plot(Variogram(model.5gls1, form = ~ Easting + Northing,
+     maxDist = 300), xlim = c(0,300), # Fig. 12.6
+     main = "Variogram of Residuals, model.5, Field 4.1")
```

The shape of the variogram indicates that the best model is probably spherical, although the variogram looks like it is practically pure nugget (Figure 12.6).

```
> model.5gls2 <- update(model.5gls1,
+     corr = corSpher(form = ~ Easting + Northing, nugget = TRUE))
```

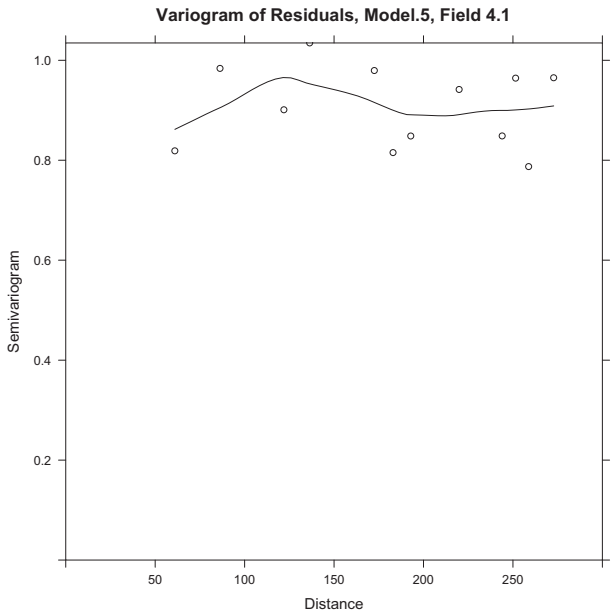

FIGURE 12.6
Variogram of the residuals of the generalized least squares model for the northern portion of Field 4.1.

Since the models are nested, we can compare them using the likelihood ratio test.

```
> anova(model.5gls1, model.5gls2)
           Model df      AIC      BIC    logLik   Test  L.Ratio p-value
model.5gls1     1  6 1344.236 1358.603 -666.1179
model.5gls2     2  8 1336.940 1356.096 -660.4700 1 vs 2 11.29591  0.0035
```

There is a significant difference, indicating that the effect of spatial autocorrelation of the errors is sufficient to make the generalized least square model preferable to the ordinary least squares model. The effect of including spatial autocorrelation on the Akaike information criterion (AIC) is relatively minor, as is shown by these excerpts from the `summary()` function.

```
> summary(model.5gls1)
       AIC      BIC   logLik
  1344.236 1358.603 -666.1179
Coefficients:
                    Value Std.Error   t-value p-value
(Intercept)     16083.290 1523.7018 10.555405       0
Clay             -324.716   37.8523 -8.578485       0
SoilP            -890.167  205.3942 -4.333945       0
I(Clay * SoilP)    24.051    5.2771  4.557588       0
Weeds            -390.776   71.2825 -5.482070       0
> summary(model.gls2)
       AIC      BIC    logLik
  1344.236 1358.603 -666.1179
Coefficients:
                    Value Std.Error   t-value p-value
(Intercept)     11815.244  55202.73  0.214034  0.8311
Clay             -201.002     51.46 -3.905613  0.0002
SoilP            -890.766    228.12 -3.904832  0.0002
I(Clay * SoilP)    21.850      5.79  3.777035  0.0003
Weeds            -255.535     77.38 -3.302258  0.0014
```

All of the coefficients except *SoilP* are reduced in magnitude in the correlated errors model. Autocorrelation of the residuals does not bias the coefficients by itself, but there are several possible things that may be going on. In Chapter 13, we will pursue this issue much more deeply.

12.6 Spatial Logistic Regression

12.6.1 Upscaling Data Set 2 in the Coast Range

This section describes a method called the *quasibinomial model* that allows the incorporation of spatial autocorrelation into regression models for binomial variables. The quasibinomial model is related to the generalized linear model discussed in Section 8.4. In Exercise 8.10, the blue oak presence-absence data of the Coast Range subset of Data Set 2 was analyzed using a multiple logistic regression model. One of the candidate models for blue oak presence versus absence in the Coast Range (as presented in the solution given

in Section 8.3) includes the explanatory variables *TempR*, *Permeab*, *Precip*, *GS32*, *PE*, and *SolRad6*. Mention was made in Section 8.3 of the complexity of earlier logistic regression models (Evett, 1994; Vayssieres et al., 2000) for this data set. It is possible that this complexity is in part due to the spatial scale of the data. Data Set 2 has over 4000 individual records. Relative to the extent of the data set, the support (Section 6.4.1) of the raw data in Data Set 2 is much smaller than that of the other data sets.

The method that we will present in Section 12.6.2 for incorporating spatial autocorrelation effects into regression models for binomial variables involves data that have been aggregated over a set of subregions. For this reason, we examine in greater detail the effect of data aggregation on analysis of the blue oak presence-absence data in the Coast Range subset of Data Set 2. There are no natural boundaries that we can use for aggregation, so the first order of business is to create a set of regions over which to aggregate. We would like to have regions that are geographically close and, since elevation is a major predictor of blue oak presence, we would like the regions to also be close in elevation. Therefore, we will search for clusters of similar values of longitude, latitude, and elevation. We will do this using *k*-means cluster analysis (Jain and Dubes, 1984). In this algorithm, *k* points in the data space are initially selected as cluster *seeds*. Clusters are formed by assigning all other points in the data space to the seed having the closest value. The means of each cluster are then selected as the new set of *k* seeds, and a set of clusters are formed around this set of seeds. This process is repeated iteratively. In theory, the process may be iterated to convergence, but in practice it is stopped after a predefined maximum set of iterations. In the present case, the data space is comprised of the triples of longitude, latitude, and elevation.

Clustering can be carried out in R using the function `kmeans()`. In this function the initial set of seeds, if it is not specified in the argument of the function, is selected at random. In our implementation of the method, we will search for $k = 50$ clusters. The `sf` object `data.Set2C` contains the Coast Range subset of Data Set 2, created in Section 7.3. We create a new version of this data set, `Set2.kmeans`, in which the quantities `Longitude`, `Latitude`, and `Elevation` are centered and scaled, and then `Elevation` is multiplied by 0.5 in order to put more emphasis on spatial contiguity.

