1 Introduction and Background

This chapter gives a broad overview of the philosophy and techniques of ecological modeling. A small data set on seed removal illustrates the three most common frameworks for statistical modeling in ecology: frequentist, likelihood-based, and Bayesian. The chapter also reviews what you should know to get the most out of the book, discusses the R language, and spells out a step-by-step process for building models of ecological systems.

If you're impatient with philosophical discussion, you can read Section 1.4 and the R supplement at the end of the chapter and move on to Chapter 2.

1.1 Introduction

This book is about combining models with data to answer ecological questions. Pursuing this worthwhile goal will lead to topics ranging from basic statistics, to the cutting edge of modern statistics, to the nuts and bolts of computer programming, to the philosophy of science. Remember as we go along not to miss the ecological forest for the statistical trees; all of these complexities are in the service of answering ecological questions, and the most important thing is to keep your common sense about you and your focus on the biological questions you set out to answer. "Does this make sense?" and "What does this answer really mean?" are the two questions you should ask constantly. If you cannot answer them, back up to the last point you understood.

If you want to combine models with data, you need to use statistical tools. Ecological statistics has gotten much more complicated in the last few decades. Research papers in ecology now routinely refer to likelihood, Markov chain Monte Carlo, and other arcana. This new complexity arises from the explosion of cheap computing power, which allows us to run complicated tests quickly and easily—or at least more easily than before. But there is still a lot to know about how these tests work, which is what this book is about. The good news is that we can now develop statistical methods that directly answer our ecological questions, adapting statistics to the data rather than vice versa. Instead of asking "What is the probability of observing at least this much variability among the arcsine-square-root-transformed counts of seeds in different treatments?" we can ask "Is the number of seeds removed consistent with standard foraging theory, and what are the attack rates and handling times of predators? Do the attack rates or handling times increase with mean seed size? With the time that the seeds have been available? Is there evidence for variability among seeds?" By customizing statistical tests we can squeeze more information, and more relevant information, from expensive data. Building your own statistical tests is not easy, but it is really no harder than using any of the other tools ecologists have picked up in their ongoing effort to extract meaning from the natural world (stable isotope techniques, radiotelemetry, microsatellite population genetics, geographic information systems, otolith analysis, flow cytometry, mist netting ... you can probably identify several more from your own field). Custom statistical techniques are just another set of tools in the modern ecologist's toolbox; the information this book presents should show you how to use them on your own data, to answer your own questions.

For example, Sandin and Pacala (2005) combined population counts through time with remote underwater video monitoring to analyze how the density of reef fishes in the Caribbean affected their risk of predation. The classic approach to this problem would be to test for a significant correlation between density and mortality rate, or between density and predator activity. A positive correlation between prey population density and the number of observed predator visits or attacks would suggest that prey aggregations attract predators. If predator attacks on the prey population are proportional to population density, then the predation rate per prey *individual* will be independent of density; predator attacks would need to accelerate with increasing population density in order for predators to regulate the prey population. One could test for positive correlations between prey density and per capita mortality to see whether this is so.

However, correlation analysis assumes the data are bivariate normally distributed, while linear regression assumes a linear relationship between a predictor variable and a normally distributed response variable. Although one can sometimes transform data to satisfy these assumptions, or simply ignore minor violations, Sandin and Pacala took a more powerful approach: they built explicit models to describe how the absolute and per capita predator visits or mortality depended on prey population density. For example, the absolute mortality probability would be $r_0 + r_1 n$ and the per capita mortality probability would be $(r_0 + r_1 n)/n$ if predator visits are proportional to prey density. They also used realistic binomial and Poisson probability distributions to describe the variation in the data, rather than assuming normality (a particularly awkward assumption when there are lots of zeros in the data). By doing so, they were able to choose among a variety of possible models and conclude that predators induce *inverse* density dependence in this system (i.e., that smaller prey populations experience higher per capita mortality, because predators are present at relatively constant numbers independent of prey density). Because they fitted models rather than running classical statistical tests on transformed data, they were also able to estimate meaningful parameter values, such as the increase in predator visits per hour for every additional prey individual present. These values are more useful than p (significance) values, or than regression slopes from transformed data, because they express statistical information in ecological terms.

1.2 What This Book Is Not About

1.2.1 What You Should Already Know

To get the most out of the material presented here you should already have a good grasp of basic statistics, be comfortable with computers (e.g., have used Microsoft Excel to deal with data), and have some rusty calculus. But attitude and aptitude are more important than previous classroom experience. Getting into this material requires some hard work at the outset, but it will become easier as you brush up on basic concepts.*

STATISTICS

I assume that you've had the equivalent of a one-semester undergraduate statistics course. The phrases *hypothesis test*, *analysis of variance*, *linear regression*, *normal distribution* (maybe even *Central Limit Theorem*) should be familiar to you, even if you don't remember all of the details. The basics of experimental design—the meaning of and need for randomization, control, independence, and replication in setting up experiments, the idea of statistical power, and the concept of pseudoreplication (Hurlbert, 1984; Hargrove and Pickering, 1992; Heffner et al., 1996; Oksanen, 2001)—are essential tools for any working ecologist, but you can learn them from a good introductory statistics class or textbook such as Gotelli and Ellison (2004) or Quinn and Keough (2002).†

Further reading: If you need to review statistics, try Crawley (2002), Dalgaard (2003), or Gotelli and Ellison (2004). Gonick and Smith's 1993 *Cartoon Guide to Statistics* gives a gentle introduction to some basic concepts, but you will need to go beyond what they cover. Sokal and Rohlf (1995), Zar (1999), and Crawley (2005, 2007) cover a broader range of classical statistics. For experimental design, try Underwood (1996), Scheiner and Gurevitch (2001), or Quinn and Keough (2002) (the latter two discuss statistical analysis as well).

COMPUTERS

This book will teach you how to use computers to understand data. You will be writing a few lines of R code at a time rather than full-blown computer programs, but you will have to go beyond pointing and clicking. You need to be comfortable with computers, and with using spreadsheets like Excel to manipulate data. Familiarity with a mainstream statistics package like SPSS or SAS will be useful, although you

* After teaching with Hilborn and Mangel's excellent book *The Ecological Detective* (1997) I wanted to write a book that included enough nitty-gritty detail for students to tackle their own problems. If this book feels too hard for you, consider starting with *The Ecological Detective*—but consider reading *ED* in any case.

† Ideally, you would think about how you will analyze your data before you go into the field to collect it. This rarely happens. Fortunately, if your observations are adequately randomized, controlled, independent, and replicated, you will be able to do *something* with your data. If they aren't, no fancy statistical techniques can help you.

should definitely use R to work through this book instead of falling back on a familiar software package. (If you have used R already, you'll have a big head start.) You needn't have done any programming.

MATH

Having "rusty" calculus means knowing what a derivative and an integral are. While it would be handy to remember a few of the formulas for derivatives, a feeling for the meanings of logarithms, exponentials, derivatives, and integrals is more important than the formulas (you'll find the formulas in the appendix). In working through this book you will have to *use* algebra, as much as calculus, in a routine way to solve equations and answer questions. Most of the people who have taken my classes were very rusty when they started.

Further reading: Adler (2004) gives a very applied review of basic calculus, differential equations, and probability, while Neuhauser (2003) covers calculus in a more rigorous and traditional way, but still with a biological slant.

ECOLOGY

I have assumed you know some basic ecological concepts, since they are the foundation of ecological data analysis. You should be familiar, for example, with exponential and logistic growth from population ecology; functional responses from predator-prey ecology; and competitive exclusion from community ecology.

Further reading: For a short introduction to ecological theory, try Hastings (1997) or Vandermeer and Goldberg (2004) (the latter is more general). Gotelli (2001) is more detailed. Begon et al. (1996) gives an extremely thorough introduction to general ecology, including some basic ecological models. Case (1999) provides an illustrated treatment of theory, while Roughgarden (1997) integrates ecological theory with programming examples in MATLAB. Mangel (2006) and Otto and Day (2007), two new books, both give basic introductions to the "theoretical biologist's toolbox."

1.2.2 Other Kinds of Models

Ecologists sometimes want to "learn how to model" without knowing clearly what questions they hope the models will answer, and without knowing what kind of models might be useful. This is a bit like saying "I want to learn to do experiments" or "I want to learn molecular biology": Do you want to analyze microsatellites? Use RNA inactivation to knock out gene function? Sequence genomes? What people usually mean by "I want to learn how to model" is "I have heard that modeling is a powerful tool and I think it could tell me something about my system, but I'm not really sure what it can do."

