8

Data Exploration Using Non-Spatial Methods: The Linear Model

8.1 Introduction

The previous chapter focused on graphical means of data exploration. In the analysis of non-spatial data sets, the methods discussed in this chapter are not necessarily considered exploratory. We classify these methods as exploratory here because in those cases in which a parameter may be estimated or a hypothesis may be tested, spatial data may violate the assumptions underlying these operations, and therefore the estimation or the significance test or both may not be valid. We often do not use the methods to obtain final confirmatory results (parameter estimates and/or significance tests), but rather to gain further insights into the relationships among the data. Confirmatory analysis will be carried out in later chapters in which the analysis incorporates assumptions about the spatial relationships among the data.

All is not bad news, however, when it comes to the impact of spatial autocorrelation on data analysis. Although we must be aware of possible bias in parameter estimates and we may lose the ability to carry out hypothesis tests in the same way that we would with non-spatial data, we have already seen in Chapter 7 that this loss is often more than compensated by our acquisition of the ability to glean further understanding of data relationships through the use of maps and other devices. We will see further examples of this throughout the present and subsequent chapters. In this chapter, Section 8.2 provides an introduction to multiple linear regression and develops the approach to model selection that we will use. Section 8.3 applies this approach to the construction of alternative multiple linear regression models for yield in Field 1 of Data Set 4. Section 8.4 introduces generalized linear models and shows how these are applied to data from Data Sets 1 and 2.

8.2 Multiple Linear Regression

8.2.1 The Many Perils of Model Selection

In a seminar I attended many years ago, a prominent ecologist characterized multiple regression as "the last refuge of scoundrels." He was not actually referring to multiple regression per se, but rather to model selection, the process of using multiple regression to select the "best" variables to explain a particular observed phenomenon. In this context, he had a point. Model selection can be a very useful tool for exploratory purposes, to eliminate some alternatives from a wide variety of competing possible explanations, but

it must be used with extreme caution and not abused. In this subsection, we will briefly review some of the traditional automated methods of model selection. The primary purpose of this review is to indicate where they may have problems and to extract any ideas from them that may seem suitable. After that, we discuss two graphical techniques that are often useful in constructing a good multiple regression model: added variable plots and partial residual plots. The third and final subsection reviews the pitfalls associated with model selection and describes the approach that will be used in this book. Readers unfamiliar with the theory of simple linear regression and ordinary least squares (OLS) should consult Appendix A.2.

There are a number of problems in the interpretation of multiple regression results that do not arise with simple linear regression. Two in particular are very important. The first is that when there are more than one explanatory variable it becomes difficult or impossible to visualize the interrelationships among the variables, and to identify influential data records. Second, the explanatory variables themselves may be correlated, a phenomenon called *multicollinearity*. Because of this, it may become difficult or impossible to distinguish between the direct impact a process has on the response variable and the indirect effect it has due to its effect on another explanatory variable. This is why controlled experiments are much more powerful than observational studies. A second negative impact of multicollinearity is its impact on the matrix equation (A.34) in Appendix A defining the computation of the regression coefficients. This computation may become subject to considerable error. The bad effects of multicollinearity, as well as the difficulty of identifying influential data records, are part of what makes the process of selecting explanatory variables to include in a multiple regression model so difficult and dangerous.

In comparing multiple regression methods, one must always keep in mind the objective of the analysis. Generally, the two possibilities are prediction and interpretation. The former objective is much easier to quantify than the latter, and as a result one can, based on the establishment of a precise measure, compare methods and determine which method is optimal according to that measure. If the objective is interpretation, as it generally is in this book, it is not even clear what would define an "optimal" method. The unfortunate fact is that one is using a tool in a way for which it itself is not optimal, and the best one can do is to be careful and try to avoid making serious errors. With these caveats, we can begin discussing the methods.

We will express the multiple linear regression model as (cf. Appendix A.2, Equation A.37)

$$Y_i = \beta_0 + \beta_1 X_{i1} + \beta_2 X_{i2} + \ldots + \beta_{p-1} X_{i,p-1} + \varepsilon_i,\ i = 1, \ldots, n, \tag{8.1}$$

where the error terms ε_i are assumed to be normally distributed, of constant variance σ^2, and mutually independent. This last condition, independent errors, is often violated when the data represent measurements at nearby spatial locations. In Chapters 11 through 13 we will discuss the consequences of this failure of the assumptions and the means to deal with them. For the moment, we will simply say that one must consider the results of ordinary least squares linear regression applied to spatial data as strictly preliminary.

The process of selecting which explanatory variables to retain and which to eliminate from a multiple linear regression model can be viewed as a search for balance in the *bias-variance trade-off* (Hastie et al., 2009, p. 37). This concept can be illustrated using an example with artificial data. Our discussion follows that of Kutner et al. (2005, p. 357). Using the notation of Appendix A, let $\hat{Y}_i$ be the value of the response variable predicted by the regression at data location i, that is,

$$\hat{Y}_i = b_0 + b_1 X_{i1} + b_2 X_{i2} + \ldots + b_{p-1} X_{i,p-1},\ i = 1, \ldots, n. \tag{8.2}$$

Suppose that a linear model with three explanatory variables fits the data exactly, that is, that if we define $\mu_i = E\{Y_i\}$, then

$$\mu_i = \beta_0 + \beta_1 X_{i1} + \beta_2 X_{i2} + \beta_3 X_{i3},\ i = 1, \ldots, n. \tag{8.3}$$

Suppose, however, that we have measured five explanatory variables, two of which have no association with the response variable, so that $\beta_4 = \beta_5 = 0$. Suppose there are ten measurements of each variable. With five explanatory variables, this means there are only two measurements per variable. This low number is chosen on purpose, as we will see later. We first generate an artificial data set that implements this setup.

```
> set.seed(123)
> X <- matrix(rnorm(50), ncol = 5, byrow = TRUE)
> beta.lm <- matrix(c(1,1,1,0,0), ncol = 1)
> mu <- X %*% beta.lm
```

The matrix `X`, with five columns of ten rows each, contains the data values X_{ij}. The X_i are normally distributed random variables that are uncorrelated with each other. The matrix `beta.lm`, with one column, contains the β_j in Equation 8.3, and the last line of code implements the equation $\mu = X\beta$. By the way, "beta" is the name of a function in R, which is why this name is not used.

The total error of the fitted value $\hat{Y}_i$ of the regression is given by $\hat{Y}_i - \mu_i$. Adding and subtracting the expected value $E\{\hat{Y}_i\}$, this error may be written $(\hat{Y}_i - E\{\hat{Y}_i\}) + (E\{\hat{Y}_i\} - \mu_i)$. This divides the error into two components. The *bias component* is given by

$$bias_i = E\{\hat{Y}_i\} - \mu_i, \tag{8.4}$$

and represents the error caused by the model not fitting the data correctly. The *random error component* is given by

$$random\ error_i = \hat{Y}_i - E\{\hat{Y}_i\}, \tag{8.5}$$

which represents the deviation due to random error in the data. The mean square error of the fitted values is given by

$$\begin{aligned} MSE &= \sum_{i=1}^{n} E\{(\hat{Y}_i - \mu_i)\}^2 \\ &= \sum_{i=1}^{n} E\{(E\{\hat{Y}_i\} - \mu_i) + (\hat{Y}_i - E\{\hat{Y}_i\})^2 \\ &= \sum_{i=1}^{n} E\{(bias_i + random\ error_i)^2\} \end{aligned} \tag{8.6}$$

It turns out (Kutner et al., 2005, p. 357) that this equation can be simplified to

$$\begin{aligned} MSE &= \sum_{i=1}^{n} (E\{\hat{Y}_i\} - \mu_i)^2 + \sum_{i=1}^{n} (\hat{Y}_i - E\{\hat{Y}_i\})^2 \\ &= \sum_{i=1}^{n} bias_i^2 + \sum_{i=1}^{n} \text{var}\{\hat{Y}_i\} \end{aligned} \tag{8.7}$$

Thus, the total mean square error is the sum of the total squared bias plus the sample variance of the $\hat{Y}_i$. As the complexity of the regression model, measured by the number $p-1$ of explanatory variables, increases, the bias tends to decrease, because with more explanatory variables in the model the fit will be more precise and therefore the quantity $\Sigma(E\{\hat{Y}_i\} - \mu_i)^2$ will decrease. Conversely, as the complexity increases, the variance $\Sigma(\hat{Y}_i - E\{\hat{Y}_i\})^2$ tends to increase. This decreasing bias and increasing variance as explanatory variables are added to the model represents the bias-variance trade-off.

We can view the bias-variance trade-off from another perspective by using the example begun above. We will fit models with varying numbers of explanatory variables to the data set $\{Y_i\}$ generated by $Y_i = \mu_i + \varepsilon_i$, where μ_i satisfies Equation 8.3 and ε_i is a normally distributed random variable uncorrelated with any of the X_i. We then estimate the squared bias and variance of the fit using Monte Carlo simulation.

First, we need a function `regress()` that computes the regression model.

```
> regress <- function(X, mu, pminus1){
+    Y <- mu + 0.25 * rnorm(10)
+    Yhat <- predict(lm(Y ~ X[,1:pminus1]))
+ }
```

The computation of `Yhat` in this function takes advantage of the fact that the function `lm()` interprets a matrix with $p-1$ columns on the right-hand side of the formula statement as specifying that the formula consists of the sum of the $p-1$ explanatory variables making up those columns. Now we are ready to run the Monte Carlo simulation.

```
> b2 <- numeric(5)
> v <- numeric(5)
> for (pminus1 in 1:5){
+    set.seed(123)
+    Yhat <- replicate(1000, regress(X, mu, pminus1))
+    EYhat <- rowMeans(Yhat)
+    b2[pminus1] <- sum((EYhat - mu)^2) / 1000
+    v[pminus1] <- sum((Yhat - EYhat)^2) / 1000
+ }
```

The array `Yhat` generated in the fifth line consists of ten rows and one thousand columns. Each column is the set of predicted values $\hat{Y}_i$, $i = 1,...,10$ of one regression. The values $E\{\hat{Y}_i\}$ are the row means of `Yhat`. The average value of the squared bias and the variance are therefore computed by dividing the sums in Equation 8.7 by the number of replications. The code that created `X` and `mu` is listed at the start of the section.

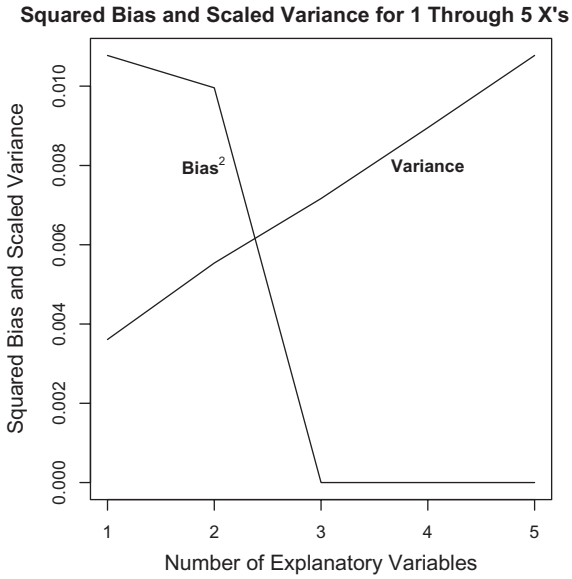

FIGURE 8.1
Plots of squared bias and scaled variance for a regression model with one through five explanatory variables based on artificial data.

Figure 8.1 shows a plot of the squared bias `b2` and the variance `v`, with `v` scaled to have the same maximum value as `b2`. The figure shows that as the number of explanatory variables increases, the bias decreases and the variance increases. In this particular example, the model with three or more variables fits the data, so there is no bias.

Pursuing this example a bit further, let's examine a plot of the fits $\hat{Y}_i$ against the observed values Y_i for the models with one, three, and five explanatory variables. (True confession: this example is specifically rigged, in a way shown below, to emphasize the effect and make a point). Once again, we create an artificial data set satisfying Equation 8.3, that is, with three explanatory variables that determine Y and two that are measured but have no effect on Y. First, we fit a regression line to ten data sets, using models with one, three, and five explanatory variables. Figure 8.2a shows the fitted values $\hat{Y}_i$ versus Y_i for the three models. The models with $p-1=3$ and $p-1=5$ both provide a better fit than the model with $p-1=1$; the simplest model *underfits* the data. Next, we generate a new data set and try to fit it with the same three models. Here is where the rigging takes place: to emphasize the effect, the values of X_{i5} are multiplied by ten. Figure 8.2b shows the results. The model with $p-1=5$ does not provide as good a fit as the model with $p-1=3$. The most complex model *overfits* the data. That is, when variables that should not be in the model are included, the effect of these variables is unique to the data set that generates the model and is inappropriate for other data sets. Thus, high bias is associated with underfitting and high variance with overfitting.

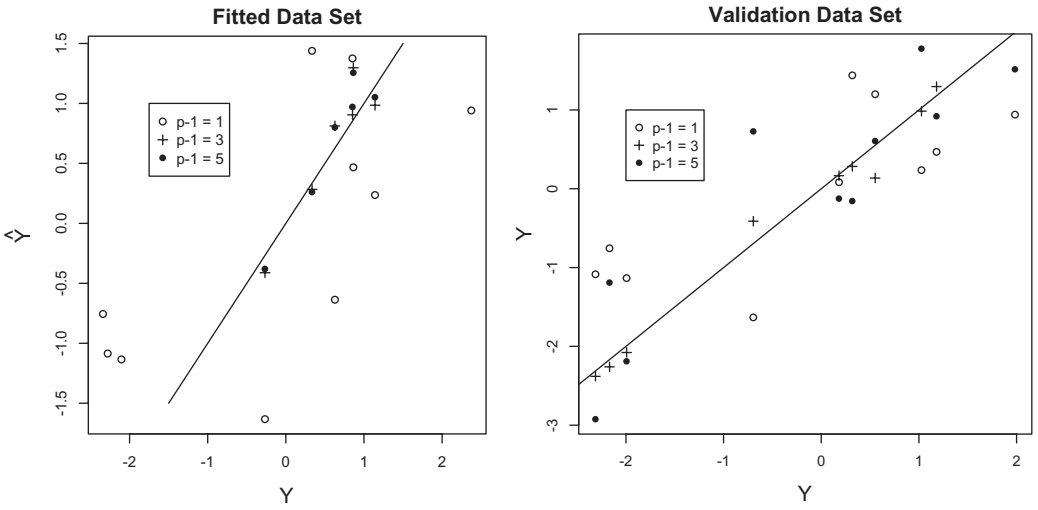

FIGURE 8.2
Regression plots of three linear models constructed with artificial data. The model with $p = 1$ underfits the data; the model with $p = 3$ fits the data; and the model with $p = 5$ overfits the data. (a) Data used to fit the models. (b) Data used to validate the models.

Three statistics are widely used as a means of measuring the trade-off between bias and variance. These are Mallows' C_p (Mallows, 1973), the Akaike information criterion (AIC) (Akaike, 1974), and the Bayesian Information Criterion (BIC) (Schwarz, 1978). These shown in Table 8.1. Each has the form measure = bias penalty + complexity penalty. The value of the *bias penalty* is an error sum of squares *SSE* (equation A.27, Appendix A.2), and the *complexity penalty* is an increasing function of the number of explanatory variables. Mallows' C_p is derived and discussed by Kutner et al. (2005, p. 357), and the AIC and BIC are discussed by Ramsey and Schafer (2002, p. 356). The use of the AIC and BIC is relatively simple: one tries to make them as small (or as negative) as possible. The BIC differs from the AIC only in the complexity penalty. The use of Mallows' C_p is slightly more complex: one searches for the model having the smallest value of C_p that approximately equals the value of p.

In general, automated model selection methods fall into two broad categories: all possible subsets methods and stepwise methods. All possible subsets methods use a statistic such

TABLE 8.1
Statistics that Can be Used in Model Selection

Name	Symbol	Bias Penalty	Complexity Penalty
Mallows' C_p	C_p	$\frac{SSE}{MSE_{full}}$	$2p - n$
Akaike info. criterion	AIC	$n \ln SSE$	$2p$
Bayesian info. criterion	BIC	$n \ln SSE$	$p \ln n$

Note: Each of these is the sum of a bias penalty that is computed from the error sum of squares (*SSE*), and a complexity penalty that is computed from the number n of observations and the number p of regression coefficients. In the formula for *Cp*, the term *MSEfull* represents the mean squared error of the full model with all explanatory variables Included.

as Mallows' C_p to compare many or all possible combinations of the explanatory variables. The R package `leaps` (Lumley and Miller, 2017) contains a widely used set of functions for carrying out all possible subsets model selection. Stepwise methods use a statistic such as a p value or the Akaike information criterion to add or remove variables one at a time until the "best" version is obtained. Forward selection starts with no explanatory variables and adds them one at a time, backward selection starts with the full model and removes variables one at a time, and bidirectional selection combines forward and backward selection. If one's goal is to preserve the statistical properties of the full model as well as possible, then backward selection, in which one starts with the full model and removes variables one at a time, may be the best procedure (Snedecor and Cochran, 1989). One problem with stepwise methods is that they may converge to a model that is locally optimal, that is, one that is better than similar models but is not globally optimal over all of the explanatory variables. There are other, more subtle problems with stepwise selection as well (Harrell, 2001, p. 56).