```
> Set2.kmeans <- with(data.Set2C, data.frame(scale(Longitude),
+   scale(Latitude), 0.5 * scale(Elevation)))
> cluster.k <- 50
> set.seed(123)
> cl <- kmeans(Set2.kmeans, cluster.k, iter.max = 100)
> data.Set2C$clusID <- cl$cluster
```

Figure 12.7 shows the resulting clusters at the southern end of the Coast Range (a plot of the full Coast Range would be impossibly busy). Cluster membership of each point is represented by the point's symbol, and the cluster membership symbols are superimposed on the contours of a digital elevation model created by interpolating the elevation data. The heavy line is the California coast, and the map illustrates two of the problems associated with map overlay and the issues of scale discussed in Section 6.4. The first problem is that the contour map is an extrapolation in the sense of Bierkens et al. (2000, cf. Figure 6.10). That is, it extends the elevation values beyond the geographic region of the data, and therefore its extrapolated elevation values include regions where the predicted elevation of the ocean is over 200 m. The second problem is that the map of California on which the coastline is based was digitized at the extent of the entire United States, and therefore its accuracy does not match that of the site location data. This results in some tree locations

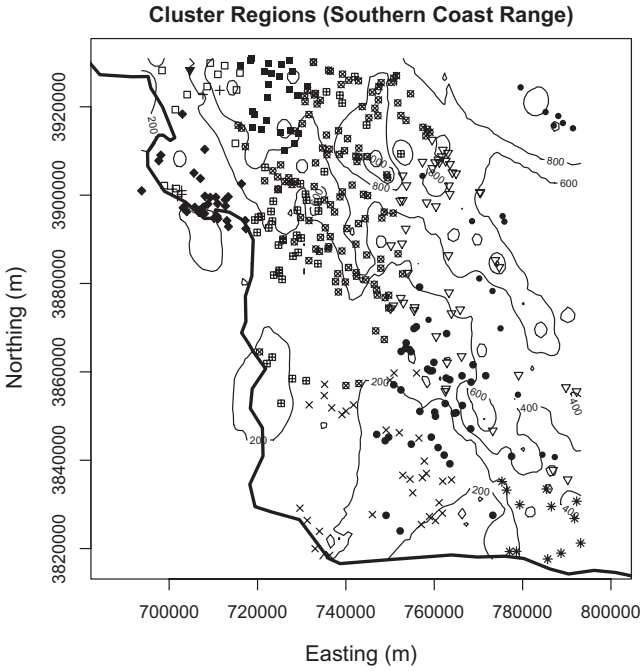

FIGURE 12.7
Cluster regions at the southern end of the Coast Range, shown superimposed over a digital elevation model of the region.

apparently being in the ocean. Since the map is used for purposes of visualization only, these problems will not affect our analysis.

A data frame called `data.Set2Cpts` is created to hold the attribute values on with the regression model will be based. We use the suffix "pts" in this case to emphasize that these are pointwise data, as opposed to clusterwise data set `data.Set2Cclus` that will be created next.

```
> data.Set2Cpts <- with(data.Set2C, data.frame(MAT, TempR, GS32,
+     Precip, PE, ET, Texture, AWCAvg, Permeab, SolRad6, SolRad12,
+     SolRad, QUDO, clusID))
> data.Set2Cpts$PM100 <- as.numeric(data.Set2C$PM100 > 0)
```

There are similar statements to the last one for each class of parent material.

Now we will create a data frame `data.Set2Cclus` to hold the aggregated data in the spatial clusters. We use the function `tapply()` to compute the averages of each data field over each cluster.

