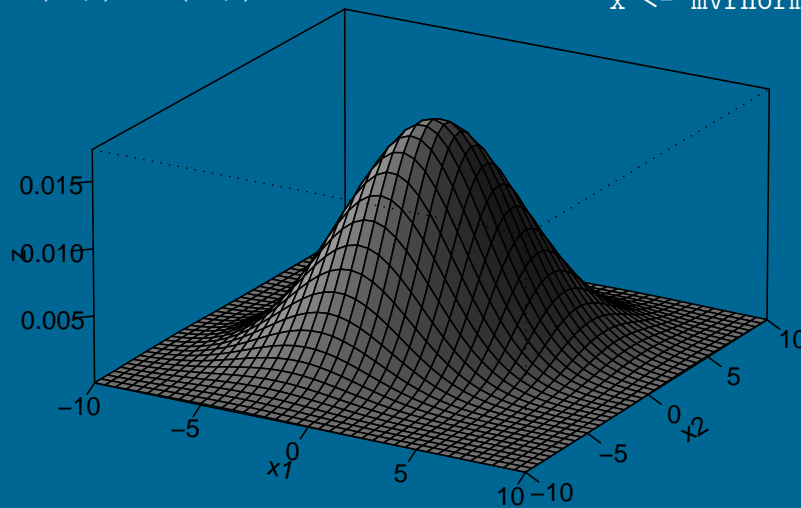# From Algorithms to Z-Scores:

Probabilistic and Statistical Modeling in Computer Science

Norm Matloff, University of California, Davis

$$f_X(t) = ce^{-0.5(t-\mu)'\Sigma^{-1}(t-\mu)}$$

```
library(MASS)
x <- mvrnorm(mu,sgm)
```


See Creative Commons license at

http://heather.cs.ucdavis.edu/ matloff/probstatbook.html

Author's Biographical Sketch

Dr. Norm Matloff is a professor of computer science at the University of California at Davis, and was formerly a professor of statistics at that university. He is a former database software developer in Silicon Valley, and has been a statistical consultant for firms such as the Kaiser Permanente Health Plan.

Dr. Matloff was born in Los Angeles, and grew up in East Los Angeles and the San Gabriel Valley. He has a PhD in pure mathematics from UCLA, specializing in probability theory and statistics. He has published numerous papers in computer science and statistics, with current research interests in machine learning, parallel processing, statistical computing, and regression methodology.

Prof. Matloff is a former appointed member of IFIP Working Group 11.3, an international committee concerned with database software security, established under the United Nations. He was a founding member of the UC Davis Department of Statistics, and participated in the formation of the UCD Computer Science Department as well. He is a recipient of the campuswide Distinguished Teaching Award and Distinguished Public Service Award at UC Davis.

Dr. Matloff is currently Editor-in-Chief of the *R Journal*, and is on the editorial board of the *Journal of Statistical Software*.

Dr. Matloff is the author of several published textbooks. His book *Statistical Regression and Classification: from Linear Models to Machine Learning*, won the Ziegel Award in 2017.

Contents

Preface

Why is this book different from all other books on mathematical probability and statistics? The key aspect is the book's consistently *applied* approach, especially important for engineering students.

The applied nature comes is manifested in a number of senses. First, there is a strong emphasis on intution, with less mathematical formalism. In my experience, defining probability via sample spaces, the standard approach, is a major impediment to doing good applied work. The same holds for defining expected value as a weighted average. Instead, I use the intuitive, informal approach of long-run frequency and long-run average. I believe this is especially helpful when explaining conditional probability and expectation, concepts that students tend to have trouble with. (They often think they understand until they actually have to work a problem using the concepts.)

On the other hand, in spite of the relative lack of formalism, all models and so on are described precisely in terms of random variables and distributions. And the material is actually somewhat more mathematical than most at this level in the sense that it makes extensive usage of linear algebra.

Second, the book stresses *real-world* applications. Many similar texts, notably the elegant and interesting book for computer science students by Mitzenmacher, focus on probability, in fact discrete probability. Their intended class of "applications" is the theoretical analysis of algorithms. I instead focus on the actual use of the material in the real world; which tends to be more continuous than discrete, and more in the realm of statistics than probability. This should prove especially valuable, as "big data" and machine learning now play a significant role in applications of computers.

Third, there is a strong emphasis on modeling. Considerable emphasis is placed on questions such as: What do probabilistic models really mean, in real-life terms? How does one choose a model? How do we assess the practical usefulness of models? This aspect is so important that there is a separate chapter for this, titled Introduction to Model Building. Throughout the text, there is considerable discussion of the real-world meaning of probabilistic concepts. For instance, when probability density functions are introduced, there is an extended discussion regarding the intuitive meaning of densities in light of the inherently-discrete nature of real data, due to the finite precision of measurement.

Finally, the R statistical/data analysis language is used throughout. Again, several excellent texts on probability and statistics have been written that feature R, but this book, by virtue of having a computer science audience, uses R in a more sophisticated manner. My open source tutorial on R programming, *R for Programmers* (`http://heather.cs.ucdavis.edu/~matloff/R/RProg.pdf`), can be used as a supplement. (More advanced R programming is covered in my book, *The Art of R Programming*, No Starch Press, 2011.)

There is a large amount of material here. For my one-quarter undergraduate course, I usually cover Chapters 2, 3, 6, 7, 8, 10, 12, 15, 16, 17, 18 and 22. My lecture style is conversational, referring to material in the book and making lots of supplementary remarks ("What if we changed the assumption here to such-and-such?" etc.). Students read the details on their own. For my one-quarter graduate course, I cover Chapters 10, 11, 19, **??**, **??**, 20, 21, 22, 23, 24 and 25.

As prerequisites, the student must know calculus, basic matrix algebra, and have some skill in programming. As with any text in probability and statistics, it is also necessary that the student has a good sense of math intuition, and does not treat mathematics as simply memorization of formulas.

The LATEXsource `.tex` files for this book are in `http://heather.cs.ucdavis.edu/~matloff/132/PLN`, so readers can copy the R code and experiment with it. (It is not recommended to copy-and-paste from the PDF file, as hidden characters may be copied.) The PDF file is searchable.

The following, among many, provided valuable feedback for which I am very grateful: Ibrahim Ahmed; Ahmed Ahmedin; Stuart Ambler; Earl Barr; Benjamin Beasley; Matthew Butner; Michael Clifford; Dipak Ghosal; Noah Gift; Laura Matloff; Nelson Max, Connie Nguyen, Jack Norman, Richard Oehrle, Michael Rea, Sana Vaziri, Yingkang Xie, and Ivana Zetko. The cover picture, by the way, is inspired by an example in Romaine Francois' old R Graphics Gallery, sadly now defunct.

Many of the data sets used in the book are from the UC Irvine Machine Learning Repository, `http://archive.ics.uci.edu/ml/`. Thanks to UCI for making available this very valuable resource.

The book contains a number of references for further reading. Since the audience includes a number of students at my institution, the University of California, Davis, I often refer to work by current or former UCD faculty, so that students can see what their professors do in research.

Chapter 1

Time Waste Versus Empowerment

I took a course in speed reading, and read War and Peace in 20 minutes. It's about Russia—comedian Woody Allen

I learned very early the difference between knowing the name of something and knowing something—Richard Feynman, Nobel laureate in physics

The main goal [of this course] is self-actualization through the empowerment of claiming your education—UCSC (and former UCD) professor Marc Mangel, in the syllabus for his calculus course

What does this really mean? Hmm, I've never thought about that—UCD PhD student in statistics, in answer to a student who asked the actual meaning of a very basic concept

You have a PhD in engineering. You may have forgotten technical details like $\frac{d}{dt}sin(t) = cos(t)$, but you should at least understand the concepts of rates of change—the author, gently chiding a friend who was having trouble following a simple quantitative discussion of trends in California's educational system

Give me six hours to chop down a tree and I will spend the first four sharpening the axe—Abraham Lincoln

The field of probability and statistics (which, for convenience, I will refer to simply as "statistics" below) impacts many aspects of our daily lives—business, medicine, the law, government and so on. Consider just a few examples:

- The statistical models used on Wall Street made the "quants" (quantitative analysts) rich—but also contributed to the worldwide financial crash of 2008.

- In a court trial, large sums of money or the freedom of an accused may hinge on whether the

judge and jury understand some statistical evidence presented by one side or the other.

- Wittingly or unconsciously, you are using probability every time you gamble in a casino—and every time you buy insurance.

- Statistics is used to determine whether a new medical treatment is safe/effective for you.

- Statistics is used to flag possible terrorists—but sometimes unfairly singling out innocent people while other times missing ones who really are dangerous.

Clearly, statistics *matters*. But it only has value when one really *understands* what it means and what it does. Indeed, blindly plugging into statistical formulas can be not only valueless but in fact highly dangerous, say if a bad drug goes onto the market.

Yet most people view statistics as exactly that—mindless plugging into boring formulas. If even the statistics graduate student quoted above thinks this, how can the students taking the course be blamed for taking that atititude?

I once had a student who had an unusually good understanding of probability. It turned out that this was due to his being highly successful at playing online poker, winning lots of cash. No blind formula-plugging for him! He really had to *understand* how probability works.

Statistics is *not* just a bunch of formulas. On the contrary, it can be mathematically deep, for those who like that kind of thing. (Much of statistics can be viewed as the Pythagorean Theorem in n-dimensional or even infinite-dimensional space.) But the key point is that *anyone* who has taken a calculus course can develop true understanding of statistics, of real practical value. As Professor Mangel says, that's empowering.

Statistics is based on probabilistic models. So, in order to become effective at data analysis, one must really master the principles of probability; this is where Lincoln's comment about "sharpening the axe" truly applies.

So as you make your way through this book, always stop to think, "What does this equation really mean? What is its goal? Why are its ingredients defined in the way they are? Might there be a better way? How does this relate to our daily lives?" Now THAT is empowering.

Chapter 2

Basic Probability Models

This chapter will introduce the general notions of probability. Most of it will seem intuitive to you, but pay careful attention to the general principles which are developed; in more complex settings intuition may not be enough, and the tools discussed here will be very useful.

2.1 ALOHA Network Example

Throughout this book, we will be discussing both "classical" probability examples involving coins, cards and dice, and also examples involving applications to computer science. The latter will involve diverse fields such as data mining, machine learning, computer networks, software engineering and bioinformatics.

In this section, an example from computer networks is presented which will be used at a number of points in this chapter. Probability analysis is used extensively in the development of new, faster types of networks.

Today's Ethernet evolved from an experimental network developed at the University of Hawaii, called ALOHA. A number of network nodes would occasionally try to use the same radio channel to communicate with a central computer. The nodes couldn't hear each other, due to the obstruction of mountains between them. If only one of them made an attempt to send, it would be successful, and it would receive an acknowledgement message in response from the central computer. But if more than one node were to transmit, a **collision** would occur, garbling all the messages. The sending nodes would timeout after waiting for an acknowledgement which never came, and try sending again later. To avoid having too many collisions, nodes would engage in random **backoff**, meaning that they would refrain from sending for a while even though they had something to send.

One variation is **slotted** ALOHA, which divides time into intervals which I will call "epochs." Each

epoch will have duration 1.0, so epoch 1 extends from time 0.0 to 1.0, epoch 2 extends from 1.0 to 2.0 and so on. In the version we will consider here, in each epoch, if a node is **active**, i.e. has a message to send, it will either send or refrain from sending, with probability p and 1-p. The value of p is set by the designer of the network. (Real Ethernet hardware does something like this, using a random number generator inside the chip.)

The other parameter q in our model is the probability that a node which had been inactive generates a message during an epoch, i.e. the probability that the user hits a key, and thus becomes "active." Think of what happens when you are at a computer. You are not typing constantly, and when you are not typing, the time until you hit a key again will be random. Our parameter q models that randomness.

Let n be the number of nodes, which we'll assume for simplicity is two. Assume also for simplicity that the timing is as follows. Arrival of a new message happens in the middle of an epoch, and the decision as to whether to send versus back off is made near the end of an epoch, say 90% into the epoch.

For example, say that at the beginning of the epoch which extends from time 15.0 to 16.0, node A has something to send but node B does not. At time 15.5, node B will either generate a message to send or not, with probability q and 1-q, respectively. Suppose B does generate a new message. At time 15.9, node A will either try to send or refrain, with probability p and 1-p, and node B will do the same. Suppose A refrains but B sends. Then B's transmission will be successful, and at the start of epoch 16 B will be inactive, while node A will still be active. On the other hand, suppose both A and B try to send at time 15.9; both will fail, and thus both will be active at time 16.0, and so on.

Be sure to keep in mind that in our simple model here, during the time a node is active, it won't generate any additional new messages.

(Note: The definition of this ALOHA model is summarized concisely on page 10.)

Let's observe the network for two epochs, epoch 1 and epoch 2. Assume that the network consists of just two nodes, called node 1 and node 2, both of which start out active. Let X_1 and X_2 denote the numbers of active nodes at the *very end* of epochs 1 and 2, *after possible transmissions*. We'll take p to be 0.4 and q to be 0.8 in this example.

Let's find $P(X_1 = 2)$, the probability that $X_1 = 2$, and then get to the main point, which is to ask what we really mean by this probability.

How could $X_1 = 2$ occur? There are two possibilities:

- both nodes try to send; this has probability p^2

- neither node tries to send; this has probability $(1 - p)^2$

1,1	1,2	1,3	1,4	1,5	1,6
2,1	2,2	2,3	2,4	2,5	2,6
3,1	3,2	3,3	3,4	3,5	3,6
4,1	4,2	4,3	4,4	4,5	4,6
5,1	5,2	5,3	5,4	5,5	5,6
6,1	6,2	6,3	6,4	6,5	6,6

Table 2.1: Sample Space for the Dice Example

Thus

$$P(X_1 = 2) = p^2 + (1-p)^2 = 0.52 \tag{2.1}$$

2.2   A "Notebook" View: the Notion of a Repeatable Experiment

It's crucial to understand what that 0.52 figure really means in a practical sense. To this end, let's put the ALOHA example aside for a moment, and consider the "experiment" consisting of rolling two dice, say a blue one and a yellow one. Let X and Y denote the number of dots we get on the blue and yellow dice, respectively, and consider the meaning of $P(X + Y = 6) = \frac{5}{36}$.

In the mathematical theory of probability, we talk of a **sample space**, which (in simple cases) consists of the possible outcomes (X, Y), seen in Table 2.1. In a theoretical treatment, we place weights of 1/36 on each of the points in the space, reflecting the fact that each of the 36 points is equally likely, and then say, "What we mean by $P(X + Y = 6) = \frac{5}{36}$ is that the outcomes (1,5), (2,4), (3,3), (4,2), (5,1) have total weight 5/36."

Unfortunately, the notion of sample space becomes mathematically tricky when developed for more complex probability models. Indeed, it requires graduate-level math, called **measure theory**.

And much worse, under the sample space approach, one loses all the intuition. In particular, **there is no good way using set theory to convey the intuition underlying conditional probability** (to be introduced in Section 2.3). The same is true for expected value, a central topic to be introduced in Section 3.5.

In any case, most probability computations do not rely on explicitly writing down a sample space. In this particular example it is useful for us as a vehicle for explaining the concepts, but we will NOT use it much. Those who wish to get a more theoretical grounding can get a start in Section

notebook line	outcome	blue+yellow = 6?
1	blue 2, yellow 6	No
2	blue 3, yellow 1	No
3	blue 1, yellow 1	No
4	blue 4, yellow 2	Yes
5	blue 1, yellow 1	No
6	blue 3, yellow 4	No
7	blue 5, yellow 1	Yes
8	blue 3, yellow 6	No
9	blue 2, yellow 5	No

Table 2.2: Notebook for the Dice Problem

5.4.

But the intuitive notion—which is FAR more important—of what $P(X + Y = 6) = \frac{5}{36}$ means is the following. Imagine doing the experiment many, many times, recording the results in a large notebook:

- Roll the dice the first time, and write the outcome on the first line of the notebook.

- Roll the dice the second time, and write the outcome on the second line of the notebook.

- Roll the dice the third time, and write the outcome on the third line of the notebook.

- Roll the dice the fourth time, and write the outcome on the fourth line of the notebook.

- Imagine you keep doing this, thousands of times, filling thousands of lines in the notebook.

The first 9 lines of the notebook might look like Table 2.2. Here 2/9 of these lines say Yes. But after many, many repetitions, approximately 5/36 of the lines will say Yes. For example, after doing the experiment 720 times, approximately $\frac{5}{36} \times 720 = 100$ lines will say Yes.

This is what probability really is: In what fraction of the lines does the event of interest happen? **It sounds simple, but if you always think about this "lines in the notebook" idea, probability problems are a lot easier to solve.** And it is the fundamental basis of computer simulation.

2.3 Our Definitions

These definitions are intuitive, rather than rigorous math, but intuition is what we need. Keep in mind that we are making <u>definitions</u> below, not listing properties.

- We assume an "experiment" which is (at least in concept) repeatable. The experiment of rolling two dice is repeatable, and even the ALOHA experiment is so. (We simply watch the network for a long time, collecting data on pairs of consecutive epochs in which there are two active stations at the beginning.) On the other hand, the econometricians, in forecasting 2009, cannot "repeat" 2008. Yet all of the econometricians' tools assume that events in 2008 were affected by various sorts of randomness, and we think of repeating the experiment in a conceptual sense.

- We imagine performing the experiment a large number of times, recording the result of each repetition on a separate line in a notebook.

- We say A is an **event** for this experiment if it is a possible boolean (i.e. yes-or-no) outcome of the experiment. In the above example, here are some events:

    * X+Y = 6
    * X = 1
    * Y = 3
    * X-Y = 4

- A **random variable** is a numerical outcome of the experiment, such as X and Y here, as well as X+Y, 2XY and even sin(XY).

- For any event of interest A, imagine a column on A in the notebook. The kth line in the notebook, k = 1,2,3,..., will say Yes or No, depending on whether A occurred or not during the kth repetition of the experiment. For instance, we have such a column in our table above, for the event {A = blue+yellow = 6}.

- For any event of interest A, we define P(A) to be the long-run fraction of lines with Yes entries.

- For any events A, B, imagine a new column in our notebook, labeled "A and B." In each line, this column will say Yes if and only if there are Yes entries for both A and B. P(A and B) is then the long-run fraction of lines with Yes entries in the new column labeled "A and B."[1]

---

[1]In most textbooks, what we call "A and B" here is written A∩B, indicating the intersection of two sets in the sample space. But again, we do not take a sample space point of view here.

- For any events A, B, imagine a new column in our notebook, labeled "A or B." In each line, this column will say Yes if and only if at least one of the entries for A and B says Yes.[2]

- For any events A, B, imagine a new column in our notebook, labeled "A | B" and pronounced "A given B." In each line:

  * This new column will say "NA" ("not applicable") if the B entry is No.
  * If it is a line in which the B column says Yes, then this new column will say Yes or No, depending on whether the A column says Yes or No.

Think of probabilities in this "notebook" context:

- P(A) means the long-run fraction of lines in the notebook in which the A column says Yes.

- P(A or B) means the long-run fraction of lines in the notebook in which the A-or-B column says Yes.

- P(A and B) means the long-run fraction of lines in the notebook in which the A-and-B column says Yes.

- P(A | B) means the long-run fraction of lines in the notebook in which the A | B column says Yes—**among the lines which do NOT say NA.**

A hugely common mistake is to confuse P(A and B) and P(A | B). This is where the notebook view becomes so important. In the dice example, compare the quantities $P(X = 1 \text{ and } S = 6) = \frac{1}{36}$ and $P(X = 1 | S = 6) = \frac{1}{5}$, where S = X+Y:[3]

- After a large number of repetitions of the experiment, approximately 1/36 of the lines of the notebook will have the property that both X = 1 and S = 6 (since X = 1 and S = 6 is equivalent to X = 1 and Y = 5).

- After a large number of repetitions of the experiment, if **we look only at the lines in which S = 6**, then **among those lines**, approximately 1/5 of **those lines** will show X = 1.

The quantity P(A|B) is called the **conditional probability of A, given B**.

Note that *and* has higher logical precedence than *or*. For example, P(A and B or C) means P[(A and B) or C]. Also, *not* has higher precedence than *and*.

Here are some more very important definitions and properties:

---

[2]In the sample space approach, this is written A ∪ B.

[3]Think of adding an S column to the notebook too

- **Definition 1** *Suppose A and B are events such that it is impossible for them to occur in the same line of the notebook. They are said to be* **disjoint** *events.*

- If A and B are disjoint events, then

$$P(A \text{ or } B) = P(A) + P(B) \tag{2.2}$$

Again, this terminology *disjoint* stems from the set-theoretic sample space approach, where it means that A ∩ B = ϕ. That mathematical terminology works fine for our dice example, but in my experience people have major difficulty applying it correctly in more complicated problems. This is another illustration of why I put so much emphasis on the "notebook" framework.

By writing

$$\{A \text{ or } B \text{ or } C\} = \{A \text{ or } [B \text{ or } C]\} = \tag{2.3}$$

(2.2) can be iterated, e.g.

$$P(A \text{ or } B \text{ or } C) = P(A) + P(B) + P(C) \tag{2.4}$$

- If A and B are not disjoint, then

$$P(A \text{ or } B) = P(A) + P(B) - P(A \text{ and } B) \tag{2.5}$$

In the disjoint case, that subtracted term is 0, so (2.5) reduces to (2.2).

- **Definition 2** *Events A and B are said to be* **stochastically independent***, usually just stated as* **independent***,*[4] *if*

$$P(A \text{ and } B) = P(A) \cdot P(B) \tag{2.6}$$

- In calculating an "and" probability, how does one know whether the events are independent? The answer is that this will typically be clear from the problem. If we toss the blue and yellow dice, for instance, it is clear that one die has no impact on the other, so events involving the blue die are independent of events involving the yellow die. On the other hand, in the ALOHA example, it's clear that events involving X_1 are NOT independent of those involving X_2.

---

[4]The term *stochastic* is just a fancy synonym for *random*.

- If A and B are not independent, the equation (2.6) generalizes to

$$P(A \text{ and } B) = P(A)P(B|A) \tag{2.7}$$

This should make sense to you. Suppose 30% of all UC Davis students are in engineering, and 20% of all engineering majors are female. That would imply that 0.30 x 0.20 = 0.06, i.e. 6% of all UCD students are female engineers.

Note that if A and B actually are independent, then $P(B|A) = P(B)$, and (2.7) reduces to (2.6).

Note too that (2.7) implies

$$P(B|A) = \frac{P(A \text{ and } B)}{P(A)} \tag{2.8}$$

2.4   "Mailing Tubes"

If I ever need to buy some mailing tubes, I can come here—friend of the author's, while browsing through an office supplies store

Examples of the above properties, e.g. (2.6) and (2.7), will be given starting in Section 2.5. But first, a crucial strategic point in learning probability must be addressed.

Some years ago, a friend of mine was in an office supplies store, and he noticed a rack of mailing tubes. My friend made the remark shown above. Well, (2.6) and 2.7 are "mailing tubes"–make a mental note to yourself saying, "If I ever need to find a probability involving *and*, one thing I can try is (2.6) and (2.7)." **Be ready for this!**

This mailing tube metaphor will be mentioned often, such as in Section 3.5.5 .

2.5   Example: ALOHA Network

Please keep in mind that the notebook idea is simply a vehicle to help you understand what the concepts really mean. This is crucial for your intuition and your ability to apply this material in the real world. But the notebook idea is NOT for the purpose of calculating probabilities. Instead, we use the properties of probability, as seen in the following.

Let's look at all of this in the ALOHA context. Here's a summary:

- We have n network nodes, sharing a common communications channel.

- Time is divided in epochs. X_k denotes the number of active nodes at the end of epoch k, which we will sometimes refer to as the **state** of the system in epoch k.

- If two or more nodes try to send in an epoch, they collide, and the message doesn't get through.

- We say a node is active if it has a message to send.

- If a node is active node near the end of an epoch, it tries to send with probability p.

- If a node is inactive at the beginning of an epoch, then at the middle of the epoch it will generate a message to send with probability q.

- In our examples here, we have n = 2 and $X_0 = 2$, i.e. both nodes start out active.

Let's say the two nodes consist of terminals, at which John and Mary are typing. Call the nodes Node John and Node Mary.

Now, in Equation (2.1) we found that

$$P(X_1 = 2) = p^2 + (1 - p)^2 = 0.52 \tag{2.9}$$

How did we get this? Let C_i denote the event that node i tries to send, i = 1,2. Then using the definitions above, our steps would be

$$
\begin{aligned}
P(X_1 = 2) &= P(\underbrace{C_1 \text{ and } C_2}_{} \text{ or } \underbrace{\text{not } C_1 \text{ and } \text{ not } C_2}_{}) & (2.10) \\
&= P(C_1 \text{ and } C_2) + P(\text{ not } C_1 \text{ and } \text{ not } C_2) \text{ (from (2.2))} & (2.11) \\
&= P(C_1)P(C_2) + P(\text{ not } C_1)P(\text{ not } C_2) \text{ (from (2.6))} & (2.12) \\
&= p^2 + (1 - p)^2 & (2.13)
\end{aligned}
$$

(The underbraces in (2.10) do not represent some esoteric mathematical operation. There are there simply to make the grouping clearer, corresponding to events G and H defined below.)

Here are the reasons for these steps:

(2.10): We listed the ways in which the event $\{X_1 = 2\}$ could occur.

(2.11): Write $G = C_1$ and C_2, $H = D_1$ and D_2, where $D_i =$ not C_i, i = 1,2. Then the events G and H are clearly disjoint; if in a given line of our notebook there is a Yes for G, then definitely there will be a No for H, and vice versa.

(2.12): The two nodes act physically independently of each other. Thus the events C_1 and C_2 are stochastically independent, so we applied (2.6). Then we did the same for D_1 and D_2.

Now, what about $P(X_2 = 2)$? Again, we break big events down into small events, in this case according to the value of X_1:

$$
\begin{aligned}
P(X_2 = 2) &= P(X_1 = 0 \text{ and } X_2 = 2 \text{ or } X_1 = 1 \text{ and } X_2 = 2 \text{ or } X_1 = 2 \text{ and } X_2 = 2) \\
&= P(X_1 = 0 \text{ and } X_2 = 2) \\
&+ P(X_1 = 1 \text{ and } X_2 = 2) \\
&+ P(X_1 = 2 \text{ and } X_2 = 2)
\end{aligned}
\tag{2.14}
$$

Since X_1 cannot be 0, that first term, $P(X_1 = 0 \text{ and } X_2 = 2)$ is 0. To deal with the second term, $P(X_1 = 1 \text{ and } X_2 = 2)$, we'll use (2.7). Due to the time-sequential nature of our experiment here, it is natural (but certainly not "mandated," as we'll often see situations to the contrary) to take A and B to be $\{X_1 = 1\}$ and $\{X_2 = 2\}$, respectively. So, we write

$$
P(X_1 = 1 \text{ and } X_2 = 2) = P(X_1 = 1)P(X_2 = 2 | X_1 = 1)
\tag{2.15}
$$

To calculate $P(X_1 = 1)$, we use the same kind of reasoning as in Equation (2.1). For the event in question to occur, either Node John would send and Node B wouldn't, or John would refrain from sending and Node B would send. Thus

$$
P(X_1 = 1) = 2p(1 - p) = 0.48
\tag{2.16}
$$

Now we need to find $P(X_2 = 2 | X_1 = 1)$. This again involves breaking big events down into small ones. If $X_1 = 1$, then $X_2 = 2$ can occur only if *both* of the following occur:

- Event A: Whichever node was the one to successfully transmit during epoch 1—and we are given that there indeed was one, since $X_1 = 1$—now generates a new message.

- Event B: During epoch 2, no successful transmission occurs, i.e. either they both try to send or neither tries to send.

Recalling the definitions of p and q in Section 2.1, we have that

$$P(X_2 = 2|X_1 = 1) = q[p^2 + (1-p)^2] = 0.41 \qquad (2.17)$$

Thus $P(X_1 = 1 \text{ and } X_2 = 2) = 0.48 \times 0.41 = 0.20$.

We go through a similar analysis for $P(X_1 = 2 \text{ and } X_2 = 2)$: We recall that $P(X_1 = 2) = 0.52$ from before, and find that $P(X_2 = 2|X_1 = 2) = 0.52$ as well. So we find $P(X_1 = 2 \text{ and } X_2 = 2)$ to be $0.52^2 = 0.27$. Putting all this together, we find that $P(X_2 = 2) = 0.47$.

