16

Analysis of Data from Controlled Experiments

16.1 Introduction

At the landscape scale, controlled experiments are often logistically or financially difficult, and precision landscape measurement tools such as remotely sensed images and yield monitors may provide observational data that supports scientific interpretation. Nevertheless, if it is possible to carry out a controlled, replicated experiment under the same conditions as an observational study, then the controlled experiment provides more powerful results. Often the results of observational studies and controlled experiments can be used in a complementary manner to address a question from different perspectives. The controlled experiment in this ideal scenario provides precise, readily interpretable answers to specific questions, and the observational study permits those answers to be extended to more general conditions than those under which the controlled experiment is conducted. One must, however, confront the same issues in the presence of spatial autocorrelation in data from controlled experiments that one confronts with observational data. The purpose of this chapter is to discuss measures that can be taken in dealing with these issues.

The three fundamental concepts of the design of a controlled experiment, as put forward by Fisher (1935, pp. 17, 21, 48), are randomization, replication, and blocking. In discussing these concepts, we will refer to the experimental unit, that is, the component that is assigned the treatment and whose response is measured, as the *plot*, and we refer to the environment that defines the response of the plot to the treatment as the *substrate*. The notion of *randomness* is itself difficult to define (Kempthorne, 1952, p. 121), and we will adopt the practical definition given by Kempthorne in the just-cited reference that a process is random if it is based on a set of random numbers. This definition works for us because the use we need to make of randomness is to randomly assign treatments to experimental plots. In this context, the treatments are assigned at random if at the start of the assignment process each plot has the same probability of being assigned every treatment (Kutner et al., 2005, p. 653). This occurs if the treatments are assigned based on a set of random numbers. The primary purpose of randomization is to avoid bias. Even in the presence of autocorrelated variability of the substrate properties, the comparison of treatment effects is unbiased if the assignment of treatments to plots is random (Kempthorne, 1952, p. 140). The effect of modifications to the experimental design to take into account heterogeneity of the substrate of a randomized experiment is not to remove bias but rather to increase precision (Brownie et al., 1993).

The primary purpose of replication, that is, of assigning each treatment to more than one plot, is to decrease the error in treatment comparisons (Kempthorne, 1952, p. 177). That said, it is necessary to have at least two replications of a treatment to be able to estimate the experimental variance at all. A second, almost equally important effect of replication is to provide what Hurlbert (1984) calls *interspersion* of the plots. This means that the plots are scattered among each other on the substrate, thus reducing the chance that a group of plots that each receive the same treatment will be located near to each other.

The incorporation of blocking into the design of an experiment involving spatial data represents an explicit recognition that spatial autocorrelation is present in the substrate. The purpose of blocking is to reduce the effect of spatial heterogeneity on the measurement of the treatment effect by making the individual blocks as homogeneous as possible (Fisher, 1935, p. 56; Kempthorne, 1952, p. 210; Kutner et al., 2005, p. 661). Having defined these three fundamental concepts of experimental design, we now turn to a discussion of the fundamentals of the analysis of variance.

16.2 Classical Analysis of Variance

Both linear regression and analysis of variance (ANOVA) are particular cases of the general linear model $Y = X\beta + \varepsilon$. In linear regression, the explanatory variables are assumed to be interval or ratio scale, or at least to function as if they had one of these measurement scales, and the model assumes a relation between the response and the explanatory variables (e.g., $Y_i = \beta_0 + \beta_1 X_{i1} + \varepsilon_i$). In ANOVA the explanatory variables, or *treatments*, may be nominal, and in any case the model does not assume a particular relationship between the treatments and the response. For example, suppose we are conducting an experiment on the effects of a type of fertilizer on crop yield, and we are going to analyze the data using the linear model. If the explanatory variable is the amount of fertilizer applied, we would probably choose (at least initially) to analyze these data using linear regression. If the explanatory variables are different types of fertilizer, then ANOVA would probably be a more appropriate choice. We will assume that the reader has been exposed to ANOVA already and simply provide a brief review to set up the notation. References are provided in Section 16.5.

We will adopt a notation based on that of Brownie et al. (1993). We restrict ourselves to rectangular plot arrangements such as that shown in Figure 16.1. The rows of plots are indexed by i, $i = 1,\ldots,l$ and the columns are indexed by j, $j = 1,\ldots,m$. For example, in Figure 16.1 $l = 12$ and $m = 6$. The treatment applied to plot ij is denoted $\tau_{k(ij)}$, signifying that plot ij receives the kth treatment, $k = 1,\ldots,r$. Assume that there are n replications per plot (we will not deal with unbalanced designs), so that $lm = rn$. The simplest ANOVA model is the *completely randomized design*. It is expressed in the so-called *effects formulation* as

$$Y_{ij} = \mu + \tau_{k(ij)} + \varepsilon_{ij}, \tag{16.1}$$

where ε_{ij} is the error term. We assume that the ε_{ij} are normal random variables with mean 0 and variance σ^2.

The analysis of variance was developed by R.A. Fisher (1935) as a result of his experience with field plot research at the Rothamsted Experiment Station in Harpenden, England. Fisher recognized the first axiom of geography (Tobler, 1970, See Section 3.1), that nearby experimental plots would generally be more similar in their properties than more distant plots, and that this effect could reduce the statistical power of field experiments. It is important to reemphasize that if the assignment of treatments to plots is randomized (i.e., if each plot has at the start of the assignment of treatments an equal probability of

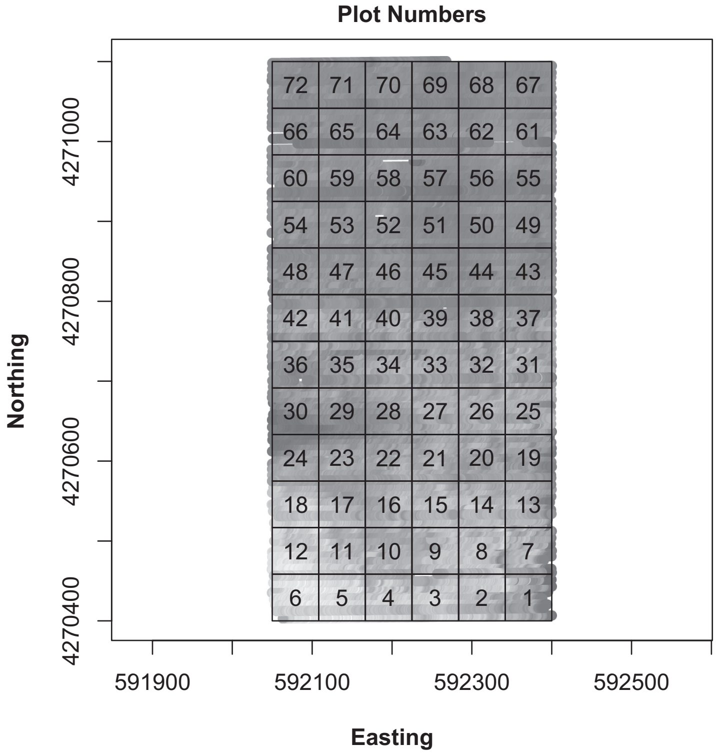

FIGURE 16.1
Plot numbers for a uniformity trial held on a rectangular subsection of Field 4.1.

being assigned any treatment), then the greater similarity of properties of nearby plots does not invalidate the results, but only reduces their power (Kempthorne, 1952, p. 135; Brownie et al., 1993). In order to alleviate this reduction in power, Fisher proposed the idea of *blocking*. The most common blocked design is the randomized complete block (RCB), which is represented as

$$Y_{ij,b} = \mu + \tau_{k(ij)} + \rho_b + \varepsilon_{ij}, \tag{16.2}$$

where all of the symbols are as before, and ρ_b is the effect of the *bth* block, to which plot *ij* belongs. In laying out an RCB experiment, each block should be as uniform as possible in the blocking factor, and encompass as much variability as possible of any other factors that might enter into the experiment, to avoid confounding these factors with the treatment.