In order to describe the approach to variable selection taken in this book, we need to develop a few more concepts. These include a more detailed look at the effects of multicollinearity as well as at two graphical tools: the added variable plot and the partial residual plot. These are the subjects of the next section.

8.2.2 Multicollinearity, Added Variable Plots, and Partial Residual Plots

Suppose that Y is a response variable and that there are two potential explanatory variables, X_1 and X_2. Consider the regression model

$$Y_i = \beta_0 + \beta_1 X_{i1} + \beta_2 X_{i2} + \varepsilon_i. \tag{8.8}$$

Both X_1 and X_2 are explanatory variables of Y, but if X_1 and X_2 are themselves correlated, that is, if multicollinearity as defined in the previous section exists in the data, then X_1 and X_2 are to some extent providing duplicate information. Before studying an example from a real data set, will first examine two extreme cases using artificial data. Our example is motivated by a similar one in Legendre and Legendre (1998, p. 591). In the first data set (Data Set A), X_1 and X_2 are random variables generated independently from a unit normal distribution (Figure 8.3a).

```
> set.seed(123)
> XA <- cbind(rnorm(100), rnorm(100))
```

Computing the sample variance-covariance matrix shows that X_1 and X_2 are almost uncorrelated.

```
> var(XA)
             [,1]         [,2]
[1,]  0.83323283 -0.04372107
[2,] -0.04372107  0.93506310
```

We construct a response variable Y by adding X_1 and X_2 together with a random error term, and then we compute the coefficients of the linear model. Note that $\beta_0 = 0$, $\beta_1 = \beta_2 = 1$, and the magnitude of the error is roughly one-tenth that of the regression coefficients. Some of the output of the function `summary()` is deleted.

```
> eps <- rnorm(100)
```

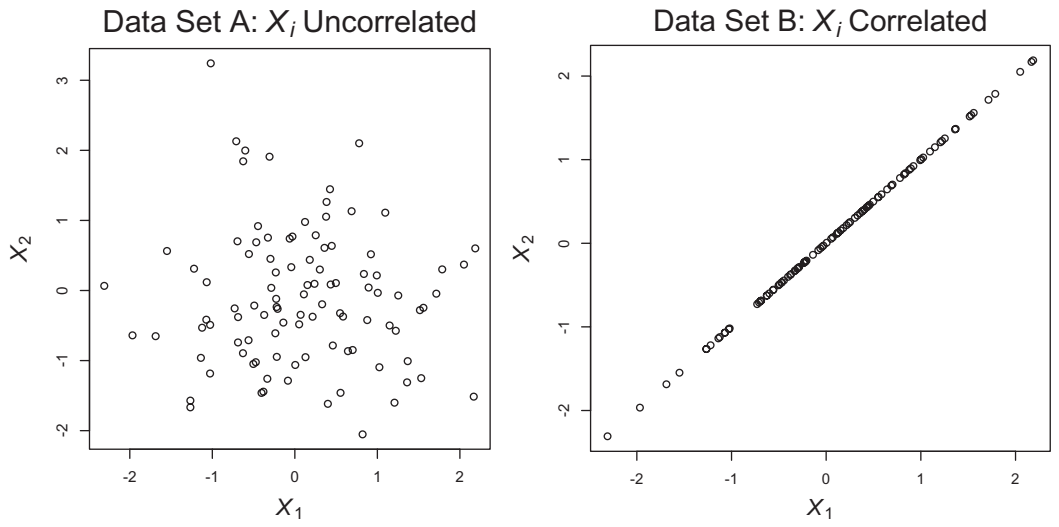

FIGURE 8.3
(a) Scatterplot of Data Set A, two uncorrelated variables X_1 and X_2 generated from a unit normal distribution. (b) Scatterplot of Data Set B, two highly correlated variables.

```
> YA <- rowSums(XA) + 0.1 * eps
> summary(lm(YA ~ XA))
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 0.013507   0.009614   1.405    0.163
XA1         0.986683   0.010487  94.087   <2e-16 ***
XA2         1.002381   0.009899 101.256   <2e-16 ***
Multiple R-squared: 0.9947,     Adjusted R-squared: 0.9946
```

Next, we remove X_2 from the model and recompute.

```
> summary(lm(YA ~ XA[,1]))
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -0.08954    0.09824  -0.911    0.364
XA[, 1]      0.93409    0.10764   8.678 8.89e-14 ***
Multiple R-squared: 0.4345,     Adjusted R-squared: 0.4288
```

Although the R^2 of the model with only X_1 is much lower than that of the model with both X_1 and X_2, the values of the estimated coefficients b_1 of X_1 are similar in both models and are quite close to the true β_1. Because they are independent, neither explanatory variable shares with the other any of the information provided about Y. Therefore, the value of b_1 does not depend on whether or not X_2 is in the model.

In the second case (Data Set B), X_1 is equal to its counterpart in Data Set A. The values of X_2 in Data Set B are a linear combination of X_1 and X_2 from Data Set A, arranged to put almost all the weight on X_1.

```
> XB <- cbind(XA[,1], XA[,1] + 0.0001 * XA[,2])
```

In this case, X_1 and X_2 are very highly correlated (Figure 8.3b).

```
> var(XB)
          [,1]      [,2]
[1,] 0.8332328 0.8332285
[2,] 0.8332285 0.8332241
```

With both variables in the model, the regression coefficients are quite different from the actual values β_i used to compute the response variable. The fit, however, is very good ($R^2 = 0.997$).

```
> YB <- rowSums(XB) + 0.1 * eps
> summary(lm(YB ~ XB))
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)   0.013507   0.009614   1.405    0.163
XB1         -22.824605  98.994169  -0.231    0.818
XB2          24.811288  98.994688   0.251    0.803
Multiple R-squared: 0.9973,     Adjusted R-squared: 0.9973
```

When X_2 is removed from the model, however, unlike the case with Data Set A, the coefficient of X_1 changes dramatically, and the change in R^2 is negligible.

```
> summary(lm(YB ~ XB[,1]))
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 0.013251   0.009514   1.393    0.167
XB[, 1]     1.986553   0.010424 190.577   <2e-16 ***
Multiple R-squared: 0.9973,     Adjusted R-squared: 0.9973
```

In model B, the explanatory variable X_2 is highly correlated with X_1 (this is the multicollinearity), while in the first model the two explanatory variables are independent. The most obvious effect of multicollinearity is that the value of the regression coefficient of one variable depends on whether the other variable is in the model.

Formally, the value of the regression coefficient b_i represents the marginal effect of the explanatory variable X_i on the response variable Y, assuming that all of the other explanatory variables are held constant (Kutner et al, 2005, p. 216). For example, if the value of X_{i1} changes by 1 unit, and the value of X_{i2} is held constant, the value of Y_i changes by b_1 units. This interpretation is still true mathematically in the presence of multicollinearity, but it no longer has any meaning. For example, suppose that X_1 represents sand content, X_2 represents clay content, and that the silt content is zero. Then $X_2 = 1 - X_1$. It makes no sense to think of holding X_1 constant while varying $1 - X_1$ (Harrell, 2001, p. 97). The best one can say is that, for example, the value of the regression coefficient b_1 represents the marginal effect of the explanatory variable X_{i1} on the response variable Y_i *given that the other variables are in the model*. It makes no sense biophysically, however, to think of holding one variable constant while varying another correlated variable.

A second deleterious effect of multicollinearity can be seen by returning to the output of the `summary()` function for the two artificial data sets displayed above. As we have already mentioned, the R^2 for the full model in Data Set B is about the same as that for the full model in Data Set A. This illustrates that multicollinearity does not substantially affect the fit of the data. Comparing the standard errors of the regression coefficients of Data Set B

with those of Data Set A, however, indicates that the standard errors of Data Set B are much higher. Multicollinearity increases the variability of the regression coefficients b_i, and in the presence of extreme multicollinearity these estimated regression coefficients can become so variable that they are almost worthless, as they are in this example. This effect exacerbates the already severe numerical difficulties that are sometimes associated with the spatial regression models discussed in Chapters 12 and 13.

For a physical example of multicollinearity, consider the two explanatory variables *Clay* and *SoilK* in Field 1 of Data Set 4. These variables both have a negative association with yield (Figure 7.25), so that in a regression model with only one explanatory variable, each generates a negative regression coefficient. The data are loaded into R as described in Appendix B.4 and Section 7.5.

```
> print(coef(lm(Yield ~ Clay, data = data.Set4.1)), digits = 4)
(Intercept)        Clay
     9454.9      -176.5
> print(coef(lm(Yield ~ SoilK, data = data.Set4.1)), digits = 3)
(Intercept)       SoilK
     6994.1       -21.5
```

In a model with both of these variables, the regression coefficient of *SoilK* is positive (and much smaller).

```
> print(coef(lm(Yield ~ Clay + SoilK, data = data.Set4.1)), digits = 2)
(Intercept)        Clay       SoilK
    9448.72     -177.09        0.15
```

This indicates that even though the effect of soil potassium by itself has a negative association with yield, the effect of increasing soil potassium given the presence of clay in the model is associated with an increase in yield. It is a bit surprising that the regression coefficient is negative when *SoilK* is the unique explanatory variable in the simple linear regression model for *Yield*. Does this mean that potassium is bad for wheat? Probably not. There are two potential explanations, which are not mutually exclusive. One is the mining effect of non-labile mineral nutrients discussed below in Section 8.3. A more likely explanation in this case, however, is the strong association of soil potassium with soil clay content itself. The area of the field with high clay content is also the area with high potassium content. It is likely that the apparent negative effect of soil potassium on yield occurs because any positive effect of soil potassium is outweighed by the negative effect of the high clay content. The sign of the regression coefficient for *SoilK* changes when *Clay* is added to the model because each variable contains information about the other one. In effect, this "loads" the coefficients of the other variable, that is, part of the effect of an explanatory variable is due to the effect on it of other variables included in the model (Fox, 1997, p. 127). Perhaps one can begin to see the reason for the "last refuge of scoundrels" comment.

The impact of multicollinearity can be assessed by means of *added variable plots*, which are also called *partial regression plots* (Kutner et al., 2005, p. 268; Fox, 1997, p. 281). To avoid confusion with partial residual plots, which are not the same thing and which are described below, we will use the former term. An added variable plot is a way of determining the effect of adding a particular explanatory variable to a model that already contains a given set of other explanatory variables. We will consider the case of two explanatory variables X_1 and X_2; the extension to more variables is straightforward. Consider a multiple regression model $Y_i = \beta_0 + \beta_1 X_{i1} + \varepsilon_i$, having one explanatory variable. We want to understand

the effect of adding a potential second explanatory variable X_2 to the model, noting that it may be correlated with X_1. The added variable plot for X_2, given that X_1 is in the model, is constructed as follows. First, we compute the residuals, denoted $e_i(Y \mid X_1)$, of the regression of Y on X_1:

$$\begin{aligned} \hat{Y}_i &= b_0 + b_1 X_{i1} \\ e_i(Y \mid X_1) &= Y_i - \hat{Y}_i. \end{aligned} \tag{8.9}$$

Next, we compute the residuals of the regression of X_2 on X_1:

$$\begin{aligned} \hat{X}_{i2} &= c_0 + c_1 X_{i1} \\ e_i(X_2 \mid X_1) &= X_{i2} - \hat{X}_{i2}. \end{aligned} \tag{8.10}$$

Intuitively, the residuals of the regression of Y on X_1 contain the information about the variability of Y not explained by X_1. The residuals of the regression of X_2 on X_1 contain the information about the variability of X_2 not explained by X_1. If X_2 adds information to the model not obtained from X_1, then there will be a pattern in the added variable plot that indicates the effect of adding the explanatory variable X_2. If there is a strong linear relationship between X_2 and X_1, then there will not be much information about the variability of Y remaining in the residuals of the regression of X_2 on X_1, and therefore the added variable plot will be more or less free from pattern.

We can illustrate added variable plots using the two artificial data sets generated above. First, consider Data Set A, in which X_1 and X_2 are independent.

```
> e.YA1 <- residuals(lm(YA ~ XA[,1]))
> e.XA21 <- residuals(lm(XA[,2] ~ XA[,1]))
> plot(e.XA21, e.YA1, #Fig. 8.4a
+     main = "Added Variable Plot of Data Set A", cex.main = 2,
+     xlab = expression(Residuals~of~italic(X)[2]),
+     ylab = expression(Residuals~of~italic(Y)),
+     cex.lab = 1.5)
```

The residuals of the regression of Y on X_1 are very closely associated with the residuals of the regression of X_2 on X_1 (Figure 8.4a). This indicates that X_2 contains a great deal of information about Y that is not contained in X_1. Repeating the same procedure with Data Set B (Figure 8.4b) indicates almost no relationship between the residuals. This indicates that almost all of the information about Y contained in X_2 has been provided by including X_1 in the model.

Another property of added variable plots is illustrated by computing the regression of $e(Y \mid X_1)$ on $e(X_2 \mid X_1)$ (see Equations 8.9 and 8.10). For convenience, the coefficients of the original regressions are also displayed.

```
> coef(lm(YA ~ XA))
(Intercept)          XA1          XA2
0.01350654   0.98668285   1.00238113
> coef(lm(YA ~ XA[,1]))
p(Intercept)     XA[, 1]
  -0.0895413   0.9340863
```

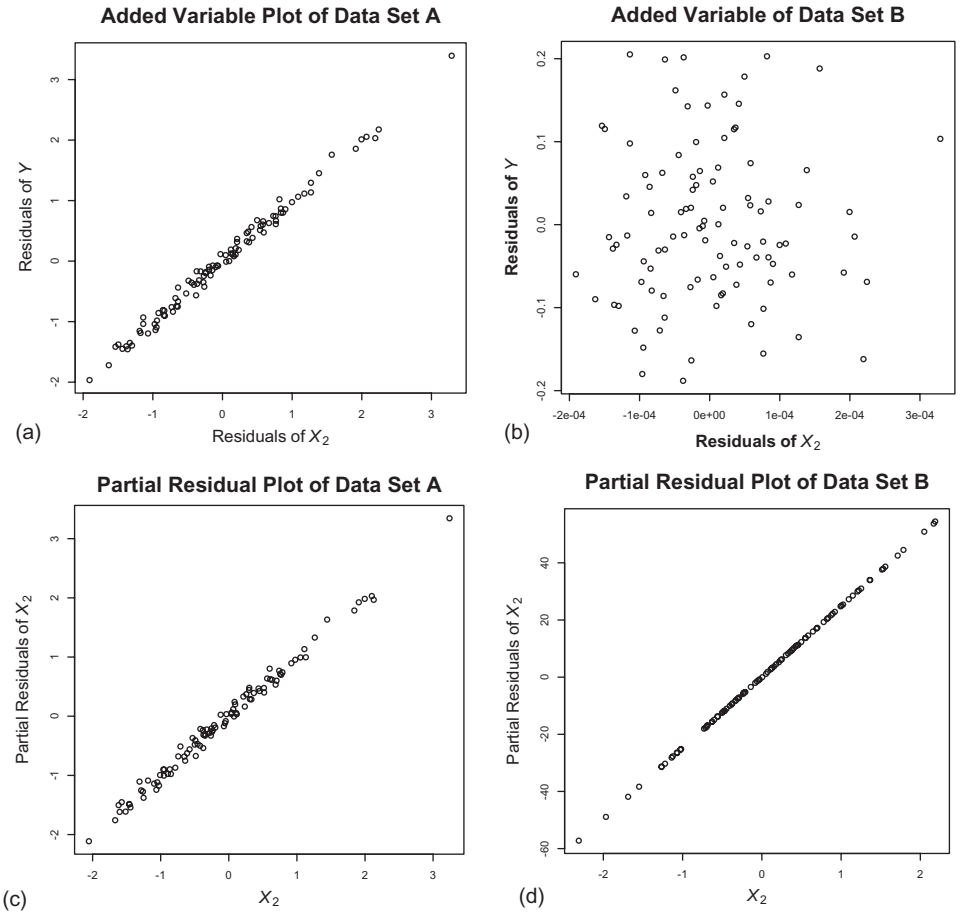

FIGURE 8.4
(a) Added variable plot of Data Set A. (b) Added variable plot of Data Set B. (c) Partial residual plot of Data Set A. (d) Partial residual plot of Data Set B.

```
> coef(lm(e.YA1 ~ e.XA21))
  (Intercept)        e.XA21
-1.334573e-17  1.002381e+00
```

The coefficient of the added variable regression, 1.002381, is equal to the coefficient b_2 of the full regression model. This is of course true for Data Set B as well.

```
> coef(lm(YB ~ XB))
(Intercept)          XB1          XB2
 0.01350654 -22.82460503  24.81128788
> coef(lm(YB ~ XB[,1]))
(Intercept)     XB[, 1]
 0.01325148  1.98655266
> coef(lm(e. YB1 ~ e.XB21))
```

```
 (Intercept)         e.XB21
2.167670e-18  2.481129e+01
```

Thus, if we use added variable regression to help build a multiple regression model, we can see the value of the regression coefficient in advance when considering whether or not to add a variable. Added variable plots are also very useful for identifying discordant data records, as we will see in the next section.

Although added variable plots can be constructed using code similar to that given above, it is much easier and more effective to use the function `avPlots()` of the `car` package (Fox and Weisberg, 2011). Not only does this function provide plots similar to (but nicer than) those of Figure 8.4, it also provides the ability to label points on the graph. This is very useful for identifying outliers.