Let's do another; let's find $P(X_1 = 1|X_2 = 2)$. [Pause a minute here to make sure you understand that this is quite different from $P(X_2 = 2|X_1 = 1)$.] From (2.8), we know that

$$P(X_1 = 1|X_2 = 2) = \frac{P(X_1 = 1 \text{ and } X_2 = 2)}{P(X_2 = 2)} \qquad (2.18)$$

We computed both numerator and denominator here before, in Equations (2.15) and (2.14), so we see that $P(X_1 = 1|X_2 = 2) = 0.20/0.47 = 0.43$.

So, in our notebook view, if we were to look only at lines in the notebook for which $X_2 = 2$, a fraction 0.43 of *those lines* would have $X_1 = 1$.

You might be bothered that we are looking "backwards in time" in (2.18), kind of guessing the past from the present. There is nothing wrong or unnatural about that. Jurors in court trials do it all the time, though presumably not with formal probability calculation. And evolutionary biologists do use formal probability models to guess the past.

And one more calculation: $P(X_1 = 2 \text{ or } X_2 = 2)$. From (2.5),

$$P(X_1 = 2 \text{ or } X_2 = 2) = P(X_1 = 2) + P(X_2 = 2) - P(X_1 = 2 \text{ and } X_2 = 2) \qquad (2.19)$$

Luckily, we've already calculated all three probabilities on the right-hand side to be 0.52, 0.47 and 0.27, respectively. Thus $P(X_1 = 2 \text{ or } X_2 = 2) = 0.72$.

Note by the way that events involving X_2 are NOT independent of those involving X_1. For instance, we found in (2.18) that

$$P(X_1 = 1|X_2 = 2) = 0.43 \qquad (2.20)$$

yet from (2.16) we have

$$P(X_1 = 1) = 0.48. \qquad (2.21)$$

2.6   Example: Dice and Conditional Probability

Note in particular that $P(B|A)$ and $P(A|B)$ are completely different quantities. The first restricts attention to lines of the notebook in which A occurs, while for the second we look at lines in which B occurs.

Suppose two dice are rolled, resulting in the random variables X and Y. Let S be the sum X+Y, and let T denote the number of dice having an even number of dots, 0, 1 or 2. Suppose we hear that S = 12, and we know nothing else. Then we know that T must be 2. In other words,

$$P(T = 2 \mid S = 12) = 1 \tag{2.22}$$

On the other hand, if we hear instead that T = 2, that does *not* imply that S is 12, i.e.

$$P(S = 12 \mid T = 2) < 1 \tag{2.23}$$

In other words

$$P(S = 12 \mid T = 2) \neq P(T = 2 \mid S = 12) \tag{2.24}$$

(In fact the reader should show that $P(S = 12 \mid T = 2) = 1/9$.

2.7   ALOHA in the Notebook Context

Think of doing the ALOHA "experiment" many, many times.

- Run the network for two epochs, starting with both nodes active, the first time, and write the outcome on the first line of the notebook.

- Run the network for two epochs, starting with both nodes active, the second time, and write the outcome on the second line of the notebook.

- Run the network for two epochs, starting with both nodes active, the third time, and write the outcome on the third line of the notebook.

- Run the network for two epochs, starting with both nodes active, the fourth time, and write the outcome on the fourth line of the notebook.

- Imagine you keep doing this, thousands of times, filling thousands of lines in the notebook.

notebook line	$X_1 = 2$	$X_2 = 2$	$X_1 = 2$ and $X_2 = 2$	$X_2 = 2	X_1 = 2$
1	Yes	No	No	No
2	No	No	No	NA
3	Yes	Yes	Yes	Yes
4	Yes	No	No	No
5	Yes	Yes	Yes	Yes
6	No	No	No	NA
7	No	Yes	No	NA

Table 2.3: Top of Notebook for Two-Epoch ALOHA Experiment

The first seven lines of the notebook might look like Table 2.3. We see that:

- Among those first seven lines in the notebook, 4/7 of them have $X_1 = 2$. After many, many lines, this fraction will be approximately 0.52.

- Among those first seven lines in the notebook, 3/7 of them have $X_2 = 2$. After many, many lines, this fraction will be approximately 0.47.[5]

- Among those first seven lines in the notebook, 2/7 of them have $X_1 = 2$ and $X_2 = 2$. After many, many lines, this fraction will be approximately 0.27.

- Among the first seven lines in the notebook, four of them do not say NA in the $X_2 = 2 | X_1 = 2$ column. **Among these four lines**, two say Yes, a fraction of 2/4. After many, many lines, this fraction will be approximately 0.52.

2.8 A Note on Modeling

Here is a question to test your understanding of the ALOHA model—not the calculation of probabilities, but rather the meaning of the model itself. What kinds of properties are we trying to capture in the model?

So, consider this question:

> Consider the ALOHA network model. Say we have two such networks, A and B. In A, the network typically is used for keyboard input, such as a user typing e-mail or editing

---

[5] Don't make anything of the fact that these probabilities nearly add up to 1.

a file. But in B, users tend to do a lot of file uploading, not just typing. Fill in the blanks: In B, the model parameter _____ is _____ than in A, and in order to accommodate this, we should set the parameter _____ to be relatively _____ in B.

In network B we have heavy traffic. Thus, when a node becomes idle, it is quite likely to have a new message to send right away.[6] Thus q is large.

That means we need to be especially worried about collisions, so we probably should set p to a low value.

2.9   Solution Strategies

The example in Section 2.5 shows typical stategies in exploring solutions to probability problems, such as:

- Name what seem to be the important variables and events, in this case X_1, X_2, C_1, C_2 and so on.

- Write the given probability in terms of those named variables, e.g.

$$P(X_1 = 2) = P(\underbrace{C_1 \text{ and } C_2} \text{ or } \underbrace{\text{not } C_1 \text{ and } \text{ not } C_2}) \tag{2.25}$$

  above.

- Ask the famous question, "How can it happen?" Break big events down into small events; in the above case the event $X_1 = 2$ can happen if C_1 and C_2 or not C_1 and not C_2.

- But when you do break things down like this, make sure to neither expand or contract the scope of the probability. Say you write something like

$$P(A) = P(B) \tag{2.26}$$

  where B might be some complicated event expression such as in the right-hand side of (2.10). Make SURE that A and B are equivalent—meaning that in every notebook line in which A occurs, then B also occurs, and *vice versa*.

---

[6]The file is likely read in chunks called disk *sectors*, so there may be a slight pause between the uploading of chunks. Our model here is too coarse to reflect such things.

- Do not write/think nonsense. For example: the expression "P(A) or P(B)" is nonsense—do you see why? Probabilities are numbers, not boolean expressions, so "P(A) or P(B)" is like saying, "0.2 or 0.5"—meaningless!

  Similarly, say we have a random variable X. The "probability" P(X) is invalid. Say X is the number of dots we get when we roll a single die. Then P(X) would mean "the probability that the number of dots," which is nonsense English! P(X = 3) is valid, but P(X) is meaningless.

  Please note that = is not like a comma, or equivalent to the English word *therefore*. It needs a left side and a right side; "a = b" makes sense, but "= b" doesn't.

- Similarly, don't use "formulas" that you didn't learn and that are in fact false. For example, in an expression involving a random variable X, one can NOT replace X by its mean. (How would you like it if your professor were to lose your exam, and then tell you, "Well, I'll just assign you a score that is equal to the class mean"?)

- Adhere to convention! Use capital letters for random variables and names of events. Use P() notation, not p() (which will mean something else in this book).

- In the beginning of your learning probability methods, meticulously write down all your steps, with reasons, as in the computation of $P(X_1 = 2)$ in Equations (2.10)ff. After you gain more experience, you can start skipping steps, but not in the initial learning period.

- Solving probability problems—and even more so, building useful probability models—is like computer programming: It's a creative process.

  One can NOT—repeat, NOT—teach someone how to write programs. All one can do is show the person how the basic building blocks work, such as loops, if-else and arrays, then show a number of examples. But the actual writing of a program is a creative act, not formula-based. The programmer must creatively combined the various building blocks to produce the desired result. The teacher cannot teach the student how to do this.

  The same is true for solving probability problems. The basic building blocks were presented above in Section 2.5, and many more "mailing tubes" will be presented in the rest of this book. But it is up to the student to try using the various building blocks in a way that solves the problem. Sometimes use of one block may prove to be unfruitful, in which case one must try other blocks.

  For instance, in using probability formulas like P(A and B) = P(A) P(B|A), there is no magic rule as to how to choose A and B.

  Moreover, if you need P(B|A), there is no magic rule on how to find it. On the one hand, you might calculate it from (2.8), as we did in (2.18), but on the other hand you may be able to reason out the value of P(B|A), as we did following (2.16). Just try some cases until you find one that works, in the sense that you can evaluate both factors. It's the same as trying various programming ideas until you find one that works.

2.10   Example: A Simple Board Game

Consider a board game, which for simplicity we'll assume consists of two squares per side, on four sides. A player's token advances around the board. The squares are numbered 0-7, and play begins at square 0. The token advances according to the roll of a single die. If a player lands on square 3, he/she gets a bonus turn.

In most problems like this, **it is greatly helpful to first name the quantities or events involved**. Toward that end, let R denote the player's first roll, and let B be his bonus if there is one, with B being set to 0 if there is no bonus.

Let's find the probability that a player has yet to make a complete circuit of the board—i.e. has not yet reached or passed 0—after the first turn (including the bonus, if any).

$$
\begin{aligned}
P(\text{doesn't reach or pass } 0) \;\; &= \;\; P(R + B \le 7) & (2.27)\\
&= \;\; P(R \le 6, R \ne 3 \text{ or } R = 3, B \le 4) & (2.28)\\
&= \;\; P(R \le 6, R \ne 3) + P(R = 3, B \le 4) & (2.29)\\
&= \;\; P(R \le 6, R \ne 3) + P(R = 3)\ P(B \le 4 \mid R = 3) & (2.30)\\
&= \;\; \frac{5}{6} + \frac{1}{6} \cdot \frac{4}{6} & (2.31)\\
&= \;\; \frac{17}{18} & (2.32)
\end{aligned}
$$

(Above we have written commas as a shorthand notation for *and*, a common abbreviation.)

Now here is a very subtle issue. The events $R + B = 3$ and $B \le 4$ are *not* independent. At first it would seem that they're independent, as we are rolling the die twice, and successful rolls are independent. That's true, they are independent, but remember, B can be 0 (no bonus), and the event $B \le 4$ includes that 0 case. And if we know $R = 3$, then we know for sure that B cannot be 0, so we see the two events are not independent.

Note that we first had to "translate" the English specification of the problem ("doesn't reach or...") to math symbols. As mentioned, this is quite helpful.

Now, here's a shorter way (there are always multiple ways to do a problem):

$$P(\text{don't reach or pass } 0) \; = \; 1 - P(\text{do reach or pass } 0) \tag{2.33}$$

$$= \; 1 - P(R + B > 7) \tag{2.34}$$

$$= \; 1 - P(R = 3, B > 4) \tag{2.35}$$

$$= \; 1 - \frac{1}{6} \cdot \frac{2}{6} \tag{2.36}$$

$$= \; \frac{17}{18} \tag{2.37}$$

Now suppose that, according to a telephone report of the game, you hear that on the player's first turn, his token ended up at square 4. Let's find the probability that he got there with the aid of a bonus roll.

Note that this a conditional probability—we're finding the probability that the player got a bonus roll, <u>given</u> that we know he ended up at square 4. The word *given* wasn't there in the statement of the problem, but it was implied.

A little thought reveals that we cannot end up at square 4 after making a complete circuit of the board, which simplifies the situation quite a bit. So, write

$$P(B > 0 \mid R + B = 4) \; = \; \frac{P(R + B = 4, B > 0)}{P(R + B = 4)} \tag{2.38}$$

$$= \; \frac{P(R + B = 4, B > 0)}{P(R + B = 4, B > 0 \text{ or } R + B = 4, B = 0)} \tag{2.39}$$

$$= \; \frac{P(R + B = 4, B > 0)}{P(R + B = 4, B > 0) + P(R + B = 4, B = 0)} \tag{2.40}$$

$$= \; \frac{P(R = 3, B = 1)}{P(R = 3, B = 1) + P(R = 4)} \tag{2.41}$$

$$= \; \frac{\frac{1}{6} \cdot \frac{1}{6}}{\frac{1}{6} \cdot \frac{1}{6} + \frac{1}{6}} \tag{2.42}$$

$$= \; \frac{1}{7} \tag{2.43}$$

We could have used Bayes' Rule to shorten the derivation a little here, but will prefer to derive everything, at least in this introductory chapter.

Pay special attention to that third equality above, as it is a frequent mode of attack in probability problems. In considering the probability P(R+B = 4, B > 0), we ask, what is a simpler—but still equivalent!—description of this event? Well, we see that R+B = 4, B > 0 boils down to R = 3, B = 1, so we replace the above probability with P(R = 3, B = 1).

Again, this is a very common approach. But be sure to take care that we are in an "if and only if" situation. Yes, R+B = 4, B > 0 implies R = 3, B = 1, but we must make sure that the converse is true as well. In other words, we must also confirm that R = 3, B = 1 implies R+B = 4, B > 0. That's trivial in this case, but one can make a subtle error in some problems if one is not careful; otherwise we will have replaced a higher-probability event by a lower-probability one.

2.11   Example: Bus Ridership

Consider the following analysis of bus ridership. (In order to keep things easy, it will be quite oversimplified, but the principles will be clear.) Here is the model:

- At each stop, each passsenger alights from the bus, independently, with probability 0.2 each.

- Either 0, 1 or 2 new passengers get on the bus, with probabilities 0.5, 0.4 and 0.1, respectively. Passengers at successive stops are independent.

- Assume the bus is so large that it never becomes full, so the new passengers can always get on.

- Suppose the bus is empty when it arrives at its first stop.

Once again, **it is greatly helpful to first name the quantities or events involved**. Let L_i denote the number of passengers on the bus as it *leaves* its ith stop, i = 1,2,3,... Let B_i denote the number of new passengers who board the bus at the ith stop.

First, let's find the probability that no passengers board the bus at the first three stops. That's easy:

$$P(B_1 = 0 \text{ and } B_2 = 0 \text{ and } B_3 = 0) = 0.5^3 \tag{2.44}$$

Now let's compute the probability that the bus leaves the second stop empty. Again, we must translate this to math first, $P(L_2 = 0)$. Now as usual, "break big events into small events":

$$
\begin{aligned}
P(L_2 = 0) &= P(B_1 = 0 \text{ and } L_2 = 0 \text{ or } B_1 = 1 \text{ and } L_2 = 0 \text{ or } B_1 = 2 \text{ and } L_2 = 0) &(2.45)\\
&= \sum_{i=0}^{2} P(B_1 = i \text{ and } L_2 = 0) &(2.46)\\
&= \sum_{i=0}^{2} P(B_1 = i)P(L_2 = 0 | B_1 = i) &(2.47)\\
&= 0.5^2 + (0.4)(0.2)(0.5) + (0.1)(0.2^2)(0.5) &(2.48)\\
&= 0.292 &(2.49)
\end{aligned}
$$

For instance, where did that first term, 0.5^2, come from? Well, $P(B_1 = 0) = 0.5$, and what about $P(L_2 = 0 | B_1 = 0)$? If $B_1 = 0$, then the bus approaches the second stop empty. For it to then *leave* that second stop empty, it must be the case that $B_2 = 0$, which has probability 0.5. In other words, $P(L_2 = 0 | B_1 = 0) = 0.5$.

As another example, suppose we are told that the bus arrives empty at the third stop. What is the probability that exactly two people boarded the bus at the first stop?

Note first what is being asked for here: $P(B_1 = 2 | L_2 = 0)$. Then we have

$$
\begin{aligned}
P(B_1 = 2 \mid L_2 = 0) &= \frac{P(B_1 = 2 \text{ and } L_2 = 0)}{P(L_2 = 0)} &(2.50)\\
&= P(B_1 = 2) \, P(L_2 = 0 \mid B_1 = 2) \, / \, 0.292 &(2.51)\\
&= 0.1 \cdot 0.2^2 \cdot 0.5/0.292 &(2.52)
\end{aligned}
$$

(the 0.292 had been previously calculated in (2.49)).

Now let's find the probability that fewer people board at the second stop than at the first:

$$
\begin{aligned}
P(B_2 < B_1) &= P(B_1 = 1 \text{ and } B_2 < B_1 \text{ or } B_1 = 2 \text{ and } B_2 < B_1) &(2.53)\\
&= 0.4 \cdot 0.5 + 0.1 \cdot (0.5 + 0.4) &(2.54)
\end{aligned}
$$

Also: Someone tells you that as she got off the bus at the second stop, she saw that the bus then left that stop empty. Let's find the probability that she was the only passenger when the bus left the first stop:

We are given that $L_2 = 0$. But we are *also* given that $L_1 > 0$. Then

$$P(L_1 = 1 | L_2 = 0 \text{ and } L_1 > 0) \quad = \quad \frac{P(L_1 = 1 \text{ and } L_2 = 0)}{P(L_2 = 0 \text{ and } L_1 > 0)} \qquad (2.55)$$

$$= \quad \frac{P(B_1 = 1 \text{ and } L_2 = 0)}{P(B_1 = 1 \text{ and } L_2 = 0 \text{ or } B_1 = 2 \text{ and } L_2 = 0)} \qquad (2.56)$$

$$= \quad \frac{(0.4)(0.2)(0.5)}{(0.4)(0.2)(0.5) + (0.1)(0.2)^2(0.5)} \qquad (2.57)$$

Equation (2.56) requires some explanation. Let's first consider how we got the numerator from the preceding equation.

Ask the usual question: How can it happen? In this case, how can the event

$$L_1 = 1 \text{ and } L_2 = 0 \qquad (2.58)$$

occur? Since we know a lot about the probabilistic behavior of the B_i, let's try to recast that event. A little thought shows that the event is equivalent to the event

$$B_1 = 0 \text{ and } L_2 = 0 \qquad (2.59)$$

So, how did the denominator in (2.56) come from the preceding equation? In other words, how did we recast the event

$$L_2 = 0 \text{ and } L_1 > 0 \qquad (2.60)$$

in terms of the B_i? Well, $L_1 > 0$ means that B_1 is either 1 or 2. Thus we broke things down accordingly in the denominator of (2.56).

The remainder of the computation is similar to what we did earlier in this example.

Here is another computation: An observer at the second stop notices that no one alights there, but it is dark and the observer couldn't see whether anyone was on the bus. Find the probability that there was one passenger on the bus at the time.

Let A_i denote the number of passengers alighting at stop i.

$$P(L_1 = 1 | A_2 = 0) \quad = \quad \frac{P(L_1 = 1 \text{ and } A_2 = 0)}{P(A_2 = 0)} \tag{2.61}$$

$$= \quad \frac{P(L_1 = 1 \text{ and } A_2 = 0)}{\sum_{j=0}^{2} P(L_1 = j \text{ and } A_2 = 0)} \tag{2.62}$$

$$= \quad \frac{0.4 \cdot 0.8}{0.5 \cdot 1 + 0.4 \cdot 0.8 + 0.1 \cdot 0.8^2} \tag{2.63}$$

2.12 Bayes' Rule

(This section should not be confused with Section 18.4. The latter is highly controversial, while the material in this section is not controversial at all.)

Following (2.18) above, we noted that the ingredients had already been computed, in (2.15) and (2.14). If we go back to the derivations in those two equations and substitute in (2.18), we have

$$P(X_1 = 1 | X_2 = 2) \quad = \quad \frac{P(X_1 = 1 \text{ and } X_2 = 2)}{P(X_2 = 2)} \tag{2.64}$$

$$= \quad \frac{P(X_1 = 1 \text{ and } X_2 = 2)}{P(X_1 = 1 \text{ and } X_2 = 2) + P(X_1 = 2 \text{ and } X_2 = 2)} \tag{2.65}$$

$$= \quad \frac{P(X_1 = 1)P(X_2 = 2 | X_1 = 1)}{P(X_1 = 1)P(X_2 = 2 | X_1 = 1) + P(X_1 = 2)P(X_2 = 2 | X_1 = 2)} \tag{2.66}$$

Looking at this in more generality, for events A and B we would find that

$$P(A|B) = \frac{P(A)P(B|A)}{P(A)P(B|A) + P(\text{not } A)P(B|\text{not } A)} \tag{2.67}$$

This is known as **Bayes' Theorem** or **Bayes' Rule**. It can be extended easily to cases with several terms in the denominator, arising from situations that need to be broken down into several subevents rather than just A and not-A.

2.13 Random Graph Models

A *graph* consists of *vertices* and *edges*. To understand this, think of a social network. Here the vertices represent people and the edges represent friendships. For the time being, assume that

friendship relations are mutual, i.e. if person i says he is friends with person j, then j will say the same about i.

For any graph, its *adjacency matrix* consists of 1 and 0 entries, with a 1 in row i, column j meaning there is an edge from vertex i to vertex j. For instance, say we have a simple tiny network of three people, with adjacency matrix

$$\begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix} \tag{2.68}$$

Row 1 of the matrix says that Person 1 is friends with persons 2 and 3, but we see from the other rows that Persons 2 and 3 are not friends with each other.

In any graph, the *degree* of a vertex is its number of edges. So, the degree of vertex i is the number of 1s in row i. In the little model above, vertex 1 has degree 2 but the other two vertices each have degree 1.

The assumption that friendships are mutual is described in graph theory as having a *undirected* graph. Note that that implies that the adjacency matrix is symmetric. However, we might model some other networks as *directed*, with adjacency matrices that are not necessarily symmetric. In a large extended family, for example, we could define edges in terms of being an elder sibling; there would be an edge from Person i to Person j if j is an older sibling of i.

Graphs need not represent people. They are used in myriad other settings, such as analysis of Web site relations, Internet traffic routing, genetics research and so on.

2.13.1   Example: Preferential Attachment Graph Model

A famous graph model is *Preferential Attachment*. Think of it again as an undirected social network, with each edge representing a "friend" relation. The number of vertices grows over time, one vertex per time step. At time 0, we have just two vertices, v_1 and v_2, with a link between them.

Thus at time 0, each of the two vertices has degree 1. Whenever a new vertex is added to the graph, it randomly chooses an existing vertex to *attach to*, creating a new edge with that existing vertex. In making that random choice, it follows probabilities in proportion to the degrees of the existing edges.

As an example of how the Preferential Attachment Model works, suppose that at just before time 2, when v_4 is added, the adjacency matrix for the graph is (2.68). Then there will be an edge created between v_4 with v_1, v_2 or v_3, with probability 2/4, 1/4 and 1/4, respectively.

Let's find $P(v_4$ attaches to $v_1)$ (in general, no longer assuming (2.68)). Let N_i denote the node

that v_i attaches to, i = 3,4,... Then, following the solution strategy "break big event down into small events," let's break this question about v_4 according to what happens with v_3:

$$
\begin{aligned}
P(N_4 = 1) &= P(N_3 = 1 \text{ and } N_4 = 1) + P(N_3 = 2 \text{ and } N_4 = 1) & (2.69) \\
&= (1/2)(2/4) + (1/2)(1/4) & (2.70) \\
&= 3/8 & (2.71)
\end{aligned}
$$

2.14 Simulation

Computer simulation essentially does in actual code what one does conceptually in our "notebook" view of probability (Section 2.2).

2.14.1 Example: Rolling Dice

If we roll three dice, what is the probability that their total is 8? We could count all the possibilities, or we could get an approximate answer via simulation:

```
1   # roll d dice; find P(total = k)
2
3   probtotk <- function(d,k,nreps) {
4      count <- 0
5      # do the experiment nreps times -- like doing nreps notebook lines
6      for (rep in 1:nreps) {
7         sum <- 0
8         # roll d dice and find their sum
9         for (j in 1:d) sum <- sum + roll()
10        if (sum == k) count <- count + 1
11     }
12     return(count/nreps)
13  }
14
15  # simulate roll of one die; the possible return values are 1,2,3,4,5,6,
16  # all equally likely
17  roll <- function() return(sample(1:6,1))
18
19  # example
20  probtotk(3,8,1000)
```

The call to the built-in R function **sample()** here says to take a sample of size 1 from the sequence of numbers 1,2,3,4,5,6. That's just what we want to simulate the rolling of a die. The code

```
for (j in 1:d) sum <- sum + roll()
```

then simulates the tossing of a die d times, and computing the sum.

2.14.2   First Improvement

Since applications of R often use large amounts of computer time, good R programmers are always looking for ways to speed things up. Here is an alternate version of the above program:

```
1   # roll d dice; find P(total = k)
2
3   probtotk <- function(d,k,nreps) {
4      count <- 0
5      # do the experiment nreps times
6      for (rep in 1:nreps)
7         total <- sum(sample(1:6,d,replace=TRUE))
8         if (total == k) count <- count + 1
9      }
10     return(count/nreps)
11  }
```

Let's first discuss the code.

```
sample(1:6,d,replace=TRUE)
```

The call to **sample()** here says, "Generate d random numbers, chosen randomly (i.e. with equal probability) from the integers 1 through 6, with replacement." Well, of course, that simulates tossing the die d times. So, that call returns a d-element array, and we then call R's built-in function **sum()** to find the total of the d dice.

Note the call to R's **sum()** function, a nice convenience.

This second version of the code here eliminates one explicit loop, which is the key to writing fast code in R. But just as important, it is more compact and clearer in expressing what we are doing in this simulation.

2.14.3   Second Improvement

Further improvements are possible. Consider this code:

```
1   # roll d dice; find P(total = k)
2
3   # simulate roll of nd dice; the possible return values are 1,2,3,4,5,6,
4   # all equally likely
5   roll <- function(nd) return(sample(1:6,nd,replace=TRUE))
6
```

```
 7   probtotk <- function(d,k,nreps) {
 8      sums <- vector(length=nreps)
 9      # do the experiment nreps times
10      for (rep in 1:nreps) sums[rep] <- sum(roll(d))
11      return(mean(sums==k))
12   }
```

There is quite a bit going on here.

We are storing the various "notebook lines" in a vector **sums**. We first call **vector()** to allocate space for it.

But the heart of the above code is the expression **sums==k**, which involves the very essence of the R idiom, **vectorization**. At first, the expression looks odd, in that we are comparing a vector (remember, this is what languages like C call an *array*), **sums**, to a scalar, **k**. But in R, every "scalar" is actually considered a one-element vector.

Fine, **k** is a vector, but wait! It has a different length than **sums**, so how can we compare the two vectors? Well, in R a vector is **recycled**—extended in length, by repeating its values—in order to conform to longer vectors it will be involved with. For instance:

```
> c(2,5) + 4:6
[1]  6 10  8
```

Here we added the vector (2,5) to (4,5,6). The former was first recycled to (2,5,2), resulting in a sum of (6,10,8).[7]

So, in evaluating the expression **sums==k**, R will recycle **k** to a vector consisting of **nreps** copies of **k**, thus conforming to the length of **sums**. The result of the comparison will then be a vector of length **nreps**, consisting of TRUE and FALSE values. In numerical contexts, these are treated at 1s and 0s, respectively. R's **mean()** function will then average those values, resulting in the fraction of 1s! That's exactly what we want.

2.14.4   Third Improvement

Even better:

```
1   roll <- function(nd) return(sample(1:6,nd,replace=TRUE))
2
3   probtotk <- function(d,k,nreps) {
4      # do the experiment nreps times
```

---

[7]There was also a warning message, not shown here. The circumstances under which warnings are or are not generated are beyond our scope here, but recycling is a very common R operation.

```
5      sums <- replicate(nreps,sum(roll(d)))
6      return(mean(sums==k))
7    }
```

R's **replicate()** function does what its name implies, in this case executing the call **sum(roll(d))**. That produces a vector, which we then assign to **sums**. And note that we don't have to allocate space for **sums**; **replicate()** produces a vector, allocating space, and then we merely point **sums** to that vector.

The various improvements shown above compactify the code, and in many cases, make it much faster.[8] Note, though, that this comes at the expense of using more memory.

2.14.5   Example: Dice Problem

Suppose three fair dice are rolled. We wish to find the approximate probability that the first die shows fewer than 3 dots, *given* that the total number of dots for the 3 dice is more than 8, using simulation.

Again, simulation is writing code the implements our "notebook" view of probability. In this case, we are working with a conditional probability, which our notebook view defined as follows. $P(B \mid A)$ is the long-run proportion of the time B occurs, *among those lines in which A occurs*. Here is the code:

```
1  dicesim <- function(nreps) {
2      count1 <- 0
3      count2 <- 0
4      for (i in 1:nreps) {
5          d <- sample(1:6,3,replace=T)
6          if (sum(d) > 8) {  # "among those lines in which A occurs"
7              count1 <- count1 + 1
8              if (d[1] < 3) count2 <- count2 + 1
9          }
10     }
11     return(count2 / count1)
12 }
```

Note carefully that we did NOT use (2.8). That would defeat the purpose of simulation, which is the model the actual process.