```
> data.Set2C.mat <- matrix(nrow = length(unique(data.Set2Cpts$clusID)),
+    ncol = ncol(data.Set2Cpts))
> for (i in 1:ncol(data.Set2Cpts)){
+    data.Set2C.mat[,i] <- tapply(data.Set2Cpts@data[,i],
+       data.Set2Cpts$clusID, mean)
+ }
> data.Set2Cclus <- data.frame(data.Set2C.mat)
> names(data.Set2Cclus) <- names(data.Set2Cpts)
```

Next, we convert the resulting matrix into a data frame and assign the appropriate name to each data field.

```
> data.Set2Cclus <- data.frame(data.Set2C.mat)
> names(data.Set2Cclus) <- names(data.Set2Cpts)
```

To summarize, let's look at the number of data records in each data frame.

```
> nrow(data.Set2Cpts)
[1] 1852
> nrow(data.Set2Cclus)
[1] 50
```

Your answer may vary slightly depending on how you constructed the Coast Range data set in Section 7.3. The data frame `data.Set2Cpts` contains the pointwise data, and the data frame `data.Set2Cclus` contains the data averaged over each cluster.

Now we are ready to construct the quasibinomial model. The response variable `QUDO` in `data.Set2Cclus` represents the fraction of sites in the cluster that have a blue oak present. The model for this type of data is also sometimes called a *logistic-binomial* model (Gelman and Hill, 2007, p. 116). Recall that in the case of the binomial model the response variable Y_i took on the value 1 with probability π_i and zero with probability $1-\pi_i$, and that the explanatory variables were related to the response variable through the link function (Equation 8.28)

$$g(\pi_i) = \ln\left(\frac{\pi_i}{1-\pi_i}\right) = \beta_0 + \beta_1 X_{i1} + \beta_2 X_{i2} + \ldots + \beta_{p-1} X_{i,p-1}. \tag{12.12}$$

The quasibinomial model uses the same link function, but now the response variable Y_i takes on any value between zero and n_i and is distributed as $binomial(n_i, \pi_i)$. In our implementation, we have divided the number of sites with blue oaks by the total number of sites in the region, so that $n_i = 1$ for all i, and Y_i is allowed to take on fractional values. This avoids the problem that different clusters have different numbers of sites (i.e., it makes Y an *intensive* variable in the sense of Section 7.2).

One of the key differences between the binomial and quasibinomial models is that the latter frequently will have a variance different from the variance of a binomial distribution. This is an example of the *overdispersion* phenomenon discussed in Section 8.4.4. The function `glm()` can be used to fit quasibinomial models by specifying the family argument as `family = quasibinomial`. This introduces a variable dispersion parameter.

Rather than using a likelihood function, the function `glm()` when fitting a quasibinomial model uses a *quasi-likelihood* function (McCulloch et al., 2008, p. 152). A major problem with this approach is that the AIC cannot be computed for quasi-likelihood functions. Burnham and Anderson (1998, p. 52) suggest a quasi-AIC (QAIC) statistic, but this is not implemented in `glm()`, and we will instead break the rules a little and use hypothesis testing to compare models.

First, we compute the coefficients of the candidate model for both the pointwise data and the aggregated data.

```
> model.glmC6clus <- glm(QUDO ~ TempR + Permeab + Precip +
+     GS32 + PE + SolRad6, data = data.Set2Cclus,
+     family = quasibinomial)
```

```
> print(coef(model.glmC6clus), digits = 2)
(Intercept)      TempR     Permeab      Precip         GS32
   -8.9e+00    1.6e-01    -1.6e+00    4.8e-05    -1.6e-02
         PE    SolRad6
    8.9e-03    2.6e-01
> model.glmC6pts <- glm(QUDO ~ TempR + Permeab + Precip +
+     GS32 + PE + SolRad6, data = data.Set2Cpts,
+     family = binomial)
> print(coef(model.glmC6pts), digits = 2)
(Intercept)      TempR     Permeab      Precip         GS32
    -7.7864     0.2966     -0.7391    -0.0015     -0.0213
         PE    SolRad6
     0.0093     0.1783
```

The coefficient of *Precip* changed sign. Perhaps more importantly from a model selection perspective (since we have agreed to use hypothesis testing for model selection purposes) is the effect on significance. Unlike the pointwise case, there is no significant difference (at the $\alpha = 0.05$ level) between aggregated data models with three and six explanatory variables.

