5

Sampling and Data Collection

5.1 Introduction

Spatial data consist of two components, a spatial component and an attribute component (Haining, 2003). Spatial sampling can refer to the collection of information about either of these two components. An example of the sampling of the spatial component of a data set is the use of a GPS to record locations in a landscape of point features such as trees. An example of sampling of the attribute component is the collection of soil properties such as clay content, cation exchange capacity, and so forth, according to some sampling pattern. One could also sample for both the spatial location and the attribute values. All four of the data sets used in this book conform to the second type of sampling, in which the location of the sample is specified and some attribute value or values are recorded. It is implicitly assumed in this sort of sampling that the locations of the sites are measured without error. This assumption is not always valid in real data sets, but it is a fundamental one on which much of the theory is based. Since the true location is usually close to the measured location, the usual way to take location uncertainty into account is to augment attribute uncertainty.

In the sampling plans described in this chapter, the locations at which one collects data are generated according to a rule. Sometimes the rule involves some form of randomization, although this is not the case with any of the four data sets in this book. Sometimes the rule involves the systematic selection of locations, as is the case with Data Set 4. Alternative methods of site selection, neither random nor systematic, also exist. These are sometimes imposed by the conditions under which the data are collected. Cochran (1977, p. 16) describes some of them.

1. The sample may be restricted to that part of the population that is readily accessible. The observation sites of western yellow-billed cuckoos in Data Set 1 were collected by floating down the river in a boat, playing bird calls, and listening for a response. Because of this collection method, the sites were restricted to areas near the water.
2. The sample may be selected haphazardly. The data in Data Set 2, which involve the distribution of oak trees in the foothills of the Sierra Nevada and the factors that affect this distribution, were collected during the 1930s in a partly systematic and partly haphazard manner (Wieslander, 1935).

3. The sampler may select "typical" members of a population. This was more or less the case in the selection of sample sites within the rice fields of Data Set 3.
4. The sample may consist of those members of the population who volunteer to be sampled. This is often the case in sampling campaigns involving private property owners. The data in Data Set 3 involves samples collected in the fields of a collection of Uruguayan rice farmers, all of whom agreed to have the samples collected in their fields.

To quote Cochran (1977, p. 16), "Under the right conditions, any of these methods can give useful results." This is a good thing, because at some level virtually every data set collected in the course of a field experiment or observational study can be described at least partly as having one or more of these characteristics (Gelfand et al., 2006). Agricultural researchers collect data on the farms of cooperating growers or on an experiment station. Researchers travel to sites that they can reach. Locations that appear anomalous may be, perhaps subconsciously, excluded. However, it is frequently the case that, once one has either haphazardly or for purposes of convenience selected a site, one can carry out the sampling campaign on that site according to a well-considered plan. This is the case with Data Sets 3 and 4. The locations of fields themselves were selected in consultation with the cooperating farmers, but once the fields were selected the investigators were free to collect data within each field according to a plan of their choice.

In interpreting the results of the analysis of data from a study that suffers from an imperfection, such as one of the four mentioned above, the most important issue to keep in mind is whether the imperfections have introduced bias relative to the population that the sample is supposed to represent (i.e., to the scope of inference). The yellow-billed cuckoo sample in Data Set 1 may favor areas that are more easily accessible to humans. The haphazard selection of sites to sample for oak trees in Data Set 2 may favor a certain type of site, such as those that are more accessible. Data collected from a group of participating farmers such as that in Data Set 3 may be a biased representation of a population of all farmers if only the more "progressive" farmers cooperate with researchers. When reporting these results, care should be taken to describe the conditions under which the sample was taken as precisely as possible so that the reader may form a judgment as to the validity of the sample.

In many ecological applications, the data collected by sampling at point locations on the site may be accurately modeled as varying continuously in space (for example, mean annual precipitation or soil clay content). This chapter is concerned with sampling plans for such spatially continuous data. Although the data are assumed to vary continuously by position, and the samples may be treated mathematically as if they occurred at a single point, this is obviously not really the case. Any real sample, be it a soil core, an electrical conductivity measurement, a pixel in a remotely sensed image, a yield monitor data value, a visual observation, or whatever, is taken over a finite area. The size of area over which the sample is extracted is called the *support* (Webster and Oliver, 1990, p. 29; Isaaks and Srivastava, 1989, p. 190; see Section 6.4.1). In the case of a soil core, the support may be a centimeter or two for a single core, or a meter or two in the case of a composite sample. The other measurements just mentioned typically have supports ranging from one to tens of meters. As an example, Data Set 1 includes locations of bird calls taken near a river. Precise identification of the geographic location of the source of this observation is in many cases impossible, and so the support may be quite large. It is often the case that the support of the sample is different from the size of the area that the sample is intended to represent. For example, many of the analyses described in this book assume that the data are defined on a mosaic of polygons.

Soil core data with a support of a few meters may be used to represent the value of that quantity in polygons that are each tens or hundreds of square meters in area. The distribution of a soil property averaged over a set of polygons of larger area than the support will generally be much smoother than the distribution of the same property averaged over a support of a few square meters. Thus, the polygons should not be considered as the support of these data. Isaacs and Srivastava (1989, p. 190) discuss the issue of mismatch between sample size and area of intended use of the sample data in some detail. Precise definitions of support and other related terms are given in Section 6.4.1, and the effect of this difference between the actual and represented support is discussed in Section 11.5.

In the process of sampling spatial data, the first step is to delimit the area in which to sample. This explicitly defines the *population*, the set of values from which the sample is drawn. The population could be either finite, meaning that it can be indexed by a set of N integers; it may be countably infinite, meaning that it can be indexed by a set of integers, but that set of integers is infinite; or it may be uncountably infinite, meaning that it varies continuously and cannot be indexed. In terms of sampling a spatial area, only the second and third cases generally are relevant unless N is very large. If the value of every member of the population could be determined, then the process would be a *census* and not a *sample*, so it is implicit in the use of this latter term that not every member of the population will be measured. Rather, the properties of the subset that constitutes the sample will be evaluated and used to estimate properties or characteristics of the population.

According to the use of the term given in the preceding paragraph, the population from which the sample is drawn is restricted to the region of sampling. In many if not most cases, however, the results of the sampling and of the analysis of the data from the sample are actually extrapolated to a larger population that is contained in a larger region. For example, the analysis of Data Set 3 is not intended to be restricted to the 16 participating farmers. These results are implicitly extrapolated to a wider collection of similar locations. The definition of similarity in this context, however, is more a scientific issue than a statistical one, and the extrapolation process often depends on the ecological or agronomic skills of the investigator in selecting experimental sites.

In determining a spatial sampling plan, there are a number of issues that one must address (Haining, 2003 p. 94). These include (1) determining the precise objective of the sampling program, (2) determining the spatial form of the sampling pattern, and (3) determining the sample size. Haining (1990, p. 171) describes three categories into which the objective in a spatial sampling campaign may fall.

> *Category I* has as its objective the estimation of a global, non-spatial statistic such as the value of the mean or the probability that the mean exceeds some predetermined threshold.
>
> *Category II* includes objectives that require knowledge of the variation of the quantity sampled. These include the computation of graphical summaries such as the variogram as well as the development of interpolated maps.
>
> *Category III* includes objectives that involve the creation of a thematic map via classification. An example is ground-truthing the classification of vegetation types in a remotely sensed image.

Our initial comparison of methods will be based on how well they satisfy an objective in Category I, namely, the estimation of the population mean. While this objective may not

arise frequently in ecological research involving spatial data, it turns out that sampling plans that are good at estimating the global mean are often good at other things as well. The problems of selecting placement and number of locations may be separated into two stages. The first stage is the selection of the form of locations' spatial patterns, and the second stage is the selection of the intensity of sampling using that pattern, that is, the number of samples. Most practical spatial sampling patterns fall into one of a few general types. Therefore, it is possible to simplify the process by choosing from a relatively small number of candidate patterns.

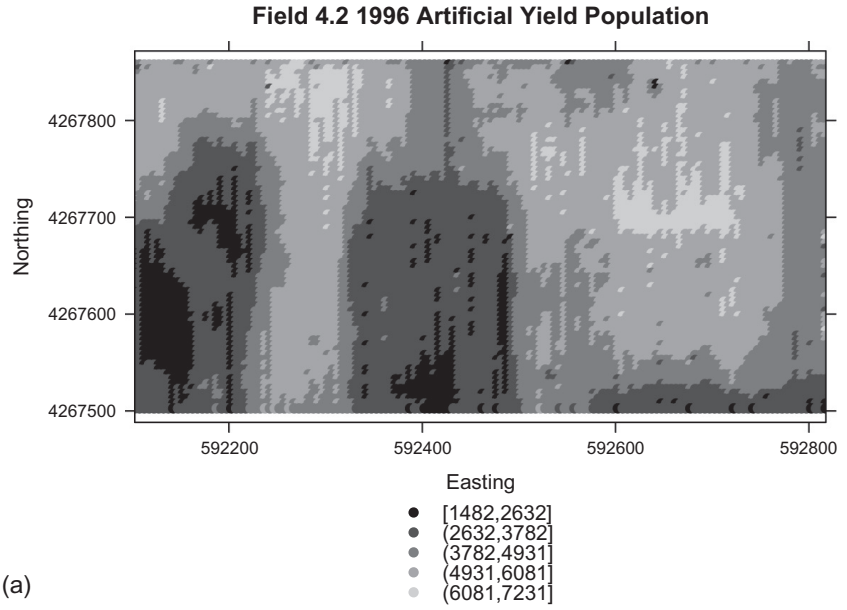
The method we use for comparison of sampling patterns is to construct an artificial population, run Monte Carlo simulations on the application of various spatial sampling patterns to this population, and compare the results. This approach does not imply any generality of the comparisons, because it only considers one example. The comparison is useful as a means of gaining insight into the strengths and weaknesses of the patterns, but not as a means of developing general conclusions about the effectiveness of the patterns in sampling other populations. Nevertheless, it provides a useful and intuitive device for increasing understanding of the issues associated with spatial sampling. In Section 5.2, we set up the artificial pattern and formulate the ground rules for the comparison of patterns.In Section 5.3 we develop a set of sampling plans to collect data and compare these plans according to the rules developed in Section 5.2. Section 5.4 contains a discussion of sampling for variogram estimation, and Section 5.5 describes methods for estimating the appropriate sample size based on a targeted confidence interval. Section 5.6 briefly discusses sampling when the intent is to construct a thematic map. Section 5.7 describes the concept of model-based sampling.