---

[8]You   can   measure   times   using   R's   **system.time()**   function, e.g.   via   the   call   **system.time(probtotk(3,7,10000))**.

2.14.6   Use of runif() for Simulating Events

To simulate whether a simple event occurs or not, we typically use R function **runif()**. This function generates random numbers from the interval (0,1), with all the points inside being equally likely. So for instance the probability that the function returns a value in (0,0.5) is 0.5. Thus here is code to simulate tossing a coin:

```
if (runif(1) < 0.5) heads <- TRUE else heads <- FALSE
```

The argument 1 means we wish to generate just one random number from the interval (0,1).

2.14.7   Example: ALOHA Network (cont'd.)

Following is a computation via simulation of the *approximate* values of $P(X_1 = 2)$, $P(X_2 = 2)$ and $P(X_2 = 2 | X_1 = 1)$.

```
1   # finds P(X1 = 2), P(X2 = 2) and P(X2 = 2|X1 = 1) in ALOHA example
2   sim <- function(p,q,nreps) {
3      countx2eq2 <- 0
4      countx1eq1 <- 0
5      countx1eq2 <- 0
6      countx2eq2givx1eq1 <- 0
7      # simulate nreps repetitions of the experiment
8      for (i in 1:nreps) {
9         numsend <- 0  # no messages sent so far
10        # simulate A and B's decision on whether to send in epoch 1
11        for (j in 1:2)
12           if (runif(1) < p) numsend <- numsend + 1
13        if (numsend == 1)   X1 <- 1
14        else X1 <- 2
15        if (X1 == 2) countx1eq2 <- countx1eq2 + 1
16        # now simulate epoch 2
17        # if X1 = 1 then one node may generate a new message
18        numactive <- X1
19        if (X1 == 1 && runif(1) < q) numactive <- numactive + 1
20        # send?
21        if (numactive == 1)
22           if (runif(1) < p) X2 <- 0
23           else X2 <- 1
24        else {   # numactive = 2
25           numsend <- 0
26           for (i in 1:2)
27              if (runif(1) < p) numsend <- numsend + 1
28           if (numsend == 1) X2 <- 1
29           else X2 <- 2
30        }
31        if (X2 == 2) countx2eq2 <- countx2eq2 + 1
32        if (X1 == 1) {   # do tally for the cond. prob.
33           countx1eq1 <- countx1eq1 + 1
```

```
34              if (X2 == 2) countx2eq2givx1eq1 <- countx2eq2givx1eq1 + 1
35         }
36      }
37      # print results
38      cat("P(X1 = 2):",countx1eq2/nreps,"\n")
39      cat("P(X2 = 2):",countx2eq2/nreps,"\n")
40      cat("P(X2 = 2 | X1 = 1):",countx2eq2givx1eq1/countx1eq1,"\n")
41   }
```

Note that each of the **nreps** iterations of the main **for** loop is analogous to one line in our hypothetical notebook. So, to find (the approximate value of) $P(X_1 = 2)$, divide the count of the number of times $X_1 = 2$ occurred by the number of iterations.

Again, note especially that the way we calculated $P(X_2 = 2|X_1 = 1)$ was to count the number of times $X_2 = 2$, **among those times that** $X_1 = 1$, just like in the notebook case.

Also: Keep in mind that we did NOT use (2.67) or any other formula in our simulation. We stuck to basics, the "notebook" definition of probability. This is really important if you are using simulation to confirm something you derived mathematically. On the other hand, if you are using simulation because you CAN'T derive something mathematically (the usual situation), using some of the mailing tubes might speed up the computation.

2.14.8 Example: Bus Ridership (cont'd.)

Consider the example in Section 2.11. Let's find the probability that after visiting the tenth stop, the bus is empty. This is too complicated to solve analytically, but can easily be simulated:

```
1    nreps <- 10000
2    nstops <- 10
3    count <- 0
4    for (i in 1:nreps) {
5       passengers <- 0
6       for (j in 1:nstops) {
7          if (passengers > 0)
8             for (k in 1:passengers)
9                if (runif(1) < 0.2)
10                   passengers <- passengers - 1
11          newpass <- sample(0:2,1,prob=c(0.5,0.4,0.1))
12          passengers <- passengers + newpass
13       }
14       if (passengers == 0) count <- count + 1
15    }
16   print(count/nreps)
```

Note the different usage of the **sample()** function in the call

```
sample(0:2,1,prob=c(0.5,0.4,0.1))
```

Here we take a sample of size 1 from the set {0,1,2}, but with probabilities 0.5 and so on. Since the third argument for **sample()** is **replace**, not **prob**, we need to specify the latter in our call.

2.14.9   Example: Board Game (cont'd.)

Recall the board game in Section 2.10. Below is simulation code to find the probability in (2.38):

```
1    boardsim <- function(nreps) {
2       count4 <- 0
3       countbonusgiven4 <- 0
4       for (i in 1:nreps) {
5          position <- sample(1:6,1)
6          if (position == 3) {
7             bonus <- TRUE
8             position <- (position + sample(1:6,1)) %% 8
9          } else bonus <- FALSE
10         if (position == 4) {
11            count4 <- count4 + 1
12            if (bonus) countbonusgiven4 <- countbonusgiven4 + 1
13         }
14      }
15      return(countbonusgiven4/count4)
16   }
```

2.14.10   Example: Broken Rod

Say a glass rod drops and breaks into 5 random pieces. Let's find the probability that the smallest piece has length below 0.02.

First, what does "random" mean here? Let's assume that the break points, treating the left end as 0 and the right end as 1, can be modeled with **runif()**. Here then is code to do the job:

```
# random breaks the rod in k pieces, returning the length of the
# shortest one
minpiece <- function(k) {
   breakpts <- sort(runif(k-1))
   lengths <- diff(c(0,breakpts,1))
   min(lengths)
}

# returns the approximate probability that the smallest of k pieces will
# have length less than q
bkrod <- function(nreps,k,q) {
   minpieces <- replicate(nreps,minpiece(k))
   mean(minpieces < q)
```

```
}

> bkrod(10000,5,0.02)
[1] 0.35
```

So, we generate the break points according to the model, then sort them in order to call R's **diff()** function. The latter finds differences between successvie values of its argument, which in our case will give us the lengths of the pieces. We then find the minimum length.

2.14.11   Example: Toss a Coin Until k Consecutive Heads

We toss a coin until we get k heads in a row. Let N denote the number of tosses needed, so that for instance the pattern HTHHH gives $N = 5$ for $k = 3$. Here is code that finds the approximate probability that $N > m$:

```
ngtm <- function(k,m,nreps) {
   count <- 0
   for (rep in 1:nreps) {
      consech <- 0
      for (i in 1:m) {
         toss <- sample(0:1,1)
         if (toss) {
            consech <- consech + 1
            if (consech == k) break
         } else consech <- 0
      }
      if (consech < k) count <- count + 1
   }
   return(count/nreps)
}
```


2.14.12   How Long Should We Run the Simulation?

Clearly, the large the value of **nreps** in our examples above, the more accurate our simulation results are likely to be. But how large should this value be? Or, more to the point, what measure is there for the degree of accuracy one can expect (whatever that means) for a given value of **nreps**? These questions will be addressed in Chapter 16.

2.15 Combinatorics-Based Probability Computation

And though the holes were rather small, they had to count them all—from the Beatles song, *A Day in the Life*

In some probability problems all the outcomes are equally likely. The probability computation is then simply a matter of counting all the outcomes of interest and dividing by the total number of possible outcomes. Of course, sometimes even such counting can be challenging, but it is simple in principle. We'll discuss two examples here.

2.15.1 Which Is More Likely in Five Cards, One King or Two Hearts?

Suppose we deal a 5-card hand from a regular 52-card deck. Which is larger, P(1 king) or P(2 hearts)? Before continuing, take a moment to guess which one is more likely.

Now, here is how we can compute the probabilities. **The key point is that all possible hands are equally likely, which implies that all we need to do is count them.** There are $\binom{52}{5}$ possible hands, so this is our denominator. For P(1 king), our numerator will be the number of hands consisting of one king and four non-kings. Since there are four kings in the deck, the number of ways to choose one king is $\binom{4}{1} = 4$. There are 48 non-kings in the deck, so there are $\binom{48}{4}$ ways to choose them. Every choice of one king can be combined with every choice of four non-kings, so the number of hands consisting of one king and four non-kings is $4 \cdot \binom{48}{4}$. Thus

$$P(1 \ \text{king}) = \frac{4 \cdot \binom{48}{4}}{\binom{52}{5}} = 0.299 \tag{2.72}$$

The same reasoning gives us

$$P(2 \ \text{hearts}) = \frac{\binom{13}{2} \cdot \binom{39}{3}}{\binom{52}{5}} = 0.274 \tag{2.73}$$

So, the 1-king hand is just slightly more likely.

Note that an unstated assumption here was that all 5-card hands are equally likely. That *is* a realistic assumption, but it's important to understand that it plays a key role here.

By the way, I used the R function **choose()** to evaluate these quantities, running R in interactive mode, e.g.:

```
> choose(13,2) * choose(39,3) / choose(52,5)
[1] 0.2742797
```

R also has a very nice function **combn()** which will generate all the $\binom{n}{k}$ combinations of k things chosen from n, and also will at your option call a user-specified function on each combination. This allows you to save a lot of computational work. See the examples in R's online documentation.

Here's how we could do the 1-king problem via simulation:

```
1   # use simulation to find P(1 king) when deal a 5-card hand from a
2   # standard deck
3
4   # think of the 52 cards as being labeled 1-52, with the 4 kings having
5   # numbers 1-4
6
7   sim <- function(nreps) {
8      count1king <- 0  # count of number of hands with 1 king
9      for (rep in 1:nreps) {
10         hand <- sample(1:52,5,replace=FALSE)  # deal hand
11         kings <- intersect(1:4,hand)  # find which kings, if any, are in hand
12         if (length(kings) == 1) count1king <- count1king + 1
13      }
14      print(count1king/nreps)
15   }
```

Here the **intersect()** function performs set intersection, in this case the set 1,2,3,4 and the one in the variable **hand**. Applying the `length()` function then gets us number of kings.

2.15.2   Example: Random Groups of Students

A class has 68 students, 48 of which are CS majors. The 68 students will be randomly assigned to groups of 4. Find the probability that a random group of 4 has exactly 2 CS majors.

$$\frac{\binom{48}{2}\binom{20}{2}}{\binom{68}{4}}$$

2.15.3   Example: Lottery Tickets

Twenty tickets are sold in a lottery, numbered 1 to 20, inclusive. Five tickets are drawn for prizes. Let's find the probability that two of the five winning tickets are even-numbered.

Since there are 10 even-numbered tickets, there are $\binom{10}{2}$ sets of two such tickets. Continuing along these lines, we find the desired probability to be.

$$\frac{\binom{10}{2}\binom{10}{3}}{\binom{20}{5}} \tag{2.74}$$

Now let's find the probability that two of the five winning tickets are in the range 1 to 5, two are in 6 to 10, and one is in 11 to 20.

Picture yourself picking your tickets. Again there are $\binom{20}{5}$ ways to choose the five tickets. How many of those ways satisfy the stated condition?

Well, first, there are $\binom{5}{2}$ ways to choose two tickets from the range 1 to 5. Once you've done that, there are $\binom{5}{2}$ ways to choose two tickets from the range 6 to 10, and so on. So, The desired probability is then

$$\frac{\binom{5}{2}\binom{5}{2}\binom{10}{1}}{\binom{20}{5}} \tag{2.75}$$

2.15.4 "Association Rules" in Data Mining

The field of *data mining* is a branch of computer science, but it is largely an application of various statistical methods to really huge databases.

One of the applications of data mining is called the *market basket* problem. Here the data consists of records of sales transactions, say of books at Amazon.com. The business' goal is exemplified by Amazon's suggestion to customers that "Patrons who bought this book also tended to buy the following books."[9] The goal of the market basket problem is to sift through sales transaction records to produce *association rules*, patterns in which sales of some combinations of books imply likely sales of other related books.

The notation for association rules is $A, B \Rightarrow C, D, E$, meaning in the book sales example that customers who bought books A and B also tended to buy books C, D and E. Here A and B are called the **antecedents** of the rule, and C, D and E are called the **consequents**. Let's suppose here that we are only interested in rules with a single consequent.

We will present some methods for finding good rules in another chapter, but for now, let's look at how many possible rules there are. Obviously, it would be impractical to use rules with a large number of antecedents.[10] Suppose the business has a total of 20 products available for sale. What percentage of potential rules have three or fewer antecedents?[11]

For each k = 1,...,19, there are $\binom{20}{k}$ possible sets of k antecedents, and for each such set there are

---

[9]Some customers appreciate such tips, while others view it as insulting or an invasion of privacy, but we'll not address such issues here.

[10]In addition, there are serious statistical problems that would arise, to be discussed in another chapter.

[11]Be sure to note that this is also a probability, namely the probability that a randomly chosen rule will have three or fewer antecedents.

$\binom{20-k}{1}$ possible consequents. The fraction of potential rules using three or fewer antecedents is then

$$\frac{\sum_{k=1}^{3} \binom{20}{k} \cdot \binom{20-k}{1}}{\sum_{k=1}^{19} \binom{20}{k} \cdot \binom{20-k}{1}} = \frac{23180}{10485740} = 0.0022 \tag{2.76}$$

So, this is just scratching the surface. And note that with only 20 products, there are already over ten million possible rules. With 50 products, this number is 2.81×10^{16}! Imagine what happens in a case like Amazon, with millions of products. These staggering numbers show what a tremendous challenge data miners face.

2.15.5   Multinomial Coefficients

Question: We have a group consisting of 6 Democrats, 5 Republicans and 2 Independents, who will participate in a panel discussion. They will be sitting at a long table. How many seating arrangements are possible, with regard to political affiliation? (So we do not care, for instance, about permuting the individual Democrats within the seats assigned to Democrats.)

Well, there are $\binom{13}{6}$ ways to choose the Democratic seats. Once those are chosen, there are $\binom{7}{5}$ ways to choose the Republican seats. The Independent seats are then already determined, i.e. there will be only way at that point, but let's write it as $\binom{2}{2}$. Thus the total number of seating arrangements is

$$\frac{13!}{6!7!} \cdot \frac{7!}{5!2!} \cdot \frac{2!}{2!0!} \tag{2.77}$$

That reduces to

$$\frac{13!}{6!5!2!} \tag{2.78}$$

The same reasoning yields the following:

Multinomial Coefficients: Suppose we have c objects and r bins. Then the number of ways to choose c_1 of them to put in bin 1, c_2 of them to put in bin 2,..., and c_r of them to put in bin r is

$$\frac{c!}{c_1!...c_r!}, \quad c_1 + ... + c_r = c \tag{2.79}$$

Of course, the "bins" may just be metaphorical. In the political party example above, the "bins " were political parties, and "objects" were seats.

2.15.6 Example: Probability of Getting Four Aces in a Bridge Hand

A standard deck of 52 cards is dealt to four players, 13 cards each. One of the players is Millie. What is the probability that Millie is dealt all four aces?

Well, there are

$$\frac{52!}{13!13!13!13!} \tag{2.80}$$

possible deals. (the "objects" are the 52 cards, and the "bins" are the 4 players.) The number of deals in which Millie holds all four aces is the same as the number of deals of 48 cards, 9 of which go to Millie and 13 each to the other three players, i.e.

$$\frac{48!}{13!13!13!9!} \tag{2.81}$$

Thus the desired probability is

$$\frac{\frac{48!}{13!13!13!9!}}{\frac{52!}{13!13!13!13!}} = 0.00264 \tag{2.82}$$

2.16 Computational Complements

2.16.1 More on the replicate() Function

The call form of **replicate()** is

```
replicate(numberOfReplications,codeBlock
```

In our example in Section 2.14.4,

```
sums <- replicate(nreps,sum(roll(d)))
```

codeBlock was just a single statement, a call to R's **sum()** function. If more than one statement is to be executed, it must be done so in a *block*, a set of statements enclosed by braces, such as

```
f <- function()
{
   replicate(3,
      {
```

```
        x <- sample(1:10,5,replace=TRUE)
        c(mean(x),sd(x))
     }
  )
}
```

Chapter 3

Discrete Random Variables

This chapter will introduce entities called *discrete random variables*. Some properties will be derived for means of such variables, with most of these properties actually holding for random variables in general. Well, all of that seems abstract to you at this point, so let's get started.

3.1   Random Variables

Definition 3 *A* **random variable** *is a numerical outcome of our experiment.*

For instance, consider our old example in which we roll two dice, with X and Y denoting the number of dots we get on the blue and yellow dice, respectively. Then X and Y are random variables, as they are numerical outcomes of the experiment. Moreover, X+Y, 2XY, sin(XY) and so on are also random variables.

In a more mathematical formulation, with a formal sample space defined, a random variable would be defined to be a real-valued function whose domain is the sample space.

3.2   Discrete Random Variables

In our dice example, the random variable X could take on six values in the set {1,2,3,4,5,6}. We say that the **support** of X is {1,2,3,4,5,6}. This is a finite set.

In the ALOHA example, X_1 and X_2 each have support {0,1,2}, again a finite set.[1]

---

[1] We could even say that X_1 takes on only values in the set {1,2}, but if we were to look at many epochs rather than just two, it would be easier not to make an exceptional case.

Now think of another experiment, in which we toss a coin until we get heads. Let N be the number of tosses needed. Then the support of N is the set {1,2,3,...} This is a countably infinite set.[2]

Now think of one more experiment, in which we throw a dart at the interval (0,1), and assume that the place that is hit, R, can take on any of the values between 0 and 1. Here the support is an uncountably infinite set.

We say that X, X_1, X_2 and N are **discrete** random variables, while R is **continuous**. We'll discuss continuous random variables in a later chapter.

3.3   Independent Random Variables

We already have a definition for the independence of events; what about independence of random variables? Here it is:

> Random variables X and Y are said to be **independent** if for any sets I and J, the events {X is in I} and {Y is in J} are independent, i.e. P(X is in I and Y is in J) = P(X is in I) P(Y is in J).

Sounds innocuous, but the notion of independent random variables is absolutely central to the field of probability and statistics, and will pervade this entire book.

3.4   Example: The Monty Hall Problem

This is an example of how the use of random variables in "translating" a probability problem to mathematical terms can simplify and clarify one's thinking. Imagine, **this simple device of introducing named random variables into our analysis makes a problem that has vexed famous mathematicians quite easy to solve!**

The Monty Hall Problem, which gets its name from a popular TV game show host, involves a contestant choosing one of three doors. Behind one door is a new automobile, while the other two doors lead to goats. The contestant chooses a door and receives the prize behind the door.

The host knows which door leads to the car. To make things interesting, after the contestant chooses, the host will open one of the other doors not chosen, showing that it leads to a goat.

---

[2]This is a concept from the fundamental theory of mathematics. Roughly speaking, it means that the set can be assigned an integer labeling, i.e. item number 1, item number 2 and so on. The set of positive even numbers is countable, as we can say 2 is item number 1, 4 is item number 2 and so on. It can be shown that even the set of all rational numbers is countable.

Should the contestant now change her choice to the remaining door, i.e. the one that she didn't choose and the host didn't open?

Many people answer No, reasoning that the two doors not opened yet each have probability 1/2 of leading to the car. But the correct answer is actually that the remaining door (not chosen by the contestant and not opened by the host) has probability 2/3, and thus the contestant should switch to it. Let's see why.

Let

- C = contestant's choice of door (1, 2 or 3)

- H = host's choice of door (1, 2 or 3)

- A = door that leads to the automobile

We can make things more concrete by considering the case $C = 1$, $H = 2$. The mathematical formulation of the problem is then to find

$$P(A = 3 \mid C = 1, \ H = 2) = \frac{P(A = 3, \ C = 1, \ H = 2)}{P(C = 1, \ H = 2)} \tag{3.1}$$

The key point, commonly missed even by mathematically sophisticated people, is the role of the host. Write the numerator above as

$$P(A = 3, \ C = 1) \ P(H = 2 \mid A = 3, \ C = 1) \tag{3.2}$$

Since C and A are independent random variables, the value of the first factor in (3.2) is

$$\frac{1}{3} \cdot \frac{1}{3} = \frac{1}{9} \tag{3.3}$$

What about the second factor? Remember, the host knows that $A = 3$, and since the contestant has chosen door 1, the host will open the only remaining door that conceals a goat, i.e. door 2. In other words,

$$P(H = 2 \mid A = 3, \ C = 1) = 1 \tag{3.4}$$

On the other hand, if say $A = 1$, the host would randomly choose between doors 2 and 3, so that

$$P(H = 2 \mid A = 1, \ C = 1) = \frac{1}{2} \tag{3.5}$$

It is left to the reader to complete the analysis, calculating the denominator of (3.1), and then showing in the end that

$$P(A = 3 \mid C = 1, \ H = 2) = \frac{2}{3} \tag{3.6}$$

According to the "Monty Hall problem" entry in Wikipedia, even Paul Erdös, one of the most famous mathematicians in history, gave the wrong answer to this problem. Presumably he would have avoided this by writing out his analysis in terms of random variables, as above, rather than say, a wordy, imprecise and ultimately wrong solution.

3.5   Expected Value

3.5.1   Generality—Not Just for <u>Discrete</u> Random Variables

The concepts and properties introduced in this section form the very core of probability and statistics. **Except for some specific calculations, these apply to both discrete and continuous random variables.**

The properties developed for *variance*, defined later in this chapter, also hold for both discrete and continuous random variables.

3.5.1.1   Misnomer

The term "expected value" is one of the many misnomers one encounters in tech circles. The expected value is actually not something we "expect" to occur. On the contrary, it's often pretty unlikely.

For instance, let H denote the number of heads we get in tossing a coin 1000 times. The expected value, you'll see later, is 500. This is not surprising, given the symmetry of the situation, but P(H = 500) turns out to be about 0.025. In other words, we certainly should not "expect" H to be 500.

Of course, even worse is the example of the number of dots that come up when we roll a fair die. The expected value is 3.5, a value which not only rarely comes up, but in fact never does.

In spite of being misnamed, expected value plays an absolutely central role in probability and statistics.

3.5.2   Definition

Consider a repeatable experiment with random variable X. We say that the **expected value** of X is the long-run average value of X, as we repeat the experiment indefinitely.

In our notebook, there will be a column for X. Let X_i denote the value of X in the ith row of the notebook. Then the long-run average of X, i.e. the long-run average in the X column of the notebook, is

$$\lim_{n \to \infty} \frac{X_1 + ... + X_n}{n} \tag{3.7}$$

To make this more explicit, look at the partial notebook example in Table 3.1. Here we roll two dice, and let S denote their sum. E(S) is then the long-run average of the values in the "S" column.

3.5.3   Existence of the Expected Value

The above defintion puts the cart before the horse, as it presumes that the limit exists. Theoretically speaking, this might not be the case. However, it does exist if the X_i have finite lower and upper bounds, which is always true in the real world. For instance, no person has height of 50 feet, say, and no one has negative height either.

For the remainder of this book, we will usually speak of "the" expected value of a random variable without adding the qualifier "if it exists."

3.5.4   Computation and Properties of Expected Value

Here we will derive a handy computational formula for the expected value of a discrete random variable.

Suppose for instance our experiment is to toss 10 coins. Let X denote the number of heads we get out of 10. We might get four heads in the first repetition of the experiment, i.e. $X_1 = 4$, seven heads in the second repetition, so $X_2 = 7$, and so on. Intuitively, the long-run average value of X will be 5. (This will be proven below.) Thus we say that the expected value of X is 5, and write E(X) = 5.

Now let K_{in} be the number of times the value i occurs among $X_1, ..., X_n$, i = 0,...,10, n = 1,2,3,... For instance, $K_{4,20}$ is the number of times we get four heads, in the first 20 repetitions of our experiment. Then

$$E(X) \quad = \quad \lim_{n \to \infty} \frac{X_1 + ... + X_n}{n} \tag{3.8}$$

$$= \quad \lim_{n \to \infty} \frac{0 \cdot K_{0n} + 1 \cdot K_{1n} + 2 \cdot K_{2n}... + 10 \cdot K_{10,n}}{n} \tag{3.9}$$

$$= \quad \sum_{i=0}^{10} i \cdot \lim_{n \to \infty} \frac{K_{in}}{n} \tag{3.10}$$

To understand that second equation, suppose when n = 5 we have 2, 3, 1, 2 and 1 for our values of X_1, X_2, X_3, X_4, X_5,. Then we can group the 2s together and group the 1s together, and write

$$2 + 3 + 1 + 2 + 1 = 2 \times 2 + 2 \times 1 + 1 \times 3 \tag{3.11}$$

But $\lim_{n \to \infty} \frac{K_{in}}{n}$ is the long-run fraction of the time that X = i. In other words, it's P(X = i)! So,

$$E(X) = \sum_{i=0}^{10} i \cdot P(X = i) \tag{3.12}$$

So in general we have:

Property A:

The expected value of a discrete random variable X which takes values in the set A is

$$E(X) = \sum_{c \in A} c \, P(X = c) \tag{3.13}$$

Note that (3.13) is the formula we'll use. The preceding equations were derivation, to motivate the formula. Note too that (3.13) is not the *definition* of expected value; that was in (3.7). It is quite important to distinguish between all of these, in terms of goals.[3]

So, we see that E(X) is a weighted average of the values in the support of X, with the weights being the probabilities of those values. You might wonder how to reconcile this with the fact that (3.7) is an *unweighted* average. But in fact it *is* weighted, because some values of X will appear more often than others in the X column of the notebook; indeed, the relative frequencies of those values will be given by P(X = c).

---

[3]The matter is made a little more confusing by the fact that many books do in fact treat (3.13) as the definition, with (3.7) being the consequence.

By the way, note the word *discrete* above. For the case of continuous random variables, the sum in (3.13) will become an integral.

It will be shown in Section 4.3 that in our example above in which X is the number of heads we get in 10 tosses of a coin,

$$P(X = i) = \binom{10}{i} 0.5^i (1 - 0.5)^{10-i} \tag{3.14}$$

So

$$E(X) = \sum_{i=0}^{10} i \binom{10}{i} 0.5^i (1 - 0.5)^{10-i} \tag{3.15}$$

It turns out that E(X) = 5.

For X in our dice example,

$$E(X) = \sum_{c=1}^{6} c \cdot \frac{1}{6} = 3.5 \tag{3.16}$$

It is customary to use capital letters for random variables, e.g. X here, and lower-case letters for values taken on by a random variable, e.g. c here. Please adhere to this convention.

By the way, it is also customary to write EX instead of E(X), whenever removal of the parentheses does not cause any ambiguity. An example in which it would produce ambiguity is $E(U^2)$. The expression EU^2 might be taken to mean either $E(U^2)$, which is what we want, or $(EU)^2$, which is not what we want.

For S = X+Y in the dice example,

$$E(S) = 2 \cdot \frac{1}{36} + 3 \cdot \frac{2}{36} + 4 \cdot \frac{3}{36} + ...12 \cdot \frac{1}{36} = 7 \tag{3.17}$$

In the case of N, tossing a coin until we get a head:

$$E(N) = \sum_{c=1}^{\infty} c \cdot \frac{1}{2^c} = 2 \tag{3.18}$$

(We will not go into the details here concerning how the sum of this particular infinite series is computed. See Section 4.2.)

notebook line	outcome	blue+yellow = 6?	S
1	blue 2, yellow 6	No	8
2	blue 3, yellow 1	No	4
3	blue 1, yellow 1	No	2
4	blue 4, yellow 2	Yes	6
5	blue 1, yellow 1	No	2
6	blue 3, yellow 4	No	7
7	blue 5, yellow 1	Yes	6
8	blue 3, yellow 6	No	9
9	blue 2, yellow 5	No	7

Table 3.1: Expanded Notebook for the Dice Problem

Some people like to think of E(X) using a center of gravity analogy. Forget that analogy! Think notebook! **Intuitively, E(X) is the long-run average value of X among all the lines of the notebook.** So for instance in our dice example, E(X) = 3.5, where X was the number of dots on the blue die, means that if we do the experiment thousands of times, with thousands of lines in our notebook, the average value of X in those lines will be about 3.5. With S = X+Y, E(S) = 7. This means that in the long-run average in column S in Table 3.1 is 7.