```
> model.glmC3clus <- update(model.glmC6clus,
+     formula = as.formula("QUDO ~ Permeab +
+     GS32 + PE"))
> model.glmC3pts <- update(model.glmC6pts,
+     formula = as.formula("QUDO ~ Permeab +
+     GS32 + PE"))
> anova(model.glmC3clus, model.glmC6clus, test = "Chisq")
Analysis of Deviance Table
Model 1: QUDO ~ Permeab + GS32 + PE
Model 2: QUDO ~ TempR + Permeab + Precip + GS32 + PE + SolRad6
  Resid. Df Resid. Dev Df Deviance P(>|Chi|)
1        46     6.8125
2        43     5.9361  3  0.87647   0.06527 .
> anova(model.glmC3pts, model.glmC6pts, test = "Chisq")
Model 1: QUDO ~ Permeab + GS32 + PE
Model 2: QUDO ~ TempR + Permeab + Precip + GS32 + PE + SolRad6
  Resid. Df Resid. Dev Df Deviance P(>|Chi|)
1      1852     1660.5
2      1849     1554.7  3   105.76 < 2.2e-16 ***
```

Further reduction in the number of variables does produce a significant difference (the code is not shown). Thus, it appears that we can make do with an aggregated data model containing only *Permeab*, *GS32*, and *PE*. Alternative models are certainly possible as well.

The next question is how the models compare in their predictive ability. The large dark circles in Figure 12.8a represent a comparison between the predictions of the cluster model, with three explanatory variables, and the means over the clusters of the pointwise model, with six explanatory variables. The difference is not large. The smaller circles show the values of each predicted probability of the pointwise model, indicating the considerable variability within each cluster in the predicted value of blue oak presence probability. Figure 12.8b shows observed versus predicted fractions of the sites in each cluster with

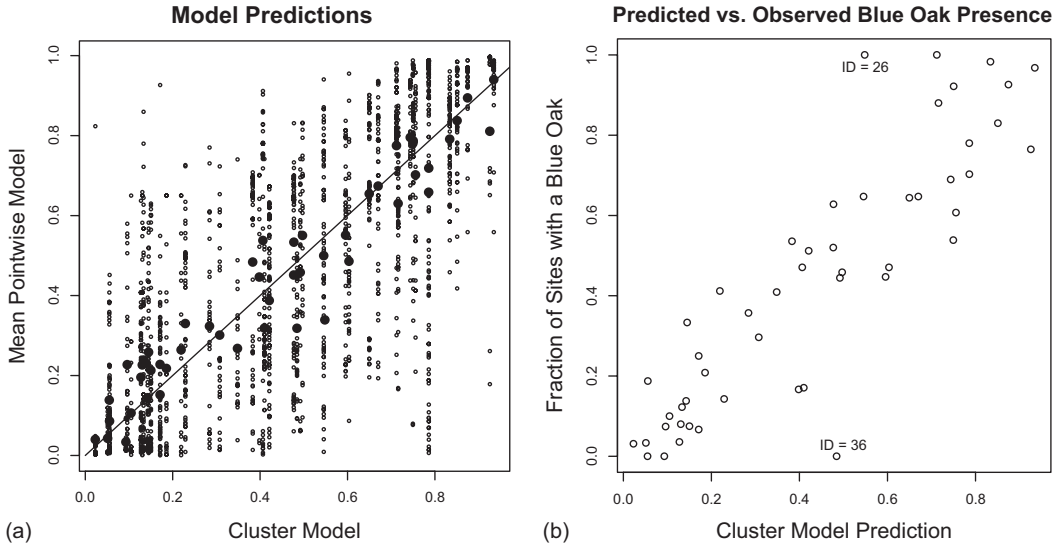

FIGURE 12.8
(a) Plot of the predicted values of the pointwise data generalized linear model of *QUDO* versus *Elevation* for the Coast Range subset of Data Set 2 versus the clustered data model. The large filled circles are based on the mean over each cluster region of the pointwise data model, and the small circles are based on the pointwise data; (b) Observed versus predicted values of the portion of sites with a blue oak for the clustered data model. Two discordant values are identified by cluster ID number.

a blue oak. By and large the fit is good. There is a pair of seeming outliers, whose cluster numbers are identified. These may be of interest if some corresponding biophysical anomaly can be identified.

In summary, the explanatory variables selected by the pointwise and aggregated data models are quite different, although their predictions are, as far as comparisons are possible, similar. The questions addressed by the two models are, however, completely different. The pointwise model predicts the probability that an individual site will contain a blue oak, and the aggregated model predicts the fraction of sites in each cluster that will contain a blue oak. Each model provides information about the ecology of blue oaks, but they cannot simply be interchanged with each other. We now move on to the incorporation of spatial autocorrelation in the cluster model.

12.6.2 The Incorporation of Spatial Autocorrelation

For the quasibinomial model used in Section 12.6.1, if Y_i represents the portion of sites in the region having a blue oak, and π_i is the probability that a site contains a blue oak, so that $E\{Y_i\} = \mu_i = \pi_i$, and if there are $p-1$ explanatory variables X_i, then by combining Equations 8.19 and 8.28, we can express the model as

$$g(E\{Y \mid X_i\}) = g(\pi_i) = \ln\left(\frac{\pi_i}{1-\pi_i}\right) = \beta_0 + \beta_1 X_{1i} + \ldots + \beta_{p-1} X_{p-1,i}, \tag{12.13}$$

where $\pi_i = \Pr\{Y_i = 1\}$ and $g(\pi_i)$ is the logit link function. The response variable Y_i is binomially distributed and normalized, so that if the measurements X_i are independent, then its variance is

$$Var\{Y_i\} = \pi_i(1-\pi_i). \tag{12.14}$$

In other words, unlike the case with the normal distribution, the variance of the binomial distribution depends on the mean, and hence indirectly on the explanatory variables. When Y_i approaches its limits of zero or one, the variance declines to zero. When the errors are spatially autocorrelated we expect that $Var\{Y_i\}$ will be larger than predicted by Equation 12.14. This equation is sometimes modified to

$$Var\{Y_i\} = a\pi_i(1-\pi_i), \tag{12.15}$$

where a is called the *scale* or *overdispersion* parameter.

In Section 12.5, we discussed the incorporation of spatial autocorrelation into linear regression equation via generalized least squares. In this context, in Equation 12.11 above, the equation $\Lambda_i = V_i R_i V_i$ describes the modification of the variance equation for the linear regression model to consider spatial autocorrelation. The analogous equation for the variance of the GLM, taking into account spatial autocorrelation, is

$$Var\{Y\} = \Lambda = aV(\mu)RV(\mu), \tag{12.16}$$

where a is again a variance scaling term, R is, as in Equation 12.11, a correlation matrix that depends on the variogram model, μ is the vector of means, and V is a diagonal matrix whose elements are $v_{ii} = \sqrt{\mu_i(1-\mu_i)}$. In the mixed model of Equation 12.10 there are two indices: the index i refers to the *group*, in which measurements are indexed by j, $j = 1,...,n_i$. In the quasibinomial model of Equation 12.13, there is only one index: i indexes the measurements of the regions, which are not grouped. In this sense, the analysis of this section is roughly analogous to that of the generalized least square model discussed in Section 12.5.

The objects `data.Set2Cpts` and `data.Set2Cclus` are data frames. We will be creating variograms, so we will also need a `SpatialPointsDataFrame` that contains the same data. Because we are working with distances, the data will be projected in UTM Zone 10 (all of the sites are in this zone), so we will need the Easting and Northing. We therefore use `spTransform()` to transform the data.