5.2 Preliminary Considerations

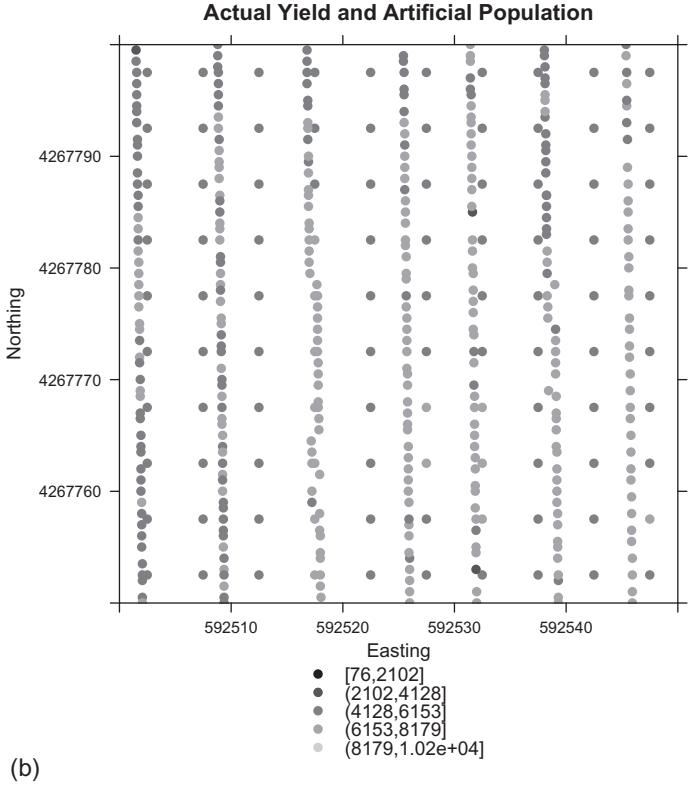
5.2.1 The Artificial Population

To test the different sampling patterns, we will create an artificial population that is somewhat idealized but is based on real data. Among all of our data sets, the yield monitor data from agricultural fields provide the best opportunity to create an artificial population whose characteristics closely resemble those of a real, continuously varying population. The data we use are taken from the 1996 yield map of Field 2 of Data Set 4, when the field was planted to wheat. Eliminating the edges of the field leaves a data set measuring 720 m long by 360 m wide. To regularize the data for sampling purposes, we create a grid of 144 by 72 points, for a total of 10,368 points on a 5m grid, and in each point we place the value of yield estimated by inverse distance weighted interpolation (Isaaks and Srivastava, 1989, p. 257; see Section 6.3.1). The resulting values are close to the original yield data, but the grid provides the regularity needed for a test of the various sampling designs. Moreover, since this is a finite population we can compute its parameters (mean, standard deviation, etc.) exactly and test the various sampling methods exactly. The creation of the artificial data set involves procedures that are not discussed until later in the book, and understanding these procedures is not essential to understanding the discussion in this chapter. For this reason, the code is not discussed, although like all of the code it is included on the book's website. The population is housed in a `SpatialPointsDataFrame` called `pop.data`.

Figure 5.1a shows a gray-scale plot of the population, and Figure 5.1b shows a close-up of the locations of the data from the artificial population and the real data on which it is based. We first compute some of the population parameters associated with the artificial



(a)



(b)

FIGURE 5.1
(a) Plot of the population of 10,368 idealized yield data values; (b) a close-up plot of the locations of a small subset of the idealized values (regular grid) together with the locations of the actual data values.

population. The summary statistics for the yield population data of Figure 5.1 are as follows (because $N = 10,368$, we can ignore the fact that `sd()` divides by $N-1$ instead of N).

```
> print(pop.mean <- mean(pop.data$Yield), digits = 5)
[1] 4529.8
> print(pop.sd <- sd(pop.data$Yield), digits = 5)
[1] 1163.3
```

We can generate a plot of the histogram of the data (Figure 5.2) using the function `hist()`.

```
> hist(data. Pop$Yield, 100, plot = TRUE,
+     main = "Histogram of Artificial Population",
+     xlab = "Yield, kg/ha", font.lab = 2, cex.main = 2,
+     cex.lab = 1.5) # Fig. 5.2
```

This reveals that the population distribution is bimodal, as might be expected from inspection of Figure 5.1a. We will see in Chapter 7 that the areas of very low yield correspond to areas of very high weed levels. Therefore, the population presumably can be considered consisting of two subpopulations, one without intense weed competition, and one with weed competition.

We must also create a boundary polygon `sampbdry.sf` that delimits the population. This follows exactly the same procedure as was used in Section 2.4.3 to create the

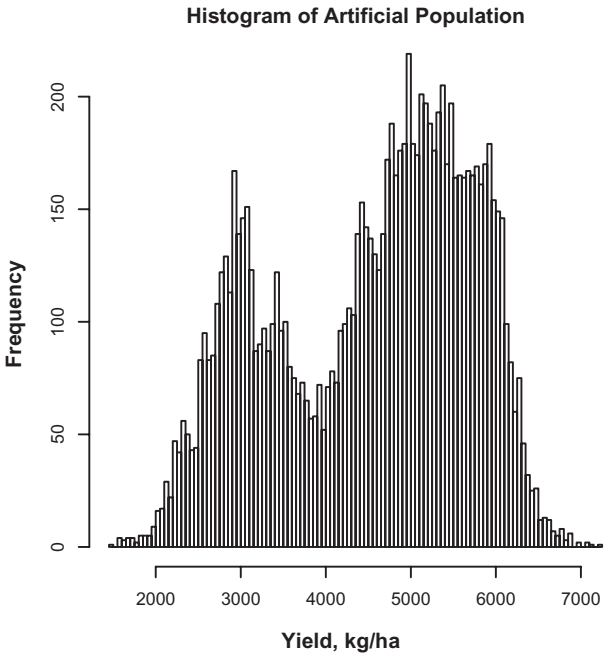


FIGURE 5.2
Frequency histogram of the yield distribution of the artificial data set shown in Figure 5.1.

boundary for the actual field. The only difference is in how the location of the boundaries is determined. For the boundary of the artificial population we extract the coordinates of the bounding box of the `SpatialPointsDataFrame` that contains the artificial population data as follows.

```
> W <- bbox(pop.data)[1,1]
> E <- bbox(pop.data)[1,2]
> S <- bbox(pop.data)[2,1]
> N <- bbox(pop.data)[2,2]
> E - W
[1] 715
> N - S
[1] 355
```

In order return to our original size, we add or subtract 2.5m as appropriate.

```
> N <- N + 2.5
> S <- S - 2.5
> E <- E + 2.5
> W <- W - 2.5
```

The remainder of the construction follows exactly the same steps as were used in Section 2.4.3.

5.2.2 Accuracy, Bias, Precision, and Variance

Sampling error is an unavoidable consequence of the fact that the data represent a sample and not a census. Suppose we are trying to use the sample mean $\bar{Y}$ to estimate the mean μ of a population. The difference $|\bar{Y} - \mu|$ is a measure of the *accuracy* of the sample (Cochran, 1977, p. 16). One source of low accuracy is a *bias* in the sample. This term will be defined more precisely later on, but intuitively it means a systematic difference between $\bar{Y}$ and μ as shown schematically in Figure 5.3c. A potential second source of low accuracy is low *precision*, as shown schematically in Figure 5.3b and 5.3d. The term refers to a large deviation of the sample means $\bar{Y}$ from the grand mean obtained by repeated applications of the sample procedure. Of course, in a real sampling campaign one does not know the true values of the population parameters, but with our artificial population we do, and so we can compare both the accuracy and the precision of the samples.

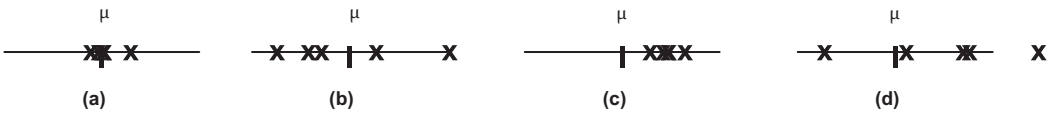


FIGURE 5.3
Four different combinations of accuracy and precision: (a) high accuracy and high precision, (b) high accuracy and low precision, (c) low accuracy due to bias and high precision, (d) low accuracy and low precision. Each x represents one application of the sampling procedure.

5.2.3 Comparison Procedures

The testing we present here employs Monte Carlo simulation (Section 3.3). The simulations generate statistics that measure the quality of the sampling plan. Two such statistical measures are used. The first is a measure of accuracy, and the second is a measure of precision. The measure of accuracy is the percent error, defined as

$$\text{Percent error} = \left| \frac{\text{Sample mean} - \text{true mean}}{\text{True mean}} \right| \times 100\%. \tag{5.1}$$

This statistic is computed for each simulation and the mean of the simulations is displayed. The measure of precision is the experimental standard error (i.e., the standard deviation of the means) computed by the simulations. This standard error will be used to compute the *relative efficiency* of two sampling schemes. We will define the relative efficiency e_{12} of estimation of the mean of two sampling schemes of the same sample size to be

$$e_{12} = \frac{se_2}{se_1}, \tag{5.2}$$

where se_i is the standard error of the mean obtained using scheme i.

5.3 Developing the Sampling Patterns

5.3.1 Random Sampling

Webster and Oliver (1990, p. 42) describe a number of simple sampling plans, including simple random sampling, geographically stratified random sampling, and grid sampling. We will evaluate five sampling plans: these three, a second form of stratified sampling, and cluster sampling. All are tested on our finite artificial population, which has a size $N = 10,368$ on a 144 by 72 grid.

The `sp` package contains a very convenient function `spsample()` for determining sample points in a spatial context. We will use the function `spsample()` to develop several of the sampling plans in this section. The code that generates the random sampling pattern shown in Figure 5.4 is as follows.

```
set.seed(123)
spsamp.pts <- spsample(sampbdry.sp, 32, type = "random")
```

The `SpatialPolygon` object `sampbdry.sp` describes the region in which to sample and was created in Section 5.2.1. The second argument in `spsample()` above specifies the sample size. The argument `type` is specified as `"random"`. The function `spsample()` selects a random sample of the specified size from the area bounded by the polygon

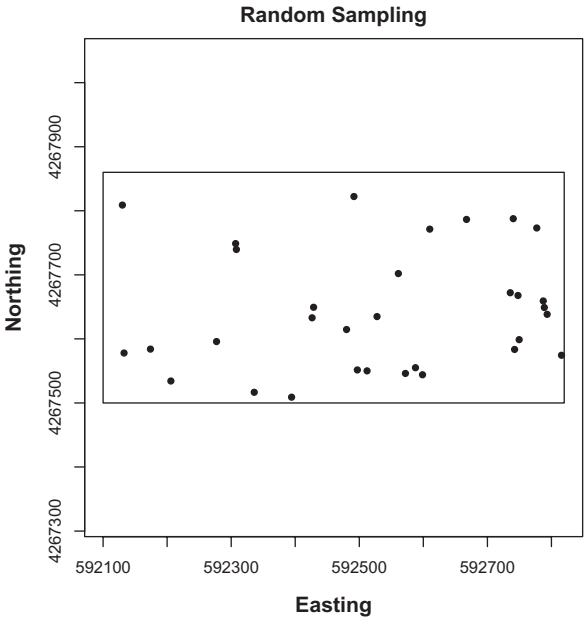


FIGURE 5.4
Thirty-two randomly sampled values from the population of size 10,368.