Of course, by symmetry, E(Y) will be 3.5 too, where Y is the number of dots showing on the yellow die. That means we wasted our time calculating in Equation (3.17); we should have realized beforehand that E(S) is $2 \times 3.5 = 7$.

In other words:

Property B:

For any random variables U and V, the expected value of a new random variable D = U+V is the sum of the expected values of U and V:

$$E(U + V) = E(U) + E(V) \tag{3.19}$$

Note carefully that U and V do NOT need to be independent random variables for this relation to hold. You should convince yourself of this fact intuitively **by thinking about the notebook notion.** Say we look at 10000 lines of the notebook, which has columns for the values of U, V and U+V. It makes no difference whether we average U+V in that column, or average U and V in their columns and then add—either way, we'll get the same result.

While you are at it, use the notebook notion to convince yourself of the following:

Properties C:

- For any random variable U and constant a, then

$$E(aU) = aEU \tag{3.20}$$

- For random variables X and Y—not necessarily independent—and constants a and b, we have

$$E(aX + bY) = aEX + bEY \tag{3.21}$$

  This follows by taking U = aX and V = bY in (3.19), and then using (3.20).

  By induction, for constants $a_1, ..., a_k$ and random variables $X_1, ..., X_k$, form the new random variable $a_1 X_1 + ... + a_k X_k$. Then

$$E(a_1 X_1 + ... + a_n X_k) = a_1 EX_1 + ... + a_n EX_k \tag{3.22}$$

- For any constant b, we have

$$E(b) = b \tag{3.23}$$

  This should make sense. If the "random" variable X has the constant value 3, say, then the "X" column in the notebook will consist entirely of 3s. Thus the long-run average value in that column will be 3, so EX = 3.

For instance, say U is temperature in Celsius. Then the temperature in Fahrenheit is $W = \frac{9}{5}U + 32$. So, W is a new random variable, and we can get its expected value from that of U by using (3.21) with $a = \frac{9}{5}$ and b = 32.

If you combine (3.23) with (3.21), we have an important special case:

$$E(aX + b) = aEX + b \tag{3.24}$$

Another important point:

Property D: If U and V *are* independent, then

$$E(UV) = EU \cdot EV \tag{3.25}$$

In the dice example, for instance, let D denote the product of the numbers of blue dots and yellow dots, i.e. D = XY. Then

$$E(D) = 3.5^2 = 12.25 \tag{3.26}$$

Equation (3.25) doesn't have an easy "notebook proof." It is proved in Section 13.3.1.

Consider a function $g()$ of one variable, and let $W = g(X)$. W is then a random variable too. Say X has support A, as in (3.13). Then W has support $B = \{g(c) : c \; \epsilon \; A\}$. (

For instance, say $g()$ is the squaring function, and X takes on the values -1, 0 and 1, with probability 0.5, 0.4 and 0.1. Then

$$A = \{-1, 0, 1\} \tag{3.27}$$

and

$$B = \{0, 1\} \tag{3.28}$$

Define

$$A_d = \{c : c \in A, g(c) = d\} \tag{3.29}$$

In our above squaring example, we will have

$$A_0 = \{0\}, \quad A_1 = \{-1, 1\} \tag{3.30}$$

Then

$$P(W = d) = P(X \in A_d) \tag{3.31}$$

so

$$\begin{aligned}
E[g(X)] &= E(W) & (3.32) \\
&= \sum_{d \in B} dP(W = d) & (3.33) \\
&= \sum_{d \in B} d \sum_{c \in A_d} P(X = c) & (3.34) \\
&= \sum_{c \in A} g(c)P(X = c) & (3.35)
\end{aligned}$$

(Going from the next-to-last equation here to the last one is rather tricky. Work through for the case of our squaring function example above in order to see why the final equation does follow.

Property E:

If E[g(X)] exists, then

$$E[g(X)] = \sum_{c \in A} g(c) \cdot P(X = c) \tag{3.36}$$

where the sum ranges over all values c that can be taken on by X.

For example, suppose for some odd reason we are interested in finding $E(\sqrt{X})$, where **X** is the number of dots we get when we roll one die. Let $W = \sqrt{X}$). Then **W** is another random variable, and is discrete, since it takes on only a finite number of values. (The fact that most of the values are not integers is irrelevant.) We want to find EW.

Well, W is a function of X, with $g(t) = \sqrt{t}$. So, (3.36) tells us to make a list of values in the support of W, i.e. $\sqrt{1}, \sqrt{2}, ..., \sqrt{6}$, and a list of the corresponding probabilities for **X**, which are all $\frac{1}{6}$. Substituting into (3.36), we find that

$$E(\sqrt{X}) = \sum_{i=1}^{6} \sqrt{i} \cdot \frac{1}{6} \tag{3.37}$$

What about a function of several variables? Say for instance you are finding E(UV), where U has support, say, 1,2 and V has support 5,12,13. In order to find E(UV), you need to know the support of UV, recognizing that it, the product UV, is a new random variable in its own right. Let's call it W. Then in this little example, W has support 5,12,13,10,24,26. Then compute

$$5P(W = 5) + 12P(W = 12) + ... = 5P(U = 1, V = 5) + 12P(U = 1, V = 12) + ...$$

Note: Equation (3.36) will be one of the most heavily used formulas in this book. Make sure you keep it in mind.

3.5.5 "Mailing Tubes"

The properties of expected value discussed above are key to the entire remainder of this book. You should notice immediately when you are in a setting in which they are applicable. For instance, if you see the expected value of the sum of two random variables, you should instinctively think of (3.19) right away.

As discussed in Section 2.4, these properties are "mailing tubes." For instance, (3.19) is a "mailing tube"–make a mental note to yourself saying, "If I ever need to find the expected value of the sum of two random variables, I can use (3.19)." Similarly, (3.36) is a mailing tube; tell yourself, "If I ever see a new random variable that is a function of one whose probabilities I already know, I can find the expected value of the new random variable using (3.36)."

You will encounter "mailing tubes" throughout this book. For instance, (3.49) below is a very important "mailing tube." Constantly remind yourself—"Remember the 'mailing tubes'!"

3.5.6 Casinos, Insurance Companies and "Sum Users," Compared to Others

The expected value is intended as a **measure of central tendency** (also called a **measure of location**, i.e. as some sort of definition of the probablistic "middle" in the range of a random variable. There are various other such measures one can use, such as the **median**, the halfway point of a distribution, and today they are recognized as being superior to the mean in certain senses. For historical reasons, the mean plays an absolutely central role in probability and statistics. Yet one should understand its limitations. (This discussion will be general, not limited to discrete random variables.)

(**Warning:** The concept of the mean is likely so ingrained in your consciousness that you simply take it for granted that you know what the mean means, no pun intended. But try to take a step back, and think of the mean afresh in what follows.)

First, the term *expected value* itself is a misnomer. We do not e̲x̲p̲e̲c̲t̲ the number of dots D to be 3.5 in the die example in Section 3.5.1.1; in fact, it is impossible for W to take on that value.

Second, the expected value is what we call the **mean** in everyday life. And the mean is terribly overused. Consider, for example, an attempt to describe how wealthy (or not) people are in the city of Davis. If suddenly Bill Gates were to move into town, that would skew the value of the mean beyond recognition.

But even without Gates, there is a question as to whether the mean has that much meaning. After

all, what is so meaningful about summing our data and dividing by the number of data points? The median has an easy intuitive meaning, but although the mean has familiarity, one would be hard pressed to justify it as a measure of central tendency.

What, for example, does Equation (3.7) mean in the context of people's heights in Davis? We would sample a person at random and record his/her height as X_1. Then we'd sample another person, to get X_2, and so on. Fine, but in that context, what would (3.7) mean? The answer is, not much. So the significance of the mean height of people in Davis would be hard to explain.

For a casino, though, (3.7) means plenty. Say X is the amount a gambler wins on a play of a roulette wheel, and suppose (3.7) is equal to \$1.88. Then after, say, 1000 plays of the wheel (not necessarily by the same gambler), the casino knows from 3.7 it will have paid out a total of about \$1,880. So if the casino charges, say \$1.95 per play, it will have made a profit of about \$70 over those 1000 plays. It might be a bit more or less than that amount, but the casino can be pretty sure that it will be around \$70, and they can plan their business accordingly.

The same principle holds for insurance companies, concerning how much they pay out in claims. With a large number of customers, they know ("expect"!) approximately how much they will pay out, and thus can set their premiums accordingly. Here the mean has a tangible, practical meaning.

The key point in the casino and insurance companies examples is that they are interested in *totals*, such as *total* payouts on a blackjack table over a month's time, or *total* insurance claims paid in a year. Another example might be the number of defectives in a batch of computer chips; the manufacturer is interested in the *total* number of defectives chips produced, say in a month. Since the mean is by definition a *total* (divided by the number of data points), the mean will be of direct interest to casinos etc.

By contrast, in describing how wealthy people of a town are, the total income of all the residents is not relevant. Similarly, in describing how well students did on an exam, the sum of the scores of all the students doesn't tell us much. (Unless the professor gets \$10 for each point in the exam scores of each of the students!) A better description for heights and exam scores might be the median height or score.

Nevertheless, the mean has certain mathematical properties, such as (3.19), that have allowed the rich development of the fields of probability and statistics over the years. The median, by contrast, does not have nice mathematical properties. In many cases, the mean won't be too different from the median anyway (barring Bill Gates moving into town), so you might think of the mean as a convenient substitute for the median. The mean has become entrenched in statistics, and we will use it often.

3.6   Variance

As in Section 3.5, the concepts and properties introduced in this section form the very core of probability and statistics. **Except for some specific calculations, these apply to both discrete and continuous random variables.**

3.6.1   Definition

While the expected value tells us the average value a random variable takes on, we also need a measure of the random variable's variability—how much does it wander from one line of the notebook to another? In other words, we want a measure of **dispersion**. The classical measure is **variance**, defined to be the mean squared difference between a random variable and its mean:

Definition 4 *For a random variable U for which the expected values written below exist, the* **variance** *of U is defined to be*

$$Var(U) = E[(U - EU)^2] \tag{3.38}$$

For X in the die example, this would be

$$Var(X) = E[(X - 3.5)^2] \tag{3.39}$$

Remember what this means: We have a random variable **X**, and we're creating a new random variable, $W = (X - 3.5)^2$, which is a function of the old one. We are then finding the expected value of that new random variable W.

In the notebook view, $E[(X - 3.5)^2]$ is the long-run average of the W column:

line	X	W
1	2	2.25
2	5	2.25
3	6	6.25
4	3	0.25
5	5	2.25
6	1	6.25

To evaluate this, apply (3.36) with $g(c) = (c - 3.5)^2$:

$$Var(X) = \sum_{c=1}^{6}(c - 3.5)^2 \cdot \frac{1}{6} = 2.92 \qquad (3.40)$$

You can see that variance does indeed give us a measure of dispersion. In the expression $Var(U) = E[(U - EU)^2]$, if the values of U are mostly clustered near its mean, then $(U - EU)^2$ will usually be small, and thus the variance of U will be small; if there is wide variation in U, the variance will be large.

Property F:

$$Var(U) = E(U^2) - (EU)^2 \qquad (3.41)$$

The term $E(U^2)$ is again evaluated using (3.36).

Thus for example, if X is the number of dots which come up when we roll a die. Then, from (3.41),

$$Var(X) = E(X^2) - (EX)^2 \qquad (3.42)$$

Let's find that first term (we already know the second is 3.5^2). From (3.36),

$$E(X^2) = \sum_{i=1}^{6} i^2 \cdot \frac{1}{6} = \frac{91}{6} \qquad (3.43)$$

Thus $Var(X) = E(X^2) - (EX)^2 = \frac{91}{6} - 3.5^2$

Remember, though, that (3.41) is a shortcut formula for finding the variance, not the *definition* of variance.

Below is the derivation of (3.41). Keep in mind that EU is a constant.

$$
\begin{aligned}
Var(U) &= E[(U - EU)^2] & (3.44) \\
&= E[U^2 - 2EU \cdot U + (EU)^2] \text{ (algebra)} & (3.45) \\
&= E(U^2) + E(-2EU \cdot U) + E[(EU)^2] \text{ (3.19)} & (3.46) \\
&= E(U^2) - 2EU \cdot EU + (EU)^2 \text{ (3.20), (3.23)} & (3.47) \\
&= E(U^2) - (EU)^2 & (3.48)
\end{aligned}
$$

An important behavior of variance is:

Property G:

$$Var(cU) = c^2 Var(U) \tag{3.49}$$

for any random variable U and constant c. It should make sense to you: If we multiply a random variable by 5, say, then its average squared distance to its mean should increase by a factor of 25.

Let's prove (3.49). Define V = cU. Then

$$
\begin{aligned}
Var(V) \ &= \ E[(V - EV)^2] \text{ (def.)} & (3.50) \\
&= \ E\{[cU - E(cU)]^2\} \text{ (subst.)} & (3.51) \\
&= \ E\{[cU - cEU]^2\} \text{ ((3.21))} & (3.52) \\
&= \ E\{c^2[U - EU]^2\} \text{ (algebra)} & (3.53) \\
&= \ c^2 E\{[U - EU]^2\} \text{ ((3.21))} & (3.54) \\
&= \ c^2 Var(U) \text{ (def.)} & (3.55)
\end{aligned}
$$

Shifting data over by a constant does not change the amount of variation in them:

Property H:

$$Var(U + d) = Var(U) \tag{3.56}$$

for any constant d.

Intuitively, the variance of a constant is 0—after all, it never varies! You can show this formally using (3.41):

$$Var(c) = E(c^2) - [E(c)]^2 = c^2 - c^2 = 0 \tag{3.57}$$

The square root of the variance is called the **standard deviation**.

Again, we use variance as our main measure of dispersion for historical and mathematical reasons, not because it's the most meaningful measure. The squaring in the definition of variance produces some distortion, by exaggerating the importance of the larger differences. It would be more natural to use the **mean absolute deviation** (MAD), $E(|U - EU|)$. However, this is less tractable mathematically, so the statistical pioneers chose to use the mean squared difference, which lends

itself to lots of powerful and beautiful math, in which the Pythagorean Theorem pops up in abstract vector spaces. (See Section 19.10.2 for details.)

As with expected values, the properties of variance discussed above, and also in Section 12.1.1 below, are key to the entire remainder of this book. You should notice immediately when you are in a setting in which they are applicable. For instance, if you see the variance of the sum of two random variables, you should instinctively think of (3.75) right away, and check whether they are independent.

3.6.2 More Practice with the Properties of Variance

Suppose X and Y are independent random variables, with $EX = 1$, $EY = 2$, $Var(X) = 3$ and $Var(Y) = 4$. Let's find $Var(XY)$. (The reader should make sure to supply the reasons for each step, citing equation numbers from the material above.)

$$
\begin{aligned}
Var(XY) &= E(X^2 Y^2) - [E(XY)]^2 & (3.58) \\
&= E(X^2) \cdot E(Y^2) - (EX \cdot EY)^2 & (3.59) \\
&= [Var(X) + (EX)^2] \cdot [Var(Y) + (EY)^2] - (EX \cdot EY)^2 & (3.60) \\
&= (3 + 1^2)(4 + 2^2) - (1 \cdot 2)^2 & (3.61) \\
&= 28 & (3.62)
\end{aligned}
$$

3.6.3 Central Importance of the Concept of Variance

No one needs to be convinced that the mean is a fundamental descriptor of the nature of a random variable. But the variance is of central importance too, and will be used constantly throughout the remainder of this book.

The next section gives a quantitative look at our notion of variance as a measure of dispersion.

3.6.4 Intuition Regarding the Size of Var(X)

A billion here, a billion there, pretty soon, you're talking real money—attributed to the late Senator Everitt Dirksen, replying to a statement that some federal budget item cost "only" a billion dollars

Recall that the variance of a random variable X is supposed to be a measure of the dispersion of X, meaning the amount that X varies from one instance (one line in our notebook) to the next. But if Var(X) is, say, 2.5, is that a lot of variability or not? We will pursue this question here.

3.6.4.1 Chebychev's Inequality

This inequality states that for a random variable X with mean μ and variance σ^2,

$$P(|X - \mu| \geq c\sigma) \leq \frac{1}{c^2} \tag{3.63}$$

In other words, X strays more than, say, 3 standard deviations from its mean at most only 1/9 of the time. This gives some concrete meaning to the concept of variance/standard deviation.

You've probably had exams in which the instructor says something like "An A grade is 1.5 standard deviations above the mean." Here c in (3.63) would be 1.5.

We'll prove the inequality in Section 3.14.

3.6.4.2 The Coefficient of Variation

Continuing our discussion of the magnitude of a variance, look at our remark following (3.63):

> In other words, X does not often stray more than, say, 3 standard deviations from its mean. This gives some concrete meaning to the concept of variance/standard deviation.

Or, think of the price of, say, widgets. If the price hovers around a $1 million, but the variation around that figure is only about a dollar, you'd say there is essentially no variation. But a variation of about a dollar in the price of a hamburger would be a lot.

These considerations suggest that any discussion of the size of Var(X) should relate to the size of E(X). Accordingly, one often looks at the **coefficient of variation**, defined to be the ratio of the standard deviation to the mean:

$$\text{coef. of var.} = \frac{\sqrt{Var(X)}}{EX} \tag{3.64}$$

This is a scale-free measure (e.g. inches divided by inches), and serves as a good way to judge whether a variance is large or not.

3.7   A Useful Fact

For a random variable X, consider the function

$$g(c) = E[(X - c)^2] \tag{3.65}$$

Remember, the quantity $E[(X - c)^2]$ is a number, so g(c) really is a function, mapping a real number c to some real output.

We can ask the question, What value of c minimizes g(c)? To answer that question, write:

$$g(c) = E[(X - c)^2] = E(X^2 - 2cX + c^2) = E(X^2) - 2cEX + c^2 \tag{3.66}$$

where we have used the various properties of expected value derived in recent sections.

To make this concrete, suppose we are guessing people's weights—without seeing them and without knowing anything about them at all. (This is a somewhat artificial question, but it will become highly practical in Chapter **??**.) Since we know nothing at all about these people, we will make the same guess for each of them.

What should that guess-in-common be? Your first inclination would be to guess everyone to be the mean weight of the population. If that value in our target population is, say, 142.8 pounds, then we'll guess everyone to be that weight. Actually, that guess turns out to be optimal in a certain sense, as follows.

Say X is a person's weight. It's a random variable, because these people are showing up at random from the population. Then $X - c$ is our prediction error. How well will do in our predictions? We can't measure that as

$$E(\text{error}) \tag{3.67}$$

because that quantity is 0! (What mailing tube is at work here?)

A reasonable measure would be

$$E(|X - c|) \tag{3.68}$$

However, due to tradition, we use

$$E[(X - c)^2] \tag{3.69}$$

Now differentiate with respect to c, and set the result to 0. Remembering that $E(X^2)$ and EX are constants, we have

$$0 = -2EX + 2c \tag{3.70}$$

so the minimizing c is c = EX!

In other words, the minimum value of $E[(X - c)^2]$ occurs at c = EX. Our intuition was right!

Moreover: Plugging c = EX into (3.66) shows that the minimum value of g(c) is $E(X - EX)^2]$, which is Var(X)!

In notebook terms, think of guessing many, many people, meaning many lines in the notebook, one per person. Then (3.69) is the long-run average squared error in our guesses, and we find that we minimize that by guessing everyone's weight to be the population mean weight.

But why look at average squared error? It accentuates the large errors. Instead, we could minimize (3.68). It turns out that the best *c* here is the population *median* weight.

3.8   Covariance

This is a topic we'll cover fully in Chapter 12, but at least introduce here.

A measure of the degree to which U and V vary together is their **covariance**,

$$Cov(U, V) = E[(U - EU)(V - EV)] \tag{3.71}$$

Except for a divisor, this is essentially **correlation**. If U is usually large (relative to its expectation) at the same time V is small (relative to its expectation), for instance, then you can see that the covariance between them will be negative. On the other hand, if they are usually large together or small together, the covariance will be positive.

For example, suppose U and V are the height and weight, respectively, of a person chosen at random from some population, and think in notebook terms. Each line shows the data for one person, and we'll have columns for U, V, U - EU, V - EV and (U - EU) (V - EV). Then (3.71) is the long-run average of that last column. Will it be positive or negative? Reason as follows:

Think of the lines in the notebook for people who are taller than average, i.e. for whom $U - EU > 0$. Most such people are also heavier than average, i.e. $V - EV > 0$, so that $(U - EU)(V - EV) > 0$. On the other hand, shorter people also tend to be lighter, so most lines with shorter people will have $U - EU < 0$ and $V - EV < 0$—but still $(U - EU)(V - EV) > 0$. In other words, the long-run average of the (U - EU) (V - EV) column will be positive.

The point is that, if two variables are positively related, e.g. height and weight, their covariance should be positive. This is the intuitive underlying defining covariance as in (3.71).

Again, one can use the properties of E() to show that

$$Cov(U, V) = E(UV) - EU \cdot EV \tag{3.72}$$

Again, this will be derived fully in Chapter **??**, but you think about how to derive it yourself. Just use our old mailing tubes, e.g. E(X+Y) = EX + EY, E(cX) for a constant c, etc. Note that EU and EV are constants!

Also

$$Var(U + V) = Var(U) + Var(V) + 2Cov(U, V) \tag{3.73}$$

and more generally,

$$Var(aU + bV) = a^2 Var(U) + b^2 Var(V) + 2ab Cov(U, V) \tag{3.74}$$

for any constants a and b.

(3.72) imply that Cov(U,V) = 0. In that case,

$$Var(U + V) = Var(U) + Var(V) \tag{3.75}$$

By the way, (3.75) is actually the Pythagorean Theorem in a certain esoteric, infinite-dimesional vector space (related to a similar remark made earlier). This is pursued in Section 19.10.2 for the mathematically inclined.

Generalizing (3.74), for constants $a_1, ..., a_k$ and random variables $X_1, ..., X_k$, form the new random variable $a_1 X_1 + ... + a_k X_k$. Then

$$Var(a_1 X_1 + ... + a_k X_k) = \sum_{i=1}^{k} a_i^2 Var(X_i) + 2 \sum_{1 \leq i < j \leq k} a_i a_j Cov(X_i, X_j) \tag{3.76}$$

If the X_i are independent, then we have the special case

$$Var(a_1 X_1 + ... + a_k X_k) = \sum_{i=1}^{k} a_i^2 Var(X_i) \tag{3.77}$$

3.9   Indicator Random Variables, and Their Means and Variances

Definition 5 *A random variable that has the value 1 or 0, according to whether a specified event occurs or not is called an* **indicator random variable** *for that event.*

You'll often see later in this book that the notion of an indicator random variable is a very handy device in certain derivations. But for now, let's establish its properties in terms of mean and variance.

   Handy facts: Suppose X is an indicator random variable for the event A. Let p denote P(A). Then

$$E(X) = p \tag{3.78}$$

$$Var(X) = p(1 - p) \tag{3.79}$$

These two facts are easily derived. In the first case we have, using our properties for expected value,

$$EX = 1 \cdot P(X = 1) + 0 \cdot P(X = 0) = P(X = 1) = P(A) = p \tag{3.80}$$

The derivation for Var(X) is similar (use (3.41)).

For example, say Coin A has probability 0.6 of heads, Coin B is fair, and Coin C has probability 0.2 of heads. I toss A once, getting X heads, then toss B once, getting Y heads, then toss C once, getting Z heads. Let W = X + Y + Z, i.e. the total number of heads from the three tosses (W ranges from 0 to 3). Let's find P(W = 1) and Var(W).

The first one uses old methods:

$$
\begin{aligned}
P(W = 1) &= P(X = 1 \text{ and } Y = 0 \text{ and } Z = 0 \text{ or } ...) & (3.81)\\
&= 0.6 \cdot 0.5 \cdot 0.8 + 0.4 \cdot 0.5 \cdot 0.8 + 0.4 \cdot 0.5 \cdot 0.2 & (3.82)
\end{aligned}
$$

For Var(W), let's use what we just learned about indicator random variables; each of X, Y and Z are such variables. Var(W) = Var(X) + Var(Y) + Var(Z), by independence and (3.75). Since X is an indicator random variable, $Var(X) = 0.6 \cdot 0.4$, etc. The answer is then

$$0.6 \cdot 0.4 + 0.5 \cdot 0.5 + 0.2 \cdot 0.8 \tag{3.83}$$

3.9.1 Example: Return Time for Library Books, Version I

Suppose at some public library, patrons return books exactly 7 days after borrowing them, never early or late. However, they are allowed to return their books to another branch, rather than the branch where they borrowed their books. In that situation, it takes 9 days for a book to return to its proper library, as opposed to the normal 7. Suppose 50% of patrons return their books to a "foreign" library. Find Var(T), where T is the time, either 7 or 9 days, for a book to come back to its proper location.

Note that

$$T = 7 + 2I, \tag{3.84}$$

where I is an indicator random variable for the event that the book is returned to a "foreign" branch. Then

$$Var(T) = Var(7 + 2I) = 4Var(I) = 4 \cdot 0.5(1 - 0.5) \tag{3.85}$$

3.9.2 Example: Return Time for Library Books, Version II

Now let's look at a somewhat more general model. Here we will assume that borrowers return books after 4, 5, 6 or 7 days, with probabilities 0.1, 0.2, 0.3, 0.4, respectively. As before, 50% of patrons return their books to a "foreign" branch, resulting in an extra 2-day delay before the book arrives back to its proper location. The library is open 7 days a week.

Suppose you wish to borrow a certain book, and inquire at the library near the close of business on Monday. Assume too that no one else is waiting for the book. You are told that it had been checked out the previous Thursday. Find the probability that you will need to wait until Wednesday evening to get the book. (You check every evening.)

Let B denote the time needed for the book to arrive back at its home branch, and define I as before. Then

$$P(B = 6 \mid B > 4) \quad = \quad \frac{P(B = 6 \text{ and } B > 4)}{P(B > 4)} \tag{3.86}$$

$$= \quad \frac{P(B = 6)}{P(B > 4)} \tag{3.87}$$

$$= \quad \frac{P(B = 6 \text{ and } I = 0 \ \text{ or } \ B = 6 \ \text{ and } \ I = 1)}{1 - P(B = 4)} \tag{3.88}$$

$$= \quad \frac{0.5 \cdot 0.3 + 0.5 \cdot 0.1}{1 - 0.5 \cdot 0.1} \tag{3.89}$$

$$= \quad \frac{4}{19} \tag{3.90}$$

Here is a simulation check:

```
libsim <- function(nreps) {
   # patron return time
   prt <- sample(c(4,5,6,7),nreps,replace=T,prob=c(0.1,0.2,0.3,0.4))
   # indicator for foreign branch
   i <- sample(c(0,1),nreps,replace=T)
   b <- prt + 2*i
   x <- cbind(prt,i,b)
   # look only at the relevant notebook lines
   bgt4 <- x[b > 4,]
   # among those lines, what proportion have B = 6?
   mean(bgt4[,3] == 6)
}
```

Note that in this simulation. all **nreps** values of I, B and the patron return time are generated first. This uses more memory space (though not an issue in this small problem), but makes things easier to code, as we can exploit R's vector operations. Those not only are more convenient, but also faster running.

3.9.3   Example: Indicator Variables in a Committee Problem

A committee of four people is drawn at random from a set of six men and three women. Suppose we are concerned that there may be quite a gender imbalance in the membership of the committee. Toward that end, let M and W denote the numbers of men and women in our committee, and let D = M-W. Let's find E(D), in two different ways.

D has support consisting of the values 4-0, 3-1, 2-2 and 1-3, i.e. 4, 2, 0 and -2. So from (3.13)

$$ED = -2 \cdot P(D = -2) + 0 \cdot P(D = 0) + 2 \cdot P(D = 2) + 4 \cdot P(D = 4) \tag{3.91}$$

Now, using reasoning along the lines in Section 2.15, we have

$$P(D = -2) = P(M = 1 \text{ and } W = 3) = \frac{\binom{6}{1}\binom{3}{3}}{\binom{9}{4}} \tag{3.92}$$

After similar calculations for the other probabilities in (3.91), we find the $ED = \frac{4}{3}$.

Note what this means: If we were to perform this experiment many times, i.e. choose committees again and again, on average we would have a little more than one more man than women on the committee.