Ecological modeling has many facets. This book covers only one: statistical modeling, with a bias toward mechanistic descriptions of ecological patterns. The next section briefly reviews a much broader range of modeling frameworks and gives some starting points in the modeling literature in case you want to learn more about other kinds of ecological models.

1.3 Frameworks for Modeling

This book is primarily about how to combine models with data and how to use them to discover the answers to theoretical or applied questions. To help fit statistical models into the larger picture, Table 1.1 presents a broad range of dichotomies that cover some of the kinds and uses of ecological models. The discussion of these dichotomies starts to draw in some of the statistical, mathematical, and ecological concepts I suggested you should know. However, if a few are unfamiliar, don't worry—the next few chapters will review the most important concepts. Part of the challenge of learning the material in this book is a chicken-and-egg problem: to know why certain technical details are important, you need to know the big picture, but the big picture itself involves knowing some of those technical details. Iterating, or cycling, is the best way to handle this problem. Most of the material introduced in this chapter will be covered in more detail in later chapters. If you don't completely get it this time around, hang on and see if it makes more sense the second time.

1.3.1 Scope and Approach

The first set of dichotomies in the table subdivides models into two categories, one (theoretical/strategic) that aims for general insight into the workings of ecological processes and one (applied/tactical) that aims to describe and predict how a particular system functions, often with the goal of forecasting or managing its behavior. Theoretical models are often mathematically difficult and ecologically oversimplified, which is the price of generality. Paradoxically, although theoretical models are defined in terms of precise numbers of individuals, because of their simplicity they are usually used only for qualitative predictions. Applied models are often mathematically simpler (although they can require complex computer code) but tend to capture more of the ecological complexity and quirkiness needed to make detailed predictions about a particular place and time. Because of this complexity their predictions are often less general.

The dichotomy of mathematical versus statistical modeling says more about the culture of modeling and how different disciplines go about thinking about models than about how we should actually model ecological systems. A mathematician is more likely to produce a deterministic, dynamic process model without thinking very much about noise and uncertainty (e.g., the ordinary differential equations that make up the Lotka-Volterra predator-prey model). A statistician, on the other hand, is more likely to produce a stochastic but static model that treats noise and uncertainty carefully but focuses more on static patterns than on the dynamic processes that produce them (e.g., linear regression).*

* Of course, both mathematicians and statisticians are capable of more sophisticated models than the simple examples given here.

TABLE 1.1
Modeling dichotomies

Scope and approach	
abstract	concrete
strategic	tactical
general	specific
theoretical	applied
qualitative	quantitative
descriptive	predictive
mathematical	statistical
mechanistic	phenomenological
pattern	process
Technical details	
analytical	computational
dynamic	static
continuous	discrete
population-based	individual-based
Eulerian	Lagrangian
deterministic	stochastic
Sophistication	
simple	complex
crude	sophisticated

Each column contrasts a different qualitative style of modeling. The loose association of descriptors in each column gets looser as you work downward.

The important difference between phenomenological (pattern) and mechanistic (process) models will be with us throughout the book. Phenomenological models concentrate on observed patterns in the data, using functions and distributions that are the right shape and/or sufficiently flexible to match them; mechanistic models are more concerned with the underlying processes, using functions and distributions based on theoretical expectations. As usual, shades of gray abound; the same function could be classified as either phenomenological or mechanistic depending on why it was chosen. For example, you could use the function $f(x) = ax/(b+x)$ (a Holling type II functional response) as a mechanistic model in a predator-prey context because you expected predators to attack prey at a constant rate and be constrained by handling time, or as a phenomenological model of population growth simply because you wanted a function that started at zero, was initially linear, and leveled off as it approached an asymptote (see Chapter 3). All other things being equal, mechanistic models are more powerful since they tell you about the underlying processes driving patterns. They are more likely to work correctly when extrapolating beyond the observed conditions. Finally, by making more assumptions, they allow you to extract more information from your data—with the risk of making the *wrong* assumptions.*

Examples of theoretical models include the Lotka-Volterra or Nicholson-Bailey predator-prey equations (Hastings, 1997); classical metapopulation models for single (Hanski, 1999) and multiple (Levins and Culver, 1971; Tilman, 1994) species; simple food web models (May, 1973; Cohen et al. 1990); and theoretical ecosystem models (Agren and Bosatta, 1996). Applied models include forestry and biogeochemical cycling models (Blanco et al. 2005), fisheries stock-recruitment models (Quinn and Deriso, 1999), and population viability analysis (Morris and Doak, 2002; Miller and Lacy, 2005).

Further reading: Books on ecological modeling overlap with those on ecological theory listed on p. 4. Other good sources include Nisbet and Gurney (1982; a well-written but challenging classic), Gurney and Nisbet (1998; a lighter version), Haefner (1996; broader, including physiological and ecosystem perspectives), Renshaw (1991; good coverage of stochastic models), Wilson (2000; simulation modeling in C), and Ellner and Guckenheimer (2006; dynamics of biological systems in general).

1.3.2 Technical Details

Another set of dichotomies characterizes models according to the methods used to analyze them or according to the decisions they embody about how to represent individuals, time, and space.

An analytical model is made up of equations solved with algebra and calculus. A computational model consists of a computer program which you run for a range of parameter values to see how it behaves.

Most mathematical models and a few statistical models are dynamic; the response variables at a particular time (the state of the system) feed back to affect the response variables in the future. Integrating dynamical and statistical models is challenging (see Chapter 11). Most statistical models are static; the relationship between predictor and response variables is fixed.

One can specify how models represent the passage of time or the structure of space (both can be continuous or discrete); whether they track continuous population densities (or biomass or carbon densities) or discrete individuals; whether they consider individuals within a species to be equivalent or divide them by age, size, genotype, or past experience; and whether they track the properties of individuals

* For an alternative, classic approach to the tradeoffs between different kinds of models, see Levins (1966) (criticized by Orzack and Sober (1993); Levins's (1993) defense invokes the fluidity of model-building in ecology).

(individual-based or Eulerian) or the number of individuals within different categories (population-based or Lagrangian).

Deterministic models represent only the average, expected behavior of a system in the absence of random variation, while stochastic models incorporate noise or randomness in some way. A purely deterministic model allows only for qualitative comparisons with real systems; since the model will never match the data *exactly*, how can you tell if it matches closely enough? For example, a deterministic food web model might predict that introducing pike to a lake would cause a trophic cascade, decreasing the density of phytoplankton (because pike prey on sunfish, which eat zooplankton, which in turn consume phytoplankton); it might even predict the expected magnitude of the change. To test this prediction with real data, however, you would need some kind of statistical model to estimate the magnitude of the average change in several lakes (and the uncertainty), and to distinguish between observed changes due to pike introduction and those due to other causes (measurement error, seasonal variation, weather, nutrient dynamics, population cycles, etc.).

Most ecological models incorporate stochasticity crudely, by simply assuming that there is some kind of (perhaps normally distributed) variation, arising from a combination of unknown factors, and estimating the magnitude of that variation from the variation observed in the field. We will go beyond this approach, specifying different sources of variability and something about their expected distributions. More sophisticated models of variability enjoy some of the advantages of mechanistic models: models that make explicit assumptions about the underlying causes of variability can both provide more information about the ecological processes at work and get more out of your data.

There are essentially three kinds of random variability:

- *Measurement error* is the variability imposed by our imperfect observation of the world; it is always present, except perhaps when we are counting a small number of easily detected organisms. It is usually modeled by the standard approach of adding normally distributed variability around a mean value.
- *Demographic stochasticity* is the innate variability in outcomes due to random processes even among otherwise identical units. In experimental trials where you flip a coin 20 times you might get 10 heads, or 9, or 11, even though you're flipping the same coin the same way each time. Likewise, the number of tadpoles out of an initial cohort of 20 eaten by predators in a set amount of time will vary between experiments. Even if we controlled everything about the environment and genotype of the predators and prey, we would still see different numbers dying in each run of the experiment.
- *Environmental stochasticity* is variability imposed from "outside" the ecological system, such as climatic, seasonal, or topographic variation. We usually reserve environmental stochasticity for unpredictable variability, as opposed to predictable changes (such as seasonal or latitudinal changes in temperature) which we can incorporate into our models in a deterministic way.