(Bivand et al., 2013b, p. 146). The coordinates of the sample points are contained in the `coords` slot of the `SpatialPointsDataFrame` object `spsamp.pts` and are accessible through the extractor function `coordinates()`.

Since we are dealing with a finite population defined at 10,368 locations, the sample points randomly selected by `spsample()` do not in general match the locations of the members of the population. Therefore, we will create a function `closest.point()` to determine the member of the artificial population whose location in (x, y) coordinates is closest to that of the sample point. The function `which.min()` in the fourth line returns the index of the member of the population with the smallest geographic distance to the given point.

```
closest.point <- function(sample.pt, grid.data){
   dist.sq <- (coordinates(grid.data)[,1] - sample.pt[1])^2 +
      (coordinates(grid.data)[,2] - sample.pt[2])^2
   return(which.min(dist.sq))
}
```

There is an `sp` function `spDistsN1()` that can be used to perform this task as well (Exercise 5.1).

We now create a function `rand.samp()` that computes the sample mean and the percent error (Equation 5.1) of a random sample drawn from the population. This function is rather long, so a liberal sprinkling of comments is included.

```
> rand.samp <- function (samp.size){
+ # Create the locations of the random sample
```

```
+    spsamp.pts <- spsample(sampbdry.sp, samp.size, type = "random")
+ # Extract a two column array of the x and y coords
+    sample.coords <- coordinates(spsamp.pts)
+ # Apply the function closest.point() to each row of the
+ # array sample.coords (i.e., each sample location)
+    samp.pts <- apply(sample.coords, 1, closest.point,
+       grid.data = pop.data)
+ # Each element of samp.pts is the index of the population value
+ # closest to the corresponding location in sample.coords
+    data.samp <- pop.data[samp.pts,]
+    samp.mean <- mean(data.samp$Yield)
+    prct.err <- abs(samp.mean - pop.mean) / pop.mean
+    return(c(samp.mean, prct.err))
+}
```

We are now ready to carry out the Monte Carlo simulation of this process. The function `closest.point()` remains in memory and is accessed from inside the function `rand.samp()` through lexical scoping (Section 2.5).

```
> samp.size <- 32
> set.seed(123)
> U <- replicate(1000,rand.samp(samp.size))
> print(mean(U[2,]), 3) # Equation (5.1)
[1] 0.0365
> print(sd(U[1,]), 4)
[1] 205.9
```

The mean percent error is 3.7%, and the standard deviation of the means is 205.9 kg ha^{-1}. The histogram (not shown) of the set of means is relatively normal in appearance, as it should be according to the central limit theorem (Section 3.3). One of the assumptions of the central limit theorem as stated in Larsen and Marx (1986, p. 322) is the independence of the random variables. However, the theorem can also be shown to hold in a form sufficient for our uses (Bolthausen, 1982; Guyon, 1995, p. 111; Haining, 2003, p. 274) when the data are spatially autocorrelated. Specifically, Bolthausen (1982) shows that the central limit theorem remains valid for spatially autocorrelated data provided the autocorrelation tends to zero as the distance between data values increases.

By changing the value of the sample size `samp.size` in the first line of the code sequence above, one can determine the effect of this quantity on the sample accuracy and precision. The mean percent errors of the sample of means of sizes 162 and 288 are 1.6% and 1.2%, respectively. The standard deviations of the sample of means of sizes 162 and 288 are 90 and 72 kg ha^{-1}, respectively. Even in the case of non-spatial data, when the data have some form of structure, random sampling may produce an imprecise result (Cochran, 1977, p. 89). With spatial data, it is often a poor choice. In the next subsections we will compare common spatial sampling patterns with random sampling and with each other.

5.3.2 Geographically Stratified Sampling

There are two primary problems with random sampling of a spatial region. The first is that it is time consuming to plot out a path and travel to the randomly selected points, and the second (much more serious) problem is that large portions of the region may happen to go

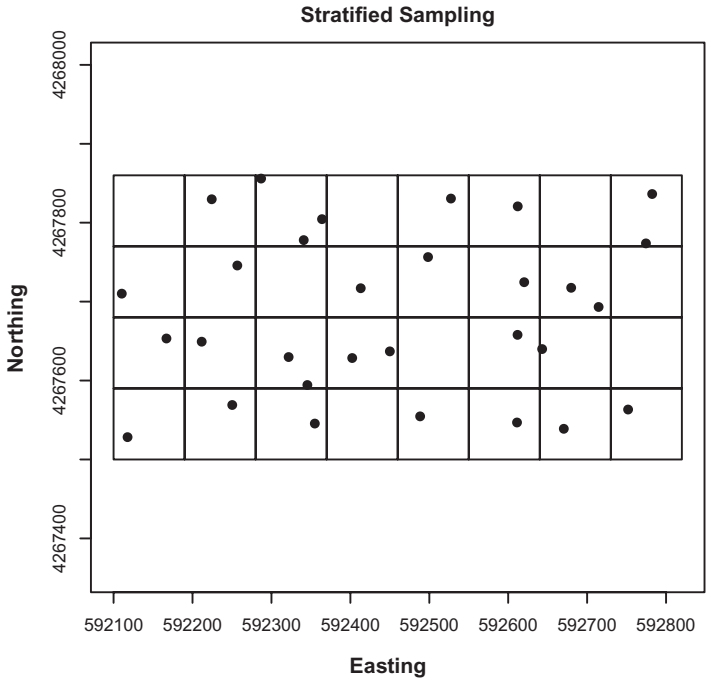


FIGURE 5.5
Stratified sampling on the basis of a subdivision of the field into 32 strata. The strata are generated internally in `spsample()` and do not conform exactly to the strata illustrated here.

unsampled, and some regions may be oversampled. For example, in the random sample illustrated in Figure 5.4, few data are collected in the western part of the field, and many of the sample points at the eastern end are so close together as to be almost duplicative. One way to increase the dispersion of the sample sites is to stratify the field geographically and to sample randomly within each stratum (Webster and Oliver, 1990, p. 43). In Figure 5.5 the field has been divided into 32 equal square strata.

A square or rectangular stratification such as that pictured in the figure is very easily constructed using the `sp` function `GridTopology()` (Bivand et al., 2013b, p. 48). First, we calculate `x.size` and `y.size`, the cell size in the x and y directions based on an 8 by 4 arrangement of the cells and the boundary of the sampling region as defined in Section 5.2.1.

```
> print(x.size <- (E - W) / 8)
[1] 90
> print(y.size <- (N - S) / 4)
[1] 90
```

In this case, the lattice cells are 90 m by 90 m squares. Next, we use the function `GridTopology(cellcenter.offset, cellsize, cells.dim)` to generate the geographic strata. Each of the arguments is a vector of two components, representing the x and y directions. The first argument represents the position of the center of the lower left cell relative to the point (0,0); the second argument represents the cell side length in the x and y direction, and the third argument represents the number of cells in each direction.

```
> x.offset <- W + 0.5 * x.size
> y.offset <- S + 0.5 * y.size
> samp.gt <- GridTopology(c(x.offset, y.offset),
+   c(x.size, y.size), c(8,4))
> samp.sp <- as(samp.gt,"SpatialPolygons")
```

After constructing the `GridTopology` object `samp.gt`, the next line uses the coercion function `as()` to convert this into a `SpatialPolygons` object.

For a sample of size 32, there is one sample per stratum; for a sample of size 288, there are 9 samples per stratum. The advantage of stratified sampling of autocorrelated data is that the sample does not miss any large geographic regions. Geographically stratified sampling of agricultural fields is discussed by Wollenenhaupt et al. (1994, 1997). The function `spsample()` accepts `"stratified"` as one of the possible values of the argument `type`. In this case, a geographic stratification is computed based on the sample size, and the appropriate number of samples are randomly located in each stratum. The stratification is carried out internally and does not perfectly match the division of the field into square regions as shown in Figure 5.5, but it is evident that the sample points are more evenly distributed than those of Figure 5.4.

Once again, we can conduct a Monte Carlo simulation with 1000 runs. The only change necessary in the previous Monte Carlo simulation test is to change the value of `type` in the arguments of `spsample()` to `"stratified"`. For samples of size 32, 162, and 288 the mean percent errors are 2.6%, 0.8%, and 0.6% (Table 5.1). The standard deviations of the sample of means of geographically stratified samples of sizes 32, 162, and 288 are 143, 44, and 31 kg ha^{-1}, respectively, indicating an improvement in precision over that of random sampling. Using the relative efficiency defined in Equation 5.2, the results of our Monte Carlo simulation indicate that the relative efficiency in estimation of the global mean of random sampling to stratified sampling is between 32% for a sample size of 32, and 42% for a sample of size 288.

5.3.3 Sampling on a Regular Grid

Stratified random sampling alleviates the second of the two objections raised against simple random sampling, the poor coverage of large portions of the field, but it does not alleviate the first, the complexity of the sampling plan. The simplest sampling plan one could carry out is to sample the data on a regular grid. For example, we can sample the field on a 12 by 24 grid to obtain 288 sample points, a 9 by 18 grid to get 162 points, and a 4 by 8 grid

TABLE 5.1

Mean Percent Error (Equation 5.1) and Standard error of the Sample Mean Obtained in 1000 Monte Carlo Simulation Process for Each of Three Random Sampling Methods.