Now let's use our "mailing tubes" to derive ED a different way:

$$
\begin{aligned}
ED &= E(M - W) & (3.93)\\
&= E[M - (4 - M)] & (3.94)\\
&= E(2M - 4) & (3.95)\\
&= 2EM - 4 \ \ (\text{from } (3.21)) & (3.96)
\end{aligned}
$$

Now, let's find EM by using indicator random variables. Let G_i denote the indicator random variable for the event that the ith person we pick is male, i = 1,2,3,4. Then

$$M = G_1 + G_2 + G_3 + G_4 \tag{3.97}$$

so

$$
\begin{aligned}
EM &= E(G_1 + G_2 + G_3 + G_4) & (3.98)\\
&= EG_1 + EG_2 + EG_3 + EG_4 \ \ [\text{ from } (3.19)] & (3.99)\\
&= P(G_1 = 1) + P(G_2 = 1) + P(G_3 = 1) + P(G_4 = 1) \ \ [\text{ from } (3.78)] & (3.100)
\end{aligned}
$$

Note carefully that the second equality here, which uses (3.19), is true in spite of the fact that the G_i are not independent. Equation (3.19) does not require independence.

Another key point is that, due to symmetry, $P(G_i = 1)$ is the same for all i. Note that we did not write a *conditional* probability here! Once again, think of the notebook view: **By definition**, $(P(G_2 = 1)$ is the long-run proportion of the number of notebook lines in which $G_2 = 1$—regardless of the value of G_1 in those lines.

Now, to see that $P(G_i = 1)$ is the same for all i, suppose the six men that are available for the committee are named Alex, Bo, Carlo, David, Eduardo and Frank. When we select our first person, any of these men has the same chance of being chosen (1/9). *But that is also true for the second pick.* Think of a notebook, with a column named "second pick." In some lines, that column will say Alex, in some it will say Bo, and so on, and in some lines there will be women's names. But in that column, Bo will appear the same fraction of the time as Alex, due to symmetry, and that will be the same fraction as for, say, Alice, again 1/9.

Now,

$$P(G_1 = 1) = \frac{6}{9} = \frac{2}{3} \tag{3.101}$$

Thus

$$ED = 2 \cdot (4 \cdot \frac{2}{3}) - 4 = \frac{4}{3} \tag{3.102}$$

3.9.4   Example: Spinner Game

In a certain game, Person A spins a spinner and wins S dollars, with mean 10 and variance 5. Person B flips a coin. If it comes up heads, Person A must give B whatever A won, but if it comes up tails, B wins nothing. Let T denote the amount B wins. Let's find $Var(T)$.

We can use the reasoning in (3.60), in this case with X = I, where I is an indicator variable for the event that B gets a head, and with Y = S. Then $T = I \cdot S$, and I and S are independent, so

$$Var(T) = Var(IS) = [Var(I) + (EI)^2] \cdot [Var(S) + (ES)^2] - (EI \cdot ES)^2 \tag{3.103}$$

Then use the facts that I has mean 0.5 and variance 0.5(1-0.5) (Equations (3.78) and (3.79)), with S having the mean 10 and variance 5, as given in the problem.

3.10 Expected Value, Etc. in the ALOHA Example

Finding expected values etc. in the ALOHA example is straightforward. For instance,

$$EX_1 = 0 \cdot P(X_1 = 0) + 1 \cdot P(X_1 = 1) + 2 \cdot P(X_1 = 2) = 1 \cdot 0.48 + 2 \cdot 0.52 = 1.52 \qquad (3.104)$$

Here is R code to find various values approximately by simulation:

```
1   # finds E(X1), E(X2), Var(X2), Cov(X1,X2)
2   sim <- function(p,q,nreps) {
3      sumx1 <- 0
4      sumx2 <- 0
5      sumx2sq <- 0
6      sumx1x2 <- 0
7      for (i in 1:nreps) {
8         numtrysend <-
9            sum(sample(0:1,2,replace=TRUE,prob=c(1-p,p)))
10        if (numtrysend == 1)  X1 <- 1
11        else X1 <- 2
12        numactive <- X1
13        if (X1 == 1 && runif(1) < q) numactive <- numactive + 1
14        if (numactive == 1)
15           if (runif(1) < p) X2 <- 0
16           else X2 <- 1
17        else {  # numactive = 2
18           numtrysend <- 0
19           for (i in 1:2)
20              if (runif(1) < p) numtrysend <- numtrysend + 1
21           if (numtrysend == 1) X2 <- 1
22           else X2 <- 2
23        }
24        sumx1 <- sumx1 + X1
25        sumx2 <- sumx2 + X2
26        sumx2sq <- sumx2sq + X2^2
27        sumx1x2 <- sumx1x2 + X1*X2
28     }
29     # print results
30     meanx1 <- sumx1 /nreps
31     cat("E(X1):",meanx1,"\n")
32     meanx2 <- sumx2 /nreps
33     cat("E(X2):",meanx2,"\n")
34     cat("Var(X2):",sumx2sq/nreps - meanx2^2,"\n")
35     cat("Cov(X1,X2):",sumx1x2/nreps - meanx1*meanx2,"\n")
36  }
```

As a check on your understanding so far, you should find at least one of these values by hand, and see if it jibes with the simulation output.

3.11   Example: Measurements at Different Ages

Say a large research program measures boys' heights at age 10 and age 15. Call the two heights X and Y. So, each boy has an X and a Y. Each boy is a "notebook line", and the notebook has two columns, for X and Y. We are interested in Var(Y-X). Which of the following is true?

(i) $Var(Y - X) = Var(Y) + Var(X)$

(ii) $Var(Y - X) = Var(Y) - Var(X)$

(iii) $Var(Y - X) < Var(Y) + Var(X)$

(iv) $Var(Y - X) < Var(Y) - Var(X)$

(v) $Var(Y - X) > Var(Y) + Var(X)$

(vi) $Var(Y - X) > Var(Y) - Var(X)$

(vii) None of the above.

Use the mailing tube (3.74):

$$Var(Y - X) = Var[Y + (-X)] = Var(Y) + Var(X) - 2Cov(X, Y) \tag{3.105}$$

Since X and Y are positively correlated, their covariance is positive, so the answer is (iii).

3.12   Example: Bus Ridership Model

In the bus ridership model, Section 2.11, let's find $Var(L_1)$:

$$Var(L_1) = E(L_1^2) - (EL_1)^2 \tag{3.106}$$

$$EL_1 = EB_1 = 0 \cdot 0.5 + 1 \cdot 0.4 + 2 \cdot 0.1 \tag{3.107}$$

$$E(L_1^2) = 0^2 \cdot 0.5 + 1^2 \cdot 0.4 + 2^2 \cdot 0.1 \tag{3.108}$$

Then put it all together.

3.13 Distributions

The idea of the **distribution** of a random variable is central to probability and statistics.

Definition 6 *Let U be a discrete random variable. Then the distribution of U is simply the support of U, together with the associated probabilities.*

Example: Let X denote the number of dots one gets in rolling a die. Then the values X can take on are 1,2,3,4,5,6, each with probability 1/6. So

$$\text{distribution of } X = \{(1, \frac{1}{6}), (2, \frac{1}{6}), (3, \frac{1}{6}), (4, \frac{1}{6}), (5, \frac{1}{6}), (6, \frac{1}{6})\} \tag{3.109}$$

Example: Recall the ALOHA example. There X_1 took on the values 1 and 2, with probabilities 0.48 and 0.52, respectively (the case of 0 was impossible). So,

$$\text{distribution of } X_1 = \{(0, 0.00), (1, 0.48), (2, 0.52)\} \tag{3.110}$$

Example: Recall our example in which N is the number of tosses of a coin needed to get the first head. N has support 1,2,3,..., the probabilities of which we found earlier to be 1/2, 1/4, 1/8,... So,

$$\text{distribution of } N = \{(1, \frac{1}{2}), (2, \frac{1}{4}), (3, \frac{1}{8}), ...\} \tag{3.111}$$

It is common to express this in functional notation:

Definition 7 *The **probability mass function** (pmf) of a discrete random variable V, denoted p_V, as*

$$p_V(k) = P(V = k) \tag{3.112}$$

for any value k in the support of V.

(Please keep in mind the notation. It is customary to use the lower-case p, with a subscript consisting of the name of the random variable.)

Note that $p_V()$ is just a function, like any function (with integer domain) you've had in your previous math courses. For each input value, there is an output value.

3.13.1   Example: Toss Coin Until First Head

In (3.111),

$$p_N(k) = \frac{1}{2^k}, k = 1, 2, ... \tag{3.113}$$

3.13.2   Example: Sum of Two Dice

In the dice example, in which S = X+Y,

$$p_S(k) = \begin{cases} \frac{1}{36}, & k = 2 \\ \frac{2}{36}, & k = 3 \\ \frac{3}{36}, & k = 4 \\ ... \\ \frac{1}{36}, & k = 12 \end{cases} \tag{3.114}$$

It is important to note that there may not be some nice closed-form expression for p_V like that of (3.113). There was no such form in (3.114), nor is there in our ALOHA example for p_{X_1} and p_{X_2}.

3.13.3   Example: Watts-Strogatz Random Graph Model

Random graph models are used to analyze many types of link systems, such as power grids, social networks and even movie stars. We saw our first example in Section 2.13.1, and here is another, a variation on a famous model of that type, due to Duncan Watts and Steven Strogatz.

3.13.3.1   The Model

We have a graph of n nodes, e.g. in which each node is a person).[4] Think of them as being linked in a circle—we're just talking about relations here, not physical locations—so we already have n links. One can thus reach any node in the graph from any other, by following the links of the circle. (We'll assume all links are bidirectional.)

We now randomly add k more links (k is thus a parameter of the model), which will serve as "shortcuts." There are $\binom{n}{2} = n(n-1)/2$ possible links between nodes, but remember, we already

---

[4]The word *graph* here doesn't mean "graph" in the sense of a picture. Here we are using the computer science sense of the word, meaning a system of vertices and edges. It's common to call those *nodes* and *links*.

have n of those in the graph, so there are only $n(n-1)/2 - n = n^2/2 - 3n/2$ possibilities left. We'll be forming k new links, chosen at random from those $n^2/2 - 3n/2$ possibilities.

Let M denote the number of links attached to a particular node, known as the **degree** of a node. M is a random variable (we are choosing the shortcut links randomly), so we can talk of its pmf, p_M, termed the **degree distribution** of M, which we'll calculate now.

Well, $p_M(r)$ is the probability that this node has r links. Since the node already had 2 links before the shortcuts were constructed, $p_M(r)$ is the probability that r-2 of the k shortcuts attach to this node.

This problem is similar in spirit to (though admittedly more difficult to think about than) kings-and-hearts example of Section 2.15.1. Other than the two neighboring links in the original circle and the "link" of a node to itself, there are n-3 possible shortcut links to attach to our given node. We're interested in the probability that r-2 of them are chosen, and that k-(r-2) are chosen from the other possible links. Thus our probability is:

$$p_M(r) = \frac{\binom{n-3}{r-2}\binom{n^2/2-3n/2-(n-3)}{k-(r-2)}}{\binom{n^2/2-3n/2}{k}} = \frac{\binom{n-3}{r-2}\binom{n^2/2-5n/2+3}{k-(r-2)}}{\binom{n^2/2-3n/2}{k}} \tag{3.115}$$

3.13.3.2   Further Reading

UCD professor Raissa D'Souza specializes in random graph models. See for instance Beyond Friend-ship: Modeling User activity Graphs on Social Network-Based Gifting Applications, A. Nazir, A. Waagen, V. Vijayaraghavan, C.-N. Chuah, R. M. D'Souza, B. Krishnamurthy, *ACM Internet Measurement Conference (IMC 2012)*, Nov 2012.

3.14   Proof of Chebychev's Inequality (optional section)

To prove (3.63), let's first state and prove Markov's Inequality: For any nonnegative random variable Y and positive constant d,

$$P(Y \geq d) \leq \frac{EY}{d} \tag{3.116}$$

To prove (3.116), let Z be the indicator random variable for the event $Y \geq d$ (Section 3.9).

notebook line	Y	dZ	$Y \geq dZ$?
1	0.36	0	yes
2	3.6	3	yes
3	2.6	0	yes

Table 3.2: Illustration of Y and Z

Now note that

$$Y \geq dZ \tag{3.117}$$

To see this, just think of a notebook, say with d = 3. Then the notebook might look like Table 3.2.

So

$$EY \geq dEZ \tag{3.118}$$

(Again think of the notebook. The long-run average in the Y column will be $\geq$ the corresponding average for the dZ column.)

The right-hand side of (3.118) is $dP(Y \geq d)$, so (3.116) follows.

Now to prove (3.63), define

$$Y = (X - \mu)^2 \tag{3.119}$$

and set $d = c^2\sigma^2$. Then (3.116) says

$$P[(X - \mu)^2 \geq c^2\sigma^2] \leq \frac{E[(X - \mu)^2]}{c^2\sigma^2} \tag{3.120}$$

Since

$$(X - \mu)^2 \geq c^2\sigma^2 \text{ if and only if } |X - \mu| \geq c\sigma \tag{3.121}$$

the left-hand side of (3.120) is the same as the left-hand side of (3.63). The numerator of the right-hand size of (3.120) is simply Var(X), i.e. σ^2, so we are done.

Chapter 4

Discrete Parametric Distribution Families

The notion of a *parametric family* of distributions is a key concept that will recur throughout the book.

Consider plotting the curves $g_{a,b}(t) = (t-a)^2 + b$. For each a and b, we get a different parabola, as seen in this plot of three of the curves:
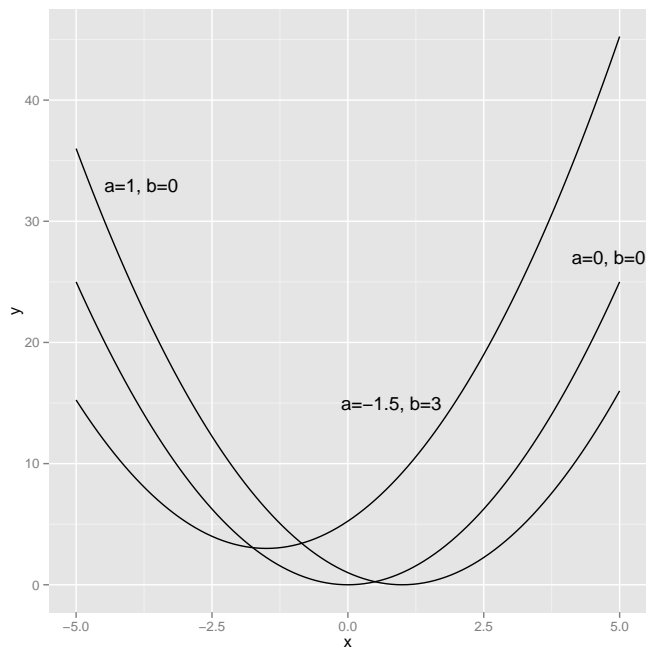
This is a family of curves, thus a family of functions. We say the numbers a and b are the **parameters** of the family. Note carefully that t is not a parameter, but rather just an argument of each function. The point is that a and b are indexing the curves.

4.1   The Case of Importance to Us: Parameteric Families of pmfs

Probability mass functions are still functions.[1] Thus they too can come in parametric families, indexed by one or more parameters. We had an example in Section 3.13.3. Since we get a different function p_M for each different values of k and n, that was a parametric family of pmfs, indexed by k and n.

Some parametric families of pmfs have been found to be so useful over the years that they've been given names. We will discuss some of those families here. But remember, they are famous just because they have been found useful, i.e. that they fit real data well in various settings. **Do not jump to the conclusion that we always "must" use pmfs from some family.**

4.2   The Geometric Family of Distributions

To explain our first parametric family of pmfs, recall our example of tossing a coin until we get the first head, with N denoting the number of tosses needed. In order for this to take k tosses, we need k-1 tails and then a head. Thus

$$p_N(k) = (1 - \frac{1}{2})^{k-1} \cdot \frac{1}{2}, k = 1, 2, ... \tag{4.1}$$

We might call getting a head a "success," and refer to a tail as a "failure." Of course, these words don't mean anything; we simply refer to the outcome of interest (which of course we ourselves choose) as "success."

Define M to be the number of rolls of a die needed until the number 5 shows up. Then

$$p_M(k) = \left(1 - \frac{1}{6}\right)^{k-1} \frac{1}{6}, k = 1, 2, ... \tag{4.2}$$

reflecting the fact that the event {M = k} occurs if we get k-1 non-5s and then a 5. Here "success" is getting a 5.

---

[1]The domains of these functions are typically the integers, but that is irrelevant; a function is a function.

The tosses of the coin and the rolls of the die are known as **Bernoulli trials**, which is a sequence of independent events. We call the occurrence of the event **success** and the nonoccurrence **failure** (just convenient terms, not value judgments). The associated indicator random variable are denoted B_i, i = 1,2,3,... So B_i is 1 for success on the ith trial, 0 for failure, with success probability p. For instance, p is 1/2 in the coin case, and 1/6 in the die example.

In general, suppose the random variable W is defined to be the number of trials needed to get a success in a sequence of Bernoulli trials. Then

$$p_W(k) = (1-p)^{k-1}p, k = 1, 2, ... \tag{4.3}$$

Note that there is a different distribution for each value of p, so we call this a **parametric family** of distributions, indexed by the parameter p. We say that W is **geometrically distributed** with parameter p.[2]

It should make good intuitive sense to you that

$$E(W) = \frac{1}{p} \tag{4.4}$$

This is indeed true, which we will now derive. First we'll need some facts (which you should file mentally for future use as well):

Properties of Geometric Series:

(a) For any $t \neq 1$ and any nonnegative integers $r \leq s$,

$$\sum_{i=r}^{s} t^i = t^r \frac{1 - t^{s-r+1}}{1 - t} \tag{4.5}$$

This is easy to derive for the case r = 0, using mathematical induction. For the general case, just factor out t^r.

(b) For $|t| < 1$,

$$\sum_{i=0}^{\infty} t^i = \frac{1}{1 - t} \tag{4.6}$$

To prove this, just take r = 0 and let $s \to \infty$ in (4.5).

---

[2]Unfortunately, we have overloaded the letter p here, using it to denote the probability mass function on the left side, and the unrelated parameter p, our success probability on the right side. It's not a problem as long as you are aware of it, though.

(c) For $|t| < 1$,

$$\sum_{i=1}^{\infty} i t^{i-1} = \frac{1}{(1-t)^2} \tag{4.7}$$

This is derived by applying $\frac{d}{dt}$ to (4.6).[3]

Deriving (4.4) is then easy, using (4.7):

$$EW = \sum_{i=1}^{\infty} i(1-p)^{i-1} p \tag{4.8}$$

$$= p \sum_{i=1}^{\infty} i(1-p)^{i-1} \tag{4.9}$$

$$= p \cdot \frac{1}{[1-(1-p)]^2} \tag{4.10}$$

$$= \frac{1}{p} \tag{4.11}$$

Using similar computations, one can show that

$$Var(W) = \frac{1-p}{p^2} \tag{4.12}$$

We can also find a closed-form expression for the quantities $P(W \leq m)$, m = 1,2,... (This has a formal name $F_W(m)$, as will be seen later in Section 7.3.) For any positive integer m we have

$$
\begin{aligned}
F_W(m) &= P(W \leq m) & (4.13)\\
&= 1 - P(W > m) & (4.14)\\
&= 1 - P(\text{the first m trials are all failures}) & (4.15)\\
&= 1 - (1-p)^m & (4.16)
\end{aligned}
$$

By the way, if we were to think of an experiment involving a geometric distribution in terms of our notebook idea, the notebook would have an infinite number of columns, one for each B_i. Within each row of the notebook, the B_i entries would be 0 until the first 1, then NA ("not applicable") after that.

---

[3]To be more careful, we should differentiate (4.5) and take limits.

4.2.1   R Functions

You can simulate geometrically distributed random variables via R's **rgeom()** function. Its first argument specifies the number of such random variables you wish to generate, and the second is the success probability p.

For example, if you run

```
> y <- rgeom(2,0.5)
```

then it's simulating tossing a coin until you get a head (**y[1]**) and then tossing the coin until a head again (**y[2]**). Of course, you could simulate on your own, say using **sample()** and **while()**, but R makes it convenient for you.

Here's the full set of functions for a geometrically distributed random variable X with success probability p:

- **dgeom(i,p)**, to find $P(X = i)$

- **pgeom(i,p)**, to find $P(X \leq i)$

- **qgeom(q,p)**, to find c such that $P(X \leq c) = q$

- **rgeom(n,p)**, to generate n variates from this geometric distribution

Important note: Some books define geometric distributions slightly differently, as the number of failures before the first success, rather than the number of trials to the first success. The same is true for software—both R and Python define it this way. Thus for example in calling **dgeom()**, subtract 1 from the value used in our definition.

For example, here is P(N = 3) for a geometric distribution under our defintion, with p = 0.4:

```
> dgeom(2,0.4)
[1] 0.144
> # check
> (1-0.4)^(3-1) * 0.4
[1] 0.144
```

Note that this also means one must *add* 1 to the result of **rgeom()**.

4.2.2   Example: a Parking Space Problem

Suppose there are 10 parking spaces per block on a certain street. You turn onto the street at the start of one block, and your destination is at the start of the next block. You take the first parking space you encounter. Let D denote the distance of the parking place you find from your destination, measured in parking spaces. Suppose each space is open with probability 0.15, with the spaces being independent. Find ED.

To solve this problem, you might at first think that D follows a geometric distribution. **But don't jump to conclusions!** Actually this is not the case; D is a somewhat complicated distance. But clearly D is a function of N, where the latter denotes the number of parking spaces you see until you find an empty one—and N *is* geometrically distributed.

As noted, D is a function of N:

$$D = \begin{cases} 11 - N, & N \le 10 \\ N - 11, & N > 10 \end{cases} \tag{4.17}$$

Since D is a function of N, we can use (3.36) with g(t) as in (4.17):

$$ED = \sum_{i=1}^{10}(11 - i)(1 - 0.15)^{i-1}0.15 + \sum_{i=11}^{\infty}(i - 11)0.85^{i-1}0.15 \tag{4.18}$$

This can now be evaluated using the properties of geometric series presented above.

Alternatively, here's how we could find the result by simulation:

```
1  parksim <- function(nreps) {
2     # do the experiment nreps times, recording the values of N
3     nvals <- rgeom(nreps,0.15) + 1
4     # now find the values of D
5     dvals <- ifelse(nvals <= 10,11-nvals,nvals-11)
6     # return ED
7     mean(dvals)
8  }
```

Note the vectorized addition and recycling (Section 2.14.2) in the line

```
nvals <- rgeom(nreps,0.15) + 1
```

The call to **ifelse()** is another instance of R's vectorization, a vectorized if-then-else. The first argument evaluates to a vector of TRUE and FALSE values. For each TRUE, the corresponding

element of **dvals** will be set to the corresponding element of the vector **11-nvals** (again involving vectorized addition and recycling), and for each false, the element of **dvals** will be set to the element of **nvals-11**.

Let's find some more, first $p_N(3)$:

$$p_N(3) = P(N = 3) = (1 - 0.15)^{3-1}0.15 \tag{4.19}$$

Next, find P(D = 1):

$$
\begin{aligned}
P(D = 1) &= P(N = 10 \text{ or } N = 12) & (4.20)\\
&= (1 - 0.15)^{10-1}0.15 + (1 - 0.15)^{12-1}0.15 & (4.21)
\end{aligned}
$$

Say Joe is the one looking for the parking place. Paul is watching from a side street at the end of the first block (the one before the destination), and Martha is watching from an alley situated right after the sixth parking space in the second block. Martha calls Paul and reports that Joe never went past the alley, and Paul replies that he did see Joe go past the first block. They are interested in the probability that Joe parked in the second space in the second block. In mathematical terms, what probability is that? Make sure you understand that it is $P(N = 12 \mid N > 10 \text{ and } N \leq 16)$. It can be evaluated as above.

Or consider a different question: Good news! I found a parking place just one space away from the destination. Find the probability that I am parked in the same block as the destination.

$$
\begin{aligned}
P(N = 12 \mid N = 10 \text{ or } N = 12) &= \frac{P(N = 12)}{P(N = 10 \text{ or } N = 12)} & (4.22)\\
&= \frac{(1 - 0.15)^{11}\ 0.15}{(1 - 0.15)^{9}\ 0.15 + (1 - 0.15)^{11}\ 0.15} & (4.23)
\end{aligned}
$$

4.3 The Binomial Family of Distributions

A geometric distribution arises when we have Bernoulli trials with parameter p, with a variable number of trials (N) but a fixed number of successes (1). A **binomial distribution** arises when we have the opposite—a fixed number of Bernoulli trials (n) but a variable number of successes (say X).[4]

---

[4]Note again the custom of using capital letters for random variables, and lower-case letters for constants.

For example, say we toss a coin five times, and let X be the number of heads we get. We say that X is binomially distributed with parameters n = 5 and p = 1/2. Let's find P(X = 2). There are many orders in which that could occur, such as HHTTT, TTHHT, HTTHT and so on. Each order has probability $0.5^2(1 - 0.5)^3$, and there are $\binom{5}{2}$ orders. Thus

$$P(X = 2) = \binom{5}{2} 0.5^2 (1 - 0.5)^3 = \binom{5}{2}/32 = 5/16 \tag{4.24}$$

For general n and p,

$$p_X(k) = P(X = k) = \binom{n}{k} p^k (1 - p)^{n-k} \tag{4.25}$$

So again we have a parametric family of distributions, in this case a family having two parameters, n and p.

Let's write X as a sum of those 0-1 Bernoulli variables we used in the discussion of the geometric distribution above:

$$X = \sum_{i=1}^{n} B_i \tag{4.26}$$

where B_i is 1 or 0, depending on whether there is success on the ith trial or not. Note again that the B_i are indicator random variables (Section 3.9), so

$$EB_i = p \tag{4.27}$$

and

$$Var(B_i) = p(1 - p) \tag{4.28}$$

Then the reader should use our earlier properties of E() and Var() in Sections 3.5 and 3.6 to fill in the details in the following derivations of the expected value and variance of a binomial random variable:

$$EX = E(B_1 + ... + B_n) = EB_1 + ... + EB_n = np \tag{4.29}$$

and from (3.75),

$$Var(X) = Var(B_1 + ... + B_n) = Var(B_1) + ... + Var(B_n) = np(1 - p) \tag{4.30}$$

Again, (4.29) should make good intuitive sense to you.

4.3.1 R Functions

Relevant functions for a binomially distributed random variable X for k trials and with success probability p are:

- **dbinom(i,k,p)**, to find $P(X = i)$

- **pbinom(i,k,p)**, to find $P(X \leq i)$

- **qbinom(q,k,p)**, to find c such that $P(X \leq c) = q$

- **rbinom(n,k,p)**, to generate n independent values of X

Our definition above of **qbinom()** is not quite tight, though. Consider a random variable X which has a binomial distribution with n = 2 and p = 0.5 Then

$$F_X(0) = 0.25, \ F_X(1) = 0.75 \tag{4.31}$$

So if q is, say, 0.33, there is no c such that $P(X \leq c) = q$. For that reason, the actual definition of **qbinom()** is the smallest c satisfying $P(X \leq c) \geq q$.

4.3.2 Example: Parking Space Model

Recall Section 4.2.2. Let's find the probability that there are three open spaces in the first block.

Let M denote the number of open spaces in the first block. This fits the definition of binomially-distributed random variables: We have a fixed number (10) of independent Bernoulli trials, and we are interested in the number of successes. So, for instance,

$$p_M(3) = \binom{10}{3} 0.15^3 (1 - 0.15)^{10-3} \tag{4.32}$$

4.4   The Negative Binomial Family of Distributions

Recall that a typical example of the geometric distribution family (Section 4.2) arises as N, the number of tosses of a coin needed to get our first head. Now generalize that, with N now being the number of tosses needed to get our r^{th} head, where r is a fixed value. Let's find P(N = k), k = r, r+1, ... For concreteness, look at the case r = 3, k = 5. In other words, we are finding the probability that it will take us 5 tosses to accumulate 3 heads.