To clarify these concepts, we will introduce the data set that will be used throughout this chapter to compare methods. It is the same data set that was used to discuss the modifiable areal unit problem in Section 11.5.1. The data are yield monitor output from Field 4.1 restricted to a 700 m by 350 m rectangular region. This region is divided into a checkerboard of 72 square plots (Figure 16.1). Our "experiment" is a *uniformity trial*, that is, an experiment in which there are plots but no applied treatments (Cochran, 1977, p. 41), or, perhaps better said, different plots are labeled as having different "treatments," but there is nothing purposefully done to distinguish these plots. Since the "treatment" effect is therefore zero, the difference between treatment means should ideally be approximately zero.

The checkerboard layout is not representative of a typical field experiment, but it is common in uniformity trials. One of the most famous agricultural data sets, and certainly one of the data sets most analyzed by statisticians, is that of Mercer and Hall (1911), which is a uniformity trial. A major advantage of the uniformity trial in terms of comparing alternative methods for dealing with spatial autocorrelation is that, because the treatments have zero effect, alternative randomization schemes can be applied in a Monte Carlo simulation study. This approach has been used by a number of authors, including Wilkinson et al. (1983), Besag and Kempton (1986), and Zimmerman and Harville (1991), in comparative analyses of methods for addressing spatial autocorrelation in experimental data. Indeed, Brownie et al. (1993) point out that if there are actual treatment effects then, because these effects are not known, it is impossible to make a completely valid comparison of the different methods.

In our example, we will have 18 treatments ($k = 1,...18$) and four blocks ($b = 1,...,4$). Figure 16.2 shows a grayscale plot of the yield values with a set of treatment numbers for a completely randomized design (CRD) experiment overlaid. Figure 16.3a shows the block layout of an RCB experiment, and Figure 16.3b shows a grayscale plot of the yield values with the RCB treatment numbers overlaid. Prior to carrying out a comparative analysis of different methods, let us examine the results of the RCB experiment and the CRD experiment in detail. The setup for this analysis follows what should by now be a familiar pattern, and not all of the code is shown. It is important to emphasize,

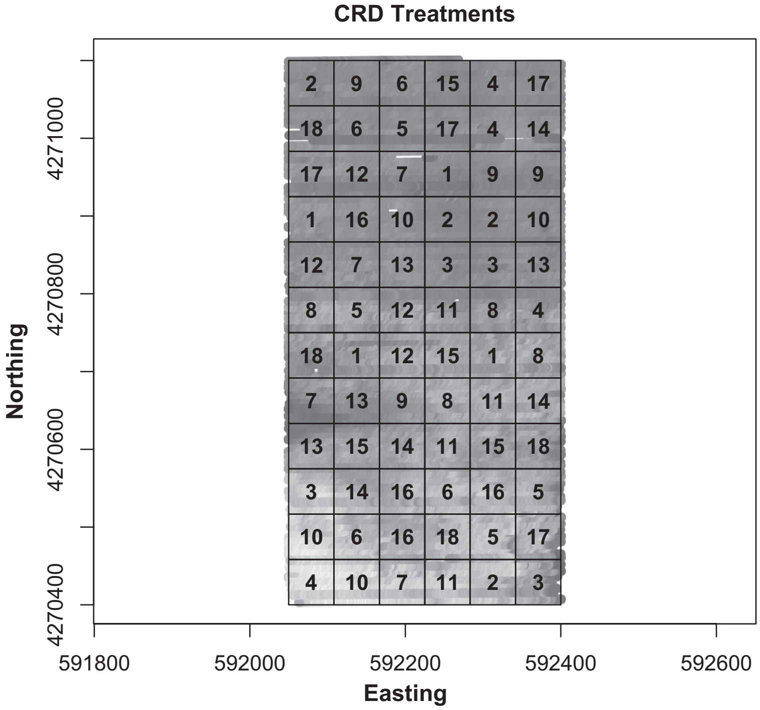

FIGURE 16.2
Treatment numbers for a completely randomized design (CRD) experiment carried out in the plots of Figure 4.1.

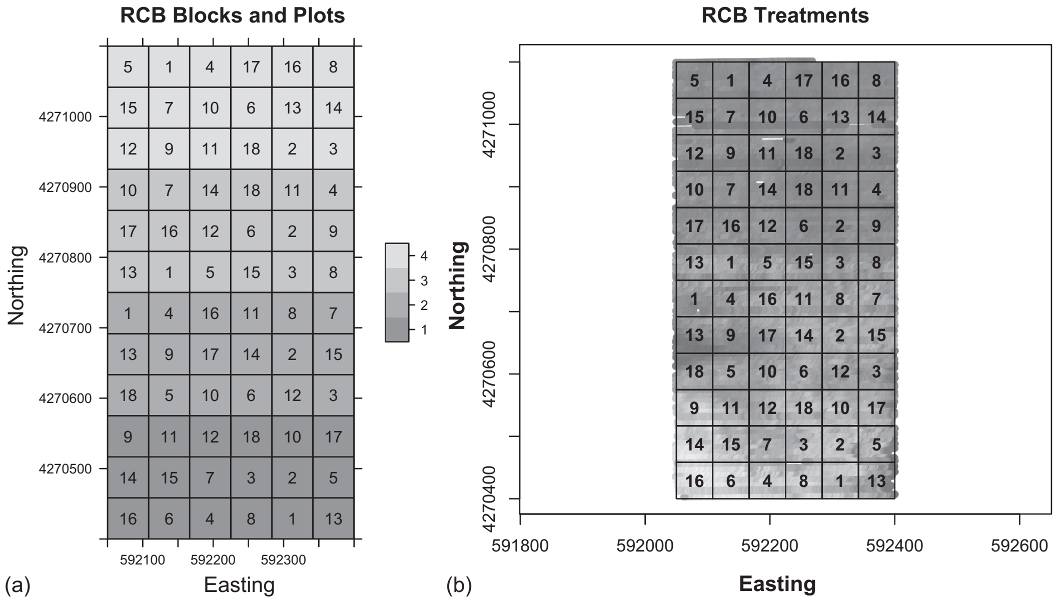

FIGURE 16.3
Layout of a randomized complete block (RCB) experiment carried out on the plots of Figure 4.1. (a) Gray scale showing blocks. (b) Gray scale showing yield.

