1 Introduction

1.1 About the book

Purpose

The purpose of this book is to teach you how to make statistical arguments using computational approaches. Such arguments are based on test-statistic probability distributions predicted by hypotheses, as in the classical approach. Unlike the classical approach, these predicted probability distributions are calculated by direct simulation of the hypotheses themselves. This approach enables you to use anything that can be calculated (or observed) from your data as a test statistic, and hypothesize any probabilistic mechanism that can generate data sets similar in structure to the one you observed. This approach frees you from the constraints, mysterious formulas, and sophisticated mathematics that classical statistics entails, and enables you to take personal control of your statistical arguments. To access this power, you will need to learn to program a computer (if you do not already know how). This task is greatly simplified through the use of spreadsheet macros, which enable the organization and input of data, as well as the output of results, using the spreadsheet itself. Many of you are already familiar with spreadsheets. The macros you will need to program will serve mostly to perform calculations, so that you will need to learn only a small sub-set of the programming language. In this book, I discuss basic hypothesis-testing statistical argument, data structures, choice of test statistics, some probability theory and its use in formulating hypotheses, and enough programming techniques to specify the calculations that simulate data sets using probabilistic hypotheses. Much of the discourse is with natural examples. Although this computational approach to statistical argument is widely applicable, these examples are mostly drawn from anthropology, ecology, evolution, and paleontology, which are my areas of interest, and those of most of my students.

Intended readers

This book is intended for readers who aspire to become, are already becoming, or who have become, research scientists who would like to feel more in control of their statistical arguments. It does not expect you to have prior training in statistics or computer programming, but some of my students who have had such training found this book valuable because it provided a very different view of these subjects, especially of statistics. Earlier versions of this book have existed in unpublished form for the past decade. I used them to teach my course on computational approaches to statistical argument at the University of Michigan. My students have been mostly Ph.D. students (about half in biological anthropology, and the others in paleontology, ecology, and evolution, with a few other areas occasionally represented). However, a few masters students, undergraduates planning to attend graduate school, post-doctoral fellows, and even fellow faculty have also participated.

Spreadsheets – This book will teach you how to make statistical arguments using a computational approach. To access this power, this text will show you all you need to know to program macros for an EXCEL spreadsheet. Here you will find examples of complete EXCEL macros. You can cut and paste them into your own EXCEL workbook and run them there. However, to do just this would miss the point. These examples show you how to program specific tasks, and are intended to be part of your personal programming manual, together with the more general explanations in the text. After all, you are reading this text because you want to free yourself from dependency on canned programs, stat packages, etc., to take responsibility for your own statistical arguments. Many students find it effective to copy examples into their macro editor using their keyboard, and then experiment with modifications of them to solidify their understanding. Later, when programming macros for your own research applications, you will be able to re-use blocks of statements from these examples.

Some of my students, who felt anxious at the beginning of my course because they did not already know how to program, bought books about programming, or downloaded things from the web about programming, or got help from a friend who had taken a course on programming. Because this book presents only a small fraction of the macro programming language, there is less to learn; extra, unnecessary information leads to

confusion and delay. This book has all you need to know to do the things you need to do to start taking a computational approach to statistical argument using a spreadsheet. A few of my students, after they had mastered the concepts of this book, decided that they wanted to climb the steep learning curve to master an object-oriented, general programming language, such as JAVA. This enabled them to participate more fully in the present-day world of computing, but did not enhance their ability to take a computational approach to statistical arguments.

Why use computation

Over the past decade, the use of computational approaches to statistical argument in research publications has increased, although it is still not widely used, because, in part, many scientists lack the ability to access these techniques, and those who can mostly include those who are also very well grounded in the techniques of classical statistics. As a consequence, such computational approaches are often seen as especially sophisticated, when in fact they are much easier to master than a strong grounding in classical statistics. Many early career scientists remain unable to access computational approaches to statistical argument mostly because there is no effective, entry-level instructional book to teach them. This is such a book.