```
> coordinates(data.Set2Cpts) <- c("Longitude", "Latitude")
> proj4string(data.Set2Cpts) <- CRS("+proj=longlat +datum=WGS84")
> data.Set2Cutm <- spTransform(data.Set2Cpts,
+    dCRS("+proj=utm +zone=10 +ellps=WGS84"))
> data.Set2Cpts$Easting <- coordinates(data.Set2Cutm)[,1]
> data.Set2Cpts$Northing <- coordinates(data.Set2Cutm)[,2]
```

The mean values of the Easting and Northing of the points in each cluster region will be used to identify the coordinates of that region in `data.Set2Cclus`.

```
> data.Set2C.mat <- matrix(nrow = length(unique(data.Set2Cpts$clusID)),
+    ncol = ncol(data.Set2Cpts))
> for (i in 1:ncol(data.Set2Cpts)){
+    data.Set2C.mat[,i] <- tapply(data.Set2Cpts@data[,i],
+       data.Set2Cpts$clusID, mean)
+ }
> data.Set2Cclus <- data.frame(data.Set2C.mat)
> names(data.Set2Cclus) <- names(data.Set2Cpts)
```

The analysis follows closely the example presented by Gotway and Stroup (1997) as Example 1 (see also Waller and Gotway, 2004, p. 385). Gotway and Stroup (1997) refer to this as a *marginal* formulation of the problem, because the quantity $E\{Y \mid X_i\}$ in Equation 12.13 is a marginal mean. Gotway and Stroup (1997) also discuss an alternative method called the *conditional* formulation. Shabenberger and Pierce (2002, p. 684) give an example of this approach, and Gotway and Wolfinger (2003) discuss a different type of marginal formulation analysis.

The marginal analysis procedure of Gotway and Stroup (1997) is called *iteratively reweighted generalized least squares* (IRWGLS) (see also Waller and Gotway, 2004, p. 337, 388) and is a generalization of the iteratively reweighted least squares algorithm mentioned in Section 8.4.1. It consists of four steps that are repeated iteratively in order to obtain estimates of the regression coefficients β_i in Equation 12.13.

Step 1. The algorithm is initiated by obtaining starting estimates $b_{i,0}$ for the regression coefficients. This is most easily done by simply solving the generalized linear model with independent errors using the function `glm()`. In our example, they can be carried over from the solution obtained in Section 12.6.1, which is repeated here.

```
> oaks.irwgls <- oaks.glm <- glm(QUDO ~ Permeab + GS32 + PE,
+    data = data.Set2Cclus, family = quasibinomial)
> print(b.irwgls <- coef(oaks.irwgls), digits = 3)
(Intercept)    Permeab        GS32          PE
     0.8958    -1.7213     -0.0265      0.0123
```

Now we must compute the predicted values of π_i using Equation 8.20. To do this, we use the function `predict()` to obtain the predictions p_i. Note the second argument (use `?predict.glm` for an explanation).

```
> data.Set2Cclus$p.irwgls <- p.irwgls <- predict(oaks.irwgls,
+ type = "response")
```

Step 2. The second step is to compute a variogram of the residuals of the model in order to estimate their spatial autocorrelation. The `gstat` package (Pebesma, 2004) function `variogram()` can be made to compute the generalized least square residuals of a generalized linear model by specifying the link function in the variogram formula (Pebesma, 2004).