however, that in order to obtain correct results, factors and not numbers must be used to identify the plots and blocks.

```
> set.seed(123)
> plots.spdf@data$trtmt.CRD <- factor(sample(rep((1:18), 4)))
> plots.spdf@data$block <- factor(sort(rep(1:4, 18)))
> plots.spdf@data$trtmt.RCB <- factor(unlist(tapply(rep(1:18,4),
+     sort(rep(1:4,18)), sample)))
```

As an exercise, work through the code for the assignment of plot numbers for the CRD and RCB cases and verify that they produce the plot layouts shown in Figures 16.2 and 16.3. We obtain the plot yields by applying the function over().

```
> plots.spdf@data$Yield <-
+     over(Yield.pts, plots.spdf, fn = mean)$Yield
```

The blocks in Figure 16.3a are laid out appropriately since most of the soil variability is in the north to south direction. Stroup (2002) points out that an incomplete block design is generally superior to an RCB in cases such as the present where there are many treatments. The most common such design is the lattice (Gomez and Gomez, 1984, p. 39), but this requires that the number of treatments be a square. Since our main interest is in comparison of methods for dealing with spatial autocorrelation, we will not pursue incomplete block designs.

Zimmerman and Harville (1991) used the RCB data as the standard of comparison, and we will do the same. We first tabulate the mean yields by treatment, from lowest mean response to highest.

```
> print(sort(tapply(plots.spdf@data$Yield,
+    plots.spdf@data$trtmt. RCB, mean)), digits = 4)
  13   17   18   11    3    5    9    7   12   10    4
2469 2571 2584 2757 2761 2790 2799 2801 2802 2934 2962
   2    1   15    8    6   14   16
2973 3005 3108 3158 3288 3390 3421
```

Examination of Figure 16.3b indicates that, by the luck of the draw, three of the treatment 13 plots and three of the treatment 17 plots fall in the lower yielding areas of their blocks. The analysis of variance reveals a significant block effect but no significant differences among treatments.

```
> RCB.aov <- aov(Yield ~ block + trtmt.RCB,
+   data = plots.spdf@data)
> summary(RCB.aov)
          Df   Sum Sq  Mean Sq F value Pr(>F)
block      3 68033791 22677930 72.9854 <2e-16 ***
trtmt.RCB   17  5046710   296865  0.9554 0.5189
Residuals  51 15846650   310719
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' 1
```

An aligned dot plot shows that the yield by treatment trends upward gradually from lowest yielding to highest (Figure 16.4). With this preparation we can now move on to the comparison of methods.

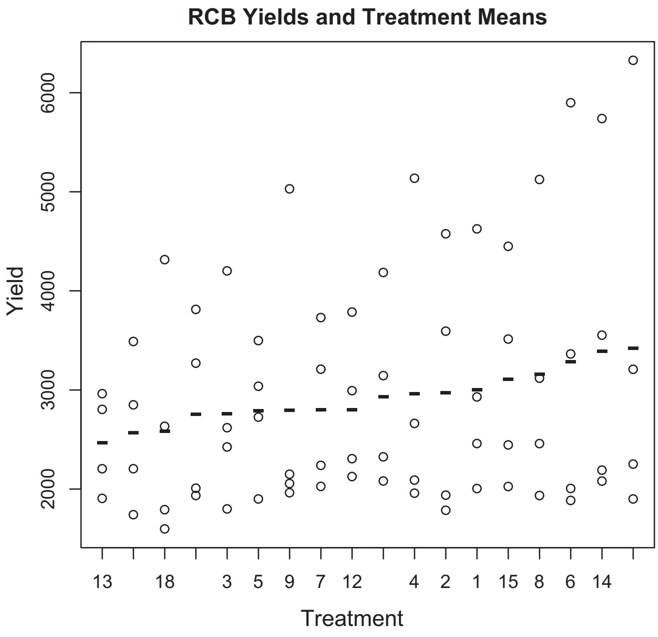

FIGURE 16.4
Dot plot of treatment responses and treatment means for the RCB experiment.

16.3 The Comparison of Methods

16.3.1 The Comparison Statistics

Zimmerman and Harville (1991) use comparison statistics that are based on a comment of Besag (1983). They are defined as follows:

$$
\begin{aligned}
EMP &= \frac{2}{r(r-1)}\sum_{i=1}^{r}\sum_{j=i+1}^{r}(\hat{\tau}_i - \hat{\tau}_j)^2 \\
PRE &= \frac{2}{r(r-1)}\sum_{i=1}^{r}\sum_{j=i+1}^{r} est.var(\hat{\tau}_i - \hat{\tau}_j),
\end{aligned}
\tag{16.3}
$$

where $\hat{\tau}_i$ is the estimated effect of treatment *i*, and $est.var(\hat{\tau}_i - \hat{\tau}_j)$ is the estimated variance of the difference in treatment effects. *EMP* stands for "empirical," and *PRE* stands for "predicted." Because the treatment effects are zero, the statistic *EMP* represents the empirical estimate of the variance of $(\tau_i - \tau_j)$, and therefore a small value of *EMP* indicates high accuracy. Put another way, since all of the τ_i should be the same, the quantities $(\hat{\tau}_i - \hat{\tau}_j)^2$ should be close to zero. *PRE* is the mean squared error of the difference, as predicted by the analysis, and should be approximately equal to *EMP* (Besag, 1983). To keep the discussion as simple as possible, we will focus on *EMP*.

Under the CRD model, the values of $\hat{\tau}_i$ are just the treatment means, computed in the last section. Let's compute the values $\hat{\tau}_i - \hat{\tau}_1$ for this model.

```
> trt.mean <- tapply(plots.spdf@data$Yield,
+   plots.spdf@data$trtmt.RCB, mean)
> print((trt.mean - trt.mean[1])[2:18], digits = 4)
      2       3       4       5       6       7       8       9      10
 -31.29 -243.50  -42.89 -214.38  283.10 -203.77  153.91 -205.44  -70.84
     11      12      13      14      15      16      17      18
-248.06 -202.35 -535.74  385.84  103.51  416.79 -433.74 -420.28
```

We can compute the value of *EMP* for the randomized complete block design directly from these treatment means. In order to facilitate the computation of the value of *EMP* for the other methods, however, we will obtain the values $\hat{\tau}_i - \hat{\tau}_j$ in a different way.