This book describes methods for formulating hypotheses and for calculating predictions from them so that they may be tested with data. These methods enable you to formulate hypotheses and to design experiments, so that you can make inferences from data, free of the burden of unwanted mathematical assumptions. Until recently, few natural scientists who wanted to reason with quantitative data could do so with a full understanding of the data analytic methods they were expected to use. Why? During the early part of the twentieth century, the inventors of statistics made assumptions and used mathematical techniques in order to avoid the impossibly large amounts of computation that would otherwise have been required. They brought a new level of rigor to inference making, which made a major contribution to the methods of natural scientists and other scholars. However, this (now) classical approach to statistics presents three problems: (1) the mathematics is too hard for most otherwise excellent research scientists to master; (2) the assumptions were NOT what scientists

needed to include in their hypotheses; and (3) as a consequence of the inaccessibility of the mathematics and the irrelevance of the assumptions, many scientists turned to a statistical expert to decide the details of forming a hypothesis and choosing a test statistic, and did not take direct responsibility for their own hypotheses and arguments.

Now, with personal computers, you can do vast quantities of calculations rapidly and painlessly. So claim power over, and responsibility for, your own statistical arguments. This book will help students (undergraduate, graduate, or established professionals) who aspire to careers that include the practice of scientific research (experimental and/or comparative) to learn how to structure data to facilitate questions, to hypothesize probabilistic mechanisms that describe variability, to choose relevant test statistics, and to instruct personal computers to put those mechanisms into motion to simulate the probability distribution predicted by their hypothesis for that statistic.

Prerequisites

Although the examples here will come primarily from biological and cultural anthropology, paleontology, ecology, and organismal evolution, the methods and concepts are more generally applicable. In addition to a strong desire to take responsibility for your own hypotheses and arguments, you should also have a (nearly) complete undergraduate major in a science, broadly construed, to be able to take full advantage of this book. A background in organismal natural science (e.g., ecology, evolution, biological anthropology, paleontology, archeology, natural resources, botany, among others) will help you recognize the examples in this book, but any social or natural science background is sufficient. Some recollection of high-school algebra will also be useful.

How to use this book

The book starts with a description of some basic principles, and programming concepts. It then uses a published case study to show you in more detail how to instruct your personal computer to carry out the computation you need to put these principles into practice. These pages are intended to give you an initial feel for this approach. Please do not expect to understand

or remember every concept on your first reading. These concepts and techniques will be visited repeatedly through the rest of the book. When you have had a little practice programming spreadsheet macros yourself, come back and read these pages again. Data structures and probability mechanisms will continue to be described so that you can calculate predictions using your personal computer. You will be shown specific programming techniques, with examples using spreadsheet macros ready for you to use as examples and modify to meet your own needs. These computational methods avoid the need to understand the mathematics, and to accept the pre-determined assumptions and test statistics of classical statistics. They enable you to understand every step of your own argument.

Although any of several computer source languages could have been used for these instructional purposes, I chose EXCEL macros. There are many good reasons for this choice. EXCEL macros are programmed in a so-called structured language, which means that data structures are explicit and instructions are hierarchically nested to reflect the logic of the algorithms. Most of you are at least somewhat familiar with spreadsheets already, even if you have never programmed macros for them. Much of what is difficult about programming in a more general language, such as PASCAL\DELPHI, C or JAVA, is managing the input of data and the output of results (so called I/O), especially when graphics interfaces (like WINDOWS and MAC operating systems) are involved. Spreadsheet macros can read data right from a spreadsheet, and can write results to a spreadsheet as well. Thus, most of the programming you will need to learn will be to instruct the computer to carry out the calculations you want it to perform. With less concern for I/O, it is much easier to learn enough about programming so that you can write spreadsheet macros to carry out the computational approach to statistical argument described here.

To get the most out of this book, it is a very good idea to copy, using your keyboard, the example programs (macros) in the text into your spreadsheet macro editor and run them yourself. Once you get started, you can experiment with your own modifications of them, and later the text will challenge you with solving problems by writing your own programs, which you will be able to do by copying from the examples you already have. To do this you will need access to a personal computer that runs a spreadsheet that enables you to write macros. Microsoft chose to remove macro programming entirely from EXCEL 2008 for MACs, and to obscure access to it in

EXCEL 2007 for WINDOWS. To take advantage of macros in EXCEL on a Mac you will need the 2004 (or earlier) version. Macro programming is readily available in earlier versions of EXCEL: (1) open EXCEL, (2) bring down the "Tools" menu, (3) open the "macros" sub-menu and (4) click on "Visual Basic Editor". The macros hiding place in EXCEL 2007 for WINDOWS is found like this: (1) open EXCEL, (2) click the "office" button in the upper left, which opens a little window where in the bottom right you can (3) click a button called "EXCEL options", which opens a menu where you can (4) click "Show developer tab" and then "OK", which closes all these little windows and leaves EXCEL open with the developer tab showing, where you can now (5) click "Visual Basic". Later, when you start programming yourself, how to use the macro editor and run macros will be explained in more detail.