The *partial residual plot* (which is also called the *component residual plot*) provides an alternative to the added variable plot to explore the effects of correlated explanatory variables (Larsen and McCleary, 1972). Again, suppose we wish to consider adding X_2 to the model given that X_1 is already in. Once again, the extension to higher values of p is direct. From Equation 8.9, the vector of residuals of the regression of X_2 on X_1 is denoted $e(X_2 \mid X_1)$. The partial residual $e^*(Y \mid X_1, X_2)$ is defined as the sum of the residuals of the full regression plus the X_2 component:

$$\begin{aligned} \hat{Y}_i &= b_0 + b_1 X_{i1} + b_2 X_{i2} \\ e_i(Y \mid X_1, X_2) &= Y_i - \hat{Y}_i \\ e_i{}^*(Y \mid X_1, X_2) &= e_i(Y \mid X_1, X_2) + b_2 X_{i2}. \end{aligned} \tag{8.11}$$

The partial residual plot is a plot of $e^*(Y \mid X_1, X_2)$ against X_2. To see why this works (Fox, 1997, p. 314; Ramsey and Schafer, 2002, p. 323) suppose that the model $Y_i = \beta_0 + \beta_1 X_{i1} + f(X_{i2}) + \varepsilon_i$ is an accurate representation of the data, but we are not sure of the form of the function $f(X_{i2})$. We can write

$$f(X_{i2}) = Y_i - \beta_0 - \beta_1 X_{i1} - \varepsilon_i. \tag{8.12}$$

We would like to plot $f(X_{i2})$ against X_2 to see its form, at least within the error term ε_i but we can't. However, if $f(X_{i2})$ is approximately linear (i.e., $f(X_{i2}) \cong b_2 X_{i2}$), then the regression of Y on X_1 and X_2 will give us estimates of b_0 and b_1 that we can use to approximate the plot. Furthermore, since $e_i(Y \mid X_1, X_2) = Y_i - \hat{Y}$ we have

$$f(X_{i2}) \cong Y_i - b_0 - b_1 X_{i1} - \varepsilon_i \tag{8.13}$$

But

$$\begin{aligned} Y_i - b_0 - b_1 X_{i1} &= \hat{Y}_i + e_i(Y \mid X_1, X_2) - b_0 - b_1 X_{i1} \\ &= b_0 + b_1 X_{i1} + b_2 X_{i2} + e_i(Y \mid X_I, X_2) - b_0 - b_1 X_{i1} \\ &= b_2 X_{i2} + e_i(Y \mid X_1, X_2) \\ &= e_i{}^*(Y \mid X_1, X_2) \end{aligned} \tag{8.14}$$

Therefore, a plot of the partial residuals may give an approximation of a plot of the function $f(X_{i2})$.

To illustrate partial residual plots, we again use the example data sets A and B. As with added variable plots, the `car` package has a function `crPlots()` (for component residuals) that would actually be used in practice, but we will first do it by hand for illustrative purposes. Here the code for Data Set A.

```
> A.lm <- lm(YA ~ XA)
> e.PRA <- residuals(A.lm)
> Y.PRA <- e.PRA + coef(A.lm)["XA2"] * XA[,2]
> plot(XA[,2], Y.PRA, main = "Partial Residual Plot of Data Set A",
+    cex.main = 2, xlab = expression(italic(X)[2]),
+    ylab = expression(Partial~Residuals~of~italic(X)[2]),
+    cex.lab = 1.5) # Fig. 8.4c
> coef(lm(Y.PRA ~ XA[,2]))
 (Intercept)      XA[, 2]
7.079840e-18 1.002381e+00
```

Your intercept term may be different from mine due to different round-off errors. Figure 8.4c shows the partial residual plot. The code for Data Set B is analogous, but the results are not.

Here are the summaries for the regression of the added variable regression and the partial residual regression.

```
> #Added variable regression
> summary(lm(e.YB1 ~ e.XB21))
Coefficients:
            Estimate Std. Error  t value Pr(>|t|)
(Intercept) 2.168e-18  9.464e-03 2.29e-16    1.000
e.XB21      2.481e+01  9.849e+01    0.252    0.802
Multiple R-squared: 0.0006472,  Adjusted R-squared: -0.00955
F-statistic: 0.06346 on 1 and 98 DF,  p-value: 0.8016
> #Partial residual regression
> summary(lm(Y.PRB ~ XB[,2]))
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 3.577e-16  9.511e-03 3.76e-14        1
XB[, 2]     2.481e+01  1.042e-02   2381   < 2e-16 ***
Multiple R-squared:    1,      Adjusted R-squared:     1
F-statistic: 5.669e+06 on 1 and 98 DF, p-value: < 2.2e-16
```

The partial residual plot for Data Set B is shown in Figure 8.4d. The two regression coefficients are the same, as the theory predicts, but the plots are somewhat different. To see why, note that due to the instability of this highly multicollinear system, the regression coefficients are very large.

```
> coef(B.lm)
 (Intercept)          XB1          XB2
  0.01350654 -22.82460503  24.81128788
```

The sum $e(Y \mid X_1, X_2) + b_2 X_2$ is approximately equal to $b_2 X_2$. Therefore, the partial residual regression is approximately a regression of $b_2 X_2$ on X_2, which yields the correct slope b_2 but

does not tell us anything explicitly about how much X_2 contributes to the model. This is an example of the numerical problems caused by multicollinearity.

Both the added variable plot and the partial residual plot potentially have other applications besides those described above, and both also have problems that in some cases reduce their effectiveness. Assuming that you want the good news first, let's look at some other potential applications. Both the added variable plot and the partial residual plot can often be used to detect nonlinearity in the contribution of a variable to the model. In addition, both plots can be used to identify the presence of outliers in the data. Partial residuals plots are most effective when the effect of the explanatory variable under consideration is small relative to that of those already in the model (Ramsey and Schafer, 2002, p. 325). Faraway (2002) compares an added variable plot and a partial residual plot for an example data set and concludes that added variable plots are better for detection of discordant data records, and partial residual plots are better for detecting the form of the relation of the response variable to the proposed added variable.

Now for the bad news (Fox. 1997, p. 315; Kutner et al., 2005, p. 389). These plots do not work well if the relation between the explanatory variables is itself nonlinear. Also, the plots may not detect interaction effects. Finally, as shown in the example of Data Set B above, high multicollinearity may invalidate the conclusions drawn from the plot. All this implies that these tools, like all tools, cannot be used blindly. Nevertheless, we shall have occasion to use both added variable and partial residual plots in the analyses of our data sets.

8.2.3 A Cautious Approach Model Selection as an Exploratory Tool

Our objective is to develop a procedure for using model selection as an exploratory tool to help determine potential explanatory variables for ecological processes. Ramsey and Schafer (2002, p. 346) give four good reasons why great caution is necessary when using model selection for this purpose:

1. Inclusion or exclusion of explanatory variables is strongly affected by multicollinearity between them.
2. The interpretation of multiple regression coefficients is difficult in the presence of multicollinearity because their value depends on the other variables in the model.
3. The interpretation of a regression coefficient as the effect of a variable given that all other variables are held constant makes no sense when there is multicollinearity.
4. Causal interpretations are in any case suspect in observational studies when there is no control over the values of explanatory variables.

A fifth problem not included in this list is that several different models often give very similar values for the measures of fit (C_p, AIC, and BIC) used in model selection.

On the other hand, graphical tools such as the scatterplot matrix and the thematic map do not provide conclusive evidence of influence either. The difference is that nobody expects them to. One of the problems with multiple regression, which is perhaps another source of the "last refuge of scoundrels" comment mentioned at the start of this section, is that with its precise numerical values and testable hypotheses, multiple regression has an aura of credibility that graphical methods lack. The trick in using multiple regression as an exploratory tool is not to take the results too seriously (or to present them as anything more than speculation). With that said, let's get on to developing our procedures. Further discussion of methods of model selection

(including the very important *shrinkage methods*, which we do not have space to cover) is contained in the references listed in Section 8.7.

We will follow the practice advocated by Henderson and Velleman (1981) and by Nicholls (1989), which is to test the variables for inclusion manually, using ecological knowledge as well as statistical tools. Our procedure follows closely that of Henderson and Velleman (1989), which is to use any available knowledge of the biophysics of the system to guide variable selection. This biophysical knowledge is augmented by tools such as the AIC, added variable plots, partial residual plots, and the general linear test (Appendix A.3) to examine the effects of the variables on the model, and to add or remove variables based on a combination of the computations and knowledge of the ecological processes involved. The procedure will be used to create a set of candidate models that can be further examined, if possible, using other analytical methods. If no further analysis is possible, the candidate models can be used to suggest experiments that can be used to distinguish between the competing models.

Burnham and Anderson (1998, p. 17) refer to the process of iteratively searching for patterns in the data as "data dredging." More recently it has come to be known by the less pejorative term "data mining." By whatever name, even Burnham and Anderson (1998) allow that it is not necessarily a bad thing, but that it must be done openly. They provide a nice analogy of model selection to an auto race (Burnham and Anderson, 1998, p. 54). Before the start of the race there is a qualifying event to determine which drivers can enter. This is analogous to the selection of the candidate models from among all the possible models. The race is run and the winner is determined. Often the winning driver is not necessary the fastest. Random events such as crashes or breakdowns may eliminate other drivers, and indeed in another race with the same set of drivers, a different driver may win. In the same way, the model determined to be "best" for a particular data set may not turn out to be the best for other data sets from the same system. Therefore, one should not read too much into the results of one particular "race" among the models. In this context, it is very important that "data dredging" not lead to "data snooping." This was defined in Section 7.1 as letting the confirmatory process be influenced inappropriately by the exploratory process through an introduced bias. If in this chapter we identify some candidate models through the exploratory process, we must not in a succeeding chapter forget where these models came from and use a hypothesis test inappropriately to assign a significance to a particular model that it does not deserve.

8.3 Building a Multiple Regression Model for Field 4.1

We will carry out our first multiple linear regression analysis on data from Field 4.1. The objective is to develop a set of candidate models for yield as a function of other variables. Our goal is not simply to predict yield but to improve our understanding of how the various explanatory variables interact to influence yield. To summarize the knowledge gained so far about Field 4.1: the field displays the greatest differences in its properties in a north to south direction. Crop yield is highest in the southern part of the field and declines markedly in the north. Dry soil infrared reflectance (Figure 7.23a) indicates that the northernmost part of the field retains water more than the southern part. Clay content is highest in the northern two-thirds of the field. This leads to the speculation that aeration stress may contribute to the low yield in some parts of the north of the field. Of the mineral

nutrients, neither soil potassium nor leaf nitrogen is at any sample point below the level at which a yield response to increased fertilization would be expected. The negative association of yield with soil nutrients evident in the scatterplots appears counterintuitive. Yield should increase with increasing nutrient levels. Webster and Oliver (2001, p. 201) point out that this seemingly paradoxical observation of higher nutrient levels with lower yields may be due to increased uptake of these nutrients in the high yielding areas, where they are not limiting. The negative association between yield and both soil nitrogen and potassium occurs in the high yield areas and is consistent with the observation in this data set that these nutrient levels exceed the minimum recommended values. However, there is an indication that soil phosphorous levels might be low in the northern part of the field (Figure 7.28a).

As a first step, we will select from Table 7.6 the explanatory variables that will be tested as candidates for inclusion in the regression model for *Yield*. Since we want to gain insight into the possible ecological processes that might influence yield, we will not consider gauge variables in the sense of Figure 7.1. The variable *SoilTOC* is almost perfectly correlated with *SoilTN* (Figure 7.25), so we will exclude it. The soil texture components sum to 100, and *Sand* has a strong negative association with *Clay* (Figure 7.25), so we will exclude it as well. The scatterplot matrix indicates that all of the endogenous variables (*CropDens, LeafN*, and *FLN*) share some degree of association with each of the exogenous variables. For now we will take the endogenous variables out of the model. We can use the term "agronomic" to refer to the variables that remain.

The scatterplot matrix of *Yield* and the agronomic variables (Figure 7.25) displays a "sideways parabola" relationship that indicates that there is an interaction between some of the explanatory variables associated with yield. Some yield values trend in one direction with the explanatory variable, and some are either not associated or trend in the opposite direction. This may indicate that yield has a different relationship with these variables in different parts of the field. In Exercise 8.3, you are asked to create these scatterplots separately for the northern 62 points (nine rows) and the southern 24 points (four rows). These plots indicate that there is a substantial geographic split, with the northern two-thirds of the field displaying a different behavior from the southern third.

The bifunctional relationship between *Yield* and the explanatory variables makes it impossible to construct a meaningful regression model that describes yield response over the entire field without including interaction terms. Once again, we have a decision to make. Since the variables have linear relationships in the northern and southern parts of the field, we could split the field into two separate areas for regression analysis, with 62 data locations in the north and 24 in the south. Unlike the analysis of Data Set 2, however, we will not split the field. The interactions may provide interesting information, and the data set is relatively small, which may lead to a degrees of freedom problem. There are lots of interactions that we could consider, and using the results of previous analyses to pick out some of them looks suspiciously like "data snooping." We will do it nevertheless, because some interactions make neither biophysical nor empirical sense. Based on the analysis in Section 7.5, the almost inescapable conclusion is that the mineral nutrients and pH interact with soil texture. Therefore, our regression model will include interaction terms of *Clay* with *SoilP*, *SoilK*, *SoilTN*, and *SoilpH*. A second possible interaction, justifiable on a biochemical basis, is between *SoilP* and *SoilpH*. In Exercise 8.8, you are asked to test for the possibility of this interaction being significant in the model.

Before we begin the regression analysis, we note that the exogenous variables *Weeds* and *Disease* and the endogenous variable *CropDens* are measured on an ordinal scale. Harrell (2001, p. 34), Kutner et al. (2005, p. 313), and Fox (1997, p. 135) all describe methods

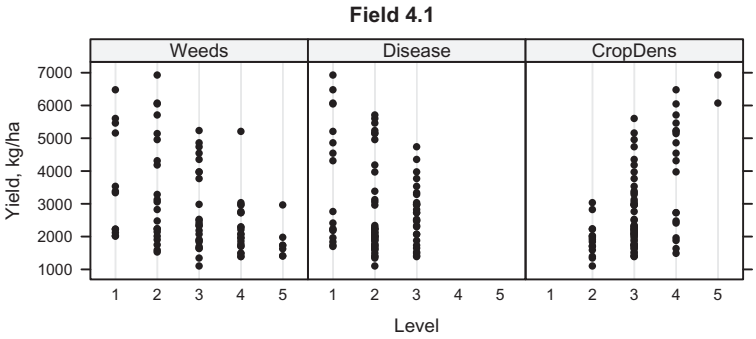

FIGURE 8.5
Dot plots of *Yield* vs. *Weeds*, *Disease*, and *CropDens* in Field 1 of Data Set 4.

of dealing with ordinal variables. These methods may not, however, capture the particular relationships between crop yield and these quantities. Dot plots of these relationships are shown in Figure 8.5. The plots of *Yield* against all three variables look reasonably linear. Mosteller and Tukey (1977, p. 103) describe a method of re-expression suggested for converting ordinal data into interval data. Similar methods may be used to suggest transformations of an interval or ratio scaled explanatory variable, and they may be applied to the response variable as well as the explanatory variables. Henderson and Velleman (1981) present an excellent example in which the response variable is gasoline mileage. They show that transforming miles per gallon to gallons per mile (i.e., inverting the response variable) makes the relationships with the explanatory variables much closer to linear. Returning to the re-expression of ordinal scale variables, it turns out that for our data the re-expression does not provide any improvement over the use of the ordinal variable itself. For this reason, we will stay with the use of *Weeds* and *Disease* in our regression model.

We are now ready to develop a multiple linear regression model for *Yield* in Field 4.1. In doing so, we need to again think about causal relationships and bias, which were touched upon in Section 8.2.3. The following discussion is taken from Fox (1997, p. 128), who in turn borrows from Goldberger (1973). Suppose we have a linear regression model with two explanatory variables:

$$Y_i = \beta_0 + \beta_1 X_{i1} + \beta_2 X_{i2} + \varepsilon_i. \tag{8.15}$$

where β_1 and β_2 are both nonzero. Fox points out that in interpreting Equation 8.15 one must distinguish between an *empirical model* and a *structural model*. In the case of the former, the only objective is to predict the value of Y and there is no assumption of a causal relation between the explanatory variables and the response variable. In the latter case, however (and this is our case), one assumes that the model is a representation of the actual relationships between the variables. Suppose that instead of fitting Equation 8.15, we fit a model with a single explanatory variable,

$$Y_i = \beta_0' + \beta_1' X_{i1} + \varepsilon_i'. \tag{8.16}$$

If X_1 and X_2 are correlated then, since the error term ε' includes the effect of X_2, it is correlated with X_1. This is an error in the specification of the model and results in β_1' being a

biased estimate of β_1. Fox (1997, p. 128) points out that the nature of the bias depends on the causal relationship between X_1 and X_2.