We will generate the same RCB analysis of variance model in the previous section, but using the function `lm()` instead of `aov()`.

```
> RCB.lm <- lm(Yield ~ trtmt.RCB + block,
+   data = plots.spdf@data)
> print(RCB.sum <- summary(RCB.lm))

Call:
lm(formula = Yield ~ trtmt. RCB + block, data = plots.spdf@data)

Residuals:
     Min       1Q   Median       3Q      Max
-1229.30  -315.90    22.78   219.38  1341.03
```

```
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  4568.41     301.04  15.175  < 2e-16 ***
trtmt.RCB2    -31.29     394.16  -0.079    0.937
trtmt.RCB3   -243.50     394.16  -0.618    0.539
   *   *   *   DELETED   *   *   *
trtmt.RCB18  -420.28     394.16  -1.066    0.291
block2      -1524.73     185.81  -8.206 6.93e-11 ***
block3      -2210.73     185.81 -11.898 2.49e-16 ***
block4      -2519.91     185.81 -13.562  < 2e-16 ***
 ---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 566.2 on 51 degrees of freedom
Multiple R-squared: 0.8168,     Adjusted R-squared: 0.745
F-statistic: 11.37 on 20 and 51 DF,  p-value: 1.709e-12
```

Notice that the values of the Estimates column of `trtmt.RCB2` through `trtmt.RCB18` are the same as the values $\hat{\tau}_i - \hat{\tau}_1$ printed in the previous output. These quantities are called the *treatment contrasts*. Without going into detail (See Kutner et al., 2005, p. 741, for information about contrasts in general and Venables and Ripley, 2002, p. 146 for information specific to R), the second contrast represents the difference $\hat{\tau}_2 - \hat{\tau}_1$ between the sample mean of treatment 2 and treatment 1, and the *ith* contrast represents the difference $\hat{\tau}_i - \hat{\tau}_1$ between the sample mean of treatment *i* and treatment 1. After the contrasts for treatments come the contrasts for blocks. One must be careful to place `trtmt.RCB` before `block` in the model specification to obtain this set of contrasts. We can then obtain the values $\hat{\tau}_i - \hat{\tau}_1$ as

```
> tau.dif1 <- coef(RCB.sum)[2:18,1]
```

We don't just need $\hat{\tau}_i - \hat{\tau}_1$, we need $\hat{\tau}_i - \hat{\tau}_j$ for all values of *j*. However, as far as the differences are concerned, the value of $\hat{\tau}_1$ doesn't matter. Each difference can be computed as $(\hat{\tau}_i - \hat{\tau}_1) - (\hat{\tau}_j \ - \hat{\tau}_1) = \hat{\tau}_i - \hat{\tau}_j$. Therefore, we can compute the quantities $\hat{\tau}_i - \hat{\tau}_j$ by just setting $\hat{\tau}_1 = 0$ and using the treatment contrasts directly.

```
> tau.hat <- c(0, tau.dif1)
```

We can use the function `vegdist()` from the `vegan` package (Oksanen et al., 2017) to compute the distances between the $\hat{\tau}_i$. Here are the distances for the first four values of $\hat{\tau}_i$.

```
> print(vegdist(tau.hat[1:4], method = "euclidean"), digits = 3)
                 trtmt.RCB2 trtmt.RCB3
trtmt.RCB2  31.3
trtmt.RCB3 243.5      212.2
trtmt.RCB4  42.9       11.6      200.6
```

Now we compute the full distance matrix.

```
> tau.dist <- vegdist(tau.hat, method = "euclidean")
```

The value of *EMP* in Equation 16.3 is computed from the sum of the squares of these values.

```
> print(EMP.RCB <- 2 * sum(tau.dist^2) / (18*17))
[1] 148432.6
```

As usual, the Monte Carlo simulation is carried out using the function `replicate()`. We first define the function `emp.calc()`.

```
> EMP.calc <- function(trt.data, form.lm){
+    trt.data$trtmt <- factor(unlist(tapply(rep(1:18,4),
+       sort(rep(1:4,18)), sample)))
+    trt.sum <- summary(lm(form.lm, data = trt.data))
+    tau.dif1 <- coef(trt.sum)[2:18,1]
+    tau.hat <-  c(0, tau.dif1)
+    tau.dist <- vegdist(tau.hat, method = "euclidean")
+    EMP <- 2 * sum(tau.dist^2) / (18*17)
+    return(EMP)
+ }
```

Now we carry out the simulation.

```
> set.seed(123)
> RCB.form <- as.formula(Yield ~ trtmt.RCB + block)
> U <- replicate(100, EMP.calc(plots.spdf@data, RCB.form))
> print(EMP.RCB <- mean(U))
[1] 148432.6
```

Following Zimmerman and Harville (1991), we will express all values of *EMP* as percentages of the RCB values. In Exercise 16.1, you are asked to repeat this exercise for the completely randomized design for this field. The result of this exercise indicates that the *EMP* value of the CRD is over three times as large as that of the RCB design.

In the next three subsections, we discuss the R implementation of three commonly used methods for dealing with spatial autocorrelation: the Papadakis nearest-neighbor method, the trend method, and the correlated errors method. Following this we review the results of published comparisons of these and other methods for dealing with a spatially autocorrelated substrate in field plot experiments.

16.3.2 The Papadakis Nearest-Neighbor Method

Next to the randomized complete block method, probably the oldest method still used for dealing with spatial autocorrelation is the *Papadakis method* (Papadakis, 1937). There are several variations on this method; we will discuss the one described by Brownie et al. (1993). Let e_{ij} denote the residual $e_{ij} = Y_{ij} - \bar{Y}_{k(ij)}$, where $\bar{Y}_{k(ij)}$ is the treatment mean of the treatment applied to cell *ij*. Then the Papadakis nearest-neighbor method is to fit the analysis of covariance (see Equation 12.1) model

$$Y_{ij} = \mu + \tau_{k(ij)} + \beta X_{ij} + \varepsilon_{ij}, \tag{16.4}$$

where, for interior plots,

$$X_{ij} = \frac{1}{4}(e_{i,j-1} + e_{i,j+1} + e_{i-1,j} + e_{i+1,j}). \tag{16.5}$$

For border plots, X_{ij} is the mean of the two or three neighboring plots. The Papadakis method does not include a blocking variable in the model, because the covariate term βX_{ij} replaces it.

Equation 16.5 can be implemented using a row normalized, rook's case spatial weights matrix as defined in Equations 3.22 and 3.23 of Section 3.4.2. Wilkinson et al. (1983) criticized the Papadakis nearest-neighbor method as being statistically inefficient, and a number of more complex methods, many of them involving iteration, have been developed. Brownie et al. (1993) and Stroup (2002) describe several of these. The improvement provided by these methods is often not very great, so we will not discuss them in detail.

In order to implement the Papadakis method, we must first compute the error terms $e_{ij} = Y_{ij} - \overline{Y}_{k(ij)}$. The treatment means are computed using the function `tapply()`, which we have seen before. The differences are computed using `sapply()`, which returns the value of the treatment mean of the appropriate treatment. The R code in this and the other subsections of this section require the results of Section 16.2 to be in memory.

```
> trt.means <- tapply(plots.spdf@data$Yield,
+   plots.spdf@data$trtmt.CRD, mean)
> Yield.e <- plots.spdf@data$Yield -
+     sapply(plots.spdf@data$trtmt.RCB, function(x) trt.means[x])
```

The X_{ij} in Equation 16.5 are computed by generating a `listw` object and using the function `listw2mat()` to convert the `listw` object into a matrix with which to multiply the error terms.

```
> library(spdep)
> nb6x12 <- cell2nb(6,12)
> W.papa <- nb2listw(nb6x12)
> X <- listw2mat(W.papa) %*% Yield.e
```

We can now apply the function `replicate()` to compute the value of *EMP*.

```
> set.seed(123)
> RCB.form <- as.formula(Yield ~ trtmt + X)
> U <- replicate(100, EMP.calc(papa.RCB, RCB.form))
> print(EMP.papa <- mean(U))
[1] 104232.1
> print(EMPbar.papa <- 100 * EMP.papa / EMP.RCB)
[1] 70.2218
```

For this data set, the Papadakis method presents, by this measure, a considerable improvement over the RCB design, with a value of *EMP* roughly 70% of that of the RCB.