To take a computational approach to statistical argument you will need to be able to instruct a computer to perform computation. However, this text is mostly not about computer programming; it is about statistical arguments, and how to make them using computation, instead of classical statistics. Toward this goal, I will discuss the nature of statistical argument, data structures, the choice of test statistics, and probability concepts. To do this, the text will rely heavily on examples, mostly taken from real questions asked by real scientists, using real data from the natural world, or using simplified data to make the calculations easier, while illustrating the concepts more clearly.

Brief overview

A discussion of argument styles places scientific statistical argument in context with other argument styles common in our culture. Scientific argument is based on hypothesis, prediction, and comparison with data. There are four parts: (1) intellectual foundation enables a question and disciplines observations; (2) data structure enables a statistical hypothesis; (3) probability concepts are part of a statistical hypothesis; and (4) a test statistic is a way to calculate something relevant from your data. A statistical hypothesis predicts a probability distribution for the text statistic.

You can calculate the value of any statistic from your data set. You can also calculate the probability distribution for that statistic predicted by any hypothesis that includes a random process capable of generating data sets

with the same structure. To do this, you write a program for your computer. To use computation to make statistical scientific arguments, you need only a few programming concepts. These few concepts are described in the context of natural examples, using spreadsheet macros. Some history of programming concepts is followed by explicit description of the concepts and language of EXCEL, the macro programming that you need to take a computational approach to statistical argument. Now you can start programming yourself, with explicit instructions to use the EXCEL macro editor.

The first example is about a plant adaptation. It integrates scientific argument with the steps you can take to write an EXCEL spreadsheet macro to calculate for a test statistic the probability distribution predicted for it by a hypothesis. Sub CARPEL presents all the programming statements of the complete macro that implements this example. However, your practice is the essential step toward being able to use the concepts and methods of this book in your own research. Next you can follow specific instructions for how to practice. An easy, but realistic, problem is posed and then solved step by step, with advice for effective programming.

The importance of choosing a relevant test statistic is discussed in the context of a specific example taken from paleontology. This example is continued to show you how to create a macro to calculate the realized significance of a relevant test statistic. Sub PERIOD presents all the programming statements of the complete macro that implements this example, and some new programming techniques are introduced.

Next, random variables and their distributions are described somewhat more formally. You are shown how spreadsheet macros use a random number generator to create and plot probability distributions, so that you can see what they look like. Arithmetic with random variables is not the same as arithmetic with numbers. You are shown how to think about and do arithmetic with random variables, using a computational approach. Expected value and variance are properties of a probability distribution that can be useful in your scientific arguments. You are shown how to write macros to estimate expected value and variance.

Another technique for hypothesizing a probability mechanism is called re-sampling data. In the context of a natural example, an unusual but appropriate test statistic is described, and you are shown how to calculate the probability distribution predicted for it by the hypothesis of re-sampling

data. Procedures are macros inside a main macro. They enable blocks of instructions to be executed in more than one place in a main macro, without copying them again. This is illustrated using the example of the shell sort. The SORT and PERMUTE procedures are used to illustrate some concepts of testing procedures to increase confidence that they are working properly before they are called by other macros.

The concept of a family of parametric distributions is introduced with the binomial distributions. A formula to calculate probabilities for distributions in the Poisson family is derived in the context of spatial distribution of trees in a savanna. Programming techniques to sample a Poisson distribution are presented. The family of normal distributions is the limiting sum of many independent samples of any random variable. A macro to sample a normal distribution is described. The negative binomial, chi square, and F distributions are briefly described. Later in the context of goodness-of-fit, chi square will be discussed in more detail. A macro to calculate the percentiles of any distribution is described.