To be specific, consider again the model discussed in Section 8.2.2 of Field 4.1 involving *Clay* and *SoilK* as explanatory variables for *Yield*. Suppose first that X_1 represents *Clay* and X_2 represents *SoilK*. Although in the very long-term soil potassium content may influence clay content (Jenny, 1941), for our purposes we may assume that X_1 influences both X_2 and Y, but neither Y nor X_2 influence X_1. In this case, the bias in b_1 simply represents an indirect effect of clay content X_1 on Y by way of soil potassium X_2. Suppose the situation is reversed, however, so that X_2 represents *Clay* and X_1 represents *SoilK*. In this case X_2, which is not in the model, influences both X_1 and Y, so that the bias in b_1 represents a spurious component of the association between X_1 and Y. In our example, it may be that soil potassium X_1 actually does not influence yield at all, but merely appears to do so because of its association with clay content X_2. In order to avoid this sort of situation as much as possible, we will try to keep in mind influence relationships among the variables. In particular, this is why we exclude the endogenous variables *CropDens*, *LeafN*, and *FLN* from the model (and why we worry about distinguishing endogenous and exogenous variables).

The data are loaded as described in Appendix B.4 and Section 7.5. We create a subset of agronomic data as follows.

```
> agron.data <- with(data.Set4.1, data.frame(Clay, Silt,
+     SoilpH, SoilTN, SoilK, SoilP, Disease, Weeds, Yield))
```

There is one more important point to cover. We are constructing a regression model based on estimated values of the response variable *Yield*. Small differences in these estimated values may have dramatic consequences in the variable selection process (see Exercise 8.6). Therefore, your results may be different from those obtained in the book. With that caveat, let's get started. To get a general picture of the various regression relationships, we will first use the function `leaps()` from the package `leaps` to carry out an all possible subsets analysis. This function uses the Mallows C_p criterion (Section 8.2.1) to select the "best" subsets of explanatory variables. If there is no bias in the model, then the expected value of C_p is p. Values of C_p larger than p are taken to indicate bias, and values less than p are taken to indicate random variation (Kutner et al., 2005, p. 357).

The function `leaps()` does not support interaction terms, so we must add these directly to the data set.

```
> agron.data$ClayP <- with(agron.data, Clay*SoilP)
> agron.data$ClayK <- with(agron.data, Clay*SoilK)
> agron.data$ClaypH <- with(agron.data, Clay*SoilpH)
> agron.data$ClayTN <- with(agron.data, Clay*SoilTN)
```

The first argument of `leaps()` is a matrix whose columns are the explanatory variables, and the second argument is a vector representing the response variable.

```
> X <- as.matrix(agron.data[,-which(names(agron.data) == "Yield")])
> Y <- agron.data$Yield
```

We can now apply the function, specifying the names of the columns.

```
> model.Cp <- leaps(X, Y, method = "Cp",
+     names = names(agron.data[,-which(names(agron.data)
+         == "Yield")]))
```

This produces a `leaps` object, `model.Cp`. This is a list, and `model.Cp$which` gives the explanatory variables in each subset, sorted by p and by C_p. A little exploration with the function `str()` indicates that `model.Cp$size` contains the size p of the model (the number of explanatory variables plus one) and `model.Cp$Cp` contains the corresponding C_p values.

Table 8.2 shows some of the output from the following statement.

```
> cbind(model.Cp$Cp, model.Cp$which)
```

Somewhat confusingly, the number in the first column is $p-1$ rather than p. The table only shows some of the best subsets, and it does not show models that contain interaction terms but not the first order terms. Figure 8.6 shows plots of C_p against $p-1$ together with the $C_p = p$ line. We are looking for the smallest value of p such that $C_p \leq p$ This occurs at $p = 6$ (i.e., $p-1 = 5$). We will return to this figure below, and discuss the models shown in the figure. First, we will carry out a stepwise process so we can compare the results.

Since we already have the interaction terms in the object `agron.data`, we will continue for now to use these rather than specifying them directly in the formula argument. To construct candidate models, we will begin by using the Akaike information criterion in the manner suggested by Burnham and Anderson (1998), along with any biophysical knowledge we can apply. We will then see how the resulting models fit in with the `leaps()` results. We start with the full model and use the function `drop1()` to drop variables to arrive at candidates via backward selection. The output of the function `drop1()` is arranged in order of increasing values the AIC would take on if the variable were dropped. In Section 8.4.2, we are going to use `add1()` to construct a regression model for Data Set 1 using forward selection. In any real data analysis project, one should use all three methods, forward, backward, and bidirectional, and draw insight from a comparison of the results.

```
> model.full <- lm(Yield ~ ., data = agron.data)
> d1 <- drop1(model.full)
> d1[order(d1[,4]),]
Single term deletions
Model:
Yield ~ Clay + Silt + SoilpH + SoilTN + SoilK + SoilP + Disease +
    Weeds + ClayP + ClayK + ClaypH + ClayTN
         Df Sum of Sq      RSS    AIC
Clay      1      7779 38814660 1143.7
SoilK     1     10444 38817325 1143.7
ClayK     1     17184 38824065 1143.7
ClaypH    1     31004 38837885 1143.8
SoilpH    1     86604 38893485 1143.9
SoilTN    1    282543 39089424 1144.3
ClayTN    1    305760 39112641 1144.4
SoilP     1    604707 39411588 1145.0
Silt      1    782611 39589492 1145.4
ClayP     1    891388 39698269 1145.7
<none>                38806881 1145.7
Disease   1    991346 39798227 1145.9
Weeds     1   9110692 47917573 1161.8
```

This indicates that the AIC is reduced the most by dropping terms involving *Clay*, *SoilK*, and *ClayK*. Right away we have occasion to use knowledge of the biophysical process,

TABLE 8.2

Output of Function `leaps()` Applied to Data of Field 4.1, Showing Lowest Mallows' C_p Values for Increasing Number $p-1$ of Explanatory Variables in the Model

```
> print(cbind(model.Cp$Cp, model.Cp$which), digits = 3)
```

		Clay	Silt	SoilpH	SoilTN	SoilK	SoilP	Disease	Weeds	ClayP	ClayK	ClaypH	ClayTN
	*	*	*	*DELETED*		*	*	*					
3	20.05	1	0	0	0	1	0	0	0	0	1	0	0
3	22.10	0	0	1	0	0	0	0	1	0	0	1	0
3	22.64	1	0	0	0	0	0	0	1	1	0	0	0
	*	*	*	*DELETED*		*	*	*					
4	5.70	1	0	0	0	0	1	0	1	1	0	0	0
4	7.58	1	0	0	0	1	0	0	1	0	1	0	0
4	10.25	1	0	0	1	0	0	0	1	0	0	0	1
	*	*	*	*DELETED*		*	*	*					
5	3.68	0	0	1	0	0	1	0	1	1	0	1	0
5	4.14	1	0	1	0	0	1	0	1	1	0	0	0
	*	*	*	*DELETED*		*	*	*					
6	3.96	0	0	1	0	0	1	1	1	1	0	1	0
6	4.35	1	0	1	0	0	1	1	1	1	0	0	0
6	4.73	1	0	0	0	0	1	1	1	1	0	1	0
	*	*	*	*DELETED*		*	*	*					
7	5.29	0	1	1	0	0	1	1	1	1	0	1	0
7	5.56	0	0	1	0	0	1	1	1	1	1	1	0
7	5.73	0	0	1	0	0	1	1	1	1	0	1	1
	*	*	*	*DELETED*		*	*	*					
8	6.42	0	1	1	1	0	0	1	1	1	0	1	1
8	6.79	0	0	1	0	1	1	1	1	1	1	1	0
8	6.79	0	0	1	1	0	1	1	1	1	0	1	1
	*	*	*	*DELETED*		*	*	*					

Note: For each value of $p-1$, only the models with the lowest C_p are shown. Models that include an interaction without including both first-order terms are also not shown.

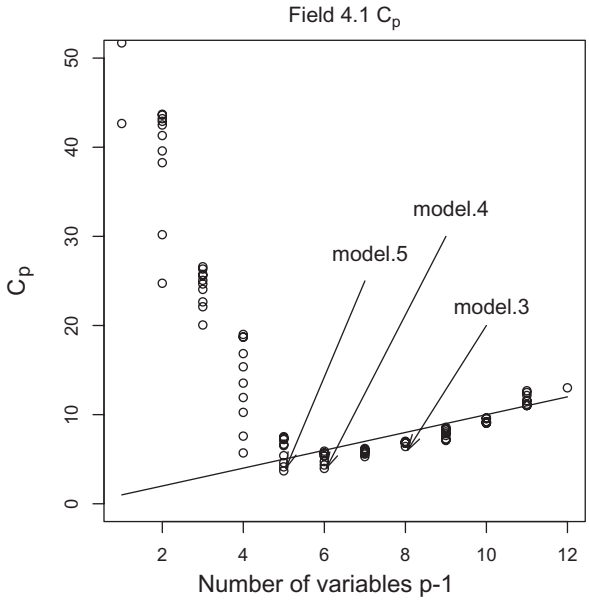

FIGURE 8.6
Plots of Mallows' C_p vs. p for the southern portion of Field 4.1, showing the C_p values of four of the models discussed in the text.

and to see the perils of a blind use of variable elimination based on numerical results. Everything we have seen in our exploration so far indicates that the variable *Clay* is of primary importance, at yet it is ranked as the first to leave. This is probably due to the effects of multicollinearity. Instead, we will drop the interaction terms *SoilK* and *ClayK*, whose resulting model has the same AIC. We will save the full model as `model.1`, since it includes all the variables.

```
> model.1 <- model.full
> model.test <- update(model.full, Yield ~ Clay + SoilpH + ClaypH +
+    Disease + SoilP + ClayP + Silt + Weeds)
```

Burnham and Anderson (1998) provide several convincing arguments and good examples for the assertion that hypothesis tests should not be used for variable selection. Nevertheless, this application provides a good opportunity to introduce the *general linear test* (Kutner et al., 2005, p. 72; Searle, 1971, p. 110), so we will provisionally ignore their good advice (spoiler alert: in this application it won't make any difference). We want to compare `model.test` and `model.full`. The general linear test is described in Appendix A.3. In brief, one can use it to test a null hypothesis of the form $H_0 : \beta = 0$, where β is a regression coefficient, against the alternative hypothesis $H_a : \beta \neq 0$. One regards the model under H_a as the *full* model and the model in which the null hypothesis is satisfied as the *restricted* model, because one or more of its parameters is restricted to a certain value. In our case, `model.full` is the full model and `model.test` is the restricted model in which the coefficients of *SoilK* and *ClayK* are restricted to the value zero. The restricted model is said to be *nested* in the full model. 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. If based on this p value the null hypothesis is not rejected, then we can take this as an indication that the restricted model provides the same functionality as the full model.

In R, the general linear test is carried out using the function `anova()`.

```
> anova(model.test, model.full)
Analysis of Variance Table
Model 1: Yield ~ Clay + Silt + SoilpH + SoilTN + SoilP + Disease +
Weeds + ClayP + ClaypH + ClayTN
Model 2: Yield ~ Clay + Silt + SoilpH + SoilTN + SoilK + SoilP +
Disease + Weeds + ClayP + ClayK + ClaypH + ClayTN
 Res.Df       RSS Df Sum of Sq      F Pr(>F)
1     75 38858288
2     73 38806881      2 51407 0.0484 0.9528
> AIC(model.full)
[1] 1391.757
> AIC(model.test)
[1] 1387.871
```

The p value of 0.95 indicates that we can accept `model.test` and continue looking for variables to drop. We invoke `drop1()` again (not shown) and as a result we drop *SoilpH* and *ClaypH*. Continuing in this way, we continue dropping variables until we reach a point where the next dropped variable makes the model worse, either because the AIC is higher or because the null hypothesis is rejected. Note in the list of dropped variables that *Clay* is now at the bottom, indicating the effects of multicollinearity.

```
> model.test <- update(model.full, Yield ~ Clay + Silt + SoilTN +
+     Weeds + ClayTN)
> AIC(model.full)
[1] 1391.757
> AIC(model.test)
[1] 1388.919
> anova(model.test, model.full)
Analysis of Variance Table
Model 1: Yield ~ Clay + Silt + SoilTN + Weeds + ClayTN
Model 2: Yield ~ Clay + Silt + SoilpH + SoilTN + SoilK + SoilP +
Disease + Weeds + ClayP + ClayK + ClaypH + ClayTN
 Res.Df       RSS Df Sum of Sq      F Pr(>F)
1     80 44185294
2     73 38806881  7   5378413 1.4453 0.2006
> d1 <- drop1(model.test)
> d1[order(d1[,4]),]
Single term deletions
Model:
Yield ~ Clay + Silt + SoilTN + Weeds + ClayTN
       Df Sum of Sq      RSS    AIC
<none>              44185294 1142.9
Silt    1   1663033 45848327 1144.0
Weeds   1  11199220 55384514 1160.3
ClayTN  1  11269156 55454450 1160.4
SoilTN  1  11335854 55521148 1160.5
Clay    1  23864449 68049743 1178.0
```

This invocation of `drop1()` indicates that there is no variable that can be dropped that would improve the model's AIC. There is no point in bothering with an application of the general linear test.

We will denote the model that we obtain by this blind use of stepwise regression as `model.2`.

```
> model.2 <- model.test
```

Returning to the Mallows' C_p output in Figure 8.6 and Table 8.2, we see that `model.2` is not included as a leading contender there. We will add three models based on the `leaps()` results.

```
> model.3 <- update(model.full, Yield ~ Clay + SoilpH +
+    SoilP + Weeds + ClayP + ClaypH)
> model.4 <- update(model.full, Yield ~ Clay + SoilpH + SoilP +
+    Weeds + ClayP)
> model.5 <- update(model.full, Yield ~ Clay + SoilP +
+    Weeds + ClayP)
```

Now we need to carry out some checks on the candidate models. We will pick `model.5` as an example, but these tests should also be carried out on the other candidate models as well.

A first question is whether there is substantial multicollinearity. The *variance inflation factor* (VIF) provides a good test for multicollinearity (Kutner et al., 2005, p. 408). This is available using the function `vif()` from the package `car`.

```
> vif(model.5)
     Clay     SoilP     Weeds     ClayP
8.953738 60.727652 1.118789 67.481848
```

Generally, one wants to keep the VIF below ten. The *Clay*×*SoilP* and *SoilP* terms are considerably above this, but interpretation of the VIF for interactions involving non-centered variables is difficult (Robinson and Schumacker, 2009).

Next, we check for influential variables. The tests for these are described in Appendix A.2.

```
> which(rowSums(influence.measures(model.5)$is.inf) > 0)
48 62 69 80 81 82
```

A call to the function `avPlots()` produces graphics (not shown) that do not support removal of any data.

```
> avPlots(model.5, id.n = 1)
```

This is confirmed by a call to the `car` function `outlierTest()`.

```
> outlierTest(model.5)
No Studentized residuals with Bonferonni p < 0.05
Largest |rstudent|:
   rstudent unadjusted p-value Bonferonni p
69 2.463534           0.015903           NA
```

Now we construct the partial residuals plot to determine whether either of the variables should have a higher order term. Calls to the function `crPlot()` (not shown) indicate that linear terms suffice.

The next question is whether there is substantial heteroscedasticity (non-constant variance) among the residuals. One test for this is a plot of the residuals vs. fits (not shown). We will carry out a Breusch-Pagan test (Kutner et al., 2005, p. 118). The function for this is in the `lmtest` package (Zeileis and Hothorn, 2002).

```
> bptest(model.5)
        studentized Breusch-Pagan test
data:  model.5
BP = 8.614, df = 4, p-value = 0.07151
```

The results indicate a bit of a problem with heteroscedasticity. As of now, however, we will not take any corrective action. The next step is to check for normality of the residuals. The *QQ* plot is a possibility (Kutner et al., 2005, p. 110), and I have included this in the code file. Personally, however, I find these somewhat hard to interpret, so instead I will do a Shapiro-Wilk normality test (Shapiro and Wilk, 1965) on the residuals.

```
> shapiro.test(residuals(model.5))
        Shapiro-Wilk normality test
data:   residuals(model.5)
W = 0.98928, p-value = 0.7065
```

Normality looks good.

The final test we will carry out involves the predicted sum of squares, or *PRESS*, statistic (Allen, 1971; Kutner et al., 2005, p. 360). This statistic provides an indication for all data records of how well a model based on a subset of the data in which the i^{th} data record has been deleted can predict the response variable value Y_i. The formula for the statistic is

$$PRESS = \sum_{i=1}^{n} (Y_i - \hat{Y}_{i(i)})^2, \tag{8.17}$$

where $\hat{Y}_{i(i)}$ is the predicted value of Y_i computed from the model when the i^{th} data record is deleted. It turns out that the statistic can be computed without actually computing each individual regression model (Kutner et al., 2005, p. 360). The *PRESS* statistic has two uses. First, it is a way to perform a final check of the model to ensure against overfitting. One can compute the value of the statistic for the proposed model and for each of the reduced models in which one explanatory variable is omitted, and eliminate any variable leading to a lower value of this statistic. The `MPV` package (Braun, 2015) contains a function `PRESS()` that does this computation.

```
> PRESS(model.full)
[1] 56576470
> PRESS(model.5)
[1] 47990460
```

The *PRESS* statistic of `model.5` is much smaller than that of the full model.