16.3.3 The Trend Method

The trend method was proposed by Federer and Schlottfeld (1954) and analyzed by Tamura et al. (1988). It involves the use of a model of the form

$$Y_{ij} = \mu + \tau_{k(ij)} + T_{ij} + \varepsilon_{ij}, \tag{16.6}$$

where T_{ij} represents a trend (Section 3.2.1). The point of trend analysis is to improve on blocking by recognizing that the properties of the substrate are likely to vary smoothly rather than discretely at boundaries of the blocks (Brownie et al., 1993). Therefore, as with the Papadakis method, a block term is not added to the model. Although, as discussed in Sections 3.2.1 and 9.2, a model of T_{ij} may be developed by a method such as median polish or a generalized additive model, almost all published investigations of trend analysis have used a polynomial model, most commonly of the form

$$T_{ij} = \beta_0 + \beta_1 x_{ij} + \beta_2 y_{ij} + \beta_3 x_{ij}^2 + \beta_4 y_{ij}^2 + \beta_5 x_{ij} y_{ij} + \varepsilon_{ij}, \tag{16.7}$$

where x_{ij} and y_{ij} are x and y coordinates taken as the location of plot ij. To implement a trend analysis using Equation 16.7, we first compute the trend, normalizing the geographic coordinates to avoid computing with very large numbers.

```
> x <- (cell.ctrs[,1] - W) / (E - W)
> y <- (cell.ctrs[,2] - S) / (N - S)
> Yield.T <- lm(Yield ~ x + y + I(x^2) +
+   I(y^2) + I(x*y), data = plots.spdf@data)
> trend.RCB <- with(plots.spdf@data, data.frame(Yield = Yield,
+    trtmt = trtmt.RCB, Trend = predict(Yield.T)))
```

Next, we replicate the model and compute the relative value of *EMP*.

```
> trend.form <- as.formula(Yield ~ trtmt + Trend)
> set.seed(123)
> U <- replicate(100, EMP.calc(trend. RCB, trend.form))
> print(EMP.trend <- mean(U))
[1] 120175.6
> print(EMPbar.trend <- 100 * EMP.trend / EMP.RCB)
[1] 80.96304
```

Although it represents a considerable improvement over the RCB, in this particular case the trend method does not fare quite as well as the Papadakis method.

16.3.4 The "Correlated Errors" Method

The work of Zimmerman and Harville (1991) introduced the concept of using a generalized least squares model of the type described in Section 12.5 to the analysis of agricultural field plot data. This work in turn motivated further comparative studies by Brownie et al. (1993) and Stroup et al. (1994). Brownie et al. (1993) used the term "correlated errors model" to describe this approach. The model may be written

$$\begin{aligned} Y_{ij} &= \mu + \tau_{(k)ij} + T_{ij} + \varepsilon_{ij}, \\ \varepsilon_{ij} &\sim N(0, \sigma^2 \Lambda). \end{aligned} \tag{16.8}$$

where Λ is a variance-covariance matrix. The model of Equation 16.8 has the form, in the terminology of Brownie et al. (1993), of a "correlated errors plus trend" model. A purely correlated errors model would be obtained by eliminating the trend term T_{ij}. Brownie et al. (1993) found that the performance of the pure correlated errors was inferior to that of the correlated errors with trend model. It would be easy to develop the former by modifying the code given here, but we will only analyze the model in which a trend is included.

Following the methods of Section 12.5, it is straightforward to implement this model using the function `gls()` of the `nlme` package (Pinhiero et al., 2011). We first compute the generalized least squares (GLS) model with independent errors.

```
> library(nlme)
> trend.RCB$x <- x
> trend.RCB$y <- y
> model.gls <- gls(Yield ~ trtmt + Trend, data = trend.RCB)
```

Next, we use the function `Variogram()` to construct the variogram of the residuals (Figure 16.5) and by inspection, set the range and nugget to 0.35 and 0.5, respectively.

```
> plot(Variogram(model.gls, form = ~ x + y,
+     maxDist = 1), xlim = c(0,1),
+     main = "Variogram of Residuals")
```

We modify the function `emp.calc()` of Section 16.3.3 to invoke the function `gls()` rather than `lm()`, and to take into account that for a `gls` object the function `coef()` returns a vector containing only the treatment contrasts. We have to use the same variogram parameters for all of the models, but the values of the model coefficients are not very sensitive to these parameters (Exercise 16.2).

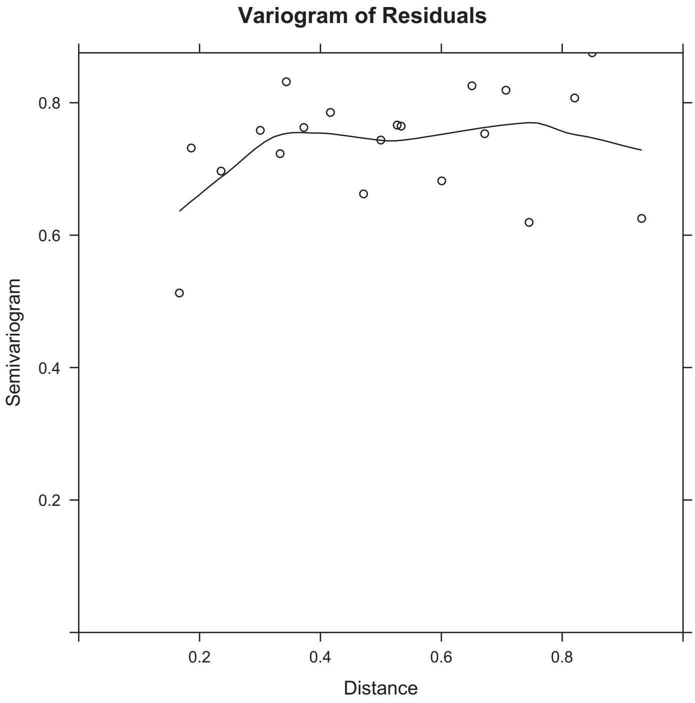

FIGURE 16.5
Experimental variogram of the residuals of the generalized least squares model for the example data set.

```
> EMP.gls <- function(trt.data){
+    trt.data$trtmt <- factor(unlist(tapply(rep(1:18,4),
+       sort(rep(1:4,18)), sample)))
+    trt.sum <- summary(gls(Yield ~ trtmt + Trend,
+       corr = corSpher(value = c(0.35, 0.5),
+       form = ~ x + y, nugget = TRUE), data = trt.data))
+    contr <- coef(trt.sum)
+    tau.hat <- c(0, contr[2:18])
+    tau.dist <- vegdist(tau.hat, method = "euclidean")
+    EMP <- 2 * sum(tau.dist^2) / (18*17)
+    return(EMP)
+ }
```

Finally, we run the simulation.

```
> set.seed(123)
> U <- replicate(100, EMP.gls(trend. RCB))
> print(EMP.gls <- mean(U))
[1] 80315.73
> print(EMPbar.gls <- 100 * EMP.gls / EMP.RCB)
[1] 54.10921
```

The results indicate that the correlated errors plus trend model outperforms both the Papadakis and the trend methods, with a value of *EMP* roughly half of that of the RCB.