The family of data structures, called a linear model, is a useful concept from classical statistics. There is a variety of ways to quantify the error made by a variety of ways to summarize data. The classical way to choose a linear model family member to summarize your data is to minimize squared error. Techniques to discover a particular linear model that minimizes the squared error in summarizing your data are explained. Matrix notation is introduced to describe the calculating formulas that emerge from this explanation. Natural examples use simplified data to better illustrate concepts. One application of a linear model determines a line that minimizes squared error, while summarizing the relationship between two different measurements of the same things. The concepts are presented in the context of a natural example, and a procedure to implement this application is described. A linear model is a way to structure and describe data. It specifies no hypothesized random process to generate data, and no test statistics. To hypothesize a random process, the computational approaches of re-sampling and permutation depart from classical statistics. A computational approach enables the use of any test statistic. A macro to test the significance of the sum of squared errors is described as an example. The concepts of two-way analysis of variance are illustrated using natural examples from ecology, and a procedure for inverting matrices is described.

An estimator is a random variable used to estimate a parameter of the distribution of some other random variable. An estimator is created by doing arithmetic with observed data, hypothesized to be sampling a member of a family of parametric distributions. Consistent, unbiased, and maximum likelihood properties of estimators are explained. Do samples of an unknown random variable (your data) seem to be sampling a given random variable? One way to answer using the classical chi square statistic is discussed in some detail. Computational techniques to compute its predicted probability distribution are presented. With these techniques any relevant statistic can be used; alternative statistics are described.

What can happen when two random processes both participate in generating data? In the artificial example here, you can test a simpler hypothesis to distinguish how two random processes interact. Macro programming techniques are shown. Subtler concerns for choice of test statistic are examined. Finally, a natural example is discussed. Random walks, more general auto-regressive series, and Markov chains, are described, with natural examples.

A computational approach to the analysis of covariance is illustrated with an example of the coevolution of two properties of the same species over time, and a macro to implement this approach. Statistical tests for difference in sexual dimorphism between two species of different size are confounded by size difference. Re-scaling techniques to address this kind of problem are described. Sub SEXDIMO presents the programming statements that implement this example.

How can you get away with peeking at data? Usually, arguments based on classical statistics are less valid if you look at data before you formulate a hypothesis that you intend to test with them. Natural examples illustrate ways to incorporate into the hypothesis the peeking itself. Using a computational approach, you can calculate valid predicted probability distributions for test statistics, even if you peeked.

Two different ways to classify the same collection of entities can be related or independent. The classical approach to testing this independence statistically is problematic with small amounts of data. A computational approach can help. ACTUS2, "Analysis of Contingency Tables Using Simulation, version 2" is an example of a computational approach to statistical argument that I have published. ACTUS2 is a program that uses computation to test for dependencies in sparse contingency tables. How

to interpret the results of ACTUS2 is described. Spreadsheet ACTUS is a macro that implements the two hypotheses of independence of ACTUS2, using a spreadsheet for input and output. This section explains how to use it, with explicit instructions. Sub ACTUS presents all the programming statements of the complete macro that implements spreadsheet ACTUS.

1.2 Basic principles

Applicability

As a research scientist, your purpose is to invent explanations for natural phenomena, contribute them to the pool of competing explanations, and argue their differential credibility. Note that human beings are a part of the natural world, and phenomena involving them may also be construed as natural; so arguments to support or question explanations of human culture or behavior can also be scientific. A scientific approach to the study of human culture and behavior is called social science. The concepts and methods presented here are also applicable to social science.

Argument style

This book is primarily concerned with the argument style that compares observations of natural phenomena (data) to predictions that follow as a logical consequence of a particular explanation. If the observed data are more or less the same as the predictions then we say the data are consistent with that explanation; if the observed data are different from the predictions then we say the data are not consistent with the explanation. Sometimes we describe results more starkly: we say the data fail to disprove an explanation, or we say the data disprove the explanation. Sound scientific argument style evaluates the credibility of any explanation that can make predictions about something observable.

Another argument style, which we might call advocacy, has been common in our western culture for centuries. When practicing advocacy, advocates look for evidence that supports their claim and ignore the rest. Advocates then use only this evidence to formulate arguments that support their claim. Perhaps another person or group does the same for a competing or contradictory claim. Each side presents its evidence and arguments in an