First note the equivalence of two events:

$$\{N = 5\} = \{2 \text{ heads in the first 4 tosses and head on the } 5^{th} \text{ toss}\} \tag{4.33}$$

That event described before the "and" corresponds to a binomial probability:

$$P(2 \text{ heads in the first 4 tosses}) = \binom{4}{2}\left(\frac{1}{2}\right)^4 \tag{4.34}$$

Since the probability of a head on the kth toss is 1/2 and the tosses are independent, we find that

$$P(N = 5) = \binom{4}{2}\left(\frac{1}{2}\right)^5 = \frac{3}{16} \tag{4.35}$$

The negative binomial distribution family, indexed by parameters r and p, corresponds to random variables that count the number of independent trials with success probability p needed until we get r successes. The pmf is

$$p_N(k) = P(N = k) = \binom{k-1}{r-1}(1-p)^{k-r}p^r, k = r, r+1, ... \tag{4.36}$$

We can write

$$N = G_1 + ... + G_r \tag{4.37}$$

where G_i is the number of tosses between the successes numbers i-1 and i. But each G_i has a geometric distribution! Since the mean of that distribution is 1/p, we have that

$$E(N) = r \cdot \frac{1}{p} \tag{4.38}$$

In fact, those r geometric variables are also independent, so we know the variance of N is the sum of their variances:

$$Var(N) = r \cdot \frac{1-p}{p^2} \tag{4.39}$$

4.4.1 R Functions

Relevant functions for a negative binomial distributed random variable X with success parameter p are:

- **dnbinom(i,size=1,prob=p)**, to find $P(X = i)$

- **pnbinom(i,size=1,prob=p)**, to find $P(X <= i)$

- **qnbinom(q,sixe=1,prob=p)**, to find c such that $P(X \leq c) = q$

- **rnbinom(n,size=1,prob=p)**, to generate n independent values of X

Here **size** is our r. Note, though, that as with the **geom()** family, R defines the distribution in terms of number of failures. So, in **dbinom()**, the argument **i** is the number of failures, and **i + r** is our X.

4.4.2 Example: Backup Batteries

A machine contains one active battery and two spares. Each battery has a 0.1 chance of failure each month. Let L denote the lifetime of the machine, i.e. the time in months until the third battery failure. Find P(L = 12).

The number of months until the third failure has a negative binomial distribution, with r = 3 and p = 0.1. Thus the answer is obtained by (4.36), with k = 12:

$$P(L = 12) = \binom{11}{2}(1 - 0.1)^9 0.1^3 \tag{4.40}$$

4.5 The Poisson Family of Distributions

Another famous parametric family of distributions is the set of **Poisson Distributions**.

This family is a little different from the geometric, binomial and negative binomial families, in the sense that in those cases there were qualitative descriptions of the settings in which such distributions arise. Geometrically distributed random variables, for example occur as the number of Bernoulli trials needed to get the first success.

By contrast, the Poisson family does not really have this kind of qualitative description.[5] It is merely something that people have found to be a reasonably accurate model of actual data in many cases. We might be interested, say, in the number of disk drive failures in periods of a specified length of time. If we have data on this, we might graph it, and if it looks like the pmf form below, then we might adopt it as our model.

The pmf is

$$P(X = k) = \frac{e^{-\lambda}\lambda^k}{k!}, k = 0, 1, 2, ... \tag{4.41}$$

It turns out that

$$EX = \lambda \tag{4.42}$$

$$Var(X) = \lambda \tag{4.43}$$

The derivations of these facts are similar to those for the geometric family in Section 4.2. One starts with the Maclaurin series expansion for e^t:

$$e^t = \sum_{i=0}^{\infty} \frac{t^i}{i!} \tag{4.44}$$

and finds its derivative with respect to t, and so on. The details are left to the reader.

The Poisson family is very often used to model count data. For example, if you go to a certain bank every day and count the number of customers who arrive between 11:00 and 11:15 a.m., you will probably find that that distribution is well approximated by a Poisson distribution for some λ.

There is a lot more to the Poisson story than we see in this short section. We'll return to this distribution family in Section 9.1.

---

[5]Some such descriptions are possible in the Poisson case, but they are complicated and difficult to verify.

4.5.1 R Functions

Relevant functions for a Poisson distributed random variable X with parameter lambda are:

- **dpois(i,lambda)**, to find $P(X = i)$

- **ppois(i,lambda)**, to find $P(X \leq i)$

- **qpois(q,lambda)**, to find c such that $P(X \leq c) = q$

- **rpois(n,lambda)**, to generate n independent values of X

4.5.2 Example: Broken Rod

Recall the example of a broken glass rod in Section 2.14.10. Suppose now that the number of breaks is random, not just the break points. A reasonable model to try would be Poisson. However, the latter's support starts at 0, and we cannot have 0 pieces, so we need to model the number of pieces minus 1 (the number of break points) as Poisson.

The code is similar to that in Section 2.14.10, but we must first generate the number of break points:

```
minpiecepois <- function(lambda) {
   nbreaks <- rpois(1,lambda)
   breakpts <- sort(runif(nbreaks))
   lengths <- diff(c(0,breakpts,1))
   min(lengths)
}

bkrodpois <- function(nreps,lambda,q) {
   minpieces <- replicate(nreps,minpiecepois(lambda))
   mean(minpieces < q)
}

> bkrodpois(10000,5,0.02)
[1] 0.4655
```

Note that in each call to **minpiecepois()**, there will be a different number of breakpoints.

4.6   The Power Law Family of Distributions

This family has attracted quite a bit of attention in recent years, due to its use in random graph models.

4.6.1   The Model

Here

$$p_X(k) = ck^{-\gamma}, \ k = 1, 2, 3, ... \tag{4.45}$$

It is required that $\gamma > 1$, as otherwise the sum of probabilities will be infinite. For γ satisfying that condition, the value c is chosen so that that sum is 1.0:

$$1.0 = \sum_{k=1}^{\infty} ck^{-\gamma} \approx c \int_1^{\infty} k^{-\gamma} \ dk = c/(\gamma - 1) \tag{4.46}$$

so $c \approx \gamma - 1$.

Here again we have a parametric family of distributions, indexed by the parameter γ.

The power law family is an old-fashioned model (an old-fashioned term for *distribution* is *law*), but there has been a resurgence of interest in it in recent years. Analysts have found that many types of social networks in the real world exhibit approximately power law behavior in their degree distributions.

For instance, in a famous study of the Web (A. Barabasi and R. Albert, Emergence of Scaling in Random Networks, *Science*, 1999, 509-512), degree distribution on the Web (a directed graph, with incoming links being the ones of interest here) it was found that the number of links leading to a Web page has an approximate power law distribution with $\gamma = 2.1$. The number of links leading out of a Web page was found to be approximately power-law distributed, with $\gamma = 2.7$.

Much of the interest in power laws stems from their **fat tails**, a term meaning that values far from the mean are more likely under a power law than they would be under a normal distribution with the same mean. In recent popular literature, values far from the mean have often been called **black swans**. The financial crash of 2008, for example, is blamed by some on the ignorance by **quants** (people who develop probabilistic models for guiding investment) in underestimating the probabilities of values far from the mean.

Some examples of real data that are, or are not, fit well by power law models are given in the paper *Power-Law Distributions in Empirical Data*, by A. Clauset, C. Shalizi and M. Newman, at

`http://arxiv.org/abs/0706.1062`. Methods for estimating the parameter γ are discussed and evaluated.

A variant of the power law model is the **power law with exponential cutoff**, which essentially consists of a blend of the power law and a geometric distribution. Here

$$p_X(k) = ck^{-\gamma}q^k \tag{4.47}$$

This now is a two-parameter family, the parameters being γ and q. Again c is chosen so that the pmf sums to 1.0.

This model is said to work better than a pure power law for some types of data. Note, though, that this version does not really have the fat tail property, as the tail decays exponentially now.

4.6.2 Further Reading

There is nice paper on fitting (or not fitting) power law models:

Power-Law Distributions in Empirical Data, *SIAM Review*, A. Clauset, C.R. Shalizi, and M.E.J. Newman, 51(4), 661-703, 2009.

4.7 Recognizing Some Parametric Distributions When You See Them

Three of the discrete distribution families we've considered here arise in settings with very definite structure, all dealing with independent trials:

- the binomial family gives the distribution of the number of successes in a fixed number of trials

- the geometric family gives the distribution of the number of trials needed to obtain the first success

- the negative binomial family gives the distribution of the number of trials needed to obtain the kth success

Such situations arise often, hence the fame of these distribution families.

By contrast, the Poisson and power law distributions have no underlying structure. They are famous for a different reason, that it has been found empirically that they provide a good fit to many real data sets.

In other words, the Poisson and power law distributions are typically fit to data, in an attempt to find a good model, whereas in the binomial, geometric and negative binomial cases, the fundamental nature of the setting implies one of those distributions.

You should make a strong effort to get to the point at which you automatically recognize such settings when your encounter them.

4.8   Example: a Coin Game

Life is unfair—former President Jimmy Carter

Consider a game played by Jack and Jill. Each of them tosses a coin many times, but Jack gets a head start of two tosses. So by the time Jack has had, for instance, 8 tosses, Jill has had only 6; when Jack tosses for the 15th time, Jill has her 13th toss; etc.

Let X_k denote the number of heads Jack has gotten through his kth toss, and let Y_k be the head count for Jill at that same time, i.e. among only k-2 tosses for her. (So, $Y_1 = Y_2 = 0$.) Let's find the probability that Jill is winning after the 6th toss, i.e. $P(Y_6 > X_6)$.

Your first reaction might be, "Aha, binomial distribution!" You would be on the right track, but the problem is that you would not be thinking precisely enough. Just WHAT has a binomial distribution? The answer is that both X_6 and Y_6 have binomial distributions, both with p = 0.5, but n = 6 for X_6 while n = 4 for Y_6.

Now, as usual, ask the famous question, "How can it happen?" How can it happen that $Y_6 > X_6$? Well, we could have, for example, $Y_6 = 3$ and $X_6 = 1$, as well as many other possibilities. Let's write it mathematically:

$$P(Y_6 > X_6) = \sum_{i=1}^{4} \sum_{j=0}^{i-1} P(Y_6 = i \text{ and } X_6 = j) \tag{4.48}$$

Make SURE your understand this equation.

Now, to evaluate $P(Y_6 = i \text{ and } X_6 = j)$, we see the "and" so we ask whether Y_6 and X_6 are independent. They in fact are; Jill's coin tosses certainly don't affect Jack's. So,

$$P(Y_6 = i \text{ and } X_6 = j) = P(Y_6 = i) \cdot P(X_6 = j) \tag{4.49}$$

It is at this point that we finally use the fact that X_6 and Y_6 have binomial distributions. We have

$$P(Y_6 = i) = \binom{4}{i} 0.5^i (1 - 0.5)^{4-i} \tag{4.50}$$

and

$$P(X_6 = j) = \binom{6}{j} 0.5^j (1 - 0.5)^{6-j} \tag{4.51}$$

We would then substitute (4.50) and (4.51) in (4.48). We could then evaluate it by hand, but it would be more convenient to use R's **dbinom()** function:

```
1   prob <- 0
2   for (i in 1:4)
3      for (j in 0:(i-1))
4          prob <- prob + dbinom(i,4,0.5) * dbinom(j,6,0.5)
5   prob
```

We get an answer of about 0.17. If Jack and Jill were to play this game repeatedly, stopping each time after the 6th toss, then Jill would win about 17% of the time.

4.9 Example: Tossing a Set of Four Coins

Consider a game in which we have a set of four coins. We keep tossing the set of four until we have a situation in which exactly two of them come up heads. Let N denote the number of times we must toss the set of four coins.

For instance, on the first toss of the set of four, the outcome might be HTHH. The second might be TTTH, and the third could be THHT. In the situation, N = 3.

Let's find $P(N = 5)$. Here we recognize that N has a geometric distribution, with "success" defined as getting two heads in our set of four coins. What value does the parameter p have here?

Well, p is $P(X = 2)$, where X is the number of heads we get from a toss of the set of four coins. We recognize that X is binomial! Thus

$$p = \binom{4}{2} 0.5^4 = \frac{3}{8} \tag{4.52}$$

Thus using the fact that N has a geometric distribution,

$$P(N = 5) = (1 - p)^4 p = 0.057 \tag{4.53}$$

4.10   Example: the ALOHA Example Again

As an illustration of how commonly these parametric families arise, let's again look at the ALOHA example. Consider the general case, with transmission probability p, message creation probability q, and m network nodes. We will not restrict our observation to just two epochs.

Suppose $X_i = m$, i.e. at the end of epoch i all nodes have a message to send. Then the number which attempt to send during epoch i+1 will be binomially distributed, with parameters m and p.[6] For instance, the probability that there is a successful transmission is equal to the probability that exactly one of the m nodes attempts to send,

$$\binom{m}{1} p(1 - p)^{m-1} = mp(1 - p)^{m-1} \tag{4.54}$$

Now in that same setting, $X_i = m$, let K be the number of epochs it will take before some message actually gets through. In other words, we will have $X_i = m$, $X_{i+1} = m$, $X_{i+2} = m$,... but finally $X_{i+K-1} = m - 1$. Then K will be geometrically distributed, with success probability equal to (4.54).

There is no Poisson distribution in this example, but it is central to the analysis of Ethernet, and almost any other network. We will discuss this at various points in later chapters.

4.11   Example: the Bus Ridership Problem Again

Recall the bus ridership example of Section 2.11. Let's calculate some expected values, for instance $E(B_1)$:

$$E(B_1) = 0 \cdot P(B_1 = 0) + 1 \cdot P(B_1 = 1) + 2 \cdot P(B_1 = 2) = 0.4 + 2 \cdot 0.1 \tag{4.55}$$

Now suppose the company charges $3 for passengers who board at the first stop, but charges $2 for those who join at the second stop. (The latter passengers get a possibly shorter ride, thus pay

---

[6]Note that this is a conditional distribution, given $X_i = m$.

less.) So, the total revenue from the first two stops is $T = 3B_1 + 2B_2$. Let's find E(T). We'd write

$$E(T) = 3E(B_1) + 2E(B_2) \tag{4.56}$$

making use of (3.21). We'd then compute the terms as in 4.55.

Suppose the bus driver has the habit of exclaiming, "What? No new passengers?!" every time he comes to a stop at which $B_i = 0$. Let N denote the number of the stop (1,2,...) at which this first occurs. Find P(N = 3):

N has a geometric distribution, with p equal to the probability that there 0 new passengers at a stop, i.e. 0.5. Thus $p_N(3) = (1 - 0.5)^2 0.5$, by (4.3).

Let S denote the number of stops, out of the first 6, at which 2 new passengers board. For example, S would be 3 if $B_1 = 2$, $B_2 = 2$, $B_3 = 0$, $B_4 = 1$, $B_5 = 0$, and $B_6 = 2$. Find $p_S(4)$:

S has a binomial distribution, with n = 6 and p = probability of 2 new passengers at a stop = 0.1. Then

$$p_S(4) = \binom{6}{4} 0.1^4 (1 - 0.1)^{6-4} \tag{4.57}$$

By the way, we can exploit our knowledge of binomial distributions to simplify the simulation code in Section 2.14.8. The lines

```
for (k in 1:passengers)
   if (runif(1) < 0.2)
      passengers <- passengers - 1
```

simulate finding that number of passengers that alight at that stop. But that number is binomially distributed, so the above code can be compactified (and speeded up in execution) as

```
passengers <- passengers - rbinom(1,passengers,0.2)
```

4.12 Example: Flipping Coins with Bonuses

A game involves flipping a coin k times. Each time you get a head, you get a bonus flip, not counted among the k. (But if you get a head from a bonus flip, that does not give you its own bonus flip.) Let X denote the number of heads you get among all flips, bonus or not. Let's find the distribution of X.

As with the parking space example above, we should be careful not to come to hasty conclusions. The situation here "sounds" binomial, but X, based on a variable number of trials, doesn't fit the definition of binomial.

But let Y denote the number of heads you obtain through nonbonus flips. Y then has a binomial distribution with parameters k and 0.5. To find the distribution of X, we'll condition on Y.

We will as usual ask, "How can it happen?", but we need to take extra care in forming our sums, recognizing constraints on Y:

- $Y \geq X/2$

- $Y \leq X$

- $Y \leq k$

Keeping those points in mind, we have

$$
\begin{align}
p_X(m) \;&=\; P(X = m) \tag{4.58}\\[2mm]
&=\; \sum_{i=\text{ceil}(m/2)}^{\min(m,k)} P(X = m \text{ and } Y = i) \tag{4.59}\\[2mm]
&=\; \sum_{i=\text{ceil}(m/2)}^{\min(m,k)} P(X = m | Y = i)\; P(Y = i) \tag{4.60}\\[2mm]
&=\; \sum_{i=\text{ceil}(m/2)}^{\min(m,k)} \binom{i}{m-i} 0.5^i \binom{k}{i} 0.5^k \tag{4.61}\\[2mm]
&=\; 0.5^k \sum_{i=\text{ceil}(m/2)}^{\min(m,k)} \frac{k!}{(m-i)!(2i-m)!(k-i)!} 0.5^i \tag{4.62}
\end{align}
$$

There doesn't seem to be much further simplification possible here.

4.13   Example: Analysis of Social Networks

Let's continue our earlier discussion from Section 3.13.3.

One of the earliest—and now the simplest—models of social networks is due to Erdös and Renyi. Say we have n people (or n Web sites, etc.), with $\binom{n}{2}$ potential links between pairs. (We are assuming an undirected graph here.) In this model, each potential link is an actual link with probability p, and a nonlink with probability 1-p, with all the potential links being independent.

Recall the notion of degree distribution from Section 3.13.3. Clearly the degree distribution D_i here for a single node i is binomial with parameters n-1 and p.

But consider k nodes, say 1 through k, among the n total nodes, and let T denote the number of links involving these nodes. Let's find the distribution of T. That distribution is again binomial, but the number of trials must be carefully calculated. We cannot simply say that, since each of the k vertices can have as many as n-1 links, there are k(n-1) potential links, because there is overlap; two nodes among the k have a potential link with each other, but we can't count it twice. So, let's reason this out.

Say n = 9 and k = 4. Among the four special nodes, there are $\binom{4}{2} = 6$ potential links, each on or off with probability p, independently. Also each of the four special nodes has $9 - 4 = 5$ potential links with the "outside world," i.e. the five non-special nodes. So there are $4 \times 5 = 20$ potential links here, for a total of 26.

So, the distribution of T is binomial with

$$k(n - k) + \binom{k}{2} \tag{4.63}$$

trials and success probability p.

4.14 Multivariate Distributions

(I am borrowing some material here from Section 13.1, for instructors or readers who skip Chapter 13. It is important to know that multivariate distributions exist, even if one doesn't know the details.)

Recall that for a single discrete random variable X, the distribution of X was defined to be a list of all the values of X, together with the probabilities of those values. The same is done for a pair (or more than a pair) of discrete random variables U and V.

Suppose we have a bag containing two yellow marbles, three blue ones and four green ones. We choose four marbles from the bag at random, without replacement. Let Y and B denote the number

of yellow and blue marbles that we get. Then define the *two-dimensional* pmf of Y and B to be

$$p_{Y,B}(i,j) = P(Y = i \text{ and } B = j) = \frac{\binom{2}{i}\binom{3}{j}\binom{4}{4-i-j}}{\binom{9}{4}} \qquad (4.64)$$

Here is a table displaying all the values of $P(Y = i \text{ and } B = j)$:

i ↓, j →	0	1	2	3
0	0.008	0.095	0.143	0.032
1	0.063	0.286	0.190	0.016
2	0.048	0.095	0.024	0.000

So this table is the distribution of the pair (Y,B).

Recall further that in the discrete case, we introduced a symbolic notation for the distribution of a random variable X, defined as $p_X(i) = P(X = i)$, where i ranged over the support of X. We do the same thing for a pair of random variables:

Definition 8 *For discrete random variables U and V, their probability mass function is defined to be*

$$p_{U,V}(i,j) = P(U = i \text{ and } V = j) \qquad (4.65)$$

where (i,j) ranges over all values taken on by (U, V). Higher-dimensional pmfs are defined similarly, e.g.

$$p_{U,V,W}(i,j,k) = P(U = i \text{ and } V = j \text{ and } W = k) \qquad (4.66)$$

So in our marble example above, $p_{Y,B}(1,2) = 0.286$, $p_{Y,B}(2,0) = 0.048$ and so on.

4.15   Iterated Expectations

This section has an abstract title, but the contents are quite useful.

4.15.1   Conditional Distributions

Just as we can define bivariate pmfs, we can also speak of conditional pmfs. Suppose we have random variables U and V.

In our bus ridership example, for instance, we can talk about

$$E(L_2 \mid B_1 = 0) \tag{4.67}$$

In notebook terms, think of many replications of watching the bus during times 1 and 2. Then (4.67) is defined to be the long-run average of values in the L_2 column, **among those rows** in which the B_1 column is 0. (And by the way, make sure you understand why (4.67) works out to be 0.6.)

4.15.2   The Theorem

The key relation says, in essence,

> The overall mean of V is a weighted average of the conditional means of V given U. The weights are the pmf of U.

Note again that $E(V \mid U = c)$ is defined in "notebook" terms as the long-run average of V, *among those lines in which $U = c$.*

Here is the formal version:

Suppose we have random variables U and V, with U discrete and with V having an expected value. Then

$$E(V) = \sum_c P(U = c) \; E(V \mid U = c) \tag{4.68}$$

where c ranges through the support of U.

So, just as EX was a weighted average in (3.13), with weights being probabilities, we see here that the unconditional mean is a weighted average of the conditional mean.

In spite of its intimidating form, (4.68) makes good intuitive sense, as follows: Suppose we want to find the average height of all students at a university. Each department measures the heights of its majors, then reports the mean height among them. Then (4.68) says that to get the overall mean in the entire school, we should take a *weighted* average of all the within-department means, with the weights being the proportions of each department's student numbers among the entire school. Clearly, we would not want to take an unweighted average, as that would count tiny departments just as much as large majors.

Here is the derivation (reader: supply the reasons!).

$$EV \;=\; \sum_d d \, P(V = d) \tag{4.69}$$

$$=\; \sum_d d \, \sum_c P(U = c \text{ and } V = d) \tag{4.70}$$

$$=\; \sum_d d \, \sum_c P(U = c) \, P(V = d \mid U = c) \tag{4.71}$$

$$=\; \sum_d \sum_c d \, P(U = c) \, P(V = d \mid U = c) \tag{4.72}$$

$$=\; \sum_c \sum_d d \, P(U = c) \, P(V = d \mid U = c) \tag{4.73}$$

$$=\; \sum_c P(U = c) \sum_d d \, P(V = d \mid U = c) \tag{4.74}$$

$$=\; \sum_c P(U = c) \, E(V \mid U = c) \tag{4.75}$$

4.15.3   Example: Coin and Die Game

You roll a die until it comes up 5, taking M rolls to do so. You then toss a coin M times, winning one dollar for each head. Find the expected winnings, EW.

Solution: Given M = k, the number of heads has a binomial distribution with n = k and p = 0.5. So

$$E(W|M = k) = 0.5k. \tag{4.76}$$

So, from (4.68), we have

$$EW = \sum_{k=1}^{\infty} P(M = k) \, 0.5k = 0.5 \sum_{k=1}^{\infty} P(M = k)k = 0.5 \; EM \tag{4.77}$$

from (3.13). And from (4.4), we know EM = 6. So, EW = 3.

Note especially that we didn't need calculate P(M = k).

4.15.4   Example: Flipping Coins with Bonuses

Recall the example in Section 4.12. The principle of iterated expectation can easily get us EX:

$$
EX \;=\; \sum_{i=1}^{k} P(Y = i)\, E(X|Y = i) \tag{4.78}
$$

$$
=\; \sum_{i=1}^{k} P(Y = i)\, (i + 0.5i) \tag{4.79}
$$

$$
=\; E(1.5Y) \tag{4.80}
$$

$$
=\; 1.5 \cdot k/2 \tag{4.81}
$$

$$
=\; 0.75k \tag{4.82}
$$

To understand that second equality, note that if $Y = i$, X will already include those i heads, the nonbonus heads, and since there will be i bonus flips, the expected value of the number of bonus heads will be $0.5i$. The third equality uses (3.36) (in a reverse manner).

Chapter 5

Pause to Reflect

Now that we have some basics, let's step back a bit, and think about what we've learned.

5.1 A Cautionary Tale

Intuition is great, but it can lead you astray! The example in this section shows how things can go wrong.

5.1.1 Trick Coins, Tricky Example

Suppose we have two trick coins in a box. They look identical, but one of them, denoted coin 1, is heavily weighted toward heads, with a 0.9 probability of heads, while the other, denoted coin 2, is biased in the opposite direction, with a 0.9 probability of tails. Let C_1 and C_2 denote the events that we get coin 1 or coin 2, respectively.

Our experiment consists of choosing a coin at random from the box, and then tossing it n times. Let B_i denote the outcome of the ith toss, i = 1,2,3,..., where $B_i = 1$ means heads and $B_i = 0$ means tails. Let $X_i = B_1 + ... + B_i$, so X_i is a count of the number of heads obtained through the ith toss.

The question is: "Does the random variable X_i have a binomial distribution?" Or, more simply, the question is, "Are the random variables B_i independent?" To most people's surprise, the answer is No (to both questions). Why not?

The variables B_i are indeed 0-1 variables, and they have a common success probability. But they are not independent! Let's see why they aren't.

Consider the events $A_i = \{B_i = 1\}$, i = 1,2,3,... In fact, just look at the first two. By definition, they are independent if and only if

$$P(A_1 \text{ and } A_2) = P(A_1)P(A_2) \tag{5.1}$$

First, what is $P(A_1)$? **Now, wait a minute!** Don't answer, "Well, it depends on which coin we get," because this is NOT a conditional probability. Yes, the *conditional* probabilities $P(A_1|C_1)$ and $P(A_1|C_2)$ are 0.9 and 0.1, respectively, but the *unconditional* probability is $P(A_1) = 0.5$. You can deduce that either by the symmetry of the situation, or by

$$P(A_1) = P(C_1)P(A_1|C_1) + P(C_2)P(A_1|C_2) = (0.5)(0.9) + (0.5)(0.1) = 0.5 \tag{5.2}$$

You should think of all this in the notebook context. Each line of the notebook would consist of a report of three things: which coin we get; the outcome of the first toss; and the outcome of the second toss. (Note by the way that in our experiment we don't know which coin we get, but conceptually it should have a column in the notebook.) If we do this experiment for many, many lines in the notebook, about 90% of the lines in which the coin column says "1" will show Heads in the second column. But 50% of the lines *overall* will show Heads in that column.

So, the right hand side of Equation (5.1) is equal to 0.25. What about the left hand side?

$$
\begin{aligned}
P(A_1 \text{ and } A_2) &= P(A_1 \text{ and } A_2 \text{ and } C_1) + P(A_1 \text{ and } A_2 \text{ and } C_2) & (5.3)\\
&= P(A_1 \text{ and } A_2|C_1)P(C_1) + P(A_1 \text{ and} A_2|C_2)P(C_2) & (5.4)\\
&= (0.9)^2(0.5) + (0.1)^2(0.5) & (5.5)\\
&= 0.41 & (5.6)
\end{aligned}
$$

Well, 0.41 is not equal to 0.25, so you can see that the events are not independent, contrary to our first intuition. And that also means that X_i is not binomial.

5.1.2 Intuition in Retrospect

To get some intuition here, think about what would happen if we tossed the chosen coin 10000 times instead of just twice. If the tosses were independent, then for example knowledge of the first 9999 tosses should not tell us anything about the 10000th toss. But that is not the case at all. After 9999 tosses, we are going to have a very good idea as to which coin we had chosen, because by that time we will have gotten about 9000 heads (in the case of coin C_1) or about 1000 heads (in the case of C_2). In the former case, we know that the 10000th toss is likely to be a head, while

in the latter case it is likely to be tails. **In other words, earlier tosses do indeed give us information about later tosses, so the tosses aren't independent.**

5.1.3 Implications for Modeling

The lesson to be learned is that independence can definitely be a tricky thing, not to be assumed cavalierly. And in creating probability models of real systems, we must give very, very careful thought to the conditional and unconditional aspects of our models—-it can make a huge difference, as we saw above. Also, the conditional aspects often play a key role in formulating models of nonindependence.

This trick coin example is just that—tricky—but similar situations occur often in real life. If in some medical study, say, we sample people at random from the population, the people are independent of each other. But if we sample *families* from the population, and then look at children within the families, the children within a family are not independent of each other.

5.2 What About "Iterated Variance"?

In analogy to (4.68), one would think that

$$Var(V) = \sum_c P(U = c) \, Var(V \mid U = c) \tag{5.7}$$

In terms of the student heights example above, this wouldn't make sense, because it doesn't take into account the variation in means from one department to another. (This matter, because $Var(V) = E[(V - EV)^2]$.) So, it's not surprising that another term must be added to (5.7). This topic is discussed in Section **??**, but for now the point is that although good intuition is essential, we must not overrely on it..

5.3 Why Not Just Do All Analysis by Simulation?

Now that computer speeds are so fast, one might ask why we need to do mathematical probability analysis; why not just do everything by simulation? There are a number of reasons:

- Even with a fast computer, simulations of complex systems can take days, weeks or even months.

- Mathematical analysis can provide us with insights that may not be clear in simulation.