16.3.5 Published Comparisons of the Methods

Since the work of Zimmerman and Harville (1991) was the first to include a generalized least square method, we begin our discussion with their results. In their comparison tests, Zimmerman and Harville (1991) compared a variety of versions of the Papadakis nearest-neighbor method, a pair of GLS (i.e., "correlated errors") models with the assumption of anisotropic covariance, a correlated errors model with the assumption of isotropic covariance, and a few methods not discussed here. Several different blocking schemes were tested on three data sets. The correlated errors methods consistently performed the best, with the version of the Papadakis method described above doing rather less well but still better than the simple RCB.

Following on the work of Zimmerman and Harville (1991), Brownie et al. (1993), and Stroup et al. (1994) conducted further comparisons of the various methods. Both of these latter papers used data from actual field trials rather than uniformity trials, so that they could not use Monte Carlo simulation of the *EMP* and *PRE* statistics of Besag (1973) in the comparisons. Brownie et al. (1993) used the RCB as a baseline and compared the Papadakis nearest-neighbor method, the trend method, the "correlated errors" method without incorporation of a trend, and the "correlated errors" plus trend method. They used what is essentially the square root of the *PRE* statistic plus an analysis of plots of the residuals versus yield values as their primary standard of comparison. Brownie et al. (1993) compared the performance of the methods applied to several data sets. Although the results varied somewhat, they found that the trend plus correlated errors model provided the best performance, with the trend model generally second and the Papadakis method third. All of the methods outperformed the RCB. Thus, the relative ranking of the trend and Papadakis method in the trials of Brownie et al. (1993) was the opposite of that of our simple test, but there was agreement in the superiority of the correlated errors plus trend model.

Stroup et al. (1994) compared the RCB method with two versions of the Papadakis method and also conducted a limited comparison of the correlated errors model. Their comparison was based on the magnitude of the experimental error, the coefficient of variation, and the effect of the test on the relative ranking of the treatments. Like Brownie et al. (1993), Stroup et al. (1994) found that the correlated errors model performed the best, and that there was little difference between the versions of the Papadakis method. Based on these results, it seems that the generalized least squares (or "correlated errors") model incorporating a trend term is the superior method for dealing with spatial autocorrelation in replicated experiments. In addition to performing better, it is less dependent on the shape of the experimental region than the Papadakis method, and in a sense it is unlikely to do worse than the model with a trend but independent errors. Brownie et al. (1993) point out that the success of the trend and correlated errors models may depend on the accuracy of the trend model and the model for the correlation structure of the errors. We saw in Exercise 9.3 that a generalized additive model (GAM) model can provide an accurate prediction of a trend surface when the trend surface is known. In Exercise 16.4, you are asked to repeat the analysis of this section using a GAM model to predict the trend surface.

16.4 Pseudoreplicated Data and the Effective Sample Size

16.4.1 Pseudoreplicated Comparisons

In carrying out a large-scale, landscape-level experiment, it may happen that for logistical or economic reasons it is impossible to lay out the experiment using a traditional replicated plot design. For example, Lee et al. (2006) report on an experiment to compare the response of carbon sequestration rate to tillage method (conventional vs minimum till) in a commercial corn field. The experiment involves the measurement of carbon sequestration rate using the eddy covariance method. Without going into detail about this method, the fetch (i.e., the ground area measured) of the measurement instruments is sufficiently large that no more than two could be placed in a single field. The cost of the equipment precluded more than one trial at this initial stage of the program, but it was nevertheless desirable to obtain some statistical measure of the difference between various quantities, such as yield and normalized difference vegetation index (NDVI), in the conventional and minimum till plots of the field.

The characteristic of an experiment such as this is that a single large area is divided into two halves. Half the area receives treatment 1 and the other half treatment 2. The response variable is measured at n locations within each half. Under the usual ANOVA procedure, these data would be modeled using a nested design (Kutner et al., 2005, p. 662). In this case, however, the data are autocorrelated, but not perfectly so. Depending on the level of autocorrelation, it may not be appropriate to model the experiment using either a nested design or an independent errors model. In calculating degrees of freedom in a comparison of factor level means, the n pairs of measurements provide an effective sample size of n_e pairs, where $1 \le n_e \le n$ (see Section 10.1). The next subsection describes a method for the estimation of n_e, followed by the application to an unreplicated comparison.

Prior to describing the estimation of n_e we must take this opportunity to emphasize the hazards of carrying out an unreplicated experiment. As was mentioned earlier, although the nominal reason for replicating treatments in an experiment is to obtain an estimate of the variance, replication also serves to introduce interspersion (Hurlbert, 1984). In a spatial context, this means that treatment effects are measured at several different, randomly selected locations in the experimental area. Without replication, there is no way to distinguish between legitimate treatment effects and differences in the response variable due to differences in the experimental substrate. Therefore, the method described in this section must be used with extreme caution in the comparison of treatments. If there is no interest in the comparison of two treatments, the method may be considered as a two-plot uniformity trial in which the estimate of n_e, together with an indication of the spatial consistency of the sample variance, may be useful in its own right as an indication of the effect of autocorrelation on the effective sample size.

16.4.2 Calculation of the Effective Sample Size

The method of estimation of n_e was discussed by Plant (2007) and is similar to that described in Section 11.2.3. It is based on an extension of the method of Clifford and Richardson (1985) and Clifford et al. (1989) described in Section 11.2.2. A spatial error model for the comparison of two means associated with autocorrelated variables $Y_1(x,y) = Y_{1j}$ and $Y_2(x,y) = Y_{2j}$ may be written as

$$\begin{aligned} Y_{ij} &= \mu_i + \eta_{ij},\ i = 1,2,\ j = 1, n \\ \eta_{ij} &= \lambda \Sigma_k w_{jk} \eta_{ik} + \varepsilon_{ij}, \end{aligned} \tag{16.9}$$

where i indexes the factor levels, j indexes the samples, λ is a scalar measuring the strength of autocorrelation of the errors, the w_{ik} are elements of the spatial weights matrix describing the connectivity among the samples, and the ε_{ij} are independent, identically distributed normal random variables with mean 0 and variance σ^2. We will assume that the autocorrelation coefficient λ is the same for both Y_1 and Y_2.

The null hypothesis of equality of means of Y_1 and Y_2 may be tested using Student's t statistic, $t = (\bar{Y}_1 - \bar{Y}_2)/s_{12}$, where s_{12} is the square root of $s_{12}^2 = (s_1^2 + s_2^2)/2$, with $s_1^2 = \Sigma_j (Y_{1j} - \bar{Y}_1)^2/(n-1)$ and similarly for s_2^2. If the values of Y_{1j} and Y_{2j} are spatially independent, then the sampling distribution of the t statistic under the null hypothesis of no difference between means has a variance of $\sigma_t^2 = D/(D-2)$ where $D = 2n - 2$ is the number of degrees of freedom. In the case that Y_{1j} and Y_{2j} are each spatially autocorrelated, if an independent estimate $\hat{\sigma}_t^2$ of the sampling variance is available, then this can in principle be used as was done in Section 11.2 to estimate the effective sample size n_e by inverting the variance equation and substituting for n.