The latter two categories, demographic and environmental stochasticity, make up *process variability*,* which, unlike measurement error, affects the future dynamics of the ecological system. (Suppose we expect to find three individuals on an isolated

* Process variability is also called *process noise* or *process error* (Chapter 10).

island. If we make a measurement error and measure zero instead of three, we may go back at some time in the future and still find them. If an unexpected predator eats all three individuals (process variability), and no immigrants arrive, any future observations will find no individuals.) The conceptual distinction between process and measurement error is most important in dynamic models, where the process error has a chance to feed back on the dynamics.

The distinctions between stochastic and deterministic effects, and between demographic and environmental variability, are really a matter of definition. Until you get down to the quantum level, any "random" variability can in principle be explained and predicted. What determines whether a tossed coin will land heads-up? Its starting orientation and the number of times it turns in the air, which depends on how hard you toss it (Keller, 1986). What determines exactly which and how many seedlings of a cohort die? The amount of energy with which their mother provisions the seeds, their individual light and nutrient environments, and encounters with pathogens and herbivores. Variation that drives mortality in seedlings—e.g., variation in available carbohydrates among individuals because of small-scale variation in light availability—might be treated as a random variable by a forester at the same time that it is treated as a deterministic function of light availability by a physiological ecologist measuring the same plants. Climatic variation is random to an ecologist (at least on short time scales) but might be deterministic, although chaotically unpredictable, to a meteorologist. Similarly, the distinction between demographic variation, internal to the system, and environmental variation, external to the system, varies according to the focus of a study. Is the variation in the number of trees that die every year an internal property of the variability in the population or does it depend on an external climatic variable that is modeled as random noise?

1.3.3 Sophistication

I want to make one final distinction, between simple and complex models and between crude and sophisticated ones. One could quantify simplicity versus complexity by the length of the description of the analysis or by the number of lines of computer script or code required to implement a model. Crudity and sophistication are harder to recognize; they represent the conceptual depth, or the amount of *hidden* complexity, involved in a model or statistical approach. For example, a computer model that picks random numbers to determine when individuals give birth and die and keeps track of the total population size, for particular values of the birth and death rates and starting population size, is simple and crude. Even simpler, but far more sophisticated, is the mathematical theory of random walks (Okubo, 1980) which describes the same system but—at the cost of challenging mathematics—predicts its behavior for *any* birth and death rates and any starting population sizes. A statistical model that searches at random for the line that minimizes the sum of squared deviations of the data is crude and simple; the theory of linear models, which involves more mathematics, does the same thing in a more powerful and general way. Computer programs, too, can be either crude or sophisticated. One can pick numbers from a binomial distribution by virtually flipping the right number of coins and seeing how many come up heads, or by using numerical methods that arrive at the same result far more efficiently. A simple R command like `rbinom`, which picks random binomial deviates, hides a lot of complexity.

The value of sophistication is generality, simplicity, and power; its costs are opacity and conceptual and mathematical difficulty. In this book, I will take advantage of many of R's sophisticated tools for optimization and random number generation (since in this context it's more important to have these tools available than to learn the details of how they work), but I will avoid many of its sophisticated statistical tools, so that you can learn from the ground up how statistical models really work and make your models work the way you want them to rather than being constrained by existing frameworks. Having reinvented the wheel, however, we'll briefly revisit some standard statistical frameworks like generalized linear models and see how they can solve some problems more efficiently.

1.4 Frameworks for Statistical Inference

This section will explore three different ways of drawing statistical conclusions from data—frequentist, Bayesian, and likelihood-based. While the differences among these frameworks are sometimes controversial, most modern statisticians know them all and use whatever tools they need to get the job done; this book will teach you the details of those tools, and the distinctions among them.

To illustrate the ideas I'll draw on a seed predation data set from Duncan and Duncan (2000) that quantifies how many times seeds of two different species disappeared (presumably taken by seed predators, although we can't be sure) from observation stations in Kibale National Park, Uganda. The two species (actually the smallest- and largest-seeded species of a set of eight species) are *Polyscias fulva* (pol: seed mass < 0.01 g) and *Pseudospondias microcarpa* (psd: seed mass ≈ 50 g).

1.4.1 Classical Frequentist

Classical statistics, which are part of the broader *frequentist* paradigm, are the kind of statistics typically presented in introductory statistics classes. For a specific experimental procedure (such as drawing cards or flipping coins), you calculate the probability of a particular outcome, which is defined as *the long-run average frequency of that outcome in a sequence of repeated experiments*. Next you calculate a *p-value*, defined as the probability of that outcome *or any more extreme outcome* given a specified null hypothesis. If this so-called *tail probability* is small, then you reject the null hypothesis; otherwise, you fail to reject it. But you don't accept the null hypothesis if the tail probability is large; you just fail to reject it.

The frequentist approach to statistics (due to Fisher, Neyman, and Pearson) is useful and very widely used, but it has some serious drawbacks—which are repeatedly pointed out by proponents of other statistical frameworks (Berger and Berry, 1988). It relies on the probability of a series of outcomes that didn't happen (the tail probabilities), and which depend on the way the experiment is defined; its definition of probability depends on a series of hypothetical repeated experiments that are often impossible in any practical sense; and it tempts us to construct straw-man

TABLE 1.2
Seed removal data

	pol	psd
Any taken (t)	26	25
None taken	184	706
Total (N)	210	731

null hypotheses and make convoluted arguments about why we have failed to reject them. Probably the most criticized aspect of frequentist statistics is their reliance on p-values, which when misused (as frequently occurs) are poor tools for scientific inference. To abuse p-values seems to be human nature; we act as though alternative hypotheses (which are usually what we're really interested in) are "true" if we can reject the null hypothesis with $p < 0.05$ and "false" if we can't. In fact, when the null hypothesis is true we still find $p < 0.05$ one time in twenty (we falsely reject the null hypothesis 5% of the time, by definition). If $p > 0.05$, the null hypothesis could still be false but we have insufficient data to reject it. We could also reject the null hypothesis in cases where we have lots of data, even though the results are biologically insignificant—that is, if the estimated effect size is ecologically irrelevant (e.g., a 0.01% increase in plant growth rate with a 30°C increase in temperature). More fundamentally, if we use a so-called *point null hypothesis* (such as "the slope of the relationship between plant productivity and temperature is zero"), common sense tells us that the null hypothesis *must* be false, because it can't be exactly zero—which makes the p-value into a statement about whether we have enough data to detect a nonzero slope, rather than about whether the slope is actually different from zero. Working statisticians will tell you that it is better to focus on estimating the values of biologically meaningful parameters and finding their confidence limits rather than worrying too much about whether p is greater or less than 0.05 (Yoccoz, 1991; Johnson, 1999; Osenberg et al. 2002)—although Stephens et al. (2005) remind us that hypothesis testing can still be useful.

Looking at the seed data, we have a 2 × 2 table (Table 1.2). If t_i is the number of times that species i seeds disappear and N_i is the total number of observations of species i, then the observed proportions of the time that seeds disappeared for each species are (pol) $t_1/N_1 = 0.124$ and (psd) $t_2/N_2 = 0.034$. The overall proportion taken (which is not the average of the two proportions since the total numbers of observations for each species differ) is $(t_1 + t_2)/(N_1 + N_2) = 0.054$. The ratio of the predation probabilities (proportion for pol/proportion for psd) is $0.124/0.034 = 3.62$. The ecological question we want to answer is "Is there differential predation on the seeds on these two species?" (Given the sample sizes and the size of the observed difference, what do you think? Do you think the answer is likely to be statistically significant? How about biologically significant? What assumptions or preconceptions does your answer depend on?)