- Like all software, simulation programs are prone to bugs. The chance of having an uncaught bug in a simulation program is reduced by doing mathematical analysis for a special case of the system being simulated. This serves as a partial check.

- Statistical analysis is used in many professions, including engineering and computer science, and in order to conduct meaningful, <u>useful</u> statistical analysis, one needs a firm understanding of probability principles.

An example of that second point arose in the computer security research of a graduate student at UCD, Senthilkumar Cheetancheri, who was working on a way to more quickly detect the spread of a malicious computer worm. He was evaluating his proposed method by simulation, and found that things "hit a wall" at a certain point. He wasn't sure if this was a real limitation; maybe, for example, he just wasn't running his simulation on the right set of parameters to go beyond this limit. But a mathematical analysis showed that the limit was indeed real.

5.4   Reconciliation of Math and Intuition (optional section)

Here is a more theoretical definition of probability, as opposed to the intuitive "notebook" idea in this book. The definition is an abstraction of the notions of events (the sets A in $\mathcal{W}$ below) and probabilities of those events (the values of the function P(A)):

Definition 9 *Let S be a set, and let $\mathcal{W}$ be a collection of subsets of S. Let P be a real-valued function on $\mathcal{W}$. Then S, $\mathcal{W}$ and P form a* **probability space** *if the following conditions hold:*

- *P(S) = 1.*

- *$S \in \mathcal{W}$.*

- *$\mathcal{W}$ is closed under complements (if a set is in $\mathcal{W}$, then the set's complement with respect to S is in $\mathcal{W}$ too) and under unions of countably many members of $\mathcal{W}$.*

- *$P(A) \geq 0$ for any A in $\mathcal{W}$.*

- *If $A_1, A_2, ... \in \mathcal{W}$ and the A_i are pairwise disjoint, then*

$$P(\cup_i A_i) = \sum_i P(A_i) \tag{5.8}$$

A **random variable** *is any function $X : S \to \mathcal{R}$.*[1]

---

[1] The function must also have a property called **measurability**, which we will not discuss here.

Using just these simple axioms, one can prove (with lots of heavy math) theorems like the Strong Law of Large Numbers:

Theorem 10 *Consider a random variable U, and a sequence of independent random variables $U_1, U_2, ...$ which all have the same distribution as U, Then*

$$\lim_{n \to \infty} \frac{U_1 + ... + U_n}{n} = E(U) \ \text{with probability 1} \tag{5.9}$$

In other words, the average value of U in all the lines of the notebook will indeed converge to EU.

Chapter 6

Introduction to Discrete Markov Chains

(From this point onward, we will be making heavy use of linear algebra. The reader may find it helpful to review Appendix B.)

Here we introduce Markov chains, a topic covered in much more detail in Chapter **??**.

The basic idea is that we have random variables $X_1, X_2, ...$, with the index representing time. Each one can take on any value in a given set, called the **state space**; X_n is then the **state** of the system at time n. The state space is assumed either finite or **countably infinite**.[1]

We sometimes also consider an initial state, X_0, which might be modeled as either fixed or random. However, this seldom comes into play.

The key assumption is the **Markov property**, which in rough terms can be described as:

> The probabilities of future states, given the present state and the past states, depends only on the present state; the past is irrelevant.

In formal terms:

$$P(X_{t+1} = s_{t+1} | X_t = s_t, X_{t-1} = s_{t-1}, \ldots, X_0 = s_0) = P(X_{t+1} = s_{t+1} | X_t = s_t) \qquad (6.1)$$

---

[1]The latter is a mathematical term meaning, in essence, that it is possible to denote the space using integer subscripts. It can be shown that the set of all real numbers is not countably infinite, though perhaps surprisingly the set of all rational numbers *is* countably infinite.

Note that in (6.1), the two sides of the equation are equal but their common value may depend on t. We assume that this is not the case; nondependence on t is known as **stationarity**.[2] For instance, the probability of going from state 2 to state 5 at time 29 is assumed to be the same as the corresponding probability at time 333.

6.1   Matrix Formulation

We define p_{ij} to be the probability of going from state i to state j in one time step; note that this is a *conditional* probability, i.e. $P(X_{n+1} = j \mid X_n = i)$. These quantities form a matrix P,[3] whose row i, column j element is p_{ij}, which is called the **transition matrix**. Each row of P must sum to 1 (do you see why?).

For example, consider a three-state Markov chain with transition matrix

$$P = \begin{pmatrix} \frac{1}{2} & 0 & \frac{1}{2} \\ \frac{1}{4} & \frac{1}{2} & \frac{1}{4} \\ 1 & 0 & 0 \end{pmatrix} \tag{6.2}$$

This means that if we are now at state 1, the probabilities of going to states 1, 2 and 3 are 1/2, 0 and 1/2, respectively. Note that each row's probabilities must sum to 1—after all, from any particular state, we must go *somewhere*.

Actually, the mth power, P^m, of the transition matrix gives the probabilities for m-step transitions. In other words, the (i,j) element of P^m is $P(X_{t+m} = j|X_t = i)$. This is clear for the case m = 2 (after which one can use mathematical induction), as follows.

As usual, "break big events down into small events." How can it happen that $X_{t+2} = j$? Well, break things down according to where we might go first after leaving i. We might go from i to 1, say, then from 1 to j. So,

$$P(X_{t+2} = j|X_t = i) = \sum_k p_{ik} \ p_{kj} \tag{6.3}$$

In view of the rule for multiplying matrices, the expression on the right-hand side is simply the (i,j) element of P^2!

---

[2]Not to be confused with the notion of a stationary distribution, coming below.

[3]Unfortunately, we have some overloading of symbols here. Both in this book and in the field in general, we usually use the letter P to denote this matrix, yet we continue to denote probabilities by P( ). However, it will be clear from context which we mean. The same is true for our transition probabilities p_{ij}, which use a subscripted letter p, which is also the case for probably mass functions.

6.2 Example: Die Game

As our first example of Markov chains, consider the following game. One repeatedly rolls a die, keeping a running total. Each time the total exceeds 10 (not equals 10), we receive one dollar, and continue playing, resuming where we left off, mod 10. Say for instance we have a total of 8, then roll a 5. We receive a dollar, and now our total is 3.

It will simplify things if we assume that the player starts with one free point.

This process clearly satisfies the Markov property, with our state being our current total, 1, 2, ..., 10. If our current total is 6, for instance, then the probability that we next have a total of 9 is 1/6, *regardless of what happened our previous rolls.* We have p_{25}, p_{72} and so on all equal to 1/6, while for instance $p_{29} = 0$. Here's the code to find the transition matrix P:

```
1  # 10 states, so 10x10 matrix
2  # since most elements will be 0s, set them all to 0 first,
3  # then replace those that should be nonzero
4  p <- matrix(rep(0,100),nrow=10)
5  onesixth <- 1/6
6  for (i in 1:10) {  # look at each row
7     # since we are rolling a die, there are only 6 possible states we
8     # can go to from i, so check these
9     for (j in 1:6) {
10        k <- i + j  # new total, but did we win?
11        if (k > 10) k <- k - 10
12        p[i,k] <- onesixth
13     }
14  }
```

Note that since we knew that many entries in the matrix would be zero, it was easier just to make them all 0 first, and then fill in the nonzero ones.

6.3 Long-Run State Probabilities

Let N_{it} denote the number of times we have visited state i during times 1,...,t. For instance, in the die game, $N_{8,22}$ would be the number of rolls among the first 22 that resulted in our having a total of 8.

In typical applications we have that

$$\pi_i = \lim_{t \to \infty} \frac{N_{it}}{t} \tag{6.4}$$

exists for each state i, regardless of where we start.[4] Under a couple more conditions,[5] we have the stronger result,

$$\lim_{t \to \infty} P(X_t = i) = \pi_i \tag{6.5}$$

These quantities π_i are typically the focus of analyses of Markov chains.

We will use the symbol π to name the column vector of all the π_i:

$$\pi = (\pi_1, \pi_2, ...)' \tag{6.6}$$

where $'$ means matrix transpose.

6.3.1   Stationary Distribution

The π_i are called **stationary probabilities**, because if the initial state X_0 is a random variable with that distribution, then all X_i will have that distribution. Here's why:

Using (6.5), we have

$$\pi_i = \lim_{n \to \infty} P(X_n = i) \tag{6.7}$$

$$= \lim_{n \to \infty} \sum_k P(X_{n-1} = k) \, p_{ki} \tag{6.8}$$

$$= \sum_k \pi_k \, p_{ki} \tag{6.9}$$

---

[4]A more mathematically rigorous statement of (6.4) would include the qualifier "with probability 1." What does does mean?

The infinite sequeunce $X_1, X_2, ...$ is called a **sample path**. In our notebook, we have a column for each X_i, so one row is one sample path. So, what (**??**) says, with the phrase "with probability 1" added, is that the limit holds *for each row*.

[5]Basically, we need the chain to not be **periodic**. Consider a random walk, for instance: We start at position 0 on the number line, at time 0. The states are the integers. (So, this chain has an infinite state space.) At each time, we flip a coin to determine whether to move right (heads) or left (tails) 1 unit. A little thought shows that if we start at 0, the only times we can return to 0 are even-number times, i.e. $P(X_n = 0 \,|X_0 = 0)$ for all odd numbers n. This is a periodic chain. By the way, (6.4) turns out to be 0 for this chain.

So, if $P(X_0 = i) = \pi_i$, then

$$P(X_1 = i) = \sum_k P(X_0 = k) \, p_{ki} \tag{6.10}$$

$$= \sum_k \pi_k \, p_{ki} \tag{6.11}$$

$$= \pi_i \tag{6.12}$$

this last using (6.9). So, if X_0 has distribution π, then the same will be true for X_1, and contiuing in this manner we see that $X_2, X_3, \ldots$ will all have that distribution, thus demostrating the claimed stationary property for π. (This will be illustrated more concretely in Section 6.4.2.)

Of course, (6.7) holds for all states i. So in matrix terms, (6.7) says

$$\pi' = \pi'P \tag{6.13}$$

6.3.2 Calculation of π

Equation (6.13) then shows us how to find the π_i, at least in the case of finite state spaces, the subject of this section here, as follows.

First, rewrite (6.13)

$$(I - P')\pi = 0 \tag{6.14}$$

Here I is the $n \times n$ identity matrix (for a chain with n states), and again $'$ denotes matrix transpose.

This equation has infinitely many solutions; if π is a solution, then so for example is 8π. Moreover, the equation shows that the matrix there, $I - P'$, cannot have an inverse; if it did, we could multiply both sides by the inverse, and find that the unique solution is $\pi = 0$, which can't be right. This says in turn that the rows of $I - P'$ are not linearly independent.

The latter fact is quite important, for the following reason. Recall the close connection of matrix inverse and systems of linear equations. (6.14), a matrix equation, represents a system of n linear equations in n unknowns, the latter being $\pi_1, \pi_2, .., \pi_n$. So, the lack of linear independence of the rows of $I - P'$ means, in plain English, that at least one of those equations is redundant.

But we need n independent equations, and fortunately one is available:

$$\sum_i \pi_i = 1 \tag{6.15}$$

Note that (6.15) can be written as

$$O'\pi = 1 \tag{6.16}$$

where O is a vector of n 1s. Excellent, let's use it!

So, again, thinking of (6.14) as a system of linear equations, let's replace the last equation by (6.16). Switching back to the matrix view, that means that we replace the last row in the matrix $I - P'$ in (6.14) by O', and correspondingly replace the last element of the right-side vector by 1. Now we have a nonzero vector on the right side, and a full-rank (i.e. invertible) matrix on the left side. This is the basis for the following code, which we will use for finding π.

```
1  findpi1 <- function(p) {
2     n <- nrow(p)
3     # find I-P'
4     imp <- diag(n) - t(p)
5     # replace the last row of I-P' as discussed
6     imp[n,] <- rep(1,n)
7     # replace the corresponding element of the
8     # right side by (the scalar) 1
9     rhs <- c(rep(0,n-1),1)
10    # now use R's built-in solve()
11    solve(imp,rhs)
12 }
```

6.3.2.1 Example: π in Die Game

Consider the die game example above. Guess what! Applying the above code, all the π_i turn out to be 1/10. In retrospect, this should be obvious. If we were to draw the states 1 through 10 as a ring, with 1 following 10, it should be clear that all the states are completely symmetric.

6.3.2.2 Another Way to Find π

Here is another way to compute π. It is not commonly used, but **it will also help illustrate some of the concepts.**

Suppose (6.5) holds. Recall that P^m is the m-step transition matrix, so that for instance row 1 of that matrix is the set of probabilities of going from state 1 to the various states in m steps. The same will be true for row 2 and so on. Putting that together with (6.5), we have that

$$\lim_{n \to \infty} P^n = \Pi \tag{6.17}$$

where the $n \times n$ matrix Π has each of its rows equal to π'.

We can use this to find π. We take P to a large power m, and then each of the rows will approximate π. In fact, we can get an even better appoximation by averaging the rows.

Moreover, we can save a lot of computation by noting the following. Say we want the 16th power of P. We could set up a loop with 15 iterations, building up a product. But actually we can do it with just 4 iterations. We first square P, yielding P^2. But then we square *that*, yielding P^4. Square twice more, yielding P^8 and finally P^{16}. This is especially fast on a GPU (graphics processing unit).

```
# finds stationary probabilities of a Markov chain using matrix powers

altfindpi <- function(p,k) {
   niters <- ceiling(log2(k))
   prd <- p
   for (i in 1:niters) {
      prd <- prd %*% prd
   }
   colMeans(prd)
}
```

This approach has the advantage of being easy to parallelize, unlike matrix inversion.

6.4 Example: 3-Heads-in-a-Row Game

How about the following game? We keep tossing a coin until we get three consecutive heads. What is the expected value of the number of tosses we need?

We can model this as a Markov chain with states 0, 1, 2 and 3, where state i means that we have accumulated i consecutive heads so far. If we simply stop playing the game when we reach state 3, that state would be known as an **absorbing state**, one that we never leave.

We could proceed on this basis, but to keep things elementary, let's just model the game as being played repeatedly, as in the die game above. You'll see that that will still allow us to answer the

original question. Note that now that we are taking that approach, it will suffice to have just three states, 0, 1 and 2; there is no state 3, because as soon as we win, we immediately start a new game, in state 0.

6.4.1   Markov Analysis

Clearly we have transition probabilities such as p_{01}, p_{12}, p_{10} and so on all equal to 1/2. Note from state 2 we can only go to state 0, so $p_{20} = 1$.

Here's the code below. Of course, since R subscripts start at 1 instead of 0, we must recode our states as 1, 2 an 3.

```
p <- matrix(rep(0,9),nrow=3)
p[1,1] <- 0.5
p[1,2] <- 0.5
p[2,3] <- 0.5
p[2,1] <- 0.5
p[3,1] <- 1
findpi1(p)
```

It turns out that

$$\pi = (0.5714286, 0.2857143, 0.1428571) \tag{6.18}$$

So, in the long run, about 57.1% of our tosses will be done while in state 0, 28.6% while in state 1, and 14.3% in state 2.

Now, look at that latter figure. Of the tosses we do while in state 2, half will be heads, so half will be wins. In other words, about 0.071 of our tosses will be wins. And THAT figure answers our original question, through the following reasoning:

Think of, say, 10000 tosses. There will be about 710 wins sprinkled among those 10000 tosses. Thus the average number of tosses between wins will be about $10000/710 = 14.1$. In other words, the expected time until we get three consecutive heads is about 14.1 tosses.

6.4.2   Back to the word "Stationary"

Let's look at the term *stationary distribution* in the context of this game.

At time 0, we haven't done any coin flips, so $X_0 = 0$. But consider a modified version of the game, in which at time 0 you are given "credit" or a "head start. For instance, you might be allowed to

start in state 1, meaning that you already have one consecutive head, even though you actually haven't done any flips.

Now, suppose your head start is random, so that you start in state 0, 1 or 2 with certain probabilities. And suppose those probabilities are given by π! In other words, you start in state 0, 1 or 2, with probabilities 0.57, 0.29 and 0.14, as in (6.18). What, then, will be the distribution of X_1?

$$p_{X_1}(i) = P(X_1 = i) \tag{6.19}$$

$$= \sum_{j=0}^{2} P(X_0 = j) \, P(X_1 = i \mid X_0 = j) \tag{6.20}$$

$$= \sum_{j=0}^{2} \pi_j \, p(j, i) \tag{6.21}$$

This is exactly what we saw in (6.12) and the equations leading up to it. So we know that (6.21) will work out to π_i. In other words, $P(X_1 = 0) = 0.57$ and so on. (Work it out numerically if you are in doubt.) So, we see that

> If X_0 has distribution π, then the same will be true for X_1. We say that the distribution of the chain is *stationary*.

6.5 A Modified Notebook Analysis

Our previous notebook analysis (and most of our future ones, other than for Markov chains), relied on imagining performing many independent replications of the same experiment.

6.5.1 A Markov-Chain Notebook

Consider Table 2.3, for instance. There our experiment was to watch the network during epochs 1 and 2. So, on the first line of the notebook, we would watch the network during epochs 1 and 2 and record the result. On the second line, we watch a new, independent replication of the network during epochs 1 and 2, and record the results.

But instead of imagining a notebook recording infinitely many replications of the two epochs, we could imagine watching just *one* replication but over infinitely many epochs. We'd watch the network during epoch 1, epoch 2, epoch 3 and so on. Now one line of the notebook would record one epoch.

For general Markov chains, each line would record one time step. We would have columns of the notebook labeled n and X_n. The reason this approach would be natural is (6.4). In that context, π_i would be the long-run proportion of notebook lines in which $X_n = i$.

6.5.2   Example: 3-Heads-in-a-Row Game

For instance, consider the 3-heads-in-a-row game. Then (6.18) says that about 57% of the notebook lines would have a 0 in the X_n column, with about 29% and 14% of the lines showing 1 and 2, respectively.

Moreover, suppose we also have a notebook column labeled *Win*, with an entry Yes in a certain line meaning, yes, that coin flip resulted in a win, with a No entry meaning no win. Then the mean time until a win, which we found to be 14.1 above, would be described in notebook terms as the long-run average number of lines between Yes entries in the *Win* column.

6.6   Simulation of Markov Chains

Following up on Section 6.5, recall that our previous simulations have basically put into code form our notebook concept. Our simulations up until now have been based on the definition of probability, which we had way back in Section 2.3. Our simulation code modeled the notebook independent replications notion. We can do a similar thing now, based on the ideas in Section 6.5.

In a time series kind of situation such as Markov chains, since we are interested in long-run behavior in the sense of time, our simulation is based on (6.5). In other words, we simulate the evolution of X_n as n increases, and take long-run averages of whatever is of interest.

Here is simulation code for the example in Section 6.4, caculating the approximate value of the long-run time between wins (found to be about 14.1 by mathematical means above):

```
# simulation of 3-in-a-row coin toss gaem

threeinrow <- function(ntimesteps) {
   consec <- 0   # number of consec Hs
   nwins <- 0   # number of wins
   wintimes <- 0   # total of times to win
   # startplay will be the time at which we started the current game
   startplay <- 0   # time step 0
   for (i in 1:ntimesteps) {
      if (toss() == 'H') {
         consec <- consec + 1
```

```
        if (consec == 3) {
            nwins <- nwins + 1
            wintimes <-
                wintimes + i - startplay
            consec <- 0
            startplay <- i
        }
    } else consec <- 0
    }
    wintimes / nwins
}

toss <- function()
    if (runif(1) < 0.5) 'H' else 'T'

l
```

6.7 Example: ALOHA

Consider our old friend, the ALOHA network model. (You may wish to review the statement of the model in Section 2.5 before continuing.) The key point in that system is that it was "memoryless," in that the probability of what happens at time k+1 depends only on the state of the system at time k.

For instance, consider what might happen at time 6 if $X_5 = 2$. Recall that the latter means that at the end of epoch 5, both of our two network nodes were active. The possibilities for X_6 are then

- X_6 will be 2 again, with probability $p^2 + (1-p)^2$

- X_6 will be 1, with probability $2p(1-p)$

The central point here is that the past history of the system—i.e. the values of $X_1, X_2, X_3,$ and X_4—don't have any impact. We can state that precisely:

The quantity

$$P(X_6 = j | X_1 = i_1, X_2 = i_2, X_3 = i_3, X_4 = i_4, X_5 = i) \tag{6.22}$$

does not depend on $i_m, m = 1, ..., 4$. Thus we can write (6.22) simply as $P(X_6 = j | X_5 = i)$.

Furthermore, that probability is the same as $P(X_9 = j | X_8 = i)$ and in general $P(X_{k+1} = j | X_k = i)$. So, we do have a Markov chain.

Since this is a three-state chain, the p_{ij} form a 3x3 matrix:

$$P = \begin{pmatrix} (1-q)^2 + 2q(1-q)p & 2q(1-q)(1-p) + 2q^2p(1-p) & q^2[p^2 + (1-p)^2] \\ (1-q)p & 2qp(1-p) + (1-q)(1-p) & q[p^2 + (1-p)^2] \\ 0 & 2p(1-p) & p^2 + (1-p)^2 \end{pmatrix} \tag{6.23}$$

For instance, the element in row 0, column 2, p_{02}, is $q^2[p^2 + (1-p)^2]$, reflecting the fact that to go from state 0 to state 2 would require that both inactive nodes become active (which has probability q^2, and then either both try to send or both refrain from sending (probability $p^2 + (1-p)^2$.

For the ALOHA example here, with p = 0.4 and q = 0.3, the solution is $\pi_0 = 0.47$, $\pi_1 = 0.43$ and $\pi_2 = 0.10$.

So we know that in the long run, about 47% of the epochs will have no active nodes, 43% will have one, and 10% will have two. From this we see that the long-run average number of active nodes is

$$0 \cdot 0.47 + 1 \cdot 0.43 + 2 \cdot 0.10 = 0.63 \tag{6.24}$$

By the way, note that every row in a transition matrix must sum to 1. (The probability that we go from state i to *somewhere* is 1, after all, so we must have $\sum_j p_{ij} = 1$.) That implies that we can save some work in writing R code; the last column must be 1 minus the others. In our example above, we would write

```
transmat <- matrix(rep(0,9),nrow=3)
p1 <- 1 - p
q1 <- 1 - q
transmat[1,1] <- q1^2 + 2 * q * q1 *p
transmat[1,2] <- 2 * q * q1 * p1 + 2 * q^2 * p * p1
transmat[2,1] <- q1 * p
transmat[2,2] <- 2 * q * p * p1 + q1 * p1
transmat[3,1] <- 0
transmat[3,2] <- 2 * p * p1
transmat[,3] <- 1 - transmat[,1] - transmat[,2]
findpi1(transmat)
```

Note the vectorized addition and recycling (Section 2.14.2).

6.8 Example: Bus Ridership Problem

Consider the bus ridership problem in Section 2.11. Make the same assumptions now, but add a new one: There is a maximum capacity of 20 passengers on the bus.

The random variables L_i, i = 1,2,3,... form a Markov chain. Let's look at some of the transition probabilities:

$$p_{00} = 0.5 \tag{6.25}$$

$$p_{01} = 0.4 \tag{6.26}$$

$$p_{11} = (1 - 0.2) \cdot 0.5 + 0.2 \cdot 0.4 \tag{6.27}$$

$$p_{20} = (0.2)^2(0.5) = 0.02 \tag{6.28}$$

$$p_{20,20} = (0.8)^{20}(0.5 + 0.4 + 0.1) + \binom{20}{1}(0.2)^1(0.8)^{20-1}(0.4 + 0.1) + \binom{20}{2}(0.2)^2(0.8)^{18}(0.1) \tag{6.29}$$

(Note that for clarity, there is a comma in $p_{20,20}$, as p_{2020} would be confusing and in some other examples even ambiguous. A comma is not necessary in p_{11}, since there must be two subscripts; the 11 here can't be eleven.)

After finding the π vector as above, we can find quantities such as the long-run average number of passengers on the bus,

$$\sum_{i=0}^{20} \pi_i i \tag{6.30}$$

We can also compute the long-run average number of would-be passengers who fail to board the bus. Denote by A_i denote the number of passengers on the bus as it *arrives* at stop i. The key point is that since $A_i = L_{i-1}$, then (6.4) and (6.5) will give the same result, no matter whether we look at the L_j chain or the A_j chain.

Now, armed with that knowledge, let D_j denote the number of disappointed people at stop j. Then

$$ED_j = 1 \cdot P(D_j = 1) + 2 \cdot P(D_j = 2). \tag{6.31}$$

That latter probability, for instance, is

$$
\begin{aligned}
P(D_j = 2) &= P(A_j = 20 \text{ and } B_j = 2 \text{ and } G_j = 0) & (6.32)\\
&= P(A_j = 20) \; P(B_j = 2) \; P(G_j = 0 \mid A_j = 20) & (6.33)\\
&= P(A_j = 20) \cdot 0.1 \cdot 0.8^{20} & (6.34)
\end{aligned}
$$

where as before B_j is the number who wish to board at stop j, and G_j is the number who get off the bus at that stop. Using the same reasoning, one can find $P(D_j = 1)$. (A number of cases to consider, left as an exercise for the reader.)

Taking the limits as $j \to \infty$, we have the long-run average number of disappointed customers on the left, and on the right, the term $P(A_j = 20)$ goes to π_{20}, which we have and thus can obtain the value regarding disappointed customers.

Let's find the long-run average number of customers who alight from the bus. This can be done by considering all the various cases, but (4.68) really shortens our work:

$$EG_n = \sum_{i=0}^{20} P(A_n = i) E(G_n | A_n = i) \tag{6.35}$$

Given $A_n = i$, G_n has a binomial distribution with i trials and success probability 0.2, so

$$E(G_n | A_n = i) = i \cdot 0.2 \tag{6.36}$$

So, the right-hand side of 6.35 converges to

$$\sum_{i=0}^{20} \pi_i i \cdot 0.2 \tag{6.37}$$

In other words, the long-run average number alighting is 0.2 times (6.30).

6.9 Example: an Inventory Model

Consider the following simple inventory model. A store has 1 or 2 customers for a certain item each day, with probabilities v and w (v+w = 1). Each customer is allowed to buy only 1 item.

When the stock on hand reaches 0 on a day, it is replenished to r items immediately after the store closes that day.

If at the start of a day the stock is only 1 item and 2 customers wish to buy the item, only one customer will complete the purchase, and the other customer will leave emptyhanded.

Let X_n be the stock on hand at the end of day n (*after* replenishment, if any). Then $X_1, X_2, ...$ form a Markov chain, with state space 1,2,...,r.

The transition probabilities are easy to find. Take p_{21}, for instance. If there is a stock of 2 items at the end of one day, what is the (conditional) probability that there is only 1 item at the end of the next day? Well, for this to happen, there would have to be just 1 customer coming in, not 2, and that has probability v. So, $p_{21} = v$. The same reasoning shows that $p_{2r} = w$.

Let's write a function **inventory(v,w,r)** that returns the π vector for this Markov chain. It will call **findpi1()**, similarly to the two code snippets on page 110. For convenience, let's assume **r** is at least 3.[6]

```
1  inventory <- function(v,w,r) {
2     tm <- matrix(rep(0,r^2),nrow=r)
3     for (i in 3:r) {
4        tm[i,i-1] <- v
5        tm[i,i-2] <- w
6     }
7     tm[2,1] <- v
8     tm[2,r] <- w
9     tm[1,r] <- 1
10    findpi1(tm)
11 }
```

6.10 Expected Hitting Times

Consider an *n*-state Markov chain that is *irreducible*, meaning that it is possible to get from any state i to any other state j in some number of steps. Define the random variable T_{ij} to be the time

---

[6]If **r** is 2, then the expression 3:2 in the code evaluates to the vector (3,2), which would not be what we want in this case.

needed to go from state i to state j. (Note that T_{ii} is NOT 0, though it can be 1 if $p_{ii} > 0$.)

In many applications, we are interested in the mean time to reach a certain state, given that we start at some specified state. In other words, we wish to calculate ET_{ij}.

The material in Section 4.15 is useful here. In (4.68), take $V = T_{ij}$ and take U to be the first state we visit after leaving state i (which could be i again). Then

$$E(T_{ij}) = \sum_k p_{ik} E(T_{ij} \mid U = k), \quad 1 \le i, j \le n \tag{6.38}$$

Consider what happens if $U = k \ne j$. Then we will have already used up 1 time step, and *still* will need an average of ET_{kj} more steps to get to j. In other words,

$$E(T_{ij} \mid U = k \ne j) = 1 + ET_{kj} \tag{6.39}$$

Did you notice the Markov property being invoked?