Unfortunately, the resulting estimate, $\hat{n}_e = (2\hat{\sigma}_t^2 - 1)/(\hat{\sigma}_t^2 - 1)$, has very poor numerical properties because $\hat{\sigma}_t^2$ appears in both the numerator and the denominator. A better estimator for n_e is obtainable by using the point biserial correlation coefficient (Tate, 1954). This statistic is computed by arranging the vectors Y_1 and Y_2 into a single column vector Y. The point biserial correlation coefficient r_w is then the correlation coefficient between Y and a vector Z whose components corresponding to Y_1 are 1 and corresponding to Y_2 are 0 (i.e., $Z = [1 \ldots 1\ 0 \ldots 0]'$). When the components of $Y_1(x,y)$ and $Y_2(x,y)$ are independent the variance of the sampling distribution of r_w is $\sigma_r^2 = 1/2n$ (Tate, 1954). Therefore, given an independent estimate $\hat{\sigma}_r^2$ of the variance of r_w an estimate of the effective sample size can be obtained as

$$\hat{n}_e = \frac{1}{2\hat{\sigma}_r^2}. \tag{16.10}$$

As described in Section 11.4.2, one can obtain an estimate using either a block bootstrap or a parametric bootstrap method (Efron and Tibshirani, 1993; Davison and Hinkley, 1997). The bootstrap estimate $\hat{\sigma}_r^2$ is plugged into Equation 16.10 to generate $\hat{n}_e$. The t statistic for the test of the null hypothesis $r_w = 0$ is

$$t = \frac{r_w(2n-2)^{1/2}}{1-r_w^2}, \tag{16.11}$$

with $2n-2$ degrees of freedom. The corrected t statistic for the test of the null hypothesis $r_w = 0$ may then be obtained by replacing n with $\hat{n}_e$ in Equation 16.11 to obtain

$$t_{corr} = \frac{r_w(2\hat{n}_e-2)^{1/2}}{1-r_w^2} \tag{16.12}$$

with $2\hat{n}_e-2$ degrees of freedom.

As usual, we will first develop a simulation using a simple artificial data set. First, we generate two autocorrelated variables Y_1 and Y_2 on a 14 by 14 lattice and remove the outer two layers to produce a 10 by 10 lattice.

```
> library(spdep)
> lambda <- 0.6
> nlist.14 <- cell2nb(14, 14)
> IrWinv.14 <- invIrM(nlist.14, lambda)
> nlist <- cell2nb(10, 10)
> W <- nb2listw(nlist)
```

Next, we create a function `para.samp()` that generates a bootstrap replication of the point biserial correlation coefficient r_w.

```
> para.samp <- function(Y1, Y2, IrWinv1.mod, IrWinv2.mod){
+   Y1.samp <- sample(Y1, length(Y1), replace = TRUE)
+    Y1.boot <- IrWinv1.mod %*% Y1.samp
+   Y2.samp <- sample(Y2, length(Y2), replace = TRUE)
+   Y2.boot <- IrWinv2.mod %*% Y2.samp
+   Z <- c(rep(1,100), rep(0,100))
+   Y <- c(Y1.boot, Y2.boot)
+   r.hat <- cor(Y,Z)
+ }
```

Next, we generate the two variables Y_1 and Y_2.

```
> set.seed(123)
> Y1.plus <- IrWinv.14 %*% rnorm(14^2)
> Y1 <- matrix(Y1.plus, nrow = 14, byrow = TRUE)[3:12,3:12]
> Y2.plus <- IrWinv.14 %*% rnorm(14^2)
> Y2 <- matrix(Y2.plus, nrow = 14, byrow = TRUE)[3:12,3:12]
```

The uncorrected t-test yields the following.

```
> print(t.test(Y1, Y2, "two.sided")$p.value, digits = 3)
[1] 0.0632
```

Next, we fit the data to a spatial error model (Equation 16.10) and extract the residuals and the values of λ.

```
> # Fit a spatial error to the data
> Y1.mod <- errorsarlm(as.vector(Y1) ~ 1, listw = W)
> e1.hat <- residuals(Y1.mod)
> print(lambda1.hat <- Y1.mod$lambda, digits = 2)
lambda
  0.39
> Y2.mod <- errorsarlm(as.vector(Y2) ~ 1, listw = W)
> e2.hat <- residuals(Y2.mod)
> print(lambda2.hat <- Y2.mod$lambda, digits = 2)
lambda
  0.47
```

Now we generate 100 bootstrap resamples of the data, compute the point biserial correlation coefficients, and use the plug-in principle to compute the variance estimate $\hat{\sigma}_r^2$

```
> IrWinv1.mod <- invIrM(nlist, lambda1.hat)
> IrWinv2.mod <- invIrM(nlist, lambda2.hat)
> U <- replicate(200, para.samp(e1.hat, e2.hat, IrWinv1.mod,
+    IrWinv2.mod))
> print(sigmar.hat <- var(U), digits = 3)
[1] 0.0125
```

By way of comparison, the uncorrected value of the variance is $1/200 = 0.005$. Finally, we use Equations 16.10 and 16.12 to compute the corrected t statistic.

```
> print(ne.hat <- 1 / (2*sigmar.hat), digits = 3)
[1] 39.9
> Z <- c(rep(1,100), rep(0,100))
> Y <- c(Y1, Y2)
> rw <- cor(Y, Z)
> t.corr <- rw*sqrt(2*ne.hat - 2) / (1 - rw^2)
> print(p.corr <- 2 * (1 - pt(q = abs(t.corr), df = 2*ne.hat - 2)),
+    digits = 3)
[1] 0.241
```

The corrected p value is considerably higher than the uncorrected one. In Exercise 16.4, you are asked to test the method in a Monte Carlo simulation.

A spatial lag model of the form

$$Y_{ij} = \mu_i + \rho\Sigma_k w_{ik} Y_{ik} + \varepsilon_{ij},\ i = 1,2,\ j = 1, n \qquad (16.13)$$

may for certain data be more appropriate than the spatial error model. This can be used as well (Exercise 16.5).