```
> data.Set2Cgeo <- data.Set2Cclus
> coordinates(data.Set2Cgeo) <- c("Easting", "Northing")
> oaks.vgm <- variogram(log(p.irwgls / (1 - p.irwgls)) ~
+    Easting + Northing, data = data.Set2Cgeo)
```

Step 3. Based on the experimental variogram of the residuals, estimate the parameters of a variogram model. If the residuals do not display autocorrelation (i.e., if the variogram is a pure nugget), then the process can stop. The experimental variogram of the residuals is shown in Figure 12.9. Because of the discordant values of $\hat{\gamma}(h)$ at the largest values of the lag h, models were fit by eye. Based on the figure, the exponential model shown in the figure was selected.

```
> b.exp <- 1.0
> alpha.exp <- 20000
```

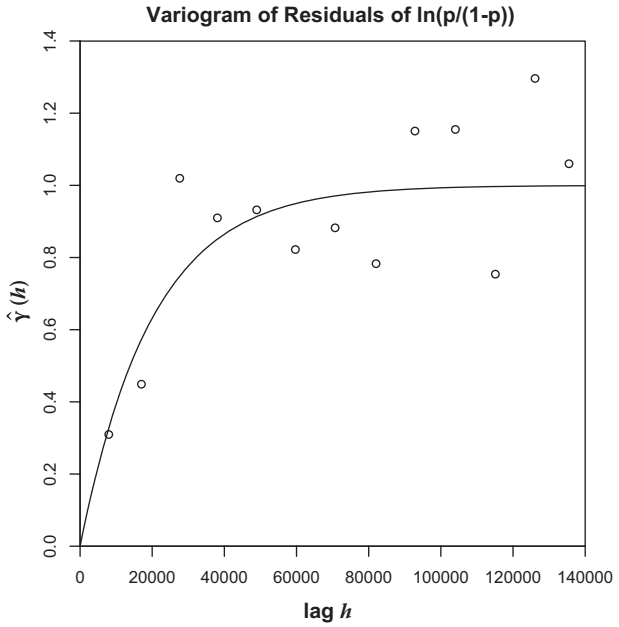

FIGURE 12.9
Variogram of the residuals of the generalized linear model of probability of a blue oak as a function of elevation. Fit by an exponential variogram model is also shown.

Step 4. This step involves the implementation of the *generalized estimating equations* to obtain an updated estimate of the regression coefficients. In order to facilitate cross-referencing with their paper, we will employ the same notation as Gotway and Stroup (1997), with one exception: where they use the symbol Z to represent their response variable, we will retain our practice of using Y. To further ease cross-referencing, we will implement the R code following their equations exactly, although it results in extremely inefficient code.

Similar to Equation 12.16, the variance of Y is written (Equation 2.13 of Gotway and Stroup, 1997)

$$Var\{Y\} = v^{1/2}(\mu)R(\alpha)v^{1/2}(\mu), \tag{12.17}$$

where $R(\alpha)$ is the correlation matrix, which depends on a parameter or parameters α, and $v^{1/2}(\mu)$ is the diagonal matrix whose diagonal elements are the square roots of products of the expected values of the Y_i with the scale factors a_i. Since Y is a binomially distributed random variable, $\mu_i = E\{Y_i\} = \pi_i$, $\text{var}\{Y_i\} = \pi_i(1-\pi_i)$, and $a_i = 1/n_i$ (Gotway and Stroup, 1997). Therefore, the matrix $v^{1/2}(\mu)$ is written

$$v^{1/2}(\mu) = \begin{bmatrix} \sqrt{\pi_1(1-\pi_1)/m_1} & 0 & \ldots & 0 \\ 0 & \sqrt{\pi_2(1-\pi_2)/m_2} & \ldots & 0 \\ \ldots & \ldots & \ldots & \ldots \\ 0 & 0 & \ldots & \sqrt{\pi_n(1-\pi_n)/m_n} \end{bmatrix} \tag{12.18}$$

where m_i is the number of data sites in polygon i. This is implemented as

```
> n.sites <- as.numeric(table(data.Set2C$clusID))
> v.half <- diag(as.vector(p.irwgls * (1 - p.irwgls) / n.sites))
```

In the first line above, the function `table()` computes a contingency table consisting of counts of the number of occurrences of each value of `data.Set2C$clusID`.

Because we are using an exponential variogram model, the correlation matrix $R(\alpha)$ in Equation 12.17 is given by

$$R(\alpha) = \begin{bmatrix} 1 & \exp(-\alpha h_{12}) & \ldots & \exp(-\alpha h_{1n}) \\ \exp(-\alpha h_{12}) & 1 & \ldots & \exp(-\alpha h_{2n}) \\ \ldots & \ldots & \ldots & \ldots \\ \exp(-\alpha h_{1n}) & 0 & \ldots & 1 \end{bmatrix} \tag{12.19}$$

where based on the results of Step 3, the parameter α of the variogram model is set at $\alpha = 20,000$ and h_{ij} is the distance from polygon i to polygon j. Here is the implementation

```
> x <- data.Set2Cclus$Easting
> y <- data.Set2Cclus$Northing
> lag.mat <- as.matrix(dist(cbind(x, y)))
> R <- exp(-lag.mat / alpha.exp)
```

The estimated variance-covariance matrix V is then computed from Equation 12.17.