The second use of the *PRESS* statistics is as one check of model validity. If the regression model is valid, one would expect the value of the *PRESS* statistic to be about the same as

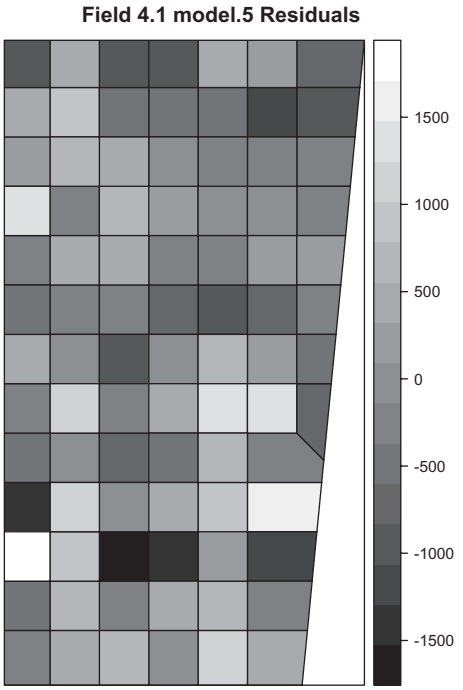

FIGURE 8.7
Thematic map of residuals of linear regression models of the northern and southern portions of Field 4.1, with residuals in the southern portion scaled to those in the north.

the mean square error (Kutner et al., 2005, p. 360). The *MSE* is computed using the R function `deviance()`.

```
> PRESS(model.5) / deviance(model.5)
[1] 1.104985
```

This is a positive indication for the validity of the model.

As a final step in this phase of the analysis, we construct a Thiessen polygon map of the residuals of `model.5` (Figure 8.7). We are looking for signs of spatial autocorrelation among the residuals. As is often the case, it is difficult to tell from the figure whether such autocorrelation exists. We will take up this issue in Chapters 12 and 13.

In summary, we have the full model (`model.1`) plus four selected models to carry forward. These four selected models are:

```
> model.2
Call:
lm(formula = Yield ~ Clay + Silt + SoilTN + Weeds + ClayTN, data =
agron.data)
Coefficients:
(Intercept)        Clay        Silt      SoilTN      Weeds      ClayTN
   19264.13    -487.46       73.43    -157.58     -319.02    4151.94
```

```
> model.3
Call:
lm(formula = Yield ~ Clay + SoilpH + SoilP + Weeds + ClayP +
   ClaypH, data = agron.data)
Coefficients:
(Intercept)       Clay     SoilpH      SoilP      Weeds      ClayP     ClaypH
  -25479.09     544.50    6752.79    -572.53    -380.41      16.66    -140.53
> model.4
Call:
lm(formula = Yield ~ Clay + SoilpH + SoilP + Weeds + ClayP, data =
agron.data)
Coefficients:
(Intercept)       Clay     SoilpH      SoilP      Weeds      ClayP
     6802.9     -314.9     1474.5     -775.7     -369.3       21.9
> model.5
Call:
lm(formula = Yield ~ Clay + SoilP + Weeds + ClayP, data = agron.data)

Coefficients:
(Intercept)       Clay      SoilP      Weeds      ClayP
   16083.29    -324.72    -890.17    -390.78      24.05
```

The same explanatory variables tend to appear in all of them. The coefficients, however, change dramatically. For example, *Clay* has a positive coefficient in `model.3`, indicating once again the effect of multicollinearity on the values of the coefficients. Nevertheless, they do provide evidence of the importance of soil texture and mineral nutrient concentration. We will continue to accumulate evidence about the effects of these variables as we move forward in the subsequent chapters. The results also show a dramatic difference between those obtained via backwards selection and those obtained by the best subsets method. Again, in a real data analysis project forward and bidirectional selection would also be tested. The function `stepAIC()` in the package `MASS` (Venables and Ripley, 2002) provides a good option for this.

8.4 Generalized Linear Models

8.4.1 Introduction to Generalized Linear Models

The response variables in Data Sets 1 and 2 are the indicators of the presence or absence at the sample location of, respectively, yellow-billed cuckoos and blue oaks. These indicator variables take on one of two values: 0, indicating absence, and 1, indicating presence. This is clearly not a normal distribution, and if one were to try to fit a linear regression model of the form of Equation 8.1 to the data, the error terms ε_i would also not be normally distributed, and the model would not be valid. Binary response variables like these are members of a class of data that can be analyzed using *generalized linear models*, or GLM (Nelder and Wedderburn, 1972; Fox, 1997, p. 438; Kutner et al., 2005, p. 555). We will introduce GLM by considering a model with a single explanatory variable. Figure 8.8 shows the distribution

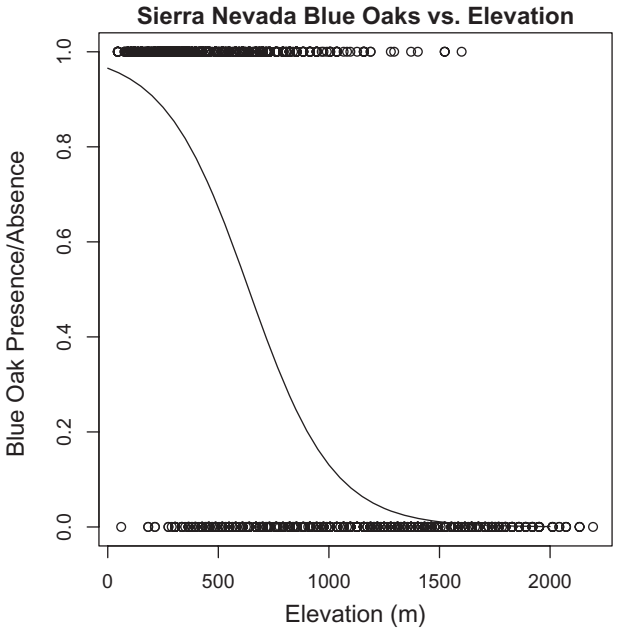

FIGURE 8.8
Plot of presence (= 1) and absence (= 0) of blue oaks as a function of *Elevation*, together with a logistic regression model of this relationship.

of *QUDO* values in the Sierra Nevada subset of Data Set 2 as a function of the single environmental variable *Elevation*. Also shown in the figure is the fit of a *logistic regression* model, which is one form of generalized linear model. The curve symbolizes the expected value of *QUDO* as a function of elevation. We already know that blue oak is almost completely absent above 1300m and that in the Sierra Nevada blue oak presence declines steadily as a function of elevation (Figure 7.11), and this is reflected in the model fit.

Consider the general case of a binary (zero or one) response variable Y_i and a single explanatory variable X_i. Let π_i represent the conditional probability that $Y_i = 1$ given that $X = X_i$, i.e.

$$\pi_i = P\{Y = Y_i \mid X = X_i\}. \tag{8.18}$$

It follows that

$$E\{Y \mid X = X_i\} = \pi_i \times 1 + (1 - \pi_i) \times 0 = \pi_i. \tag{8.19}$$

For example, in the case of the Sierra Nevada blue oak data, the expected value of $Y = QUDO$ should decline with increasing $X = Elevation$ as shown in Figure 8.8. In the case of ordinary linear regression, where the model is $Y_i = \beta_0 + \beta_1 X_i + \varepsilon_i$ and $E\{\varepsilon_i\} = 0$, we have $E\{Y_i\} = \beta_0 + \beta_1 X_i$. With generalized linear models we obtain a curvilinear relationship between $E\{Y_i\}$ and $\beta_0 + \beta_1 X_i$ like that of Figure 8.8 by applying a function like the *logistic function* to the explanatory variable. The logistic function has the form

$$\pi_i = h(\beta_0 + \beta_1 X_i) = \frac{1}{1+\exp(-[\beta_0 + \beta_1 X_i])}. \tag{8.20}$$

In the case of the blue oak vs. elevation data, β_1 will be negative, so that as X_i becomes a large positive number, $\exp(-[\beta_0 + \beta_1 X_i])$ becomes a very large positive number, and therefore $h(\beta_0 + \beta_1 X_i)$ approaches zero. As X_i becomes a large negative number, $\exp(-[\beta_0 + \beta_1 X_i])$ approaches zero, and therefore $h(\beta_0 + \beta_1 X_i)$ approaches one. Thus, the curve of $h(\beta_0 + \beta_1 X_i)$ resembles that shown in Figure 7.11 and Figure 8.8.

Using a little algebra, one can show (Kutner et al., 2005, p. 561) that the inverse of the logistic function is given by

$$g(\pi_i) = h^{-1}(\pi_i) = \ln\left(\frac{\pi_i}{1-\pi_i}\right). \tag{8.21}$$

Thus, we can estimate coefficients β_0 and β_1 by fitting the data to the model

$$g(\pi_i) = \ln\left(\frac{\pi_i}{1-\pi_i}\right) = \beta_0 + \beta_1 X_i. \tag{8.22}$$

The function g is called a *logit transformation*, and when it is used in this way in logistic regression it is called a *link function* (Kutner et al., 2005, p. 623). The quantity $\pi_i / (1-\pi_i)$ is called the *odds*.

The computation of the regression coefficients β_0 and β_1 is generally carried out using the *method of maximum likelihood*, which is discussed in Appendix A.5. Given that $P\{Y_i = 1\} = \pi_i$ and $P\{Y_i = 0\} = (1-\pi_i)$, we can represent the probability distribution of the Y_i as

$$f_i(Y_i) = \pi_i^{Y_i}(1-\pi_i)^{1-Y_i}, \quad Y_i = 0, 1, \quad i = 1,...n. \tag{8.23}$$

Since the Y_i are assumed independent (this is where we run into problems with spatial data), their joint probability distribution is the product of the individual $f_i(Y_i)$,

$$f(Y_1,...,Y_n) = \prod_{i=1}^{n} \pi_i^{Y_i}(1-\pi_i)^{1-Y_i}. \tag{8.24}$$

Taking logarithms yields

$$\begin{aligned} \ln f(Y_1,...,Y_n) &= \ln \prod_{i=1}^{n} \pi_i^{Y_i}(1-\pi_i)^{1-Y_i} \\ &= \sum_{i=1}^{n} [Y_i \ln \pi_i + (1-Y_i)\ln(1-\pi_i)] \\ &= \sum_{i=1}^{n} \left[Y_i \ln\left(\frac{\pi_i}{1-\pi_i}\right)\right] + \sum_{i=1}^{n} \ln(1-\pi_i). \end{aligned} \tag{8.25}$$

It turns out (based on algebra from earlier equations) that

$$1-\pi_i = (1+\exp(\beta_0+\beta_1 X_i))^{-1}. \tag{8.26}$$

Therefore, substituting from Equations 8.22 and 8.26 into Equation 8.25 yields the log likelihood function

$$\ln L(\beta_0,\beta_1) = l(\beta_0,\beta_1) = \sum_{i=1}^{n}\left[Y_i(\beta_0+\beta_1 X_i) - \ln(1+\exp(\beta_0+\beta_1 X_i)\right]. \tag{8.27}$$

Given a set of pairs (X_i, Y_i), $i = 1,...,n$, the maximum likelihood estimates of b_0 and b_1 of β_0 and β_1 are obtained by maximizing $l(\beta_0,\beta_1)$ in Equation 8.27. The curve in Figure 8.8 represents the maximum likelihood solution of Equation 8.27 for the Sierra Nevada blue oaks data. The solution to this nonlinear problem must be obtained numerically. The most common method for doing this is called *iteratively reweighted least squares* (McCulloch et al., 2008, p. 144).

The computation of a generalized linear model in R is accomplished with the function `glm()`, which works just like the function `lm()` except that one must also specify the family to which the data belong in order to ensure the use of the correct link function. In our case, the data are binomial. With the Sierra Nevada subset of Data Set 2 loaded into the `sf` object `data.Set2S` as described in Appendix B.2 and Section 7.3, the code to generate the model is as follows.

```
> glm.demo <- glm(QUDO ~ Elevation, data = data.Set2S,
+    family = binomial)
> summary(glm.demo)
Call:
glm(formula = QUDO ~ Elevation, family = binomial, data = data.Set2S)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.4734  -0.5620  -0.1392   0.6126   3.1737
Coefficients:
             Estimate Std. Error z value  Pr(>|z|)
(Intercept)  3.3291447  0.1722626   19.33   < 2e-16 ***
Elevation   -0.0052236  0.0002481  -21.05   < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
 (Dispersion parameter for binomial family taken to be 1)
    Null deviance: 2371.3  on 1767  degrees of freedom
Residual deviance: 1366.5  on 1766  degrees of freedom
AIC: 1370.5
Number of Fisher Scoring iterations: 6
```

It is a straightforward matter to extend the logistic regression model to multiple explanatory variables. One simply rewrites the link function in Equation 8.22 as

$$g(\pi_i) = \ln\left(\frac{\pi_i}{1-\pi_i}\right) = \beta_0 + \beta_1 X_{i1} + \beta_2 X_{i2} + ... + \beta_{p-1} X_{i,p-1}. \tag{8.28}$$

and again obtains the estimates b_i of the β_i by the method of maximum likelihood. The predicted values $\hat{\pi}_i$ are then computed as

$$\hat{\pi}_i = h(b_0 + b_1 X_{i1} + b_2 X_{i2} + \ldots + b_{p-1} X_{i,p-1i}) = \frac{1}{1 + \exp(-[b_0 + b_1 X_{i1} + b_2 X_{i2} + \ldots + b_{p-1} X_{i,p-1i}])}. \tag{8.29}$$

Equation 8.27 can be used to develop a statistic analogous to the coefficient of determination R^2 of linear regression. For a given set of regression coefficients having a likelihood function L, the *deviance* G^2 is defined as (Fox, 1997, p. 451)

$$G^2 = -2 \ln L. \tag{8.30}$$

Let G_0^2 be the deviance of the model that includes only β_0 present, and let G_1^2 be the deviance of the model being considered. Then one can define

$$R^2 = 1 - \frac{G_1^2}{G_0^2}. \tag{8.31}$$

as a measure of fit analogous to the coefficient of determination of the linear model. In this sense, the deviance can be considered as analogous to the error sum of squares of the linear model.

One cannot simply define the residuals of a generalized linear model as $e_i = Y_i - \hat{Y}_i$ in the manner of the linear model. Such a definition would produce residuals that were not normally distributed and had no meaningful interpretation. Nevertheless, if you apply the R function `residuals()` to a `glm` object, you will get a valid response. As it is with many R functions, polymorphism is implemented in the function `residuals()` by appending the name of the class to the function name. Thus, if you type `?residuals.glm`, you can view the Help file for the calculation of residuals of `glm` objects. As they often do, this help file contains a list of useful references. There are a number of definitions of residuals for the generalized linear model, of which we will focus on two: the *deviance residuals* and the *Pearson residuals*. The deviance residuals are defined as (Kutner et al., 2005, p. 592)

$$e_{Di} = sign(Y_i - \hat{\pi}_i) \sqrt{-2[Y_i \ln \hat{\pi}_i + (1 - Y_i) \ln(1 - \hat{\pi}_i)]}, \tag{8.32}$$

where the $\hat{\pi}_i$ are the fits obtained via Equation 8.29. The Pearson residuals are defined as (Kutner et al., 2005, p. 591)

$$e_{Pi} = \frac{Y_i - \hat{\pi}_i}{\sqrt{\pi_i (1 - \pi_i)}}. \tag{8.33}$$

The deviance residuals focus on the analogy of the deviance to the error sum of squares (Kutner et al., 2005, p. 68), while the Pearson residuals are related to the use of the chi-square statistic in a test of goodness of fit, which is discussed in Chapter 10.

The last step before moving on to the analysis of ecological data is to see how added variable or partial residual plots are applied to generalized linear models. There is considerable literature on the subject. In the case of partial residual plots for the logistic regression model, Landwehr et al. (1984) recommend plotting the *logit partial residuals*, which are similar to the Pearson residuals but involve division by $\hat{\pi}_i(1-\hat{\pi}_i)$ rather than by $\sqrt{\hat{\pi}_i(1-\hat{\pi}_i)}$ and which also include a term involving the explanatory variable in the plot. Nicholls (1989) recommends plotting residuals against variables not in the model. He uses the Pearson residuals. Davison and Snell (1991), on the other hand, advise against using Pearson residuals in general. Kutner et al. (2005, p. 594) compare the various types of residual plots and indicate that, at least in their example, the different types of residuals provide generally similar results. The functions `avPlots()` and `crPlots()` of the `car` package can both be used with `glm` objects. The function `avPlots()` uses a method described by Wang (1985) for generalized linear models, and the function `crPlots()` uses the method of Landwehr et al. (1984) given above, as described by Fox and Weisberg (2011).