16.4.3 Application to Field Data

As an example of the method, we will test the null hypothesis that the mean values of *SoilpH* in Field 4.2 are equal on the east and west sides of the field (Figure 16.6). This is a bit contrived, but not entirely so. Recursive partitioning picked soil acidity as a

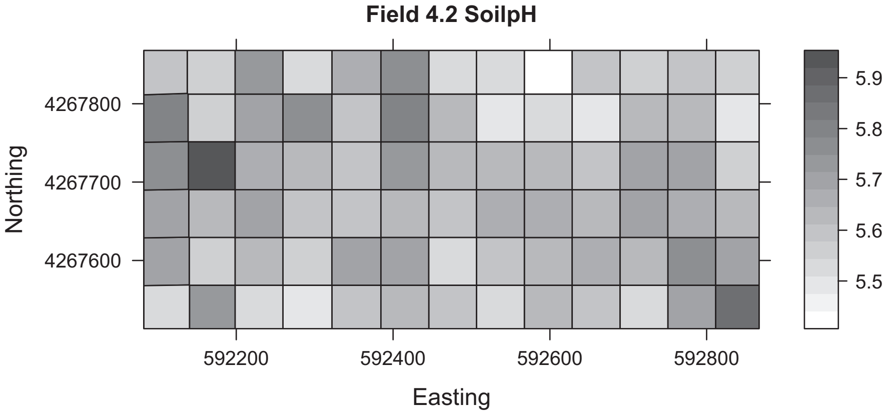

FIGURE 16.6
Thematic map of soil pH in Field 4.2.

variable associated with yield in Field 4.2 (Exercise 9.9), but higher yields are associated with more acidic soils, which is the opposite of what would be expected with wheat. There is a strong east-west yield trend in the 1996 data due to weed influence, and it is of interest to know whether soil pH might tend to be higher in the western part of the field, which could create a spurious association of soil pH with yield due to their common (and in the case of soil pH, probably coincidental) association with high weeds.

There are 13 sample locations in the east-west direction in the field, so we will eliminate the middle column and divide the field into two halves of 36 locations each.

```
> sort(unique(data.Set4.2$Easting))
 [1] 592107 592109 592110 592111 592112 592113 592168
 [8] 592229 592290 592354 592415 592476 592537 592598
[15] 592659 592720 592781 592842
> data.Set4.2W <- data.Set4.2[which(data.Set4.2$Easting < 592450),]
> data.Set4.2E <- data.Set4.2[which(data.Set4.2$Easting > 592500),]
```

For some reason, the Eastings of the westernmost sample locations are not exactly aligned (this is also evident in the slight irregularity of the Thiessen polygons on the west edge in Figure 16.6). Therefore, we have to be a bit careful in establishing the value of d2 for the function dnearneigh() and not make it too small.

```
> Y1 <- data.Set4.2W$SoilpH
> coordinates(data.Set4.2W) <- c("Easting", "Northing")
> nlist.1 <- dnearneigh(data.Set4.2W, d1 = 0, d2 = 70)
> W.1 <- nb2listw(nlist.1, style = "W")
> Y2 <- data.Set4.2E$SoilpH
> coordinates(data.Set4.2E) <- c("Easting", "Northing")
> nlist.2 <- dnearneigh(data.Set4.2E, d1 = 0, d2 = 70)
> W.2 <- nb2listw(nlist.1, style = "W")
```

Now we are ready to do the analysis. First, we will check the means and to an ordinary *t*-test.

```
> mean(data.Set4.2W$SoilpH)
[1] 5.651111
> mean(data.Set4.2E$SoilpH)
[1] 5.615833
> t.test(data.Set4.2W$SoilpH,data.Set4.2E$SoilpH, “two.sided”)
          Welch Two Sample t-test
data: data.Set4.2W$SoilpH and data.Set4.2E$SoilpH
t = 1.6709, df = 68.797, p-value = 0.09928
```

The soil is very slightly more acidic on the west side, and the difference is not significant at the $\alpha = 0.05$ level and almost certainly not important. Let's follow up and compare the outcome of this test with the test that incorporates spatial autocorrelation. We first create a function `para.samp()`almost identical to the one in Section 16.4.2. Then we set up the spatial error models in the same way as in Section 16.4.2. Finally, we generate the bootstrap resample values, calculate the variance, and estimate the effective sample size and corrected p value.

```
> set.seed(123)
> U <- replicate(200, para.samp(e1.hat, e2.hat, IrWinv1.mod,
+    IrWinv2.mod))
> print(sigmar.hat <- var(U), digits = 3)
[1] 0.0366
> print(ne.hat <- 1 / (2*sigmar.hat), digits = 3)
[1] 13.7
> Z <- c(rep(1,length(Y1)), rep(0,length(Y2)))
> Y <- c(Y1, Y2)
> rw <- cor(Y, Z)
> t.corr <- rw*sqrt(2*ne.hat - 2) / (1 - rw^2)
> print(p.corr <- 2 * (1 - pt(q = abs(t.corr), df = 2*ne.hat - 2)),
+    digits = 3)
[1] 0.315
```

The effective sample size is about one-third the total sample size. Not surprisingly, the corrected p value is much higher than the uncorrected value. The comparison of p values provides an indication of the impact of spatial autocorrelation on the data analysis.

This example does not involve a treatment but rather measures differences in the substrate itself, so there is less concern with the interspersion issue. An example of a situation in which the interspersion effect would be important is if we had applied a compound such as gypsum to, say, the west side and measured the differences in pH after this application. In this case, without other information it would be impossible to distinguish differences in soil pH due to the gypsum from those present prior to the application.

16.5 Further Reading

Kutner et al. (2005) is a good source for a basic discussion of the analysis of variance, although any other basic linear modeling text will serve equally well. Gomez and Gomez (1984) has a very extensive discussion of both complete and incomplete block designs. Stroup (2002) provides a very cogent discussion of the pros and cons of the traditional “design based” approach versus the more recent “model based” approach to the incorporation of spatial effects into replicated experiments. Brownie et al. (1993) provide a very readable review and critique of the various methods and of the assumptions that underlie them. Griffith (1978) provides further discussion. The method of dealing with unreplicated experiments with two treatments discussed in Section 16.4 has some ideas in common with *before and after control intervention* (BACI) analysis (Thomas et al., 1978). Discussions of this include Smith et al. (1993) and Evans and Coote (1993).

With the exception of variety trials, the checkerboard layout common in statistical analyses of uniformity trials is uncommon in actual field experiments, at least with row crops. Many agronomists prefer to make the plots within the blocks long and narrow. This has both agronomic and statistical advantages. Agronomically, it is generally easier to apply the same treatment to an entire row. Also, the plots encompass as much of the block variability as possible within each plot (Little and Hills, 1978, p. 285; Gomez and Gomez, 1984, p. 500). Besag (1991) provides a good discussion of agricultural plots as well as a statistical method designed for the analysis of experiments with long, narrow plots.

Exercises

16.1 Repeat using a completely randomized design (CRD) the analysis done in Section 16.3.1 for an RCB design and compute the value of *EMP* for this design.

16.2 The correlated errors method described in Section 16.3.4 requires an estimate of the parameters of the variogram model for the errors. Test this method for sensitivity to the values of this variogram model.

16.3 Apply the completely random design, the randomized complete block, and the Papadakis method to the population data used in Chapter 5 to test various sampling schemes for the data of Field 4.2.

16.4 Repeat the analysis of Section 16.3.4 using a GAM to predict the trend surface.

16.5 Carry out a Monte Carlo simulation of the method discussed in Section 16.4.2 applied to the artificial data set used in that section. Compute the experimental error rate as a function of the parameter λ.

16.6 Using a spatial lag model rather than a spatial error model, repeat the analysis of Section 16.4.3.