```
> V <- sqrt(v.half) %*% R %*% sqrt(v.half)
```

Now we are ready to compute the updated estimates for the regression coefficients. In Appendix A, it is shown that if we write the linear regression equation in matrix form as $Y = X\beta + \varepsilon$ (Equation A.33), then normal equations defining the regression coefficients may be written as $X'Xb = X'Y$. This allows the computation of the regression coefficients via the formula $b = (X'X)^{-1}X'Y$ (Equation A.35). Regression software such as R does not actually compute this inverse directly, because the computation can be done more efficiently, but in principle the regression coefficients could be computed this way.

Gotway and Stroup (1997), citing Nelder and Wedderburn (1972), Wedderburn (1974), McCullagh (1983), Liang and Zeger (1986), and McCullagh and Nelder (1989), develop the iterative approximation to the regression coefficients via IRWGLS as follows: Write the inverse link function h of equation (8.20) as $\pi = \mu = h(\eta),\ \eta = X\beta$. Then let D be the diagonal matrix whose diagonal elements are $\partial\pi_i / \eta_i$. For the binomial case, $\pi_i = h(\eta_i) = 1/(1+\exp(-\eta_i))$ and therefore $\partial\pi_i / \partial\eta_i = \pi_i(1-\pi_i)$. This is implemented as follows:

```
> D <- diag(as.vector(p.irwgls * (1 - p.irwgls)))
```

We then define the matrix W by $W = D'V^{-1}D$. This is implemented (very inefficiently) as follows:

```
> V.inv <- solve(V)
> W <- t(D) %*% V.inv %*% D
```

Now, let z^* be defined by $z^* = \eta + D^{-1}(z-\mu)$. We can estimate the μ_i using the values $h(\Sigma b_{0,j}X_{ij})$, where the $b_{0,i}$ are obtained in Step 1 and the X_{ij} are the mean values of the

explanatory variables of cluster region i. We can estimate η as $g(p_{i,0}) = \ln(p_{i,0} / (1 - p_{i,0}))$, and substitute for the z_i the values p_i representing for each polygon the fraction of sites occupied by a blue oak, and z^*.

```
> eta <- log(p.irwgls / (1 - p.irwgls))
> mu <- as.vector(p.irwgls)
> z <- p
> D.inv <- solve(D)
> z.star <- eta + D.inv %*% (z - mu)
```

The last step is to use an estimating equation given as Equation 3.1 by Gotway and Stroup (1997). This corresponds to Equation A.34 for linear regression and is written as

$$X'WX\beta = X'Wz^*. \tag{12.20}$$

Premultiplying by $(X'WX)^{-1}$ allows us to compute the values $b_{i,1}$ as

$$b_{i,1} = (X'WX)^{-1}X'Wz^*. \tag{12.21}$$

Here is the implementation.

```
> X <- with(data.Set2Cclus, cbind(rep(1,length(p.irwgls)),
+   Permeab, GS32, PE))
> XpWX.inv <- solve(t(X) %*% W %*% X)
> b.old <- b.irwgls
> t(b.irwgls)
                      Permeab           GS32           PE
[1,]    0.4304769    -1.522903    -0.02922488   0.01358672
```

Now we can compare the updated value $b_{j,1}$ of the regression coefficients with the original values.

```
> t(b.old)
      (Intercept)      Permeab           GS32           PE
[1,]    0.8957865    -1.721327     -0.0264501   0.01225389
> t(abs((b.irwgls - b.old) / b.old))
                      Permeab           GS32           PE
[1,]    0.5194426    0.1152737      0.1049063    0.1087677
```

Step 5. Since there is a fairly large change in the coefficients, we use these values to generate a second iteration by plugging $b_{j,1}$ into Step 2, working again through Steps 2, 3, 4, and 5, and continuing this iteration until reaching a sufficiently small change in the estimates for the β_i.

```
> data.Set2Cclus$p.irwgls <- p.irwgls <-
+    1 / (1 + exp(-X %*% b.irwgls))
```

In this case, the second iteration produces the following result:

```
> t(b.irwgls)
                      Permeab           GS32           PE
[1,]    0.4333026    -1.519584    -0.02928080   0.01360120
> t(b.old)
                      Permeab           GS32           PE
[1,]    0.4304769    -1.522903    -0.02922488   0.01358672
> t(abs((b.irwgls - b.old) / b.old))
                      Permeab           GS32           PE
[1,]  0.006564155  0.002179365    0.001913198   0.00106632
```

A third iteration produces this:

```
> t(b.irwgls)
                      Permeab           GS32           PE
[1,]    0.4345514    -1.519843    -0.02928375   0.01360108
> t(b.old)
                      Permeab           GS32           PE
[1,]    0.4333026    -1.519584    -0.02928080   0.01360120
> t(abs((b.irwgls - b.old) / b.old))
                      Permeab           GS32           PE
[1,]  0.002882207 0.0001701886   0.0001008558 8.808101e-06
```

We will accept this answer as the estimates for β_j. Here is the comparison with the solution obtained in Section 11.6 by ignoring spatial autocorrelation.