	Random		Geographically Stratified		Stratified by Covariate	
	Mean % error	Std. dev. of means	Mean % error	Std. dev. of means	Mean % error	Std. dev. of means
32	3.7	206	2.6	143	2.3	132
162	1.6	90	0.8	45	1.0	59
288	1.2	72	0.6	31	0.8	45

Note: For purposes of comparison, the corresponding percent errors for grid sampling with 32, 162, and 288 sample locations are 1.6%, 0.1%, and 0.3%, respectively.

to get 32 points. The function `expand.grid()` can be used to generate grid-based sampling plans. The function `spsample()` also accepts the argument `type = "regular"` to generate a regular grid (see Exercise 5.3). Here is the code to compute a 4 by 8 grid using the function `expand.grid()`.

```
> nrows <- 4
> ncols <- 8
> grid.size <- (E - W) / ncols
> grid.offset <- 0.5 * grid.size
> spsamp.pts <- expand.grid(x = seq(W+grid.offset, E,grid.size),
+   y = seq(N - grid.offset,S,-grid.size))
> samp.pts <- apply(spsamp.pts, 1, closest.point, grid.data = pop.data)
> data.samp <- pop.data[samp.pts,]
> print(abs(mean(data.samp$Yield) - pop.mean) / pop.mean, digits = 4)
[1] 0.0002843
```

The quantity `grid.offest` in line 4 of the code is computed to center the grid within the boundary of the rectangle. Figure 5.6 shows a regular square 4 by 8 grid imposed on the test population. The percent error of grid sampling for this particular experiment is 0.03% for a 32-point grid, 0.3% for a 162-point grid, and 0.09% for a 288-point grid (note that by chance the 32-point grid produces the best result). This is considerably better than the performance of the other methods (Table 5.1). Of course, as always with these numerical experiments, this result is highly dependent on the particular data set, particularly the

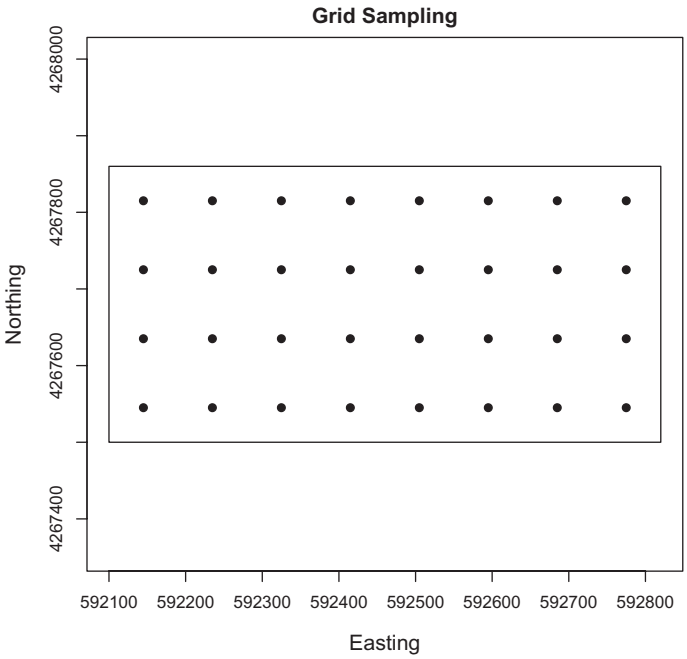


FIGURE 5.6
A regular square grid of 32 points.

very low error of the 32-point sample. Nevertheless, we shall see later that in an important way grid sampling is very close to an optimal sampling scheme.

In thinking about the possibility of carrying out a Monte Carlo simulation experiment for grid-based sampling analogous to those for simple and stratified random sampling, we can see at a glance a fundamental problem. The regularity of the grid removes the randomness of the sample points relative to one another. An element of randomness can be inserted into the sample by randomly locating the coordinates of one corner (Webster and Oliver, 1990, p. 45; Cochran 1977, p. 206). In practice, however, one ordinarily places the grid in such a way as to maximize the distance from any grid point to the field boundary, in order to minimize edge effects. In this case, there is no randomness at all, and it becomes impossible to meaningfully estimate the standard error in the same way as is done with sampling plans involving randomness in sample location. We will return to this issue in Section 5.6.

5.3.4 Stratification Based on a Covariate

The previous sampling schemes made no use of information about the properties of the site being sampled. In many cases, the investigator has information about how various aspects of the site's geography influence the response variable, and such information may be useful in increasing sampling efficiency. Indeed, in classical, non-spatial sampling, stratification is generally based on a covariate rather than on spatial location (Cochran, 1977). We can illustrate this approach for our particular site by utilizing the relationship between weed infestation level and yield. During the course of the study in Field 4.2 in 1996, it became evident that a weed infestation existed in the western part of the field that could be expected to lead to substantial yield loss (as indeed it did). We can compute the effect of weed infestation level on yield as follows. First, we use the function `closest.point()` to determine the yield of the member of the artificial population located closest to each of the 78 sample points.

```
> data. Set4.2 <- read.csv("set4\\set4.296sample.csv", header = TRUE)
> # Extract the coords
> sample.coords <- cbind(data. Set4.2$Easting, data. Set4.2$Northing)
> # Find the yield at the closest sample points
> samp.pts <- apply(sample.coords, 1, closest.point, grid.data = pop.
data)
> data. Set4.2$Yield <- pop.data$Yield[samp.pts]
> with(data. Set4.2, print(tapply(Yield, Weeds, mean), digits = 4))
   1    2    3    4    5
5419 5184 5286 4724 3129
```

The last line is an example of the use of the functions `with()` and `tapply()` (`tapply()` is described in Section 2.3.2). The function `with()` evaluates a function specified by its second argument in an environment constructed by the data in its first argument. Thus (ignoring the `print()` function) the statement is equivalent to `tapply(data. Set4.2$Yield, data. Set4.2$Weeds, mean)`.

When stratifying by a covariate one could use either random or grid-based sampling within the strata. We will employ random sampling in order to compare this sampling plan with geographic stratification. The region of high weed infestation levels was visible on the ground, and is also apparent in the infrared aerial image taken in May at the end of the season. At this time, the crop was already senescent, but the weeds were still vegetative and therefore show up as brighter in the infrared band. Figure 5.7 shows the location of high weed levels (*Weeds* = 5), as indicated by the larger circles. The locations are

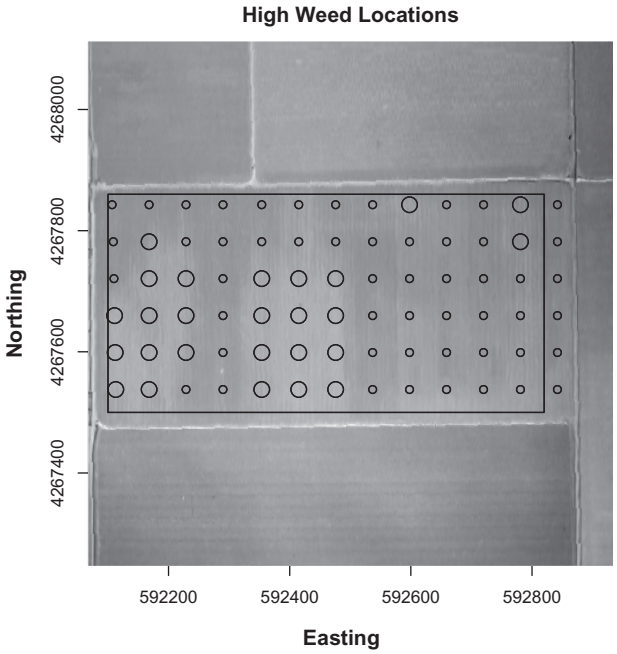


FIGURE 5.7
Locations of high weed infestation level at the sample points of Field 4.2, shown on top of the infrared band of an aerial image taken in May 1996. The image was scanned from a positive film, and there are faintly visible concentric circles in the northwest part of the image of the field and on the eastern boundary. These are Newton rings, caused by molecular interaction between the film and the scanner's glass plate.

superimposed on the infrared band of the aerial image taken in May 1996. The boundary of the sampling area is also shown. The code used to create this figure is discussed in Section 2.6.2. The first step in the process of stratified sampling based on the value of the covariate *Weeds* is to develop a boundary file that delimits the strata. At the time of the sampling campaign, the actual effect of weed level on yield was unknown. The observation of very high weed levels would, however, lead to an anticipation of yield loss, so the appropriate response would be to stratify by the high weed and low weed areas, where the high weed areas are defined as those in which the weed level has the value 5. For this application, we create a set of polygons delimiting only the high weed areas, using code similar to that used to create the boundary file, but with four individual polygons instead of one. Coordinates defining these polygons can be determined in a geographic information system or, alternatively, by using the function `locator()`.

```
> #Create boundary file
> x1 <- 592165
> x2 <- 592240
> x3 <- 592320
> x4 <- 592490
> x5 <- 592580
> x6 <- 592620
> x7 <- 592760
> y1 <- 4267690
> y2 <- 4267810
> y3 <- 4267760
```

There are four polygons defining areas of high weeds. Because they are easier to work with, we use special features to create an `sf` polygon file and then coerce this into an `sp` object. The code is a modification of that introduced in Section 2.4.3.

```
> strat1 <- matrix(c(W, y1,x1,y2,x2,y2,x2,S, W,S, W,y1), ncol = 2,
+    byrow = TRUE)
> strat2 <- matrix(c(x3,S, x3,y3,x4,y3,x4,S, x3,S), ncol = 2,
+    byrow = TRUE)
> strat3 <- matrix(c(x5,N, x6,N, x6,y2,x5,y2,x5,N), ncol = 2,
+    byrow = TRUE)
> strat4 <- matrix(c(x7,N, E,N, E,y3,x7,y3,x7,N), ncol = 2, byrow = TRUE)
> strat.pol <- st_sfc(st_polygon(list(strat1)),
+    st_polygon((list(strat2))), st_polygon(list(strat3)),
+    st_polygon(list(strat4)))
> stratbdry.sf <- st_sf(z = c(1,1,1,1), strat.pol)
> st_crs(stratbdry.sf) <- "+proj=utm +zone=10 +ellps=WGS84"
> stratbdry.sp <- as(stratbdry.sf, "Spatial")
```

Figure 5.8 shows the high weed strata created for this example.

In order to obtain an unbiased estimate using stratified sampling, the fraction of the total number of samples taken from each stratum must be equal to the relative size of that stratum in relation to the total population size (Cochran, 1977, p. 117). In our case, this means that the fraction of the samples that are carried out in the high weed stratum must

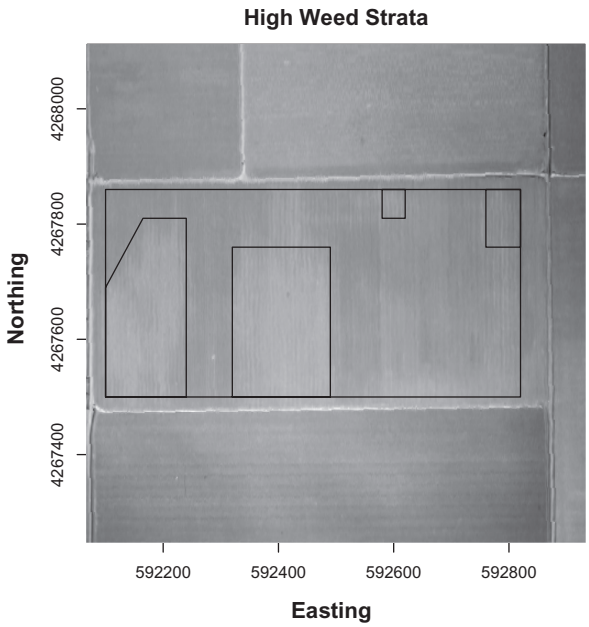


FIGURE 5.8
High weed sample strata in Field 4.2.

be equal to the fractional area occupied by this stratum. The relative size of the strata in spatial sampling can be determined from their geographic areas. It is easy to obtain the areas of the polygons of the sf objects using the function st _ area(). This returns the area of each polygon

We can also compute the total area.

```
> print(hi.area <- st_area(stratbdry.sf))
Units: m^2
[1] 39500 44200 2000 6000
> print(tot.area <- st_area(sampbdry.sf))
259200 m^2
> print(frac.hi <- as.numeric(sum(hi.area) / tot.area))
[1] 0.3537809
```

The function st _ area() returns a value in the map units, so we must use as.numeric() to coerce this into a pure number.