To get a better feel for how to use added variable and partial residual plots with generalized linear models we will develop two artificial data sets. The explanatory variables of the data sets are two independent samples drawn from a normal distribution. Each consists of 400 values, in order to make the relationships as clear as possible. The response variable Y of the first data set is a binomial random variable that depends through a link function of the form of Equation 8.22 on the quantity $X_1 + X_2$ (i.e., $\beta_0 = 0$ and $\beta_1 = \beta_2 = 1$). Here is the code for the regression model.

```
> set.seed(123)
> X <- cbind(rnorm(400), rnorm(400))
> p <- 1 / (1 + exp(-rowSums(X)))
> Y <- rbinom(numeric(length(p)), 1, p)
```

Suppose now that we have a regression model with X_1 in as an explanatory variable and we wish to test the effect of including X_2. We can create a partial residual plot with the following statement (note that the reserved word *for* needs to be enclosed in quotes).

```
> model.lin <- glm(Y ~ X[,1] + X[,2], family = binomial)
> library(car)
> crPlots(model.lin, col.lines = c("black", "black"), # Fig. 8.9a
+    main = expression(Partial~Residual~Plots~"for"~italic(X)[2]))
```

The results are shown in Figure 8.9a. There are two things about this plot that we need to discuss. First, note that the points form two clouds. This is a common feature of added variable and partial residual plots for logistic regression models and occurs because the response variable Y can only take on one of two values, zero and one, and thus the residuals are also bimodal. This feature makes the point clouds of added variable and partial residual plots very difficult to interpret, and leads to the inclusion of the second feature that must be explained. The dashed line in Figure 8.9a is a regression line fit to the point clouds, but the more important line for our purposes is the solid line. This is a called a *loess* curve. "Loess" stands for *locally weighted scatterplot smoothing* (Cleveland and Devlin, 1988; Kutner et al., 2005, p. 138; Fox, 1997, p. 417), and for this reason it is sometimes spelled

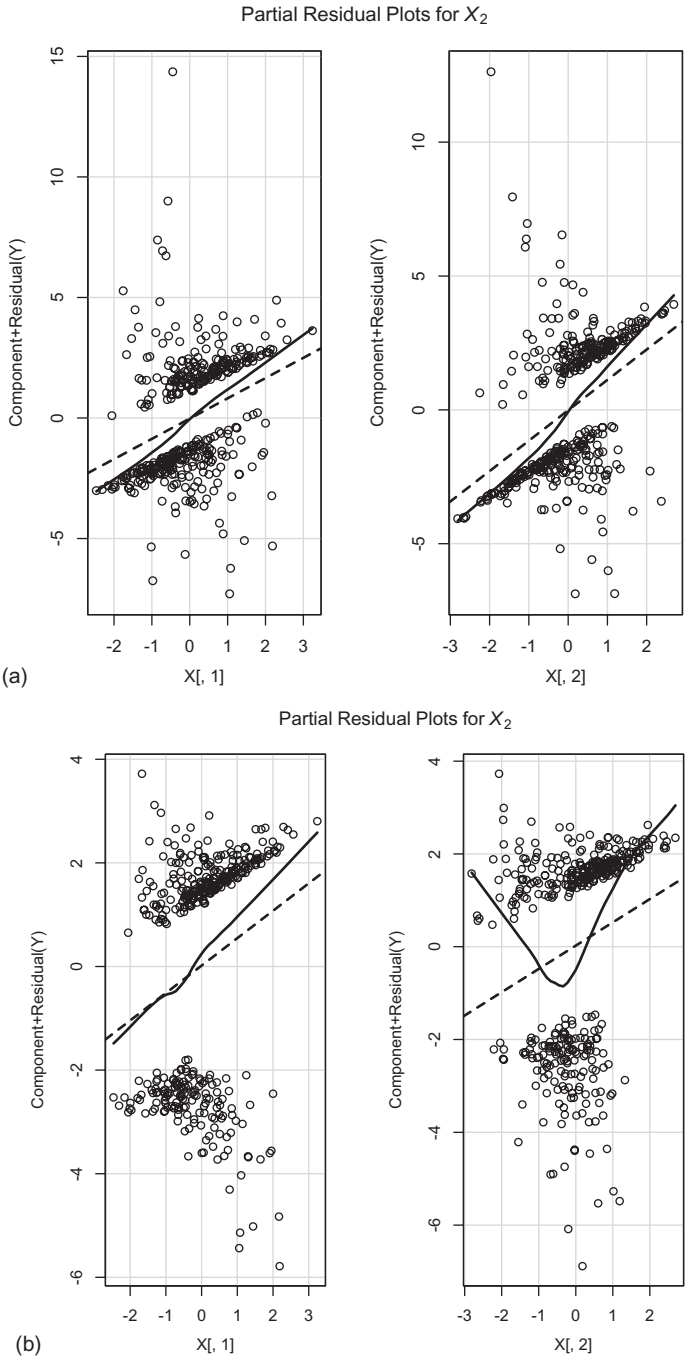

FIGURE 8.9
(a) Partial residual plots for a logistic regression model based on artificial data. (b) Partial regression plots for a model similar to that of part (a), but incorporating a quadratic term in X_2.

"lowess." In any case, unlike the soil type of the same spelling, it is pronounced "*low-ess*." Loess is a form of nonparametric regression, which means that there is no assumption made about the distribution of the regression residuals. In loess fitting, there is no estimation of parameters β_i. Instead, the curve is constructed simply to fit the data. This type of function is also called a *smoothing function*, and we will meet smoothing functions again when we discuss the generalized additive model in Section 9.2.

In Figure 8.9a, the loess curve lies almost right on top of the linear regression fit, which indicates that the contribution of X_2 to the response variable Y is linear. Now consider the following artificial data set.

```
> set.seed(123)
> X <- cbind(rnorm(400), rnorm(400))
> X <- cbind(X, X[,2]^2)
> p <- 1 / (1 + exp(-rowSums(X)))
> Y <- rbinom(numeric(length(p)), 1, p)
```

Now Y also depends on X_2^2, so that the true model would be of the form $Y_i = \beta_1 X_{i1} + \beta_2 X_{i2} + \beta_3 X_{i2}^2 + \varepsilon_i$. Suppose, however, that we again fit the data with a model that excludes the X_2^2 term and construct the partial residual plot.

```
> model.lin <- glm(Y ~ X[,1] + X[,2], family = binomial)
> crPlots(model.lin, col.lines = c("black", "black"), # Fig. 8.9b
+    main = expression(Partial~Residual~Plots~"for"~italic(X)[2]))
```

Now the loess curve shows a distinct parabolic form (Figure 8.9b), which is not visually apparent in the clouds of points. This example shows that we can use the partial residual plots to detect nonlinearity in the relationship of Y with the explanatory variables. We can use the added variable plots to detect discordant values. With this arsenal of graphical tools, we are ready to take up the construction of the logistic regression model for Data Set 2.

8.4.2 Multiple Logistic Regression Model for Data Set 2

We can summarize what we have learned so far about Data Set 2 as follows. Blue oaks are found almost exclusively at elevations less than 1300 m. The fraction of sites without a blue oak declines steadily in the Sierra Nevada but actually increases slightly up to 1100 m in the Coast Range (Figure 7.11). The fraction of sites with a blue oak present in the Sierra Nevada tends to decline steadily with increasing precipitation, while in the Coast Range the decline is more abrupt at between 700 and 900 mm per year (Figure 7.12a). In general, precipitation is more important as a determinant of blue oak presence in the Sierra Nevada than in the Coast Range because the Coast Range is drier over most of its extent. Temperature, and, to a lesser extent, soil properties also may play an important role in blue oak presence. The fraction of blue oak sites increases in both mountain ranges in areas with increasingly fine soil texture (Figure 7.12b). As stated in Chapter 7, this goes against conventional wisdom (McDonald, 1990). Sometimes it is wise to not include in a regression analysis data that violate the conventional wisdom, and indeed our application of the method of Henderson and Velleman (1981) is based on using what might be termed the conventional wisdom. In the present case, however, the conventional wisdom is based primarily on reasoning rather than data and may be incorrect. Finally, there is the "secondary" spatial variable *CoastDist*. In consideration of this quantity, we again have to deal with the fact that we are attempting to use regression for a purpose, explanation, for which it is not really suited. The quantity

CoastDist may have great utility for the purpose for which regression is best suited, namely, prediction. In Section 13.1, we discuss the various ways that spatial variables like this can enter into a model, but for now the important point is that they can affect the bias or lack thereof in the coefficients of the other quantities (in this case, the climate variables) if they are correlated with something that is not included in the model. For example, distance from the coast might be correlated with days per year of fog, which is not in the model. If days of fog has a strong influence, then its effect will be "loaded" onto the other variables, possibly biasing their coefficients. If distance from the coast is highly correlated with days per year of fog, then the former may take most of the load of the latter. It is worth noting that one might expect distance from the coast (i.e., from the ocean) to have a greater influence in the Coast Range than in the Sierra Nevada, both because of the closer proximity and because of the mitigating effects of the Central Valley, which lies between these mountain ranges.

Because of the high level of correlation among the temperature variables (Figure 7.9), we include for now only the summary variable *MAT* (mean annual temperature) among the temperature variables. We represent each parent material type with an indicator variable that simply indicates the presence of that parent material type. The Sierra Nevada subset of Data Set 2 is loaded into the spatial features object `data.Set2S.sf` as described in Appendix B.2 and Section 7.3.

```
> data.Set2S.glm <- with(data.Set2S.sf@data, data.frame(MAT, TempR,
+    Precip, PE, ET, Texture, AWCAvg, Permeab, SolRad6, SolRad12,
+    SolRad, CoastDist, QUDO))
> data.Set2S.glm$PM100 <- as.numeric(data.Set2.sfSierra$PM100 > 0)
```

The last line of code is repeated for each of the other five parent material types.

For illustrative purposes, in this analysis, unlike that of Section 8.3, we will build the model by adding explanatory variables and examining the effect of each addition (i.e., using forward selection). In practice, as was mentioned in that section, one should in an actual analysis program repeat the analysis using both selection methods as well as bidirectional selection and best subsets and compare the results. As a standard of comparison, we first create the global model that includes all of the explanatory variables.

```
> model.glmSfull <- glm(QUDO ~ ., data = data.Set2S.glm,
+   family = binomial)
```

In data sets with a very large number of observations, it is more likely that relatively unimportant explanatory variables will be assigned significance. At the end of our model selection process, we are going to compare models with different numbers of parameters. Since the BIC has a higher complexity penalty for larger data sets (Table 8.1), we will use this to compare these alternative models. According to the Help file for the R function `AIC()`, the BIC is computed using the following second argument.

```
> AIC(model.glmSfull, k = log(nrow(data.Set2Sglm)))
[1] 1134.27
```

To begin, we will create the null model as well as a formula that allows us to use the function `add1()` to examine the effects of adding different explanatory variables.

```
> model.formula <- as.formula("QUDO ~ MAT + TempR + Precip + PE + ET +
+   Texture + AWCAvg + Permeab + SolRad6 + SolRad12 + SolRad +
```

```
+   PM100 + PM200 + PM300 + PM400 + PM500 + PM600 + CoastDist")
> model.glmS0 <- glm(QUDO ~ 1, data = data.Set2S.glm, family =
+   binomial)
```

Now we are ready to start constructing the model. When we use the function `add1()`, we are comparing models of the same complexity (i.e., the same number of parameters). Therefore, only the bias term is important, and for this comparison we can use the AIC instead of the BIC. Nicholls (1989) recommends a fairly rigid procedure of variable addition based on comparing AIC values. I personally prefer the more flexible approach of Henderson and Velleman (1981), but we will nevertheless be strongly guided by the AIC results. Given all we have seen in Chapter 7, we would expect that the most important single variable would be *Precip*, but let's see what the AIC tells us.

```
> a1 <- add1(model.glmS0, model.formula)
> a1[order(a1[,3]),]
          Df Deviance    AIC
Precip     1   1222.1 1226.1
MAT        1   1404.0 1408.0
PE         1   1593.9 1597.9
CoastDist  1   1610.8 1614.8
   *   *   *   DELETED   *   *   *
SolRad12   1   2363.5 2367.5
<none>         2371.2 2373.2
PM100      1   2370.9 2374.9
```

In this case, the output gives us the increase in AIC that would result from the addition of each variable, so we want to add the term that increases the AIC the least. The variable *Precip* does indeed provide the smallest AIC, as well as making the most sense biophysically, so we bring it into the model.

```
> model.glmS1 <- update(model.glmS0,
+    formula = as.formula("QUDO ~ Precip"))
```

Parent material and *TempR* appear unimportant, so we will remove them.

```
> model.formula2 <- as.formula("QUDO ~ MAT + Precip + PE + ET +
+   Texture + AWCAvg + Permeab + SolRad6 + SolRad12 + SolRad +
+   CoastDist")
> a1 <- add1(model.glmS1, model.formula2)
> a1[order(a1[,3]),]
Single term additions
Model:
QUDO ~ Precip
         Df Deviance    AIC
MAT       1   1158.3 1164.3
ET        1   1174.1 1180.1
PE        1   1188.8 1194.8
   *   *   *   DELETED   *   *   *
```

As expected, *CoastDist* does not have a strong effect in the Sierra Nevada.

The second application of `add1()` indicates that *MAT* should enter, and this also makes sense based on what we have seen already. The variables *MAT* and *Precip* have a strong linear association, with a negative correlation coefficient.

```
> with(data.Set2S.glm, cor(Precip, MAT))
[1] -0.7488103
```

As elevation increases in the Sierra Nevada, it becomes cooler and wetter. Given this correlation, it is a bit surprising that the coefficient of *Precip* does not change too much when *MAT* is brought into the model.

```
> model.glmS1 <- update(model.glmS0,
+    formula = as.formula("QUDO ~ Precip"))
summary(model.glmS1)
Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)  8.0394180  0.3952460   20.34   < 2e-16 ***
Precip      -0.0097772  0.0004698  -20.81   < 2e-16 ***
AIC: 1226.1
> model.glmS2 <- update(model.glmS0,
+    formula = as.formula("QUDO ~ Precip + MAT"))
> summary(model.glmS2)
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -1.3701320  1.2715044  -1.078    0.281
Precip      -0.0071227  0.0005492 -12.970  < 2e-16 ***
MAT          0.5048807  0.0677577   7.451 9.25e-14 ***
AIC: 1164.3
```

Added variable and partial residual plots are not shown, but the latter indicate the possibility that the addition of a second-degree term in *Precip* would improve the model. We will generally save the analysis of higher-order terms and interactions until after the selection of first-order explanatory variables. At this point, however, we can observe something interesting. We first add a $Precip^2$ term to the model without *MAT*, and a test indicates that its coefficient is significant.

```
> model.glmS1sq <- update(model.glmS0,
+    formula = as.formula("QUDO ~ Precip + I(Precip^2)"))
> summary(model.glmS1sq)
Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)  1.064e+01  1.309e+00   8.135 4.13e-16 ***
Precip      -1.599e-02  2.945e-03  -5.431 5.61e-08 ***
I(Precip^2)  3.584e-06  1.633e-06   2.194   0.0282 *
```

The coefficient of *Precip* is negative, as expected, and the coefficient of $Precip^2$ is positive, indicating that the effect of *Precip* declines as its value increases. As mentioned in Section 3, Burnham and Anderson (1998) provide several convincing arguments and good examples for the assertion that hypothesis tests should not be used for variable selection. Nevertheless, I have found that while this is true for variable selection per se, these tests often can be useful in determining whether or not to include higher order terms or interactions. In this case, there is a significant ($p < 0.05$) difference between the model without and with $Precip^2$.

```
> anova(model.glmS1, model.glmS1sq, test = "Chisq")
Analysis of Deviance Table
Model 1: QUDO ~ Precip
Model 2: QUDO ~ Precip + I(Precip^2)
```

```
  Resid. Df Resid. Dev Df Deviance P(>|Chi|)
1      1766     1222.1
2      1765     1218.0  1   4.0801   0.04339 *
```

Let's now add *MAT* to the model that includes $Precip^2$ and see what happens.

```
> model.glmS2sq <- update(model.glmS0,
+    formula = as.formula("QUDO ~ Precip + I(Precip^2) + MAT"))
> coef(model.glmS2sq)
  (Intercept)        Precip   I(Precip^2)           MAT
 6.942303e-01 -1.167481e-02  2.568349e-06  4.964979e-01
> anova(model.glmS2, model.glmS2sq, test = "Chisq")
Analysis of Deviance Table
Model 1: QUDO ~ Precip + MAT
Model 2: QUDO ~ Precip + I(Precip^2) + MAT
  Resid. Df Resid. Dev Df Deviance P(>|Chi|)
1      1765     1158.3
2      1764     1155.9  1   2.3164    0.1280
```

The variable *MAT* apparently takes some of the "load" from $Precip^2$. There are a number of possible biophysical interpretations of this. Let's take it as a given that precipitation actually does have an effect on blue oak presence. Since *Precip* and *MAT* are negatively correlated, it is possible that temperature also has an effect, with increased temperatures reducing the probability of blue oak presence. In the alternative, the effect of *Precip* really could decline with increasing value, in which case *MAT* is serving as a surrogate for $Precip^2$. There may also be the possibility of an interaction. In any case, this is further indication that temperature as well as precipitation may play a role in blue oak presence.

We will for now only include the linear terms. We now have models with different numbers of parameters. Using the BIC to compare model 2, which includes *MAT*, with model 1, which does not, indicates a substantial improvement using model 2.

```
> AIC(model.glmS1, k = log(nrow(data.Set2S.glm)))
[1] 1237.074
> AIC(model.glmS2, k = log(nrow(data.Set2S.glm)))
[1] 1180.691
```

Figure 8.10a shows a thematic map of the predicted values of `model.glmS1`, which only includes *Precip*. Figure 8.10b shows the effect of adding *MAT* to the model. Both of these can be compared with Figure 7.8a, which shows blue oak presence and absence. Mean annual temperature in the northern Sierra foothills tends to rise as one moves north, due to increased distance from the cooling breezes of the Sacramento–San Joaquin Delta. Thus, the primary effect of adding *MAT* to the model is to slightly increase the presence probabilities in the northern foothills relative to the rest of the region.