A frequentist would translate this biological question into statistics as "What is the probability that I would observe a result this extreme, or more extreme, given the sampling procedure?" More specifically, "What proportion of possible outcomes would result in observed ratios of proportions greater than 3.62 *or*

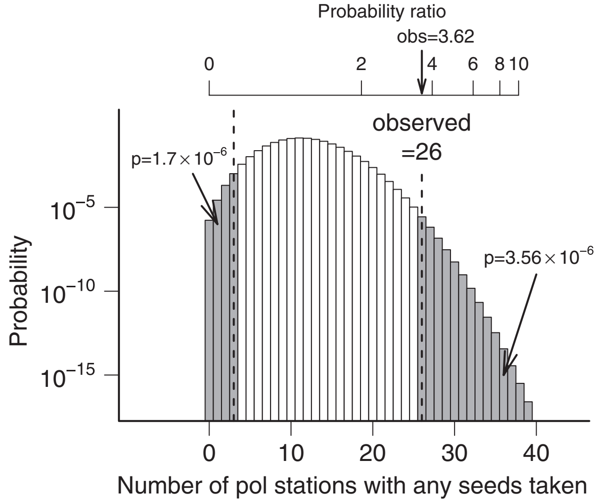

Figure 1.1 Classical frequentist analysis. Fisher's exact test calculates the probability of a given number of pol stations having seeds taken under the null hypothesis that both species have the same predation probability. The total probability that as many or more pol stations had seeds taken, *or* that the difference was more extreme in the other direction, is the two-tailed frequentist p-value ($3.56 \times 10^{-6} + 1.70 \times 10^{-6} = 5.26 \times 10^{-6}$). The top axis shows the equivalent in seed predation probability ratios. (*Note*: I put the y-axis on a log scale because the tails of the curve are otherwise too small to see, even though this change means that the area under the curve no longer represents the total probability.)

smaller than $1/3.62 = 0.276$?" (Figure 1.1). Fisher's exact test (`fisher.test` in R) calculates this probability, as a one-tailed test (proportion of outcomes with ratios greater than 3.62) or a two-tailed test (proportion with ratios greater than 3.62 or less than its reciprocal, 0.276); the two-tailed answer in this case is 5.26×10^{-6}. According to Fisher's original interpretation, this number represents the strength of evidence against the null hypothesis, or (loosely speaking) for the alternative hypothesis—that there is a difference in seed predation rates. According to the Neyman-Pearson decision rule, if we had set our acceptance cutoff at $\alpha = 0.05$, we could conclude that there was a *statistically significant* difference in predation rates.

We needn't fixate on p-values: the R command for Fisher's test, `fisher.test`, also tells us the 95% confidence limits for the difference between rates.* In terms of probability ratios, this example gives (2.073, 6.057), which as expected does not include 1. Do you think a range of a 107% to a 506% increase in seed predation probability† is significant?

* R expresses the difference in predation rates in terms of the *odds ratio*—if there are t_1 seeds taken and $N_1 - t_1$ seeds not taken for species 1, then the odds of a seed being taken are $t_1/(N_1 - t_1)$ and the odds ratio between the species is $(t_1/(N_1 - t_1))/(t_2/(N_2 - t_2))$. The odds ratio and its logarithm (the *logit* or log-odds ratio) have nice statistical properties.

† These values are the confidence limits on the probability ratios, minus 1, converted into percentages: for example, a probability ratio of 1.1 would represent a 10% increase in predation.

1.4.2 Likelihood

Most of the book will focus on frequentist statistics, but not the standard version that you may be used to. Most modern statistics uses an approach called *maximum likelihood estimation*, or approximations to it. For a particular statistical model, maximum likelihood finds the set of parameters (e.g., seed removal rates) *that makes the observed data* (e.g., the particular outcomes of predation trials) *most likely to have occurred*. Based on a model for both the deterministic and stochastic aspects of the data, we can compute the *likelihood* (the probability of the observed outcome) given a particular choice of parameters. We then find the set of parameters that makes the likelihood as large as possible, and take the resulting *maximum likelihood estimates* (MLEs) as our best guess at the parameters. So far we haven't assumed any particular definition of probability of the parameters. We could decide on confidence limits by choosing a likelihood-based cutoff, for example, by saying that any parameters that make the probability of the observed outcomes at least one-tenth as likely as the maximum likelihood are "reasonable." For mathematical convenience, we often work with the logarithm of the likelihood (the *log-likelihood*) instead of the likelihood; the parameters that give the maximum log-likelihood also give the maximum likelihood. On the log scale, statisticians have suggested a cutoff of 2 log-likelihood units (Edwards, 1992), meaning that we consider any parameter reasonable that makes the observed data at least $e^{-2} \approx 1/7.4 = 14\%$ as likely as the maximum likelihood.

However, most modelers add a frequentist interpretation to likelihoods, using a mathematical proof that says that, across the hypothetical repeated trials of the frequentist approach, the distribution of the negative logarithm of the likelihood itself follows a χ^2 (chi-squared) distribution.* This fact means that we can set a cutoff for differences in log-likelihoods based on the 95th percentile of the χ^2 distribution, which corresponds to 1.92 log-likelihood units, or parameters that lower the likelihood by a factor of $e^{1.92} = 6.82$. The theory says that the estimated value of the parameter will fall farther away than that from the true value only 5% of the time in a long series of repeated experiments. This rule is called the *Likelihood Ratio Test* (LRT).† We will see that it lets us both estimate confidence limits for parameters and choose between competing models.

Bayesians (discussed below) also use the likelihood—it is part of the recipe for computing the posterior distribution—but they take it as a measure of the information we can gain from the data, without saying anything about what the distribution of the likelihood would be in repeated trials.

How would one apply maximum likelihood estimation to the seed predation example? Lumping all the data from both species together at first, and assuming that (1) all observations are independent of each other and (2) the probability of at least one seed being taken is the same for all observations, it follows that the number of times at least one seed is removed is *binomially* distributed (we'll get to the formulas in Chapter 4). Now we want to know how the probability of observing the data (the likelihood $\mathcal{L}$) depends on the probability p_s that at least one seed was

* This result holds in the *asymptotic* case where we have lots of data, which happens less than we would like—but we often gloss over the fact of limited data and use it anyway.

† The difference between log-likelihoods is equivalent to the ratio of likelihoods.

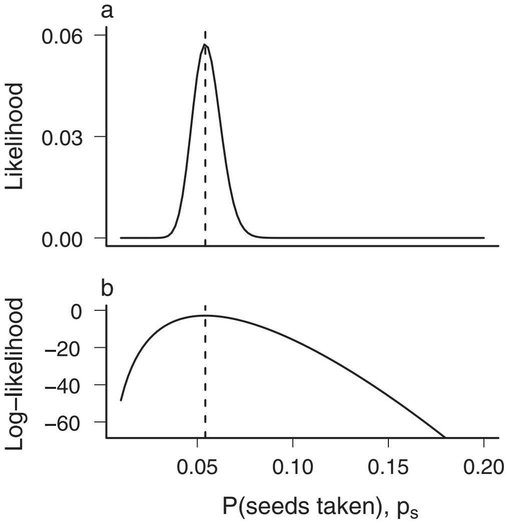

Figure 1.2 Likelihood and log-likelihood curves for removal probability p. Both curves have their maxima at the same point ($p_s = 0.054$). Log-likelihoods are based on natural ($\log_e$ or ln) logarithms.

taken from a particular station by a predator,* and what value of p_s maximizes the likelihood. The likelihood $\mathcal{L}$ is the probability that seeds were taken in 51 out of the total of 941 observations. This probability varies as a function of p_s (Figure 1.2): for $p_s = 0.05$, $\mathcal{L} = 0.048$, while for $p = 0.04$, $\mathcal{L}$ is only 6.16×10^{-3}. As it turns out, the MLE for the probability that seeds were taken in any one trial (p_s) is exactly what we'd expect—51/941, or 0.054—and the likelihood is $\mathcal{L} = 0.057$. (This likelihood is small, but it just means that the probability of any *particular* outcome—seeds being taken in 51 trials rather than 50 or 52—is small.)

To answer the questions that really concern us about the different predation probabilities for different species, we need to allow different probabilities for each species, and see how much better we can do (how much higher the likelihood is) with this more complex model. Now we take the separate values for each species (26 out of 210 and 25 out of 731) and, with a different per-observation probability for each species, compute the likelihoods of each species' data and multiply them (see Chapter 4 for basic probability calculations) or add the log-likelihoods. If we define the model in terms of the probability for psd and the ratio of the probabilities, we can plot a *likelihood profile* for the maximum likelihood we can get for a given value of the ratio (Figure 1.3).

The conclusions from this frequentist, maximum-likelihood analysis are essentially identical to those of the classical frequentist (Fisher's exact test) analyses. The maximum-likelihood estimate equals the observed ratio of the probabilities,

* One of the most confusing things about maximum likelihood estimation is that there are so many different probabilities floating around. The likelihood $\mathcal{L}$ is the probability of observing the complete data set (i.e., Prob(seeds were taken 51 times out of 941 observations)); p_s is the probability that seeds were taken in any given trial; and the (one-tailed) frequentist p-value is the probability, given a particular value of p_s, that seeds were taken 51 *or more* times out of 941 observations.