To carry out a Monte Carlo simulation using the artificial population, we must separate the artificial population into two subpopulations, one contained in the high weeds stratum and one in the low weeds stratum. This can be done with the function over()of the sp package (Pebesma, 2018).

```
> # over(pts, polygon) produces a data frame with NA
> # for points outside the polygon
> hiweeds <- over(pop.data, stratbdry.sp)
> head(hiweeds)
   z
1 NA
2 NA
3 NA
4 NA
5 NA
6 NA
> length(hiweeds$z)
[1] 10368
> length(which(!is.na(hiweeds$z)))
[1] 3668
```

The first argument of over() is pop.data, which contains the sample points and is a SpatialPointsDataFrame. The second argument stratbdry.sp, which defines the boundary of the high weed stratum, is a SpatialPolygons object. (By the way, if one of the functions in the argument of over() was originally created using an sf method and another was originally created using an sp method, you may get an error message. This is due to a trivial difference in these methods' use of the function proj4string() and can be corrected by simply setting projections when both are sp objects.)

When the first and second arguments respectively of a call to over() are members of these classes, the function returns a data frame whose one data field is a set of integers of the same length as the number of points in the first argument. Each element corresponds to a point. For those points that fall inside a polygon in the second argument (stratbdry.sp in this case), the value of the corresponding element of the data field is the index of the polygon

in which the point falls. For those points that do not fall inside a polygon of `stratbdry.sp`, the value of the corresponding element is `NA`, as it is in this case for the first six records. Since we are aggregating all polygons together as indicating high weeds, any element of the array whose value is not `NA` corresponds to a point in the high weeds stratum. We can, therefore, create a data field `hiweeds` of the artificial population data object `pop.data`, initially assign all of its records the value zero, and then assign the value 1 to all those records whose index is not `NA`.

```
> pop.data$hiweeds <- 0
> pop.data$hiweeds[which(!is.na(hiweeds$z))] <- 1
```

We next create a function `stratified.sample()` to use in the simulation in the same way that the function `random.samp()` was used earlier. In this case, the argument `samp.size` is an array with two elements. The first is the number of samples from the low weed stratum, and the second is the number of samples from the high weed stratum. The object `subpop` is created twice, to contain the elements of both subpopulations.

```
> stratified.samp <- function(samp.size){
+ # Low weed stratum
+   subpop <- pop.data$Yield[pop.data$hiweeds == 0]
+   samp <- sample(subpop, samp.size[1])
+ # High weed stratum
+   subpop <- pop.data$Yield[pop.data$hiweeds == 1]
+   samp <- c(samp, sample(subpop, samp.size[2]))
+   samp.mean <- mean(samp)
+   prct.error <- abs((samp.mean - true.mean) / true.mean)
+   return(c(samp.mean, prct.error))
+ }
```

The quantity `frac.hi` computed above is used to calculate the sample sizes in each stratum.

```
> set.seed(123)
> sample.size <- 32
> hi.size <- round(frac.hi * sample.size, 0)
> lo.size <- sample.size - hi.size
> samp.size <- c(lo.size, hi.size)
> U <- replicate(1000, stratified.samp(samp.size))
> mean(U[2,])
[1] 0.02394189
> sd(U[1,])
[1] 132.7480
```

Figure 5.9 shows a sampling pattern for stratified random sampling with 32 sample locations.

Table 5.1 contains the mean percent error and the standard deviation of the estimated means of 1000 Monte Carlo simulations for each of the three random sampling methods we have tested. For this particular case, geographic stratification and stratification

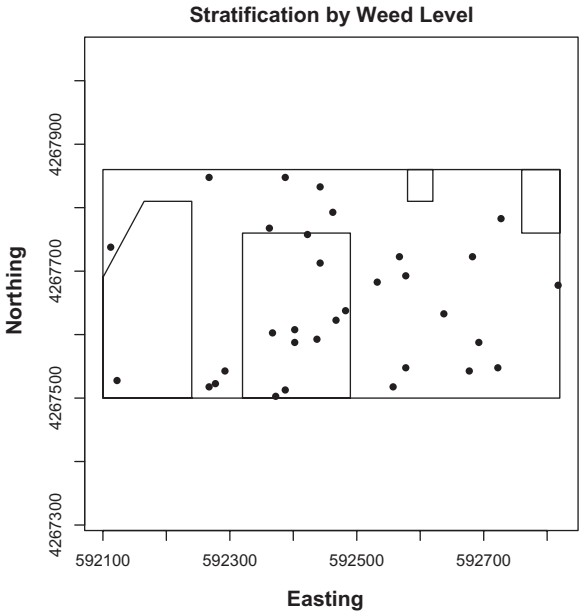


FIGURE 5.9
A sampling plan with 32 sample locations stratified by weed level.

by a covariate give similar results. This is not true in every case, but it is generally true that stratification provides better results than simple random sampling. Depending on how many strata are created in sampling by a covariate, part of the effect of this stratification may be to stratify geographically and spread the sample points more evenly.

5.3.5 Cluster Sampling

In every ecosystem, spatial variability exists on multiple scales. In Section 6.4.1, a formal model is provided for this phenomenon, but even without such a formal model one can recognize the value of gathering data on the spatial relationships of nearby locations within a site as well as relatively distant locations. Cluster sampling provides, in theory, a means of gaining precision in the estimation of short range spatial interaction effects without requiring an impossibly large number of samples. For a fixed sample size, the potentially increased local precision of cluster sampling comes at the expense of reduced coverage density of the entire site. As with stratification based on a covariate, cluster sampling can be carried out using either a random or a grid-based plan, and indeed it could be combined with any of the plans already discussed. Because its primary utility is in the improved estimation of statistical properties involving short range variability, we would not expect cluster sampling to necessarily provide improved estimates of a global statistic such as the mean. Therefore, there seems to be no particular advantage to testing a random

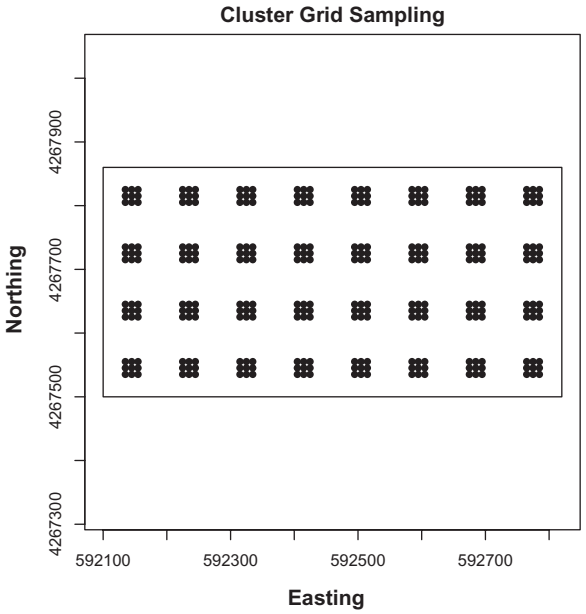


FIGURE 5.10
A cluster sampling grid consisting of 32 clusters with centers separated by 50 m, with nine sample locations in each cluster, separated by 10 m.

sampling plan, so instead we will develop a grid-based cluster sampling plan. We will be testing cluster sampling on the estimation of the mean, as with the other plans, but the real interest in this sampling plan is its performance in variogram estimation, which is discussed in Section 5.4.

Figure 5.10 shows a cluster sampling plan consisting of 32 clusters of 9 sample points each, for a total of 288 sample locations. Within each cluster, the nine sample locations are 10 m apart. The sampling plan has an error in estimation of the mean of 0.7%. This is more accurate than any of the randomization methods of Table 5.1, but for this data set it is not as accurate as the 0.3% error of the regular grid at 288 locations.

5.4 Methods for Variogram Estimation

The estimation of a single, globally based statistic such as the mean does not tell us anything about the spatial properties of the sample domain. For this we need to estimate a spatial descriptor such as one of those discussed in Chapter 4. The vast majority of the literature on this subject is devoted to estimation of the variogram (Section 4.6.1). Recall that the equation of the experimental variogram is given by (Equation 4.21),

$$\hat{\gamma}(h) = \frac{1}{2m(h)} \sum_{i=1}^{m(h)} [Y(x_i) - Y(x_i + h)]^2, \tag{5.3}$$

where h represents a lag and $m(h)$ is the number of lag vectors of lag h. As discussed in Section 4.6.1, lag vectors are generally grouped into lag groups $H(k)$ defined by those of magnitude $kh_0 \le h < (k+1)h_0$ for some fixed h_0. A reasonable size of the lag group is required to compute a good estimate $\hat{\gamma}$, and $\hat{\gamma}$ must be estimated for several lag groups, so it is evident that a lot of data values are needed to adequately estimate the variogram. Webster and Oliver (1990, 1992) state that between 100 and 200 samples should be used to generate an adequate experimental variogram.

In comparing sample plans for estimation of the variogram, it is necessary to define the standard of comparison. One possibility is to consider some direct measure of the variogram itself, such as a confidence interval. Russo (1984), for example, proposes a measure of quality based on the aggregation of the measurement accuracies of the individual values of the variogram estimate $\hat{\gamma}(h)$ at each lag h. It is generally simpler and more satisfactory, however, to consider the role of the variogram in kriging (Section 6.3.2) and to use the kriging variance (Isaaks and Srivastava, 1989; see Section 6.3.2) as the standard of comparison. The kriging variance does not depend directly on the values of the response variable $Y(x, y)$ but only on the number and locations of the sample points (x, y) and the values of the variogram $\hat{\gamma}(h)$ (Cressie, 1991, p. 315; Webster and Oliver, 1990, p. 262). The values of $Y(x, y)$ enter into the kriging variance through their influence on $\hat{\gamma}(h)$. An important special case is that in which the data are isotropic, that is, their properties relative to one another do not depend on direction (Section 3.2.2). It may be necessary to modify the grid in the case that the data are anisotropic. Cressie (1991, p. 323), and Webster and Oliver, (1990, p. 277) discuss this case.