Continuing on in this way, we come to candidate models that also include *PE*, *SolRad*, *AWCAvg*, and *Permeab*. Thus, our analysis has identified candidate models that include precipitation, some temperature-related variable or variables, possibly a solar radiation variable, and some variable that has to do with soil water availability.

The last step is to consider *Elevation*. While it is true that many of the explanatory variables in the model are affected by elevation, it is also true that elevation may also affect

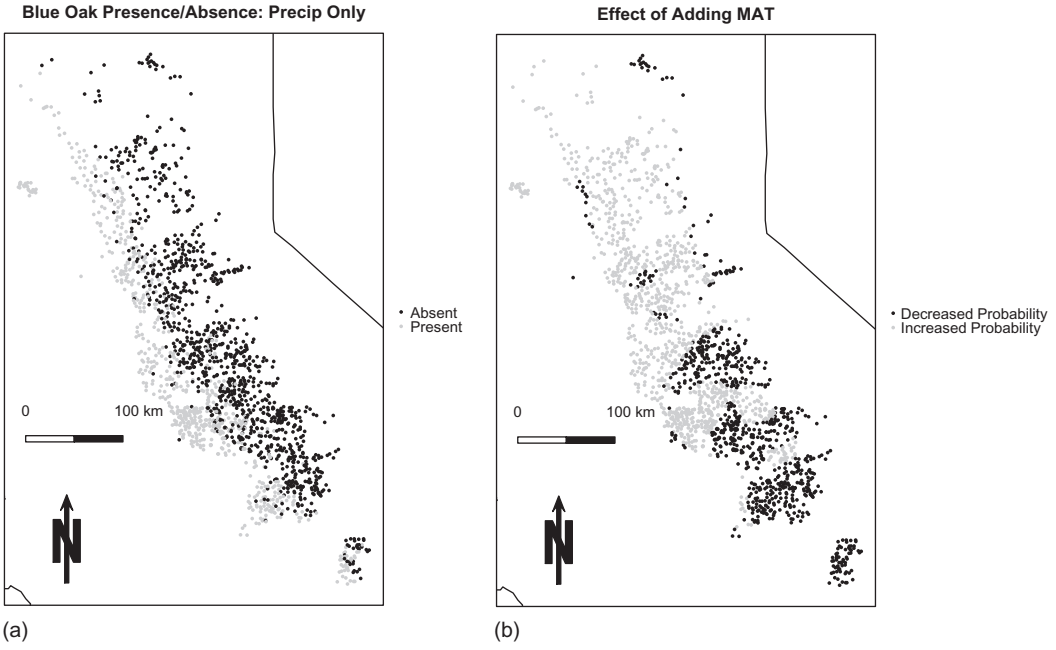

FIGURE 8.10
(a) Thematic map of blue oak presence and absence. (b) Thematic map showing the difference in predicted probability of blue oak occurrence between logistic regression models incorporating and not incorporating mean annual temperature.

other quantities not included in the data set that also impact blue oak presence. Therefore, we will test a model that incorporates representatives of all of the variables in our candidate models plus *Elevation*.

```
> data.Set2SglmE <- data.frame(cbind(data.Set2S.glm,
+    Elevation = data.Set2S$Elevation))
> model.glmS5E <- glm(QUDO ~ Precip + MAT + SolRad +
+    AWCAvg + Permeab + Elevation, data = data.Set2SglmE,
+    family = binomial)
> AIC(model.glmS5, k = log(nrow(data.Set2Sglm)))
[1] 1104.036
> AIC(model.glmS5E, k = log(nrow(data.Set2Sglm)))
[1] 1076.657
> summary(model.glmS5E)
Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -1.070e+00  1.760e+00  -0.608  0.54300
Precip      -5.531e-03  6.187e-04  -8.939  < 2e-16 ***
MAT          2.393e-01  9.251e-02   2.587  0.00968 **
SolRad       7.684e-04  9.652e-05   7.961 1.70e-15 ***
AWCAvg      -6.683e-03  1.458e-03  -4.585 4.53e-06 ***
Permeab     -3.022e-01  6.642e-02  -4.551 5.35e-06 ***
Elevation   -2.511e-03  4.269e-04  -5.880 4.09e-09 ***
```

The model that includes *Elevation* is a substantial improvement over the model that excludes it. We can then use `drop1()` to determine which variables are removed if *Elevation* is included.

```
> print(d1 <- drop1(model.glmS5E))
Single term deletions
Model:
QUDO ~ Precip + MAT + SolRad + AWCAvg + Permeab + Elevation
          Df Deviance    AIC
<none>         1024.3 1038.3
Precip     1   1129.6 1141.6
MAT        1   1031.3 1043.3
SolRad     1   1096.2 1108.2
AWCAvg     1   1046.1 1058.1
Permeab    1   1045.7 1057.7
Elevation  1   1059.2 1071.2
```

The indication is that none should be dropped, although, surprisingly, *Precip* and *MAT* are the closest. This may, again, be due to the effect of multicollinearity.

The Coast Range data display less correlation among the temperature variables (Figure 7.9c). A similar model development process for the Coast Range (Exercise 8.12) produces the following model. Once again, the high intercorrelation among variables means that this actually represents several candidate models.

```
> summary(model.glmC6)
Call:
glm(formula = QUDO ~ TempR + Permeab + Precip + GS32 + PE + SolRad6,
    family = binomial, data = data.Set2Cglm)
Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -7.812171   1.525324  -5.122 3.03e-07 ***
TempR        0.295183   0.039035   7.562 3.97e-14 ***
Permeab     -0.737872   0.077277  -9.548  < 2e-16 ***
Precip      -0.001528   0.000334  -4.577 4.72e-06 ***
GS32        -0.021310   0.002750  -7.749 9.23e-15 ***
PE           0.009384   0.001261   7.443 9.83e-14 ***
SolRad6      0.179429   0.040775   4.400 1.08e-05 ***
```

Given the complexity of the models and their preliminary nature, we will postpone any attempts to interpret the results.

The logistic regression model for blue oak presence constructed by Evett (1994) applied to the entire data set; that is, it did not separate the regions into different ranges. The explanatory variables in Evett's model includes *JaMean*, $JaMean^2$, *Precip*, *MAT*, *PM100*, *PM200*, *PM300*, *SolRad6*, $SolRad6^2$, $SolRad6^3$, *Texture*, $(JaMmean \times MAT)^2$, and *JuMax*. Vayssières et al. (2000) developed a generalized linear model based on 2,085 data records covering all of the ranges. Their explanatory variable set included both exogenous and endogenous variables (i.e., they included *Elevation* and *CoastDist* where we did not). Their model includes *JaMean*, $JaMean^2$, *Precip*, *MAT*, MAT^2, *PM100*, *PM400*, *PM500*, *PM600*, *SolRad6*, *Elevation*, $Elevation^2$, *JuMean*, *Texture*, and *ET*. Even given that we have not yet made an effort to include interactions or higher-order terms, there is obviously a considerable range of explanatory variables between the models.

8.4.3 Logistic Regression Model of Count Data for Data Set 1

In Section 7.2, we carried out a preliminary analysis of Data Set 1 as a test of the California Wildlife Habitat Relationships (CWHR) model for habitat suitability for the yellow-billed cuckoo. The model is expressed in terms of *scores* for four explanatory variables given in Table 7.1. The variables *PatchWidth* and *PatchArea* are self-explanatory. The height ratio (*HtRatio*) is the ratio of area with tall trees to total area. The *AgeRatio* is the ratio of low floodplain age (<60 years) to total area. This is a surrogate for the cover fraction of mixed cottonwood/willow. A fifth variable, cover class, is also included in the data set but does not appear to play a major role in determining habitat suitability.

The habitat scores are computed by first computing an individual patch score for each variable according to Table 7.1. The scores in Table 7.1 are non-monotonic (i.e., not strictly increasing or strictly decreasing) in both *HtRatio* and *AgeRatio*, and a regression model for these data will have to take this non-monotonicity into account. The model studied in Section 7.2 was expressed as a contingency table in which the overall habitat score of each patch was computed as the product of the individual values of each of the four variable scores. Any patch with a nonzero habitat score was judged to be suitable. Thus, in this model, any patch is unsuitable if it is unsuitable in any one of the four suitability criteria. The contingency table for this model is as follows (see Section 7.2).

```
> print(cont.table <- matrix(c(length(SP),length(SA),
+    length(UP),length(UA)), nrow = 2, byrow = TRUE,
+    dimnames = list(c("Suit.", "Unsuit."),c("Pres.", "Abs."))))
         Pres. Abs.
Suit.        5    1
Unsuit.      2   12
```

The results of Exercise 7.6 indicate that the same contingency table can be obtained with a model including only *AreaScore* and *AgeScore*, but that this table is not obtained with any one single explanatory variable.

In this section, we will apply GLM analysis to this data set. We will start with a logistic regression model for presence/absence. Before we begin the regression modeling, it is worthwhile to try to determine whether we really have any chance of distinguishing between the effects of the variable *PatchWidth* and *PatchArea*. We saw in Section 7.2 that these are highly correlated, and the only hope of distinguishing their effects would be if there is some really long, narrow patch that has a low patch width but a high patch area. Figure 8.11 is a plot of scaled values of *PathWidth* vs. *PatchArea*. Except for the smallest areas, which are all unsuitable, there is a virtually linear relationship between *PatchArea* and *PatchWidth*. The two exceptions are Patch 7 and Patch 16. We will use the data frame `Set1.obs3` created in Section 7.2. We again delete Patch 191, which has incomplete geographic coverage. Examining the patch habitat scores of the patches further gives us the following.

```
> d.f <- with(Set1.obs3[-which(Set1.obs3$PatchID == 191),],
+    data.frame(obsID, AreaScore, WidthScore, AgeScore, HeightScore,
+    PresAbs))
> d.f[order(d.f$obsID),]
   obsID AreaScore WidthScore AgeScore HeightScore PresAbs
16     2      0.33       0.33     0.00        0.00       0
15     3      0.00       0.33     0.00        0.00       0
```

14	4	0.33	0.33	0.66	1.00	1
13	5	1.00	0.66	0.66	0.66	1
12	6	0.33	0.33	0.00	0.33	0
11	7	0.66	0.66	0.66	0.66	0
10	8	1.00	0.66	0.00	0.00	0
19	9	0.00	0.00	0.00	0.33	0
9	10	0.00	0.33	0.33	0.33	0
8	11	0.66	0.33	0.00	0.00	1
7	12	0.00	0.33	0.00	0.66	0
6	13	0.00	0.33	0.00	0.66	0
20	14	0.66	0.33	0.00	0.66	0
18	15	0.00	0.00	0.33	0.33	0
4	16	0.66	0.66	0.00	0.66	0
5	17	1.00	0.66	0.33	0.66	1
17	18	0.00	0.00	0.33	0.33	1
3	19	0.33	0.33	0.33	0.33	1
2	20	0.00	0.00	0.00	0.00	0
1	21	1.00	0.66	0.33	1.00	1

Patch 16 is unsuitable because of its age score. There are four patches (*obsID* values 3, 10, 12, 13) that have a positive *WidthScore* and a zero *AreaScore*, and none that have the opposite combination. These patches all have about the same width as the patch with *obsID* value 19, which has a slightly larger area (Figure 8.11). The one small area patch (*obsID* = 18) that has a cuckoo present is accounted for in Exercise 7.5. There seems to be no way that we can separate the effects of patch width from those of patch area. Therefore, we will eliminate one of them from the logistic regression. We tentatively choose *PatchWidth* for elimination because, based on Figure 8.11, *PatchArea* seems to be a better predictor of presence vs. absence. Just to be sure, however, we will double check below.

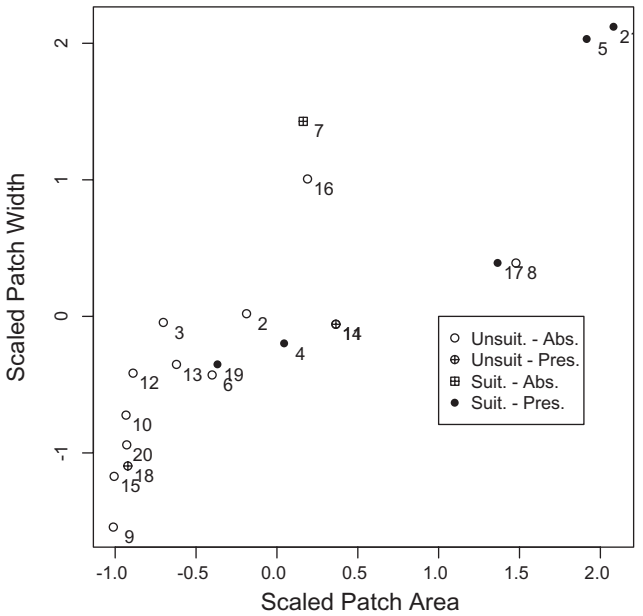

FIGURE 8.11
Scatterplot of patch area vs. patch width for the data of Data Set 1.

To improve the numerical properties of the model, we will center and scale the variables *PatchArea* and *PatchWidth*. The other two variables, *HtRatio* and *AgeRatio*, are fractions whose values range from zero to one, so we will leave these alone. We will not pursue analysis with the variable *CoverRatio.* Here is the code to create the new data set, based on the data frame `Set1.corrected` created in Section 7.2.

```
> Set1.norm1 <- with(Set1.corrected, data.frame(PresAbs = PresAbs,
+      PatchArea = scale(PatchArea), PatchWidth = scale(PatchWidth),
+      HtRatio = HtRatio, AgeRatio = AgeRatio))
```

We will start by generating the null model with nothing but the intercept on the right-hand side.

```
> Set1.logmodel0 <- glm(PresAbs ~ 1, data = Set1.norm1,
+    family = binomial)
```

Next, we will incorporate *PatchArea* and compare this to the null model. Only a portion of the output is shown in the following listings.

```
> Set1.logArea <- update(Set1.logmodel0,
+   formula = as.formula("PresAbs ~ PatchArea"))
> AIC(Set1.logArea)
[1] 25.14631
> anova(Set1.logmodel0, Set1.logArea, test = "Chisq")
Analysis of Deviance Table
Model 1: PresAbs ~ 1
Model 2: PresAbs ~ PatchArea
  Resid. Df Resid. Dev Df Deviance P(>|Chi|)
1        19     25.898
2        18     21.146  1   4.7516   0.02927 *
```

There is indeed a declared significant difference. Just to check that *PatchWidth* does not add to the model, we will add it and compare the two.

```
> Set1.logAreaWidth <- update(Set1.logmodel0,
+   formula = as.formula("PresAbs ~ PatchArea + PatchWidth"))
> AIC(Set1.logAreaWidth)
[1] 26.44410
> anova(Set1.logArea,Set1.logAreaWidth, test = "Chisq")
Analysis of Deviance Table
Model 1: PresAbs ~ PatchArea
Model 2: PresAbs ~ PatchArea + PatchWidth
  Resid. Df Resid. Dev Df Deviance P(>|Chi|)
1        18     21.146
2        17     20.444  1   0.7022    0.4020
```

We cannot use `anova()` to compare a model that only includes *PatchArea* with one that only includes *PatchWidth* because they are not nested. We can, however, use the AIC for such a comparison.

```
> Set1.logWidth <- update(Set1.logmodel0,
+   formula = as.formula("PresAbs ~ PatchWidth"))
> AIC(Set1.logWidth)
[1] 28.03604
```

The AIC of the model including *PatchArea* is substantially lower than that of the model including *PatchWidth*, providing further evidence that we should keep *PatchArea* rather than *PatchWidth*.

The other variable besides *PatchArea* that played a prominent role in the contingency table analysis of Section 7.2 is *AgeRatio*, the ratio of area of the patch floodplain age less than 60 years to total patch area. Figure 7.6 shows a plot of the habitat suitability score as a function of *AgeRatio*. One way to approximate this relationship is with a parabolic function. In Exercise 8.13, you are asked to carry out this analysis. In our analysis, instead of using the raw *AgeRatio* as the second variable in the logistic model, we will use the age suitability score.