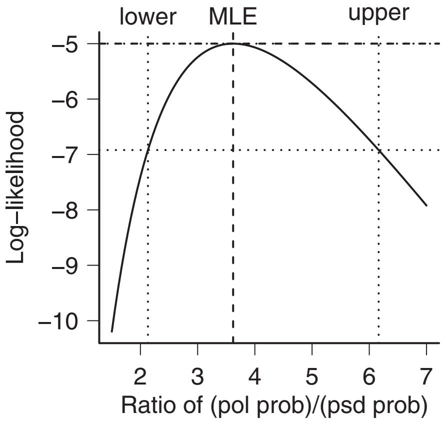

Figure 1.3 Likelihood curve for the ratio of the predation probabilities, showing the maximum likelihood estimate and 95% confidence limits. The null hypothesis value (ratio equal to 1) is just below the lower limit of the horizontal axis.

3.62; the confidence limits are (2.13, 6.16), which do not include 1; and the LRT-based p-value for rejecting the null hypothesis that the probabilities are the same is 3.83×10^{-6}.

Likelihood and classical frequentist analysis share the same philosophical underpinnings. Likelihood analysis is really a particular flavor of frequentist analysis, one that focuses on writing down a likelihood model and then testing for significant differences in the likelihood ratio rather than applying frequentist statistics directly to the observed outcomes. Classical analyses are usually easier because they are built into common statistics packages, and they may make fewer assumptions than likelihood analyses (e.g., Fisher's test is exact while the LRT is valid only for large data sets), but likelihood analyses are often better matched with ecological questions.

1.4.3 Bayesian

Frequentist statistics assumes that there is a "true" state of the world (e.g., the ratio of the species' predation probabilities) which gives rise to a distribution of possible experimental outcomes. The Bayesian framework says instead that the experimental outcome—what we actually saw happen—is the truth, while the parameter values or hypotheses have probability distributions. The Bayesian framework solves many of the conceptual problems of frequentist statistics: answers depend on what we actually saw and not on a range of hypothetical outcomes, and we can legitimately make statements about the probability of different hypotheses or parameter values.

The major fly in the ointment of Bayesian statistics is that in order to make it work we have to specify our *prior beliefs* about the probability of different hypotheses, and these prior beliefs actually affect our answers! One hard-core frequentist ecologist says "Bayesianism means never having to say you're wrong" (Dennis, 1996). It is indeed possible to cheat in Bayesian statistics by setting unreasonably strong priors.* The standard solution to the problem of subjectivity is to assume you are

* But if you really want to cheat with statistics you can do it in any framework!

completely ignorant before the experiment (setting a *flat prior*, or "letting the data speak for themselves"), although for technical reasons this isn't always possible. For better or worse, Bayesian statistics operates in the same way as we typically do science: we downweight observations that are too inconsistent with our current beliefs, while using those in line with our current beliefs to strengthen and sharpen those beliefs (statisticians are divided on whether this is good or bad).

The big advantages of Bayesian statistics, besides ease of interpretation, come (1) when we actually have data from prior observations we want to incorporate; (2) in complex models with missing data and several layers of variability; (3) when we are trying to make management decisions based on our data (the Bayesian framework makes it easier to incorporate the effect of unlikely but catastrophic scenarios in decision making). The only big disadvantage (besides the problem of priors) is that problems of small to medium complexity are actually harder with Bayesian approaches than with frequentist approaches—at least in part because most statistical software is geared toward classical statistics.

How would Bayesians answer our question about predation rates? First of all, they would say (without looking at the data) that the answer is "yes"—the true difference between predation rates is certainly not zero. (This discrepancy reflects the difference in perspective between frequentists, who believe that the true value is a fixed number and uncertainty lies in what you observe [or might have observed], and Bayesians, who believe that observations are fixed numbers and the true values are uncertain.) Then they might define a parameter, the ratio of the two proportions, and ask questions about the *posterior distribution* of that parameter—our best estimate of the probability distribution given the observed data and some prior knowledge of its distribution (see Chapter 4). What is the mode (most probable value) of that distribution? What is its expected value, or mean? What is the *credible interval*, which is the interval with equal probability cutoffs below and above the mean within which 95% of the probability falls?

The Bayesian answers, in a nutshell: when using a flat prior distribution, the posterior mode is 3.48 (near the observed proportion of 3.62). The posterior mean is 3.87, slightly larger than the posterior mode since the posterior probability density is slightly asymmetric—the density is skewed to the right (Figure 1.4).* The 95% credible interval, from 2.01 to 6.01, doesn't include 1, so Bayesians would say that there was good evidence against the hypothesis: even more strongly, they could say that the posterior probability that the predation ratio is greater than 1 is 0.998 (the probability that it is less than 1 is 0.002).

If the details of Bayesian statistics aren't perfectly clear at this point, don't worry. We'll explore Bayes' Rule and revisit Bayesian statistics in future chapters.

In this example all three statistical frameworks gave very similar answers, but they don't always. Ecological statisticians are still hotly debating which framework is best, or whether there is a single best framework. While it is important to be clear on the differences among the approaches, and knowing what question each is trying to answer, statisticians commonly move back and forth among them. My own approach is eclectic, agreeing with the advice of Crome (1997) and Stephens et al. (2005) to

* While Figure 1.1 showed the probability of each possible discrete outcome (number of seeds taken), Figure 1.4 shows a posterior probability *density* of a continuous parameter, i.e. the relative probability that the parameter lies in a particular range. Chapter 4 will explain this distinction more carefully.

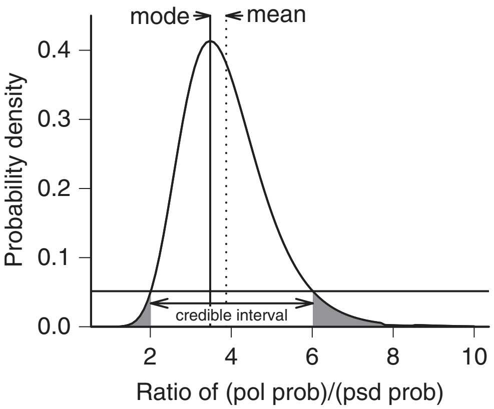

Figure 1.4 Bayesian analysis of seed predation. We calculate the probability density of the ratio of proportions of seeds taken being equal to some particular value, based on our prior (flat, assuming perfect ignorance—and in this case *improper* because it doesn't integrate to 1 [Chapter 4]) and on the data. The most probable value is the mode; the expected value is the mean. The shaded areas contain 5% of the area under the curve and cut off at the same height (probability density); the range between them is therefore the 95% credible interval.

try to understand the strengths and weaknesses of several different approaches and use each one as appropriate.

We will revisit these frameworks in more detail later. Chapter 4 will cover Bayes' Rule, which underpins Bayesian statistics; Chapters 6 and 7 will return to a much more detailed look at the practical details of maximum likelihood and Bayesian analysis. (Textbooks like Dalgaard (2003) cover classical frequentist approaches very well.)

1.5 Frameworks for Computing

To construct your own models, you will need to learn some of the basics of statistical computing. There are many computer languages and modeling tools with built-in statistical libraries (MATLAB, Mathematica) and several statistics packages with serious programming capabilities (SAS, IDL). We will use a system called R, that is both a statistics package and a computing language.

1.5.1 What Is R*?*

R's developers call it a "language and environment for statistical computing and graphics." This awkward phrase gets at the idea that R is more than just a statistics package. R is closest in spirit to other higher-level modeling languages like MATLAB or MathCAD. It is a dialect of the S computing language, which was written at Bell Labs in the 1980s as a research tool in statistical computing. MathSoft, Inc. (now Insightful Corporation), bought the rights to S and developed it into S-PLUS, a commercial package with a graphical front end. In the 1990s two New Zealand statisticians, Ross Ihaka and Robert Gentleman, rewrote S from scratch, again as a research project. The rewritten (and free) version became immensely popular and is now maintained by an international "core team" of about a dozen well-respected statisticians and computer scientists.

1.5.2 Why Use R*?*

R is an extremely powerful tool. It is a full-fledged modern computer language with sophisticated data structures; it supports a wide range of computations and statistical procedures; it can produce graphics ranging from exploratory plots to customized publication-quality graphics.

R is free in the sense that you can download it from the Internet, make as many copies as you want, and give them away.* While I don't begrudge spending money on software for research, it is certainly convenient not to have to pay—or to deal with licensing paperwork. This cheapness is vital, rather than convenient, for teachers, independent researchers, people in less-developed countries, and students who are frustrated with limited student versions (or pirated versions) of commercial software.