```
> print(t(b.glm <- coef(oaks.glm)), digits = 3)
     (Intercept) Permeab    GS32     PE
[1,]       0.896   -1.72 -0.0265 0.0123
> print(t(b.irwgls), digits = 3)
           Permeab    GS32     PE
[1,] 0.435   -1.52 -0.0293 0.0136
> print(t((b.irwgls - b.glm) / b.glm), digits = 3)
            Permeab  GS32   PE
[1,] -0.515  -0.117 0.107 0.11
```

Since this has been a long analysis, it is worthwhile to look back and recall our objective. This was to determine the effect of the incorporation of spatial autocorrelation on the coefficients of a model for blue oak distribution. Rather than using pointwise data, in which each point takes on a value of 0 or 1, we used aggregated data in which for each region the value represents the fraction of sites in a that region in which an oak is present. Our conclusion is that there is about a 10% difference in each of the coefficients between the models incorporating and not incorporating spatial autocorrelation.

Shabenberger and Pierce (2002, p. 686) point out an obvious problem with the IRWGLS method. Even though the method could be made much more efficient numerically by not following exactly the equations of Gotway and Stroup (1997) as we have done here, nevertheless, it remains necessary in Equation 12.21 to invert an $n \times n$ matrix, where n may

be quite large. Shabenberger and Pierce (p. 691) compare several methods, including IRWGLS, other marginal methods, and a conditional method, and find that in many cases they provide similar answers.

Finally, we note in passing that the term *logistic regression* has another different but (possibly confusingly) similar application. This is in the fitting of a logistic function to a set of data that are not probabilities but whose values follow a logistic shaped (also known as sigmoid) curve. An example is a plant growth curve (Hunt, 1982), in which an annual plant's growth rate, measured, for example, in biomass or leaf area index, starts out slowly, increases through a "grand growth phase," and eventually declines to zero as the plant matures. Such data may be fit with a logistic equation, but the asymptotic values are typically not 0 and 1, and the assumptions associated with the error terms are different from those of the probabilistic model discussed in this section. Pinhiero and Bates (2000) describe software for the fitting of this form of data using nonlinear regression (see Exercise 12.5).

12.7 Further Reading

The precise definitions of the terms *fixed effect* and *random effect* vary considerably in the literature, despite efforts to constrain them (Littell et al., 2002, p. 92). Gelman and Hill (2007, p. 245) identify five different uses of the terms, and someone who was really interested in such things could no doubt find some more.

Pinhiero and Bates (2000) is the single most important book for all the material in this chapter except Section 12.6. For the numerical method used by `lme()` to solve a mixed model such as Equation 12.7, see Pinhiero and Bates (2000, p. 202). Other books on the mixed model that provide a useful supplement are Searle (1971), Searle et al. (1992), and McCulloch et al. (2008). In particular, Searle (1971, p. 458) provides a good introduction to the application of the maximum likelihood method to mixed models. McCulloch et al. (2008) provide a good discussion of the differences between ML and REML. Diggle et al. (2002) discuss the use of the mixed model in the analysis of data models with autocorrelated errors. The specific case of unbalanced data such as that encountered in Data Set 3 is well covered by Searle (1987). Rutherford (2001) gives an excellent introduction to the analysis of covariance using least squares.

Zuur et al. (2009) and Bolker et al. (2009) provide very well organized discussions of the application of mixed models to ecology. Waller and Gotway (1994) and Schabenberger and Pierce (2002) provide good discussions of the generalized linear model with autocorrelated errors. Carl and Kuhn (2007) discuss the use of generalized estimating equations in species distribution models.

The process of variogram estimation for the generalized linear model as used in `gstat` is described by Christensen (1991) and ver Hoef and Cressie (1993). In addition to Shabenberger and Pierce (2002), McCulloch et al. (2008) discuss the conditional versus marginal formulation. Venables and Ripley (2002) also discuss the generalized linear mixed model and in particular the function `glmmPQL()`, which addresses overdispersion in binomial and Poisson models. Hadfield (2010) provides in the `MCMCglmm` package a very nice implementation of the Markov Chain Monte Carlo (MCMC) method, discussed in Chapter 14, for the solution of generalized linear mixed models.

Exercises

12.1 Create a scatterplot of the regression coefficients of the intercepts computed in Section 12.4 using the mixed model and the mean yields of each of the fields.

12.2 Repeat the analysis using `gls()` of Section 12.5 for a model describing the data in Field 4.2.

12.3 Load the `debug` package (Bravington, 2013). Use the function `debug()` to monitor the progression of code to predict values of π_i for the model `glm.demo` of Section 9.4.1.

12.4 Repeat the IRWGLS analysis of Section 12.6.2 for a model of the data in the Sierra Nevada subset of Data Set 2.

12.5 Read about the function `nls()` in Pinhiero and Bates (2000). Use this function to fit the Coast Range clusters data in Section 12.6.2. Compare this fit with the one obtained using the quasibinomial model. What happens to the `nls()` model as the explanatory variable approaches ∞ and $-\infty$?