To carry out an informal comparison of sampling plans for variogram estimation, we again use the artificial yield population of Figure 5.1a. In addition to (1) the population variogram, the following five sampling methods are tested: (2) a regular 12 by 24 grid (288 samples), (3) a regular 8 by 16 grid (128 samples), (4) a regular 6 by 12 grid (72 samples), (5) a sample of 288 randomly selected locations, and (6) the cluster sample of Figure 5.10. Figure 5.11 shows the results of the comparison of each of these with the variogram of the entire population.

In this example, the random sample (Figure 5.11d) generally performs about as well as the regular grid sample (Figure 5.11a) of the same size. The 128 and 72 point samples generally perform worse. Cluster sampling does much worse, even at small lag distances. Cressie (1991, p. 317) points out that cluster sampling is often poorly adapted to spatial data.

McBratney et al. (1981) carried out a comparison of sampling plans in accuracy of estimation of the variogram. Their results can be interpreted intuitively as indicating the optimal plan is one in which the sample points are mutually as far away from each other as possible. The absolute maximum mutual distance is achieved by using an equilateral triangular sampling grid, but a nearly equal mutual distance is achieved with a regular square grid, which is much easier to lay out and execute (McBratney et al., 1981).

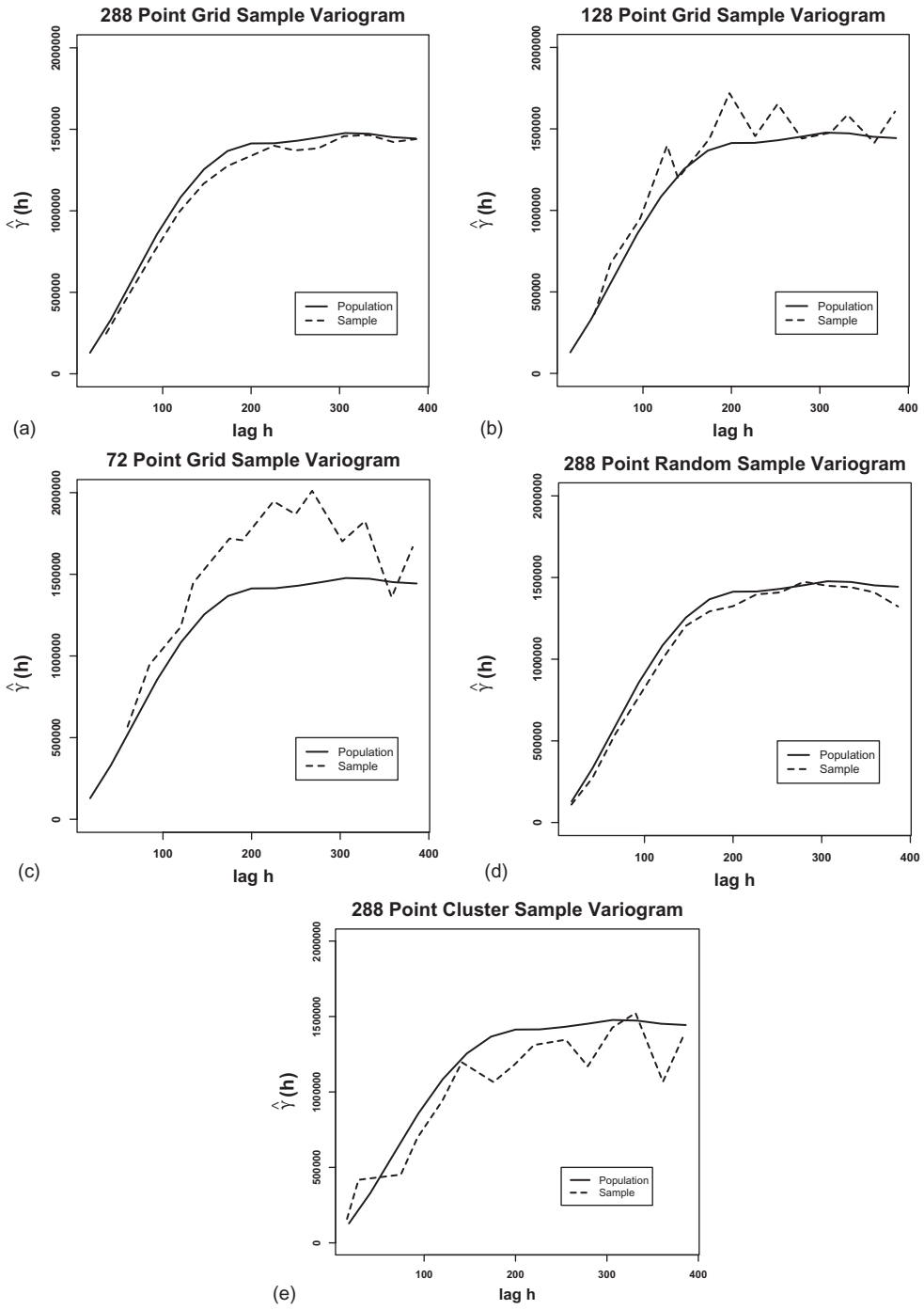


FIGURE 5.11
Plots of the comparison between the population variogram $\gamma(h)$ and the experimental variograms $\hat{\gamma}(h)$ for the artificial yield data. The variogram of the full population is shown as the solid line and that of the sample as a dashed line; (a) grid sample of size 288; (b) grid sample of size 128; (c) grid sample of size 72; (d) random sample of size 288; (e) cluster sample as in Figure 5.8.

5.5 Estimating the Sample Size

Once a sampling pattern has been established, the next step is to establish the sample size. As a practical matter, every project with which I have ever been involved used the following formula to determine the sample size:

$$\text{Number of samples} = \frac{\text{Money in the budget}}{\text{Cost per sample}} \tag{5.4}$$

Nevertheless, it is very useful to calculate an estimated sample size, if only to determine how close one can come to the ideal. Also, the process of estimating the sample size introduces a very important concept: the relationship between sample size and data variability. The number of samples necessary to estimate a parameter or test a hypothesis depends on the desired precision and the variability of the data. Many methods for estimating sample size use an independent estimate of variability, as measured by the standard error, to determine the number of samples required to achieve a given level of precision in the estimate. We consider first a simple example that illustrates some of these principles. The example is the estimation of the mean μ of a normally distributed population from a set of *independent*, identically distributed random variables drawn from the population. Assume that we want to estimate μ with a $(1-\alpha)\times 100$ percent confidence interval of width no larger than 2δ. The formula for the confidence interval is (Sokal and Rohlf, 1981, p. 148; Kutner et al., 2005, p. 1306)

$$\bar{Y} - t(1-\alpha/2; n-1)s\{\bar{Y}\} \le \mu \le \bar{Y} + t(1-\alpha/2; n-1)s\{\bar{Y}\}, \tag{5.5}$$

where $\bar{Y}$ is the sample mean, $t(1-\alpha/2; n-1)$ is the $1-\alpha/2$ percentile of the t distribution with $n-1$ degrees of freedom, and $s\{\bar{Y}\}$ is the standard error, given by

$$s\{\bar{Y}\} = \frac{s}{\sqrt{n}}, \; s = \sqrt{\frac{\Sigma(Y_i - \bar{Y})^2}{n-1}}. \tag{5.6}$$

Therefore, if the confidence interval is to have a width no larger than 2, we must have $2\delta \ge 2t(1-\alpha/2; n-1)s/\sqrt{n}$, which implies

$$n \ge \frac{t(1-\alpha/2; n-1)^2 s^2}{\delta^2}. \tag{5.7}$$

Thus, an estimate of the sample variance s^2, usually drawn from a preliminary sample, is required to estimate the sample size. The important principle is that Equation 5.6, which defines a measure of the variability of the mean in terms of the sample size n, is inverted to form Equation 5.7, which expresses the sample size n in terms of the variability. One then uses some independent estimate of variability to compute the sample size.

One method of implementing inequality (Equation 5.7) to estimate sample size in developing a sampling plan is called *Stein's method* (Steel et al., 1997, p. 124). This method requires a preliminary sample of size n_0. From this sample, one computes the sample variance s_0^2 and establishes n as the smallest integer greater than or equal to $t(1-\alpha/2; n_0-1)s_0^2/\delta^2$.

There is a considerable literature on sample size estimation, much of which is discussed by Cochran (1977, Ch. 4). This literature, however, is valid for samples of independent data but may not retain its validity for spatial data.

A common but simplistic means of estimating the spatial density of a grid sample is to use range of the variogram, that is, the value of the lag h at which the variogram $\gamma(h)$ approaches its asymptote (Section 4.6.1). This represents a measure of the spatial distance between which data values are uncorrelated. As such, the sites in a sampling grid should be separated by a distance no greater than the range if some measure of the entire sampling region is to be obtained. This is a fairly blunt instrument, however. The number of data values required for a sample is also dependent on the objective of the sampling campaign. In the variogram estimation example of Section 5.4, the sample spacings for the 288, 128, and 72 point samples were 30, 45, and 60 m, respectively. Inspection of Figure 5.11 indicates that the range is about 200 m. Thus, the estimation of the variogram requires sampling at a spacing considerably smaller than the range. On the other hand, the estimation of the global mean can sometimes be done with reasonable accuracy using a larger grid spacing.

One way to determine the appropriate sample size for a spatially autocorrelated population is to estimate the *effective sample size* n_e. This quantity, which is discussed at considerable length in Chapter 10, is the size of the sample that would yield the same sample variance if the population was not autocorrelated as a sample of size n of the actual population. If one can compute an estimate $\hat{n}_e$ of n_e, then one can determine the appropriate sample size n by substituting $\hat{n}_e$ into the right-hand side of inequality (Equation 5.7). Chapter 10 provides a simple, *ad hoc* method of computing $\hat{n}_e$ using bootstrapping. Griffith (2005) describes several methods based on the theories developed in Chapters 12 and 13.

5.6 Sampling for Thematic Mapping

In many applications, the objective is not to estimate a statistic but rather to develop a thematic map of the sample region based on one or more themes (Webster and Oliver 1990, p. 273). For example, in an ecological application one may wish to develop a thematic map of a meteorological quantity such as mean annual precipitation or maximum July temperature. An example in agriculture is the so-called "management zones" of site-specific crop management (Lowenberg-DeBoer and Erickson, 2000). We will characterize the general problem of thematic mapping as devising a sampling plan such that some attribute or attributes may be classified at each location according to a finite set of categories. The main point is that we want the estimate to be as accurate as possible at all locations, as opposed to merely seeking an estimate of a global statistic or a representation of variability.