More important, R is also free in the sense that you can inspect any of the code and change it in any way that you want.† This form of freedom is probably abstract to you at this point—you probably won't need to modify R in the course of your modeling career—but it is a part of the same basic philosophy of the free exchange of information that underlies scientific and academic research in general.

R is the choice of many academic and industrial statisticians, who work to improve it and to write extension packages. If a statistical method has made it into print, the odds are good that there's an R package somewhere that implements it.

R runs well on many computer platforms, including the "big three" (Microsoft Windows, Mac OS X, and Linux). There are only tiny, mostly cosmetic differences in the way that R runs on different machines. You can nearly always move data files and code between operating systems and get the same answers.

R is rapidly gaining popularity. The odds are good that someone in your organization is using R, and there are many resources on the Internet including a very active mailing list. A growing number of introductory books use R (Dalgaard, 2003; Verzani, 2005; Crawley, 2005). There are also books of examples (Maindonald and Braun, 2003; Heiberger and Holland, 2004; Everitt and Hothorn, 2006), more advanced and encyclopedic books covering a range of statistical approaches (Venables and Ripley, 2002; Crawley, 2002), and books on specific topics such as regression analysis (Fox, 2002; Faraway, 2004), mixed-effect models (Pinheiro and Bates, 2000), phylogenetics (Paradis, 2006), and generalized additive models (Wood, 2006) that are geared toward R and S-PLUS users.

1.5.3 Why Not Use R*?*

R is more difficult than mainstream statistics packages like SYSTAT or SPSS, because it does much more. It would be hard to squeeze all of R's capabilities into a

* In programming circles, this freedom is called "gratis" or "free as in beer."

† "Libre" or "free as in speech."

simple graphical user interface (GUI) with menus to guide you through the process of analyzing your data. R's creators haven't even tried very hard to write a GUI, because they have a do-it-yourself philosophy that emphasizes knowing procedures rather than letting the program try to tell you what to do next. John Fox has written a simple GUI for R (called `Rcmdr`), and the commercial version of R, S-PLUS, does have a graphical user interface—if you can afford it. However, for most of what we will be doing in this book a GUI would not be very useful.

While R comes with a lot of documentation, it's mostly good for reminding you of the syntax of a command rather than for finding out how to do something. Unlike SAS, for which you can buy voluminous manuals that tell you the details of various statistical procedures and how to run them in SAS, R typically assumes that you have a general knowledge of the procedure you want to use and can figure out how to make it work in R by reading the online documentation or a separately published book (including this one).

R is slower than so-called lower-level languages like C and FORTRAN because it is an *interpreted* language that processes strings of commands typed in at the command line or stored in a text file, rather than a *compiled* language that first translates commands into machine code. However, computers are so fast these days that there's speed to burn. For most problems you will encounter the limiting factor will be how fast and easily you can write (and debug) the code, not how long the computer takes to process it. Interpreted languages make writing and debugging faster.

R is memory-hungry. Unlike SAS, which was developed with a metaphor of punch cards being processed one at a time, R tries to operate on the whole data set at once. If you are lucky enough to have a gigantic data set, with hundreds of thousands of observations or more, you will need to find ways (such as using R's capability to connect directly to database software) to do your analysis in chunks rather than loading it all into memory at once.

Unlike some other software such as Maple or Mathematica, R can't do *symbolic* calculation. For example, it can't tell you that the integral of x^2 is $x^3/3 + C$, although it can compute some simple derivatives (using the `deriv` or `D` function).

No commercial organization supports R—which may not matter as much as you think. The largest software company in the world supports Microsoft Excel, but Excel's statistical procedures are notoriously unreliable (McCullough and Wilson, 2005). On the other hand, the community of researchers who build and use R are among the best in the world, and R compares well with commercial software (Keeling and Pavur, 2007). While every piece of software has bugs, the core components of R have been used so extensively by so many people that the chances of your finding a bug in R are about the same as the chances of finding a bug in a commercial software package like SAS or SPSS—and if you do find one and report it, it will probably be fixed within a few days.

It is certainly possible to do the kinds of modeling presented in this book with other computing platforms—particularly MATLAB (with appropriate toolboxes), Mathematica, SAS (using the macro language), Excel in combination with Visual Basic, and lower-level languages such as Delphi, Java, C, or FORTRAN. However, I have found R's combination of flexibility, power, and cost make it the best—although not the only—option for statistical modeling in ecology.

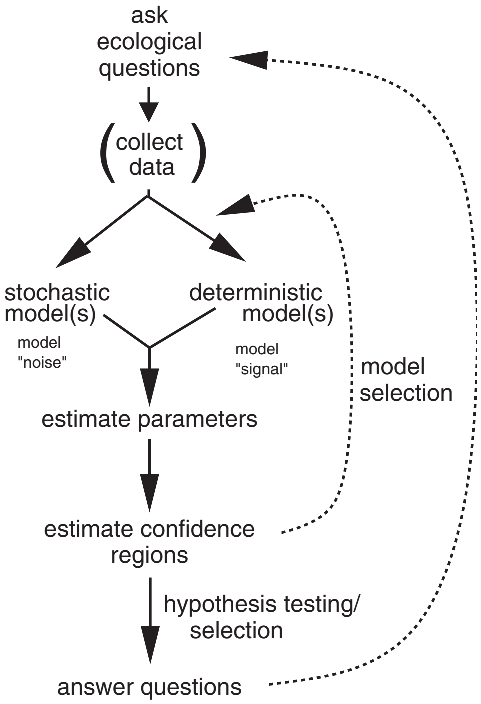

Figure 1.5 Flow of the modeling process.

1.6 Outline of the Modeling Process

After all these caveats and admonitions and before jumping into the nitty-gritty details of modeling particular data, we need an outline or road map of the modeling process (Figure 1.5).

1. **Identify the ecological question.** You have to know what you want to find out before you can start trying to model. You should know what your question is both at a general, conceptual level ("Does disease select against cannibalism in tiger salamander populations?") and at a specific level ("What is the percentage difference in probability of becoming a cannibal for tiger salamander individuals taken from populations *A* and *B*?"). Practice switching back and forth between these two levels. Being either too vague ("I want to explore the population genetics of cannibalism") or too specific ("What is the difference in the intercepts of these two linear regressions?") can impede your progress. Ultimately, knowing how to ask good questions is one of the fundamental skills for any ecologist, or indeed any scientist, and (unfortunately) no recipe can tell you how to do it. Even though I can't teach you to ask good questions, I included it in the list because it is the

first and most important step of any analysis and motivates all the other steps.*

2. **Choose deterministic model(s).** Next, you need to choose a particular mathematical description of the pattern you are trying to describe. The *deterministic* part is the average, or expected pattern in the absence of any kind of randomness or measurement error. It's tempting to call this an "ecological" model, since traditional ecological models are described in deterministic terms, but ecological models can be either deterministic or stochastic.

   The deterministic model can be phenomenological (as simple as "predator density is a linear function of prey density, or $P = a + bV$"); mechanistic (e.g., a type II functional response for predation rate); or even a complex individual-based simulation model. Chapter 3 will remind you of, or introduce you to, a broad range of mathematical models that are useful building blocks for a deterministic model, and provide general tools for getting acquainted with the mathematical properties of deterministic models.
3. **Choose stochastic model(s).** To estimate the parameters of a model, you need to know not just the expected pattern but also something about the variation around the expected pattern. Typically, you describe the stochastic model by specifying a reasonable *probability distribution* for the variation. For example, we often assume that variation that comes from measurement error is normally distributed, while variation in the number of plants found in a quadrat of a specific size is Poisson distributed. Ecologists tend to be less familiar with stochastic building blocks (e.g., the negative binomial or Gamma distributions) than with deterministic building blocks (e.g., linear or Michaelis-Menten functions). The former are frequently covered in the first week of introductory statistics courses and then forgotten as you learn standard statistical methods. Chapter 4 will (re)introduce some basics of probability as well as a wide range of probability distributions useful in building stochastic models.
4. **Fit parameters.** Once you have defined your model, you can estimate both the deterministic parameters (slope, attack rate, handling time, etc.) and stochastic parameters (e.g., the variance or parameters controlling the variance). This step is a purely technical exercise in figuring out how to get the computer to fit the model to the data. Unlike the previous steps, it provides no particular insight into the basic ecological questions. The fitting step does require ecological insight both as input (for most fitting procedures, you must start with some order-of-magnitude idea of reasonable parameter values) and output (the fitted parameters are essentially the answers to your ecological question). Chapters 6 and 7 will go into great detail about the practical aspects of fitting: the basic methods, how to make them work in R, and troubleshooting tips.
5. **Estimate confidence intervals/test hypotheses/select models.** You need to know more than just the best-fit parameters of the model (the *point estimates*, in statistical jargon). Without some measurement of uncertainty, such estimates

* In an ideal world, you would identify ecological questions before you designed your experiments and gathered data (!), but in this book I will assume you've already got data (either your own or someone else's) to work with and think about.

are meaningless. By quantifying the uncertainty in the fit of a model, you can estimate confidence limits for the parameters. You can also test ecological hypotheses, from both an ecological and a statistical point of view (e.g., can we tell the difference statistically between the handling times on two different prey types? are these differences large enough to make any practical difference in the population dynamics?). You also need to quantify uncertainty in order to choose the best out of a set of competing models, or to decide how to weight the predictions of different models. All of these procedures—estimating confidence limits, testing the differences between parameters in two models or between a parameter and a null-hypothesis value such as zero, and testing whether one model is significantly better than another—are closely related aspects of the modeling process that we will discuss in Chapter 6.