The fact that the suitability scores in the contingency table model of Section 7.2 enter into the model as a product rather than a sum is probably best represented in a logistic regression model via an interaction term. This leads us to test a model that includes *PatchArea*, *AgeScore*, and the interaction between the two. We start by introducing *AgeScore* by itself

```
> Set1.logAgeScore <- update(Set1.logmodel0,
+   formula = as.formula("PresAbs ~ AgeScore"))
> summary(Set1.logAgeScore)
Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  -1.8153     0.8123  -2.235   0.0254 *
AgeScore      5.3393     2.5237   2.116   0.0344 *
> anova(Set1.logmodel0, Set1.logAgeScore, test = "Chisq")
Analysis of Deviance Table
Model 1: PresAbs ~ 1
Model 2: PresAbs ~ AgeScore
  Resid. Df Resid. Dev Df Deviance P(>|Chi|)
1        19     25.898
2        18     19.793  1   6.1045   0.01348 *
```

The variable *AgeScore* plays a role analogous to that of *PatchArea*. Now let's try combining the two, in two steps. First, we test *AgeScore* by itself.

```
> Set1.logAgeScore <- update(Set1.logmodel0,
+   formula = as.formula("PresAbs ~ AgeScore"))
> AIC(Set1.logAgeScore)
[1] 23.79338
```

Interestingly, this produces a still lower AIC. Next, we add *PatchArea* but omit the interaction.

```
> Set1.logAreaAge <- update(Set1.logmodel0,
+   formula = as.formula("PresAbs ~ PatchArea + AgeScore"))
> summary(Set1.logAreaAge)
Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  -1.7308     0.8444  -2.050   0.0404 *
PatchArea     0.9876     0.6570   1.503   0.1328
AgeScore      4.8061     2.7064   1.776   0.0758 .
AIC: 23.086
> anova(Set1.logAgeScore, Set1.logAreaAge, test = "Chisq")
Analysis of Deviance Table
Model 1: PresAbs ~ AgeScore
Model 2: PresAbs ~ PatchArea + AgeScore
```

```
  Resid. Df Resid. Dev Df Deviance P(>|Chi|)
1        18     19.793
2        17     17.086  1   2.7069   0.09991 .
```

Both *PatchArea* and *AgeScore* enter in a fairly meaningful way when combined additively, and the model that includes both is somewhat better than the one that only includes *AgeScore*. Remember that the fact that the *p* value of *AgeScore* is slightly lower than that of *PatchArea* should not be interpreted as necessarily meaning that the former is somehow more important than the latter The evidence, however, is beginning to point in that direction. Incorporating the interaction produces the following.

```
> Set1.logAreaAgeInt <- update(Set1.logmodel0,
+   formula = as.formula("PresAbs ~ PatchArea + AgeScore +
+   I(PatchArea * AgeScore)"))
> summary(Set1.logAreaAgeInt)
Coefficients:
                        Estimate Std. Error z value Pr(>|z|)
(Intercept)              -1.7247     0.8553  -2.017   0.0437 *
PatchArea                 0.9555     1.0612   0.900   0.3679
AgeScore                  4.7816     2.7771   1.722   0.0851 .
I(PatchArea * AgeScore)   0.1403     3.6523   0.038   0.9694
AIC: 25.085
> anova(Set1.logAreaAge, Set1.logAreaAgeInt, test = "Chisq")
Model 1: PresAbs ~ PatchArea + AgeScore
Model 2: PresAbs ~ PatchArea + AgeScore + I(PatchArea * AgeScore)
  Resid. Df Resid. Dev Df  Deviance P(>|Chi|)
1        17     17.086
2        16     17.085  1 0.0014860    0.9693
```

This model has a higher AIC than the one without the interaction, and the likelihood ratio test could not be less significant.

In summary, the use of logistic regression as an exploratory tool provides further support for the speculation that patch area and patch age structure, which is a surrogate for species composition, play the most important roles in determining habitat suitability in this stretch of the Sacramento River. In the next sub-section, we use the rather dubious abundance data as a means to introduce the analysis of the zero-inflated Poisson regression model for count data.

8.4.4 Analysis of the Counts of Data Set 1: Zero-Inflated Poisson Data

When the data set of cuckoo responses was created, along with the fact of a bird response to the *kowlp* call, the number of such responses was also recorded. Among our four data sets, this represents the best example of count data, which is often important in ecological studies. It also represents the worst example of reliability in a data set. To recapitulate the problems discussed in Section 7.2, it is impossible to determine whether multiple responses signify multiple observations of the same bird or multiple observations of different birds. Moreover, since the bird count data is an extensive property, it is sensitive to patch area, but patch area is itself one of the explanatory variables so we cannot adjust for it, and moreover it would not make sense to do so. Despite all this, we will press on in this section with an analysis of the abundance data. We do this for two reasons. First, the data provide the opportunity to discuss the analysis of count data. Second, weak as they might be, the data may make some contribution to our understanding of the ecological system.

As in Section 8.4.3, the data are loaded and set up using code from Appendix B.1 and Section 7.2. Before beginning with the analysis, however, we have do decide what to do with the data record that places 135 cuckoos at the southernmost observation point (Figure 7.2d). After pondering this for a bit, I decided on a course of action that might raise a few eyebrows but that seems to be not totally inappropriate. This is to carry out a sort of Winsorization (Section 6.2.1).

If we compare the southernmost habitat patch, with 135 recorded observations, to the northernmost patch, with 16 observations (Figure 7.2a), we can see that they are somewhat similar. From what we know about the cuckoos' perspective so far, both patches seem like pretty good habitat. Both are large and have a reasonable mix of young and old areas. We excluded the northernmost patch because we could not measure the floodplain age over its full extent, but now we will take its response variable value, 16, and substitute it for the value 135 in the southernmost patch.

```
> Set1.norm2 <- with(Set1.corrected, data.frame(Abund = Abund,
+       PatchArea = scale(PatchArea), PatchWidth = scale(PatchWidth),
+       HtRatio = HtRatio, AgeRatio = AgeRatio, CoverRato = CoverRatio,
+       AgeScore = AgeScore))
> mean(Set1.norm2$Abund)
[1] 7.65
> Set1.norm3 <- Set1.norm2
> Set1.norm3$Abund[1] <- 16
> mean(Set1.norm3$Abund)
[1] 1.7
```

The resulting data are contained in the data frame `Set1.norm3`. This actually produces a reasonable distribution of values. Figure 8.12 is a plot of patch abundance vs. patch area for the data set. Positive vs. zero age ratio suitability scores are indicated by filled vs. open

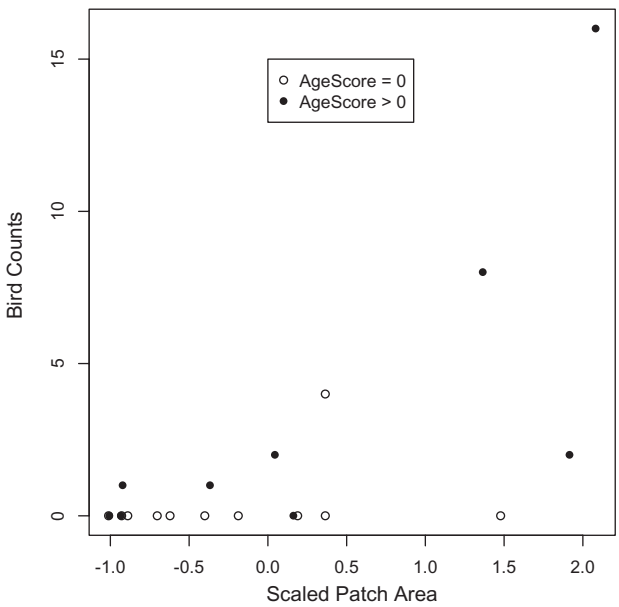

FIGURE 8.12
Scatterplot of abundance vs. scaled patch area of Data Set 1.

circles. The figure leads us to expect that, once again, we will see a regression involving *PatchArea* and *AgeScore*.

The standard method for constructing a regression model for count data is to use Poisson regression (Kutner et al., 2005, p. 618). This is carried out in exactly the same manner as logistic regression, but with a different link function. Instead of the link function given in Equation 8.22 for logistic regression, Poisson regression employs a link function of the form

$$g(Y_i) = \ln Y_i = \beta_0 + \beta_1 X_{i1} + \beta_2 X_{i2} + \ldots + \beta_{p-1} X_{i,p-1}. \tag{8.34}$$

There are, however, two complications. Recall (Larsen and Marx, 1986, p. 187) that the variance of the Poisson distribution equals its mean. The mean of the Winsorized distribution of cuckoo abundances is given above as 1.7.

Figure 8.13 shows a bar plot of the number of observed cuckoo abundance values compared with those predicted for a Poisson distribution with this mean. One feature that strikes the eye is that there are far more counts with zero observations than expected. This is very common with ecological count data. A data set such as this that has more zeroes than expected is said to be *zero-inflated*. A second feature that strikes the eye is that there seems to be a greater spread to the actual data than to the expected count numbers of the Poisson distribution. This is confirmed by a calculation of the sample variance.

```
> var(Set1.norm3$Abund)
[1] 15.16842
```

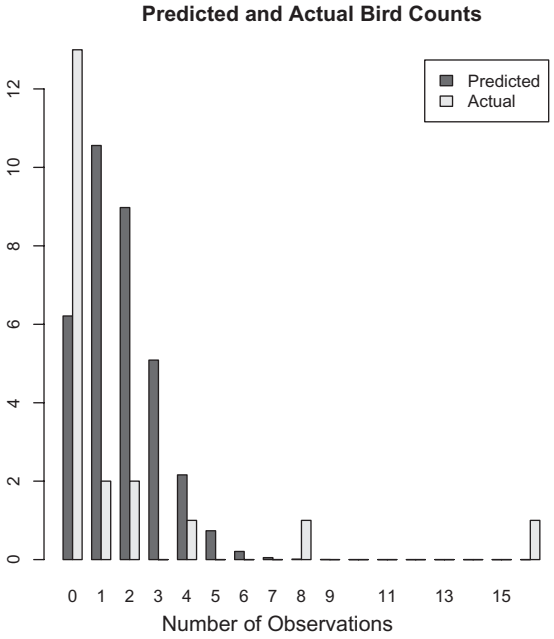

FIGURE 8.13
Bar chart of distribution of bird counts vs. that predicted by a Poisson distribution.

This large variance, called *overdispersion*, is also a common feature of ecological data (Bolker, 2008).

There are a few ways to construct regression models for overdispersed, zero-inflated count data (Cameron and Trivedi, 1998, p. 125). We will only present one: the zero-inflated Poisson, or ZIP method (Lambert, 1992). Briefly, in this method one fits a mixture of two models, a model for the probability that a count will be equal to zero, and a second model for the value of the count when it is greater than zero. This can be expressed as (Gelman and Hill, 2007, p. 127)

$$Y_i = \begin{cases} 0 \ \ if\ S_i = 0 \\ overdispersed\ Poisson(\lambda, \beta)\ if\ \ S_i = 1 \end{cases}. \tag{8.35}$$

Here S_i is a random variable that determines whether Y_i is zero or nonzero, λ is the Poisson parameter, and β is an overdispersion parameter that is used to enlarge the variance. The value of S_i can then be modeled using logistic regression.

The R package `pscl` (Zeileis et al., 2008), among others, contains functions that can be used to construct zero-inflated overdispersed Poisson models in a completely analogous way to that in which the function `glm()` is used to construct standard generalized linear models. Here is a sequence using the function `zeroinfl()`, which fits the ZIP model.

```
> library(pscl)
> Set1.zimodel0 <- zeroinfl(Abund ~ 1, data = Set1.norm3)
> AIC(Set1.zimodel0)
[1] 81.99746
> Set1.zimodelArea <- zeroinfl(Abund ~ PatchArea, data = Set1.norm3)
> AIC(Set1.zimodelArea)
[1] 60.31453
> summary(Set1.zimodelAreaAge)
Call:
zeroinfl(formula = Abund ~ PatchArea + AgeScore + I(PatchArea *
AgeScore), data = Set1.norm3)
Pearson residuals:
     Min       1Q   Median       3Q      Max
-1.09855 -0.29880 -0.14634 0.06209  2.00401
Count model coefficients (poisson with log link):
                       Estimate Std. Error z value Pr(>|z|)
(Intercept)              0.6147     0.6411   0.959 0.337586
PatchArea                2.0875     0.6285   3.321 0.000896 ***
AgeScore                -0.7066     1.4721  -0.480 0.631261
I(PatchArea * AgeScore)  -2.7942     1.5872  -1.760 0.078332.
Zero-inflation model coefficients (binomial with logit link):
                       Estimate Std. Error  z value Pr(>|z|)
(Intercept)               1.803      1.290    1.397    0.162
PatchArea                -0.476      1.572   -0.303    0.762
AgeScore               -259.720   1521.219   -0.171    0.864
I(PatchArea * AgeScore) -267.380   1663.725   -0.161    0.872
```

The coefficient for *PatchArea* is highly significant, that for *AgeScore* is not significant, and that for the interaction is marginally significant. This is pretty much the opposite of the result obtained above using a logistic model for presence and absence. Some insight into this can be gained by constructing plots analogous to that of Figure 8.13 for cuckoo presence/absence, and also by placing *AgeScore* on the abscissa.

Thus far, our initial regression results for Data Set 1 are somewhat equivocal. As with the contingency table analysis of Section 7.2, they are consistent with the interpretation that for this data set, patch area and patch age structure (and hence possibly vegetation composition) play the most important role in determining habitat suitability, and that the interaction between these variables may be important.

8.5 Further Reading

If one wishes to carry out a purely statistical approach to model selection (i.e., one that is based more on the statistical properties of the data than on their scientific interpretation), then one must read a good book on model selection. Some of the best are Fox (1997), Burnham and Anderson (1998), and Harrell (2001). Some authors tend to be a bit zealous in their belief that one or another method is the best. Hastie et al. (2009) provide an even-handed comparison of the methods. Mosteller and Tukey (1977, Ch 13) provide an excellent discussion (appropriately titled "Woes of Regression Coefficients") of the issues associated with interpreting regression coefficients generated from observational data. Venables (2000) provides additional insights into the use of linear models, both in regression and analysis of variance. Belseley et al. (1980) provide a comprehensive discussion of regression diagnostics that is still very relevant, despite its age.

An important class of methods for model fitting and variable selection not discussed here involves a process called *shrinkage*. These are particularly useful when the objective is an accurate prediction of the value of the response variable, as opposed to an improved understanding of process. Shrinkage methods work by accepting a small amount of bias in exchange for a substantial reduction in variance. The best known such method is ridge regression (Kutner et al., 2005, p. 431). Hastie et al. (2009) provide a good discussion. The R package `rms` (Harrell, 2011) implements several shrinkage methods. This package also contains many other tools for exploratory and confirmatory regression analysis. If you plan to do a lot of regression modeling, it would be a very good idea to learn to use it. Fox (1997) and Fox and Weisberg (2011) are very good sources for added variable and partial regression plots. Wang (1985) and Pregibon (1985) develop somewhat different approaches to added variable plots, and Hines et al. (1993) also present an alternative approach.

Kutner et al. (2005) and Fox (1997) provide excellent introductions to the generalized linear model. The papers of Austin et al. (1983, 1984) are seminal works on use of generalized linear models in vegetation ecology. Nicholls (1989) provides an expository treatment of this subject. Venables and Ripley (2002, Ch. 7) discuss generalized linear model approaches to regression when the response variable is ordinal. Bolker (2008) provides an excellent introduction to zero-inflated models and overdispersion. Cunningham and

Lindenmayer (2005) provide further discussion. It is important to note that just because a data set has too many zeroes does not necessarily mean that it is zero-inflated in the technical sense of the term. Warton (2005) provides a discussion of this issue.

Exercises

8.1 Compute added variable and partial regression plots of Field 4.1 with just *Clay* and *SoilK* and interpret them.

8.2 (a) Use the function `stepAIC()` from the `MASS` package to carry out a stepwise multiple regression (use `direction = "both"`) to select the "best" combination of explanatory variables to construct a linear regression model (use only first-order terms) for yield; (b) find five other linear regression models whose AIC is within 1% of the "best" model; (c) compare the predictions of these models in the context of the variables they contain. (Hint: the function `leaps()` from the `leaps` package is helpful for part b).

8.3 Create scatterplot maps of the agronomic variables for the northern and southern portions of Field 4.1.

8.4 Use the function `leaps()` to carry out a best subsets analysis of the agronomic data for the northern portion of Field 4.1.

8.5 Develop three candidate models for the relationship between *Yield* and the explanatory variables in the northern (first 62 locations) and southern (remaining locations) regions of Field 4.1.

8.6 Use the function `jitter()` to perturb the values of *Yield* by a small amount in the backward selection process of Section 8.3 and observe the effect if you blindly follow the recommendations of the function `drop1()`.

8.7 Let X be a vector of 40 elements, with $X_i = a_i + 0.1\varepsilon_i$, where $a_i = 0$ for $1 \le i \le 20$ and $a_i = 2$ for $21 \le i \le 40$, and let $Y_i = X_i + 0.1\varepsilon_i'$, with both ε_i and ε_i' being unit normal random variables. (a) Plot Y against X and compute the coefficients of the linear regression of Y on X; (b) Now suppose points 1 through 20 are in the southern half of a field and points 21 through 40 are in the northern half. Compute the regression coefficients of the two halves of the field separately; (c) What relevance does this example have to the analysis of Field 4.1?

8.8 On biophysical grounds one might expect there to be an interaction between *SoilP* and *SoilpH* in a model for *Yield* in Field 4.1. Determine whether such a model is empirically justified.

8.9 In this exercise, we examine the relationship of the endogenous variables to *Yield* and to exogenous variables. (a) Determine which of the two variables *LeafN* and *FLN* is most closely associated with *Yield*; (b) Develop a regression model for the variable that you found in part (a); (c) Repeat part (b) for *CropDens*.

8.10 Develop a regression model for grain protein in terms of the exogenous variables in Field 4.1.

8.11 (a) Construct a scatterplot matrix for Field 4.2; (b) Construct a linear regression model for Field 4.2 in terms of the exogenous data.

8.12 Carry out a regression analysis for the Coast Range subset of Data Set 2 using `glm()`.

8.13 Using Data Set 1, develop a logistic regression model of cuckoo presence/absence using a quadratic term in *AgeRatio* and interpret the results.

8.14 Using Data Set 1, develop logistic regression models of cuckoo presence/absence that includes height class, and determine their significance relative to models that do not include this variable.

8.15 Develop a Poisson regression model for cuckoo abundance in Data Set 1 and compare it with the zero-inflated model developed in Section 8.4.4.