The sampled quantity Y commonly takes on values $Y(x, y)$ continuously, or piecewise continuously (i.e., allowing for jumps at some locations) as a function of position. For example, soil clay content or precipitation could be assumed to have this property. If the scale is large enough, quantities that are not continuously distributed in this way, such as species population density, can be modeled for the purpose of mapping as if they were continuously varying in geographical space. The mapping problem in this case becomes one of finding the best interpolation method to estimate values of Y at locations other than the sample locations. This is a fundamental problem of geostatistics, and is briefly discussed in Section 6.4. Interpolation is well covered in other texts (e.g., Isaaks and Srivastava,

1989; Goovaerts, 1997; Webster and Oliver, 1990; Webster and Oliver, 2001; Cressie, 1991). A number of interpolation methods are described in these texts, including inverse distance weighted interpolation, kriging, and spline interpolation. In terms of sampling plan development, kriging has the advantage of permitting the estimation of the kriging variance, which permits one to determine the sample patterns that, for example, minimize the maximum value of the kriging variance. Webster and Oliver (1990, p. 272), Cressie (1991, p. 318), and McBratney et al. (1981) discuss optimization of sample patterns in this context. The general conclusion is that under most circumstances a regular grid provides the best combination of simplicity of layout and accuracy of estimation.

Although not always the case, it frequently happens that the estimate of a particular variable is based on two or more measured quantities, that the measurement cost of the different quantities is quite different, and that the most expensive data to collect provide the most accurate estimate. For example, in estimating soil clay content the most accurate data comes from the extraction of soil cores, but apparent electrical conductivity is much cheaper and easier to measure. The question then becomes how to allocate the samples of the expensive quantity in such a way that the combination of all the measurements yields the most accurate map possible. In this case, sampling on a grid may not be the best approach for either commercial or scientific applications. Pocknee et al. (1996) provide a discussion of the drawbacks of grid sampling for this application. Probably the most commonly used method of estimation of the expensive-to-sample quantity via the cheap-to-sample quantity is cokriging (Isaaks and Srivastava, 1989, p. 400, Section 6.3.3). An alternative approach based on response surface analysis has been put forward by Lesch et al. (1995).

5.7 Design-Based and Model-Based Sampling

There are two separate approaches to the development of a sampling plan, the *design-based* approach and the *model-based* approach. In the design-based approach, also referred to as the *probability sampling* approach (Valliant et al., 2000), the population in the study region is viewed as having a fixed set of values (Haining 2003, p. 96). Sampling locations are selected according to some randomization scheme. This scheme is designed to ensure that it yields a parameter estimate with the desired statistical properties (e.g., unbiasedness or minimum variance). This reliance on a proper design is the source of the term "design-based" (Webster and Oliver 1990, p. 28). The random and stratified sampling plans discussed in Section 5.3 are examples of the design-based approach. There has been an increase in interest in using the model-based approach in spatial sampling (e.g., Haining, 2003; Griffiths, 2005), and for this reason we will give a simple example of how it might be applied to the artificial data set used in this chapter.

In the model-based approach, the population itself is considered to be one realization of a stochastic process. Other realizations are possible, and the population properties of interest, such as the mean and variance, are functions of the values of a random process. These properties are therefore themselves random variables and technically are predicted rather than estimated (only fixed parameters of a population are estimated). For this reason, model-based sampling is also called *prediction-based sampling* (Valiant et al., 2000). The process of developing a sampling plan involves developing a model for the random process generating the data and then estimating the parameters of this model.

Since the population in the study region is viewed as a random variable, there is no need to introduce randomness in the sampling pattern. Therefore, systematic sampling plans, such as the grid-based plans described in Section 5.3, can be studied statistically using a model-based formalism.

To make this distinction a bit clearer, consider the simple example in which a finite population of size N is being sampled, and the objective is to choose a sample of size n to estimate the population mean

$$\mu = \sum_{i=1}^{N} Y_i / N \tag{5.8}$$

by computing the sample mean

$$\bar{Y} = \sum_{i=1}^{n} Y_i / n \tag{5.9}$$

In the design-based approach, the population $\{Y_i, i = 1,...,N\}$ is considered as fixed, and the quantity μ is a fixed parameter. The sample $\{Y_i, i = 1,...,n\}$ is a random quantity dependent on the random selection of the n values to sample. The objective is to select a randomization process that generates a value $\bar{Y}$ that, according to some measure, optimally estimates μ. In the model-based approach, the population $\{Y_i, i = 1,...,N\}$ is viewed as a realization of a random process, and therefore $\mu(Y_1, Y_2, ..., Y_N)$ defined by Equation 5.8 is a random variable. The objective of sampling is to develop a sampling plan that optimally predicts μ. Suppose, for example, that the sampling plan was to sample every tenth value of the population. Under the assumptions of the design-based approach, since the population is fixed, this sampling pattern, if applied repeatedly, would yield the same sample values and the same estimate $\bar{Y}$ each time it was applied. Under the assumptions of the model-based approach each time the sampling pattern was applied it would sample a different realization of a random process, and so the sample values and the value of $\bar{Y}$ would be random variables. There is, of course, nothing special about the mean, and the same idea can be applied to the prediction of any other function of the random variable Y.

If one has a model for the relationship between Y and some explanatory variable X, then one can use this model to improve the accuracy of prediction. Valliant et al. (2000, p. 2) introduce the concept of model-based sampling with a simple example based on the number of patients discharged per day from a hospital versus the number of beds. We will begin our discussion with an analogous presentation using the relationship between wheat yield of the artificial data set and observed weed level at the nearest sample point. In their example, Valliant et al. (2000) estimate the total number of patients discharged in 33 hospitals given a sample consisting of the total number of patients discharged in 32 of them (equivalently, they estimate the number of patients discharged in the one non-sampled hospital). We will generate a similar example using the artificial yield population from Field 4.2.

To some extent, comparing design-based and model-based plans is a matter of comparing apples and oranges. Nevertheless, we will carry out an informal comparison using 32 sample points, the same value as the minimum size of the random and grid sample methods. We use the `raster` package to manipulate raster objects in the ways described by Lo and Yeung (2007, p. 183). Here we make only the simplest use of the package's capabilities to compute a simple linear regression between weed level and May infrared image digital

number. We will use the fact (Kutner et al., 2005, p. 24) that the regression line between Y and X passes through $(\bar{X}, \bar{Y})$, to estimate μ by computing the value of the regression line at $\bar{X}$). We emphasize that this is not necessarily the best way to carry out a model-based sampling plan, but it does illustrate the idea and it will enable us to demonstrate some issues associated with model-based sampling.

Figure 1.2 indicates that there is an apparently close relationship between the infrared band digital number of the May aerial image of Field 4.2 and the yield. We will base our prediction on this relationship. We start by loading the object `data.May.ras`, which contains the image band information.

```
> library(raster)
> data.May.ras <- raster("set4\\set4.20596.tif")
> class(data.May.ras)
[1] "RasterLayer"
attr(,"package")
[1] "raster"
```

By default, the function `raster()` loads band 1 of the *TIFF* file, which is the band that we want. We will use the spatial locations of the 32 grid sample points in Figure 5.6. We can apply the function `extract()` to the raster object to place the IR values of the cell containing each of the 32 sample points into the data frame `data.samp` created in Section 5.3.

```
> data.samp$IRvalue <- extract(data.May.ras, spsamp.pts)
```

Figure 5.12 shows a plot of yield vs. infrared band digital number together with the least squares regression fit. The code to generate the fit and compute the value of $\bar{Y}$ is as follows.

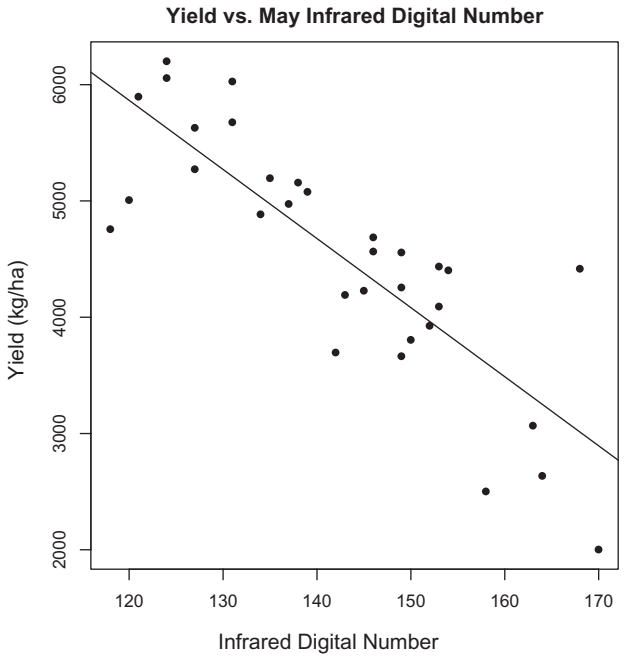


FIGURE 5.12
Linear regression model of yield vs. May IR value based on 32 sample locations.

```
> Yield.band1 <- lm(Yield ~ IRvalue, data = data.samp)
> summary(Yield.band1)
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 12996.28    1068.28  12.166 3.96e-13 ***
IRvalue       -59.42       7.46  -7.965 6.85e-09 ***
Multiple R-squared:  0.679,   Adjusted R-squared: 0.6683
```

The fit is reasonably good ($R^2 = 0.68$), which provides justification for the use of this model in our model-based sampling plan. The estimate of the population mean is obtained as the mean of the predicted values based on the linear regression model.

```
> print(Y.bar <- mean(predict(Yield.band1)), digits = 5)
[1] 4528.5
> print(abs(Y.bar - pop.mean) / pop.mean, digits = 3)
[1] 0.000284
```

The error in this particular example is less than that of grid sampling.