6. **Put the results together to answer questions/return to step #1.** Modeling is an iterative process. You may have answered your questions with a single pass through steps 1–5, but it is far more likely that estimating parameters and confidence limits will force you to redefine your models (changing their form or complexity or the ecological covariates they take into account) or even to redefine your original ecological questions. You may need to ask different questions, or collect another set of data, to further understand how your system works. Like the first step, this final step is a bit more free-form and general, but there are tools (the Likelihood Ratio test, model selection) that will help (Chapter 6).

I use this approach for modeling ecological systems every day. It answers ecological questions and, more important, it shapes the way I think about data and about those ecological questions. A growing number of studies in ecology use simple but realistic statistical models that do not fit easily into classical statistical frameworks (Butler and Burns, 1993; Ribbens et al., 1994; Pascual and Kareiva, 1996; Ferrari and Sugita, 1996; Damgaard, 1999; Strong et al., 1999; Ricketts, 2001; Lytle, 2002; Dalling et al., 2002; Ovaskainen, 2004; Tracey et al., 2005; Fujiwara et al., 2005; Sandin and Pacala, 2005; Agrawal and Fishbein, 2006; Canham and Uriarte, 2006; Horne and Garton, 2006; Ness et al., 2006; Sack et al., 2006; Wintle and Bardos, 2006). Like any tool, these tools also bias my thinking ("if you have a hammer, everything looks like a nail") and the kinds of questions I like to think about. They are most useful for ecological systems where you want to test among a well-defined set of plausible mechanisms, and where you have measured a few potentially important predictor and response variables. They work less well for generalized "fishing expeditions" where you have measured lots of variables and want to try to sort them out.

1.7 R Supplement

Each chapter ends with a set of notes on R, providing more details of the commands and ideas introduced in the chapter or examples worked in more detail. For this largely conceptual chapter, the notes are about how to get R and how to get it working on your computer.

1.7.1 Installing R; Prebasics

- *Download* R. If R is already installed on your computer, skip this step. If not, here's how to get it from the Web.* Go to the R project home page (`http://www.r-project.org`) or to CRAN, the repository for R materials (`http://cran.r-project.org`), and navigate to the binary (precompiled) distributions. Find the latest version for your operating system, download it, and follow the instructions to install it. The installation file is moderately large (the Windows installer for R version 2.5.0 was 28.5 megabytes) but should download easily over a fast connection. It should be fine to accept all the defaults in the installation process.

  R should work well on any reasonably modern computer. Version 2.5.0 requires MacOS 10.2 (or higher) or Windows 98 (or higher), or just about any version of Linux; it can also be compiled on other versions of Unix. MacOS version 10.4.4 or higher and Windows XP or higher are recommended. I developed and ran all the code in the book with R 2.5.0 on a dual-core PC laptop running at 1.66 GHz under Ubuntu Linux 7.04.

  After you have played with R a bit, you may want to take a moment to install extra packages (see below).
- *Start* R. If you are using Windows or MacOS there is probably an R icon on your desktop—click on it. Or use the menus your operating system provides to find R. If you are on a Unix system, you can probably just type `R` on the command line.
- *Play with* R *a little bit*. When you start R, you will see a *command prompt*—a > that waits for you to type something and hit ENTER. When you type in an expression, R evaluates it and prints the answer:

  ```
  > 2 * 8

  [1] 16

  > sqrt(25)

  [1] 5
  ```

  (The number `[1]` before the answer says that the answer is the first element in a vector; don't worry about this now.)

  If you use an equals sign to assign a value to a *variable*, then R will silently do what you asked. To see the value of the variable, type its name at the command prompt:

  ```
  > x = sqrt(36)
  > x

  [1] 6
  ```

  A variable name can be any sequence of alphanumeric characters, as well as "`_`" or "`.`" (but no spaces), that starts with a letter. Variable names are case-sensitive, so `x` and `X` are different variables.

* These instructions are accurate at press time—but all software, and stuff from the Web in particular, is subject to change. So details may vary.

For more information, read the *Introduction to* R that comes with your copy of R (look in the documentation section of the menus), get one of the introductory documents from the R Web site, dip into an introductory book (Dalgaard, 2003; Crawley, 2005), or get Lab 1 from `http://press.princeton.edu/titles/8709.html`.

- *Stopping* R. To stop R, type `q()` (with the empty parentheses) at the command prompt, or choose "Quit" from the appropriate menu. You can say "no" when R asks if you want to save the workspace.

  To stop a long computation without stopping R, type `ESCAPE` or click on the stop sign on the toolbar (in the R console in Windows or MacOS) or type `Control-C` (in Unix or MacOS if using the command-line version).
- *The help system*. If you type `help.start()`, R will open a Web browser with help information. If you type `?cmd`, R will open a help page with information on a particular command (e.g., `?sqrt` to get information on the square-root command). `example(cmd)` will run any examples that are included in the help page for command `cmd`. If you type `help.search("topic")` (with quotes), R will list information related to `topic` available in the base system or in any extra installed packages; use `?topic` to see the information, perhaps using `library(pkg)` to load the appropriate package first. `help(package="pkg")` will list all the help pages for a loaded package. If you type `RSiteSearch("topic")`, R will search an online database for information on `topic`. Try out one or more of these aspects of the help system.
- *Install extra packages*. R has many extra packages. You may be able to install new packages from a menu within R. You can always type

  ```
  > install.packages("plotrix")
  ```

  (this installs the `plotrix` package). You can install more than one package at a time:

  ```
  > install.packages(c("ellipse", "plotrix"))
  ```

  (`c` stands for "combine" and is the command for combining multiple things into a single object.) If the machine on which you use R is not connected to the Internet, you can download the packages to some other medium (such as a flash drive or CD) and install them later, using the menu or

  ```
  > install.packages("plotrix", repos = NULL)
  ```

  Installing packages may fail if you do not have permission to write to the folder (directory) where R is installed on your computer—which may happen if you are working on a public computer. In this case, R will ask you if it's OK to install the packages in a different location. Say yes, and ignore any warnings about R being unable to update the help index.

  Finding information about functions in R packages is a bit tricky. By default, help (or ?) only search for packages that have been loaded with `library`. The `help.search` function will tell you about the existence of functions in packages that are installed but not loaded (use `help.search("topic", agrep=FALSE)` to turn off the sometimes irritating "fuzzy" matching behavior), but to see the help information you have to load the package or specify the package with help(function, package="pkg"). RSiteSearch (or the R Site Search Sidebar for Firefox, http://addictedtor.free.fr./rsitesearch/) are the only ways to find information about functions from packages you have not installed.

Here are all the packages used in this book that are not included with R by default:

```
adapt       bbmle     chron   coda            ellipse    emdbook
gplots      gtools    gdata   MCMCpack        odesolve   plotrix
R2WinBUGS   reshape   rgl     scatterplot3d
```

If you install the emdbook package first (install.packages ("emdbook")), load it (library (emdbook)), and then run the command get.emdbook .packages() (you do need the empty parentheses), it will install these packages for you automatically.

(R2WinBUGS is an exception to R's normally seamless cross-platform operation: it depends on a Windows program called WinBUGS. WinBUGS will also run on Linux, and MacOS on Intel hardware, with the help of a program called WINE: see Chapter 6.)

Installing these packages now will save time.

1.7.2 R *Interfaces*

While R works perfectly well out of the box, some interfaces can make your R experience easier. Editors such as Tinn-R (Windows), Kate (Linux), or Emacs/ESS will color R commands and quoted material, allow you to submit lines or blocks of R code to an R session, and give hints about function arguments; the standard MacOS interface has all of these features built in. Graphical interfaces such as JGR (cross-platform) or SciViews (Windows) include similar editors and have extra functions such as a workspace browser for looking at all the variables you have defined. (All of these interfaces, which are designed to facilitate R programming, are in a different category from Rcmdr, which tries to simplify basic statistics in R.) If you are using Windows or Linux I strongly recommend that, once you have tried R a little bit, you download at least an R-aware editor and possibly a GUI to make your life easier. Links to all of these systems can be found at http://www.r-project.org/GUI/.

1.7.3 *Sample Session*

Start R. Then:

Start the Web interface to the help system:

```
> help.start()
```

Seed the pseudo-random-number generator, using an arbitrary integer, to make results match if you start a new session (it's fine to skip this step, but the particular values you get from the random-number commands will be different every time—you won't get exactly the results shown below):

```
> set.seed(101)
```

Create the variable frogs (representing the density of adult frogs in each of 20 populations) from scratch by entering 20 numbers with the c command. Create a second variable tadpoles (the density of tadpoles in each population) by generating 20 normally distributed random numbers, each with twice the mean of the corresponding frogs population and a standard deviation of 0.5:

```
> frogs = c(1.1, 1.3, 1.7, 1.8, 1.9, 2.1, 2.3, 2.4,
+     2.5, 2.8, 3.1, 3.3, 3.6, 3.7, 3.9, 4.1, 4.5,
+     4.8, 5.1, 5.3)
> tadpoles = rnorm(n = 20, mean = 2 * frogs, sd = 0.5)
```

The + at the beginning of the second line is a *continuation character*. If you hit ENTER and R recognizes that your command is unfinished, it will print a + to tell you that you can continue on the next line. Sometimes the continuation character means that you forgot to close parentheses or quotes. To discard what you've done so far and start again, type ESCAPE (on Windows or MacOS) or Control-C (on Linux) or click on the stop sign on the menu.

You can name the *arguments* (n, mean, sd above) in an R function, but R can also recognize the order: tadpoles = rnorm(20,2*frogs,0.5) will give the same answer. In general, however, it's clearer and safer to name arguments.

Notice that R doesn't tell you what's in these variables unless you ask it. Entering a variable name by itself tells R to print the value of the variable:

```
> tadpoles
```

```
 [1] 2.036982 2.876231 3.062528 3.707180 3.955385 4.786983
 [7] 4.909395 4.743633 5.458514 5.488370 6.463224 6.202578
[13] 7.913878 6.666590 7.681658 8.103331 8.575123 9.629233
[19] 9.791165 9.574846
```

(The numbers at the beginning of the line are indices.) This rule of printing a variable that is entered on a line by itself also explains why typing q rather than q() prints out R code rather than quitting R. R interprets q() as "run the function q without any arguments"; it interprets q as "print the contents of variable q."

Plot tadpoles against frogs (frogs on the *x* axis, tadpoles on the *y* axis) and add a straight line with intercept 0 and slope 2 to the plot (the result should appear in a new window, looking like Figure 1.6):

```
> plot(frogs, tadpoles)
> abline(a = 0, b = 2)
```

Try calculating the (natural) logarithm of tadpoles and plot it instead:

```
> log_tadpoles = log(tadpoles)
> plot(frogs, log_tadpoles)
```

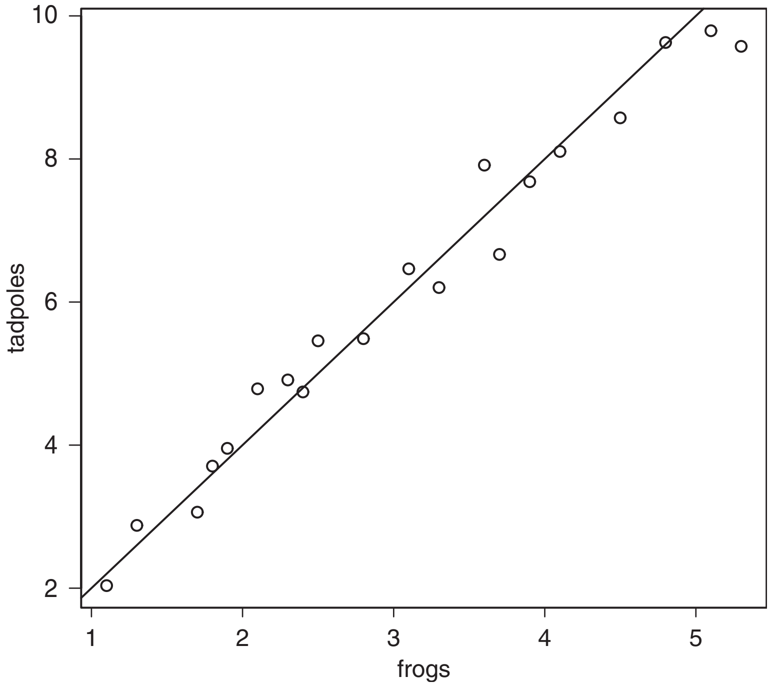

Figure 1.6 Plotting example.

You can get the same plot by typing `plot(frogs,log(tadpoles))` or a similar plot that adjusts the axes rather than the values with `plot(frogs,tadpoles,log="y")`. Use `log10(tadpoles)` to get the logarithm base 10.

Set up a variable `n` with integers ranging from 1 to 20 (the length of the `frogs` variable) and plot `frogs` against it:

```
> n = 1:length(frogs)
> plot(n, frogs)
```

(You'd get almost the same plot by typing `plot(frogs)`.)

R's default plotting character is an open circle. Open symbols are generally better than closed symbols for plotting because it is easier to see where they overlap, but you could include `pch=16` in the `plot` command if you wanted filled circles instead. Figure 1.7 shows several more ways to adjust the appearance of lines and points in R.

Calculate the mean, standard deviation, and a set of summary statistics for `tadpoles`:

```
> mean(tadpoles)

[1] 6.081341

> sd(tadpoles)

[1] 2.370449

> summary(tadpoles)
```

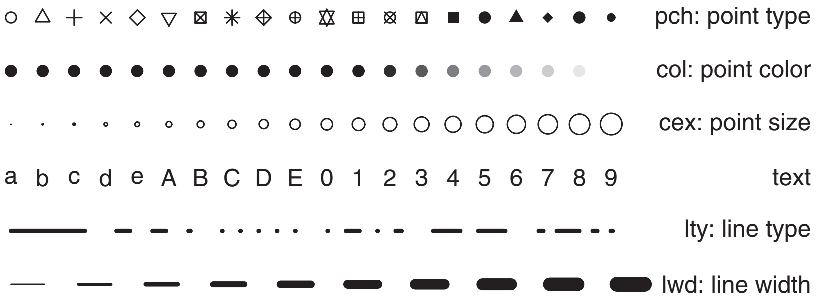

Figure 1.7 Some of R's graphics parameters. Color specification, `col`, also applies in many other contexts: all colors are set to gray scales here. See `?par` for (many more) details on graphics parameters, and one or more of `?rgb`, `?palette`, or `apropos("color")` for more on colors.

```
     Min.    1st Qu.    Median    Mean    3rd Qu.    Max.
     2.037   4.547      5.845     6.081   7.961      9.791
```

"`1st Qu.`" and "`3rd Qu.`" represent the first and third quartiles of the data. The summary statistics are displayed to only three significant digits, which can occasionally cause confusion.

Calculate the correlation between frogs and tadpoles:

```
> cor(frogs, tadpoles)

[1] 0.9870993
```

Test the statistical significance of the correlation:

```
> cor.test(frogs, tadpoles)

    Pearson's product-moment correlation

data:  frogs and tadpoles
t = 26.1566, df = 18, p-value = 8.882e-16
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 0.9669568 0.9949946
sample estimates:
      cor
0.9870993
```

The p-value here is extraordinarily low because we made up the data with very little noise: you should consider reporting it simply as $p < 0.001$. `cor.test` does a Pearson correlation test by default, but you can choose other tests; see `?cor.test`.

Look for more information on correlations:

```
> help.search("correlation")
```

Now move on to Chapter 2 to see how to deal with real data.