It is evident that the estimate of μ is highly dependent on accuracy of the model. This in turn depends in part on the choice of sample locations. This choice is important both for design-based sampling plans and for model-based sampling plans, but the use of a model-based plan provides an opportunity to illustrate this effect very graphically. We will consider three cases (Figure 5.13). The first is to sample seven points in a north to south transect in a high weed area; the second is to sample seven points in a north to south transect in a

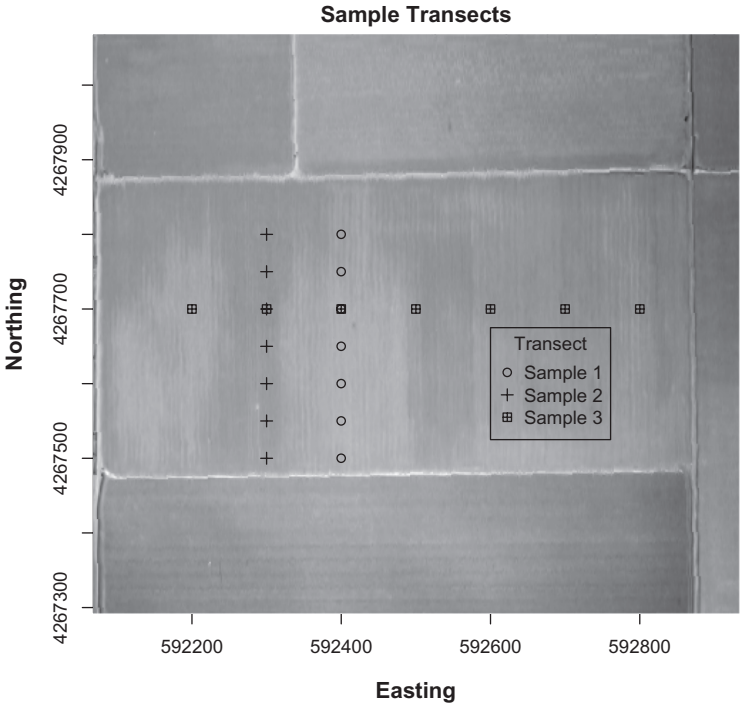


FIGURE 5.13
Three sample transects for model-based estimation of yield vs. May IR in Field 4.2.

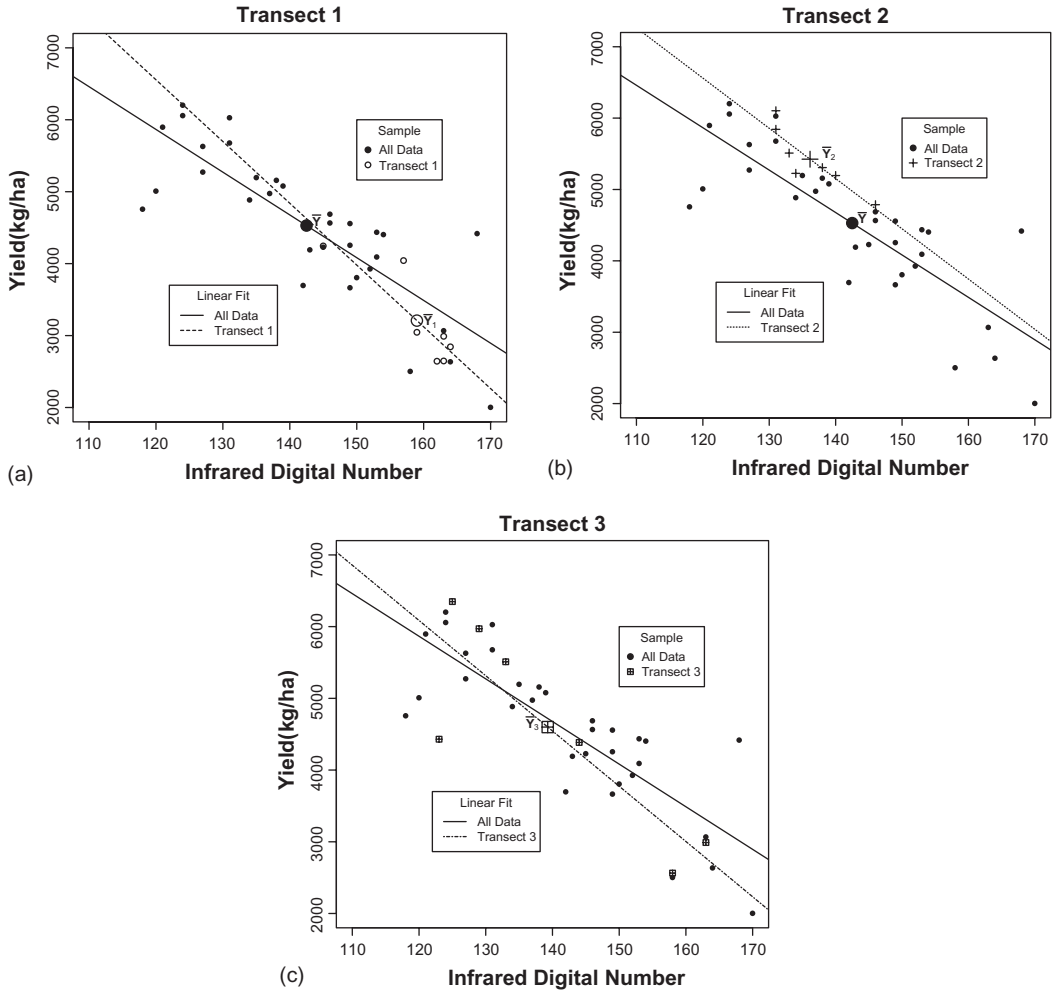


FIGURE 5.14
Regression relationships of the grid-based sample together with the models and the estimates based on the three sample transects in Figure 5.13. (a) Transect 1; (b) Transect 2; (c) Transect 3.

low weed area; and the third is to sample along an east to west transect spanning the field. The predicted yield means of the three samples are $\bar{Y}_1 = 3208$ kg/ha (29% error) for transect 1, $\bar{Y}_2 = 5424$ kg/ha (20% error) for transect 2, and $\bar{Y}_3 = 5000$ kg/ha (1.5% error) for transect 3. These results are summarized in Figure 5.14. The figure shows the data, the regression line, and the estimated mean of each of the transects. The first two transects (Figure 5.14a and b) give an incorrect representation of the yield-IR relationship because they are from regions of low and high IR value respectively, whereas the east–west transect covers the range of IR values and provides a fairly accurate representation. The estimate from the first transect (Figure 5.14a) is biased downwards, that from the second transect is biased upwards (Figure 5.14b), and that from the center transect is relatively accurate (Figure 5.14c). It is evident that care needs to be taken in selecting the sample locations. This applies equally to the results of a sampling plan using the design-based model.

The problem with both $\bar{Y}_1$ and $\bar{Y}_2$ is that they are based on samples from only a small geographic part of the field. Since the field has a strong geographic trend, this is equivalent to saying that they are based on only a small subset of the possible values of IR value and yield. Partly as a result, the regression models based on the first two sets of samples are incorrect, and since the samples are from either a higher than average range of IR values (in transect 1) or a lower than average range of IR values (in transect 2), the estimates are not robust to these incorrect models. Since the model based on the east–west transect is accurate, we cannot say anything about whether this sample is correspondingly robust to an incorrect model. We can say, however, that the most accurate estimate comes from a sample that is "representative" of the range of values of IR and yield. This concept of "representativeness" can be formalized through the property that a sample must be *balanced* (Valliant et al., 2000, p. 53), which means, roughly speaking, that each sample value represents, or is close to, about an equal fraction of the totality of values in the population.

5.8 Further Reading

Cochran (1977) is the classical reference for sampling, although it contains little in a spatial context. Valliant et al. (2013) provide an excellent recent discussion. The two books by Webster and Oliver (1990, 2001) contain a wealth of valuable material on sampling spatial data, as do Ripley (1981) and Haining (1990). Odeh et al. (1998) provide an excellent example of the use of information at multiple scales (see Chapter 6) to direct sampling. Brus (1994) describes a design-based stratified soil sampling plan. Valliant et al. (2000) provide a good introduction to model-based sampling. Olea (1984) and Lesch et al. (1995) discuss sampling plans that have model-based aspects. Edwards (2000) provides a good overview of sampling concepts for ecological data. The notion of distinguishing sampling error from nonsampling error is discussed by Biemer and Lyberg (2003). This concept goes back at least to Fisher (1935), who provides an excellent discussion of this issue.

The comparison of sampling plans in this chapter applies only to sampling a rectangular region. Van Groenigen and Stein (1998) and van Groenigen et al. (1999) provide mathematical methods for generating sampling plans on irregularly shaped regions that minimize the kriging variance. The mathematics of these plans is quite intricate, but simply by looking at the figures in the papers one can gain a good idea of how these sampling schemes relate to one that would be generated on a rectangular region. The package `spcosa` (Walvoort et al., 2010) uses a k-means clustering algorithm to develop sampling plans that distribute sample locations in an approximately uniform manner. The package is especially well suited for irregularly shaped regions, for which it can be used to construct compact strata of approximately equal area.

Exercises

5.1 Read about the sp function `spDistsN1()`. Create a new function `closest.point()` that uses this function.

5.2 a. Use the boundary file of Field 1 of Data Set 4 created in Exercise 2.11 and the function `spsample()` to create a regular grid sample plan with 100 sampling sites for the field. Use the function `points()` to add a plot of the sample point locations to the map.

b. Use the function `class()` to check the object class of the sampling plan created in part (a). Use the function `str()` to display the structure of the object. Use the function `coordinates()` to display the coordinates of the first 10 sample locations in the object.

c. Does the number of points in the sample plan equal the number you specified? Answer the question without counting the points (use the information provided by `str()`.).

5.3. Use the function `expand.grid()` to create a grid of sample points in Field 1 of Data Set 4 with the same spacing as that in Exercise 5.2. Use the function `coordinates()` to convert the object created by `expand.grid()` into a `SpatialPoints` object. Create a map showing the field boundary and the two sets of data locations, each with a different symbol.

5.4. It sometimes happens with an irregularly shaped boundary that the function `expand.grid()` creates sample locations outside the sample area boundary. Use the function `over()` to create a `SpatialPoints` object that does not include locations in the set created in Exercise 5.2 lying outside the field boundary. Create a map that shows this sample plan together with the field boundary.

5.5. Assume the 86 sample values are the entire population (i.e., $N = 86$) of clay content values in Field 1 of Data Set 4. Suppose you want to estimate the mean.

a. Compute the error in estimating the mean based on a random sample of six clay values.

b. Suppose you have EM38 values (which are much easier to measure) at all 86 locations, taken on April 25 from the beds (see Appendix B.4). You can collect six soil cores. Use the EM38 data to stratify the sample, creating two zones, one of high clay and one of low clay (make them the same size). Collect a random sample totaling seven samples within each zone and compute the error in estimating the mean. Remember that the strata cannot be of equal size.

c. Compare the result of parts (a) and (b) with estimate obtained by taking a north–south transect of seven soil cores consisting of every other data location in the middle column of the data starting from sample point 4.

5.6. Suppose in the problem of Exercise 5.4 you have taken a north–south transect of seven soil cores consisting of every other data location in the middle column of the data starting from sample point 4. Use this together with the EM 38 data to construct a model-based estimate of the mean of the 86 values of clay content.

5.7 Section 5.3.1 contains a discussion of an error in the implementation of the polymorphic function `idw()` that occurs when the `gstat` and `spatstat` packages are both loaded and R tries to implement the function from the wrong package. Ordinarily R selects polymorphic functions correctly. Why does it fail in this case?