On the other hand, the case $U = j$ is simple:

$$E(T_{ij} \mid U = j) = 1 \tag{6.40}$$

This then implies that

$$E(T_{ij}) = 1 + \sum_{k \ne j} p_{ik} E(T_{kj}), \quad 1 \le i, j \le n \tag{6.41}$$

We'll focus on the case $j = n$, i.e. look at how long it takes to get to state n. (We could of course do the same analysis for any other destination state.) Let η_i denote $E(T_{in})$, and define $\eta = (\eta_1, \eta_2, ..., \eta_{n-1})'$. (Note that η has only $n - 1$ components!) So,

$$\eta_i = 1 + \sum_{k < n} p_{ik} \eta_k, \quad 1 \le i, j \le n \tag{6.42}$$

Equation (6.42) is a system of linear equations, which we can write in matrix form. Then the code to solve the system is (remember, we have only an $(n - 1) \times (n - 1)$ system)

```
findeta <- function(p) {
   n <- nrow(p)
   q <- diag(n) - p
   q <- q[1:(n-1),1:(n-1)]
```

```
    ones <- rep(1,n-1)
    solve(q,ones)
}
```

Chapter 7

Continuous Probability Models

There are other types of random variables besides the discrete ones you studied in Chapter 3. This chapter will cover another major class, *continuous random variables*, which form the heart of statistics and are used extensively in applied probability as well. It is for such random variables that the calculus prerequisite for this book is needed.

7.1 Running Example: a Random Dart

Imagine that we throw a dart at random at the interval (0,1). Let D denote the spot we hit. By "at random" we mean that all subintervals of equal length are equally likely to get hit. For instance, the probability of the dart landing in (0.7,0.8) is the same as for (0.2,0.3), (0.537,0.637) and so on.

Because of that randomness,

$$P(u \leq D \leq v) = v - u \tag{7.1}$$

for any case of $0 \leq u < v \leq 1$.

We call D a **continuous** random variable, because its support is a continuum of points, in this case, the entire interval (0,1).

7.2   Individual Values Now Have Probability Zero

The first crucial point to note is that

$$P(D = c) = 0 \qquad\qquad (7.2)$$

for any individual point c. This may seem counterintuitive, but it can be seen in a couple of ways:

- Take for example the case c = 0.3. Then

$$P(D = 0.3) \leq P(0.29 \leq D \leq 0.31) = 0.02 \qquad\qquad (7.3)$$

  the last equality coming from (7.1).

  So, $P(D = 0.3) \leq 0.02$. But we can replace 0.29 and 0.31 in (7.3) by 0.299 and 0.301, say, and get $P(D = 0.3) \leq 0.002$. So, $P(D = 0.3)$ must be smaller than any positive number, and thus it's actually 0.

- Reason that there are infinitely many points, and if they all had some nonzero probability w, say, then the probabilities would sum to infinity instead of to 1; thus they must have probability 0.

Similarly, one will see that (7.2) will hold for any continuous random variable.

Remember, we have been looking at probability as being the long-run fraction of the time an event occurs, in infinitely many repetitions of our experiment—the "notebook" view. So (7.2) doesn't say that D = c can't occur; it merely says that it happens so rarely that the long-run fraction of occurrence is 0.

7.3   But Now We Have a Problem

But Equation (7.2) presents a problem. In the case of discrete random variables M, we defined their distribution via their probability mass function, p_M. Recall that Section 3.13 defined this as a list of the values M takes on, together with their probabilities. But that would be impossible in the continuous case—all the probabilities of individual values here are 0.

So our goal will be to develop another kind of function, which is similar to probability mass functions in spirit, but circumvents the problem of individual values having probability 0. To do this, we first must define another key function:

7.3.1   Our Way Out of the Problem: Cumulative Distribution Functions

Definition 11 *For any random variable W (including discrete ones), its* **cumulative distribution function** *(cdf), F_W, is defined by*

$$F_W(t) = P(W \leq t), -\infty < t < \infty \tag{7.4}$$

(Please keep in mind the notation. It is customary to use capital F to denote a cdf, with a subscript consisting of the name of the random variable.)

What is t here? It's simply an argument to a function. The function here has domain $(-\infty, \infty)$, and we must thus define that function for every value of t. This is a simple point, but a crucial one.

For an example of a cdf, consider our "random dart" example above. We know that, for example for t = 0.23,

$$F_D(0.23) = P(D \leq 0.23) = P(0 \leq D \leq 0.23) = 0.23 \tag{7.5}$$

Also,

$$F_D(-10.23) = P(D \leq -10.23) = 0 \tag{7.6}$$

and

$$F_D(10.23) = P(D \leq 10.23) = 1 \tag{7.7}$$

Note that *the fact that D can never be equal to 10.23 or anywhere near it is irrelevant. $F_D(t)$ is defined for all* t in $(-\infty, \infty)$, including 10.23! The definition of $F_D(10.23)$ is $P(D \leq 10.23)$), and that probability is 1! Yes, D is *always* less than or equal to 10.23, right?

In general for our dart,

$$F_D(t) = \begin{cases} 0, & \text{if } t \leq 0 \\ t, & \text{if } 0 < t < 1 \\ 1, & \text{if } t \geq 1 \end{cases} \tag{7.8}$$

Here is the graph of F_D:
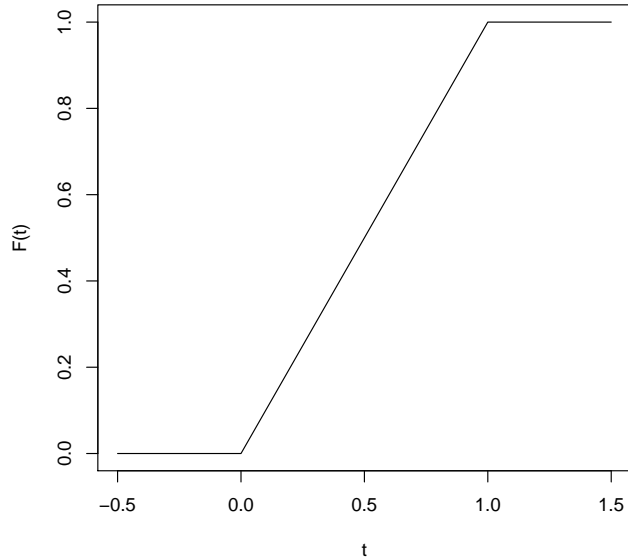
The cdf of a discrete random variable is defined as in Equation (7.4) too. For example, say Z is the number of heads we get from two tosses of a coin. Then

$$
F_Z(t) = \begin{cases} 0, & \text{if } t < 0 \\ 0.25, & \text{if } 0 \le t < 1 \\ 0.75, & \text{if } 1 \le t < 2 \\ 1, & \text{if } t \ge 2 \end{cases} \tag{7.9}
$$

For instance,

$$
\begin{aligned}
F_Z(1.2) &= P(Z \le 1.2) & (7.10) \\
&= P(Z = 0 \text{ or } Z = 1) & (7.11) \\
&= 0.25 + 0.50 & (7.12) \\
&= 0.75 & (7.13)
\end{aligned}
$$

Note that (7.11) is simply a matter of asking our famous question, "How can it happen?" Here we are asking how it can happen that $Z \le 1.2$. The answer is simple: That can happen if Z is 0 or 1. *The fact that Z cannot equal 1.2 is irrelevant.*

(7.12) uses the fact that Z has a binomial distribution with n = 2 and p = 0.5.

F_Z is graphed below.
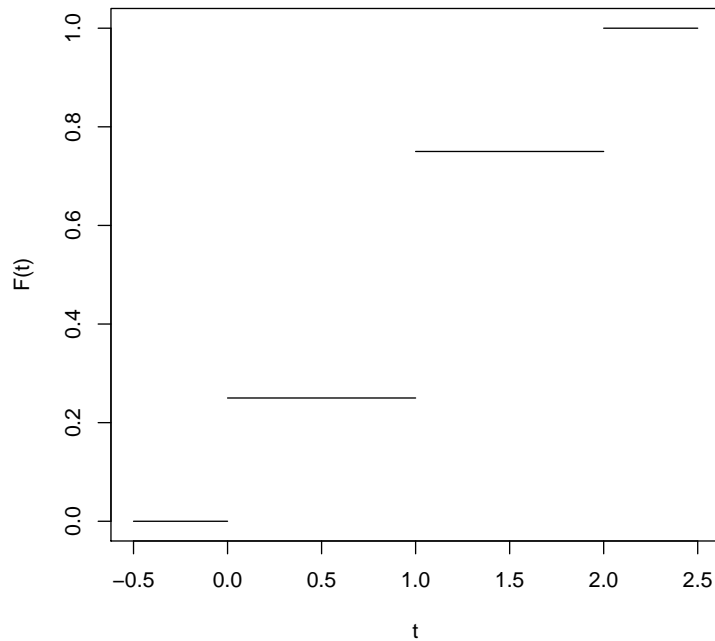
The fact that one cannot get a noninteger number of heads is what makes the cdf of Z flat between consecutive integers.

In the graphs you see that F_D in (7.8) is continuous while F_Z in (7.9) has jumps. This is another reason we call random variables such as D **continuous random variables**.

At this level of study of probability, random variables are either discrete or continuous. But some exist that are neither. We won't see any random variables from the "neither" case here, and they occur rather rarely in practice.

Armed with cdfs, let's turn to the original goal, which was to find something for continuous random variables that is similar in spirit to probability mass functions for discrete random variables.

7.3.2 Density Functions

Intuition is key here. Make SURE you develop a good intuitive understanding of density functions, as it is vital in being able to apply probability well. We will use it a lot in our course.

(The reader may wish to review pmfs in Section 3.13.)

Think as follows. From (7.4) we can see that for a discrete random variable, its cdf can be calculated by summing its pmf. Recall that in the continuous world, we integrate instead of sum. So, our continuous-case analog of the pmf should be something that integrates to the cdf. That of course is the derivative of the cdf, which is called the **density**:

Definition 12 *Consider a continuous random variable W. Define*

$$f_W(t) = \frac{d}{dt} F_W(t), -\infty < t < \infty \tag{7.14}$$

*wherever the derivative exists. The function f_W is called the **probability density function** (pdf), or just the **density** of W.*

(Please keep in mind the notation. It is customary to use lower-case f to denote a density, with a subscript consisting of the name of the random variable.)

But what *is* a density function? First and foremost, it is a tool for finding probabilities involving continuous random variables:

7.3.3   Properties of Densities

Equation (7.14) implies

Property A:

$$
\begin{aligned}
P(a < W \leq b) &= F_W(b) - F_W(a) \tag{7.15} \\
&= \int_a^b f_W(t) \, dt \tag{7.16}
\end{aligned}
$$

Where does (7.15) come from? Well, $F_W(b)$ is all the probability accumulated from $-\infty$ to b, while $F_W(a)$ is all the probability accumulated from $-\infty$ to a. The difference is the probability that X is *between* a and b.

(7.16) is just the Fundamental Theorem of Calculus: Integrate the derivative of a function, and you get the original function back again.

Since $P(W = c) = 0$ for any single point c, Property A also means:

Property B:

$$P(a < W \leq b) = P(a \leq W \leq b) = P(a \leq W < b) = P(a < W < b) = \int_a^b f_W(t) \; dt \qquad (7.17)$$

This in turn implies:

Property C:

$$\int_{-\infty}^{\infty} f_W(t) \; dt = 1 \qquad (7.18)$$

Note that in the above integral, $f_W(t)$ will be 0 in various ranges of t corresponding to values W cannot take on. For the dart example, for instance, this will be the case for $t < 0$ and $t > 1$.

Any nonnegative function that integrates to 1 is a density. A density could be increasing, decreasing or mixed. Note too that a density can have values larger than 1 at some points, even though it must integrate to 1.

7.3.4   Intuitive Meaning of Densities

Suppose we have some continuous random variable X, with density f_X, graphed in Figure 7.1.

Let's think about probabilities of the form

$$P(s - 0.1 < X < s + 0.1) \qquad (7.19)$$

Let's first consider the case of s = 1.3.

The rectangular strip in the picture should remind you of your early days in calculus. What the picture says is that the area under f_X from 1.2 to 1.4 (i.e. 1.3 ± 0.1) is approximately equal to the area of the rectangle. In other words,

$$2(0.1)f_X(1.3) \approx \int_{1.2}^{1.4} f_X(t) \; dt \qquad (7.20)$$

But from our Properties above, we can write this as

$$P(1.2 < X < 1.4) \approx 2(0.1)f_X(1.3) \qquad (7.21)$$

Similarly, for s = 0.4,

$$P(0.3 < X < 0.5) \approx 2(0.1)f_X(0.4) \tag{7.22}$$

and in general

$$P(s - 0.1 < X < s + 0.1) \approx 2(0.1)f_X(s) \tag{7.23}$$

This reasoning shows that:

> Regions in the number line (X-axis in the picture) with low density have low probabilities while regions with high density have high probabilities.

So, although densities themselves are not probabilities, they do tell us which regions will occur often or rarely. For the random variable X in our picture, there will be many lines in the notebook in which X is near 1.3, but many fewer in which X is near 0.4.

7.3.5   Expected Values

What about E(W)? Recall that if W were discrete, we'd have

$$E(W) = \sum_c cp_W(c) \tag{7.24}$$

where the sum ranges overall all values c that W can take on. If for example W is the number of dots we get in rolling two dice, c will range over the values 2,3,...,12.

So, the analog for continuous W is:

Property D:

$$E(W) = \int_t tf_W(t) \ dt \tag{7.25}$$

where here t ranges over the values W can take on, such as the interval (0,1) in the dart case. Again, we can also write this as

$$E(W) = \int_{-\infty}^{\infty} tf_W(t) \ dt \tag{7.26}$$
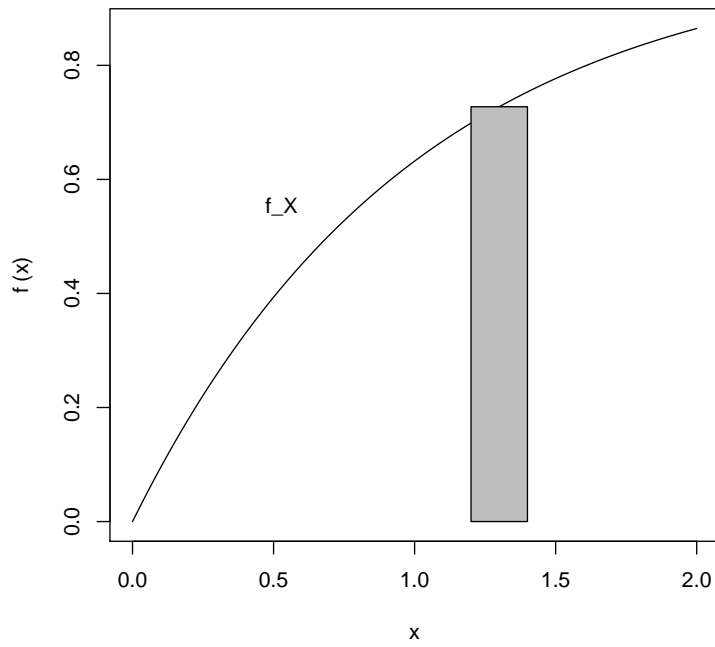
Figure 7.1: Approximation of Probability by a Rectangle

in view of the previous comment that $f_W(t)$ might be 0 for various ranges of t.

And of course,

$$E(W^2) = \int_t t^2 f_W(t) \; dt \tag{7.27}$$

and in general, similarly to (3.36):

Property E:

$$E[g(W)] = \int_t g(t) f_W(t) \; dt \tag{7.28}$$

Most of the properties of expected value and variance stated previously for discrete random variables hold for continuous ones too:

Property F:

Equations (3.19), (3.21), (3.25), (3.41) and (3.49) still hold in the continuous case.

7.4   A First Example

Consider the density function equal to 2t/15 on the interval (1,4), 0 elsewhere. Say X has this density. Here are some computations we can do:

$$EX = \int_1^4 t \cdot 2t/15 \; dt = 2.8 \tag{7.29}$$

$$P(X > 2.5) = \int_{2.5}^4 2t/15 \; dt = 0.65 \tag{7.30}$$

$$F_X(s) = \int_1^s 2t/15 \; dt = \frac{s^2 - 1}{15} \quad \text{for s in (1,4) (cdf is 0 for t < 1, and 1 for t > 4)} \tag{7.31}$$

$$
\begin{aligned}
Var(X) &= E(X^2) - (EX)^2 \quad \text{(from (3.41))} \tag{7.32}\\
&= \int_1^4 t^2 2t/15 \; dt - 2.8^2 \quad \text{(from (7.29))} \tag{7.33}\\
&= 0.66 \tag{7.34}
\end{aligned}
$$

Suppose L is the lifetime of a light bulb (say in years), with the density that X has above. Let's find some quantities in that context:

Proportion of bulbs with lifetime less than the mean lifetime:

$$P(L < 2.8) = \int_1^{2.8} 2t/15 \ dt = (2.8^2 - 1)/15 \tag{7.35}$$

Mean of 1/L:

$$E(1/L) = \int_1^4 \frac{1}{t} \cdot 2t/15 \ dt = \frac{2}{5} \tag{7.36}$$

In testing many bulbs, mean number of bulbs that it takes to find two that have lifetimes longer than 2.5:

Use (4.38) with r = 2 and p = 0.65.

7.5 The Notion of *Support* in the Continuous Case

Recall from Section 3.2 that the *support* of a discrete distribution is its "domain." If for instance X is the number of heads I get from 3 tosses of a coin, X can only take on the values 0, 1, 2 and 3. We say that that set is the support of this distribution; 8, for example, is not in the support.

The notion extends to continuous random variables. In Section 7.4, the support of the density there is the interval (1,4).

7.6 Famous Parametric Families of Continuous Distributions

7.6.1 The Uniform Distributions

7.6.1.1 Density and Properties

In our dart example, we can imagine throwing the dart at the interval (q,r) (so this will be a two-parameter family). Then to be a uniform distribution, i.e. with all the points being "equally likely," the density must be constant in that interval. But it also must integrate to 1 [see (7.18).

So, that constant must be 1 divided by the length of the interval:

$$f_D(t) = \frac{1}{r - q} \tag{7.37}$$

for t in (q,r), 0 elsewhere.

It easily shown that $E(D) = \frac{q+r}{2}$ and $Var(D) = \frac{1}{12}(r - q)^2$.

The notation for this family is U(q,r).

7.6.1.2   R Functions

Relevant functions for a uniformly distributed random variable X on (r,s) are:

- **dunif(x,r,s)**, to find $f_X(x)$

- **punif(q,r,s)**, to find $P(X \leq q)$

- **qunif(q,r,s)**, to find c such that $P(X \leq c) = q$

- **runif(n,r,s)**, to generate n independent values of X

As with most such distribution-related functions in R, **x** and **q** can be vectors, so that **punif()** for instance can be used to find the cdf values at multiple points.

7.6.1.3   Example: Modeling of Disk Performance

Uniform distributions are often used to model computer disk requests. Recall that a disk consists of a large number of concentric rings, called **tracks**. When a program issues a request to read or write a file, the **read/write head** must be positioned above the track of the first part of the file. This move, which is called a **seek**, can be a significant factor in disk performance in large systems, e.g. a database for a bank.

If the number of tracks is large, the position of the read/write head, which I'll denote as X, is like a continuous random variable, and often this position is modeled by a uniform distribution. This situation may hold just before a defragmentation operation. After that operation, the files tend to be bunched together in the central tracks of the disk, so as to reduce seek time, and X will not have a uniform distribution anymore.

Each track consists of a certain number of **sectors** of a given size, say 512 bytes each. Once the read/write head reaches the proper track, we must wait for the desired sector to rotate around and

pass under the read/write head. It should be clear that a uniform distribution is a good model for this **rotational delay**.

For example, suppose in modeling disk performance, we describe the position X of the read/write head as a number between 0 and 1, representing the innermost and outermost tracks, respectively. Say we assume X has a uniform distribution on (0,1), as discussed above. Consider two consecutive positions (i.e. due to two consecutive seeks), X_1 and X_2, which we'll assume are independent.[1] Let's find $Var(X_1 + X_2)$.

We know from Section 7.6.1.1 that the variance of a U(0,1) distribution is 1/12. Then by independence,

$$Var(X_1 + X_2) = 1/12 + 1/12 = 1/6 \tag{7.38}$$

7.6.1.4   Example: Modeling of Denial-of-Service Attack

In one facet of computer security, it has been found that a uniform distribution is actually a warning of trouble, a possible indication of a **denial-of-service attack**. Here the attacker tries to monopolize, say, a Web server, by inundating it with service requests. According to the research of David Marchette,[2] attackers choose uniformly distributed false IP addresses, a pattern not normally seen at servers.

7.6.2   The Normal (Gaussian) Family of Continuous Distributions

These are the famous "bell-shaped curves," so called because their densities have that shape.[3]

7.6.2.1   Density and Properties

Density and Parameters:

---

[1]NOT a good assumption for consecutive seeks; think about why.

[2]*Statistical Methods for Network and Computer Security*, David J. Marchette, Naval Surface Warfare Center, `rion.math.iastate.edu/IA/2003/foils/marchette.pdf`.

[3]*All that glitters is not gold"*—Shakespeare

Note that other parametric families, notably the Cauchy, also have bell shapes. The difference lies in the rate at which the tails of the distribution go to 0. However, due to the Central Limit Theorem, to be presented below, the normal family is of prime interest.

The density for a normal distribution is

$$f_W(t) = \frac{1}{\sqrt{2\pi}\sigma} \, e^{-0.5\left(\frac{t-\mu}{\sigma}\right)^2}, -\infty < t < \infty \tag{7.39}$$

Again, this is a two-parameter family, indexed by the parameters μ and σ, which turn out to be the mean[4] and standard deviation μ and σ, The notation for it is $N(\mu, \sigma^2)$ (it is customary to state the variance σ^2 rather than the standard deviation).

The normal family is so important that we have a special chapter on it, Chapter 8.

7.6.3 The Exponential Family of Distributions

Please note: We have been talking here of parametric families of distributions, and in this section will introduce one of the most famous, the family of exponential distributions. This should not be confused, though, with the term *exponential family* that arises in mathematical statistics, which includes exponential distributions but is much broader.

7.6.3.1 Density and Properties

The densities in this family have the form

$$f_W(t) = \lambda e^{-\lambda t}, 0 < t < \infty \tag{7.40}$$

This is a one-parameter family of distributions.

After integration, one finds that $E(W) = \frac{1}{\lambda}$ and $Var(W) = \frac{1}{\lambda^2}$. You might wonder why it is customary to index the family via λ rather than $1/\lambda$ (see (7.40)), since the latter is the mean. But this is actually quite natural, for reasons discussed in Section 9.1.

7.6.3.2 R Functions

Relevant functions for a uniformly distributed random variable X with parameter λ are

- **dexp(x,lambda)**, to find $f_X(x)$

- **pexp(q,lambda)**, to find $P(X \leq q)$

---

[4]Remember, this is a synonym for expected value.

- **qexp(q,lambda)**, to find c such that $P(X \leq c) = q$

- **rexp(n,lambda)**, to generate n independent values of X

7.6.3.3   Example: Refunds on Failed Components

Suppose a manufacturer of some electronic component finds that its lifetime L is exponentially distributed with mean 10000 hours. They give a refund if the item fails before 500 hours. Let M be the number of items they have sold, up to and including the one on which they make the first refund. Let's find EM and $Var(M)$.

First, notice that M has a geometric distribution! It is the number of independent trials until the first success, where a "trial" is one component, "success" (no value judgment, remember) is giving a refund, and the success probability is

$$P(L < 500) = \int_0^{500} 0.0001 e^{-0.0001t} \, dt = 0.05 \tag{7.41}$$

Then plug p = 0.05 into (4.11) and (4.12).

7.6.3.4   Example: Garage Parking Fees

A certain public parking garage charges parking fees of \$1.50 for the first hour, and \$1 per hour after that. (It is assumed here for simplicity that the time is prorated within each of those defined periods. The reader should consider how the analysis would change if the garage "rounds up" each partial hour.) Suppose parking times T are exponentially distributed with mean 1.5 hours. Let W denote the total fee paid. Let's find E(W) and Var(W).

The key point is that W is a function of T:

$$W = \begin{cases} 1.5T, & \text{if } T \leq 1 \\ 1.5 + 1 \cdot (T-1) = T + 0.5, & \text{if } T > 1 \end{cases} \tag{7.42}$$

That's good, because we know how to find the expected value of a function of a continuous random variable, from (7.28). Defining g() as in (7.42) above, we have

$$EW = \int_0^\infty g(t) \, \frac{1}{1.5} e^{-\frac{1}{1.5}t} dt = \int_0^1 1.5t \, \frac{1}{1.5} e^{-\frac{1}{1.5}t} dt + \int_1^\infty (t+0.5) \, \frac{1}{1.5} e^{-\frac{1}{1.5}t} dt \tag{7.43}$$

The integration is left to the reader.

Now, what about Var(W)? As is often the case, it's easier to use (3.41), so we need to find $E(W^2)$. The above integration becomes

$$E(W^2) = \int_0^\infty g^2(t)\ f_W(t)dt = \int_0^1 (1.5t)^2\ \frac{1}{1.5}e^{-\frac{1}{1.5}t}dt + \int_1^\infty (t+0.5)^2\ \frac{1}{1.5}e^{-\frac{1}{1.5}t}dt \qquad (7.44)$$

After evaluating this, we subtract $(EW)^2$, giving us the variance of W.

7.6.3.5   Importance in Modeling

Many distributions in real life have been found to be approximately exponentially distributed. A famous example is the lifetimes of air conditioners on airplanes. Another famous example is interarrival times, such as customers coming into a bank or messages going out onto a computer network. It is used in software reliability studies too.

One of the reasons why this family is used so widely in probabilistic modeling is that it has several remarkable properties, so many that we have a special chapter for this family, Chapter 9.

7.6.4   The Gamma Family of Distributions

7.6.4.1   Density and Properties

Suppose at time 0 we install a light bulb in a lamp, which burns X_1 amount of time. We immediately install a new bulb then, which burns for time X_2, and so on. Assume the X_i are independent random variables having an exponential distribution with parameter λ.

Let

$$T_r = X_1 + ... + X_r, \quad r = 1, 2, 3, ... \qquad (7.45)$$

Note that the random variable T_r is the time of the rth light bulb replacement. T_r is the sum of r independent exponentially distributed random variables with parameter λ. The distribution of T_r is called an **Erlang** distribution. Its density can be shown to be

$$f_{T_r}(t) = \frac{1}{(r-1)!}\lambda^r t^{r-1}e^{-\lambda t}, \ t > 0 \qquad (7.46)$$

This is a two-parameter family.

Again, it's helpful to think in "notebook" terms. Say r = 8. Then we watch the lamp for the durations of eight lightbulbs, recording T_8, the time at which the eighth burns out. We write that time in the first line of our notebook. Then we watch a new batch of eight bulbs, and write the value of T_8 for those bulbs in the second line of our notebook, and so on. Then after recording a very large number of lines in our notebook, we plot a histogram of all the T_8 values. The point is then that that histogram will look like (7.46).

We can generalize this by allowing r to take noninteger values, by defining a generalization of the factorial function:

$$\Gamma(r) = \int_0^\infty x^{r-1} e^{-x} \, dx \tag{7.47}$$

This is called the gamma function, and it gives us the gamma family of distributions, more general than the Erlang:

$$f_W(t) = \frac{1}{\Gamma(r)} \lambda^r t^{r-1} e^{-\lambda t}, \ t > 0 \tag{7.48}$$

(Note that $\Gamma(r)$ is merely serving as the constant that makes the density integrate to 1.0. It doesn't have meaning of its own.)

This is again a two-parameter family, with r and λ as parameters.

A gamma distribution has mean r/λ and variance r/λ^2. In the case of integer r, this follows from (9.1) and the fact that an exponentially distributed random variable has mean and variance $1/\lambda$ and variance $1/\lambda^2$, and it can be derived in general. Note again that the gamma reduces to the exponential when r = 1.

7.6.4.2   Example: Network Buffer

Suppose in a network context (not our ALOHA example), a node does not transmit until it has accumulated five messages in its buffer. Suppose the times between message arrivals are independent and exponentially distributed with mean 100 milliseconds. Let's find the probability that more than 552 ms will pass before a transmission is made, starting with an empty buffer.

Let X_1 be the time until the first message arrives, X_2 the time from then to the arrival of the second message, and so on. Then the time until we accumulate five messages is $Y = X_1 + ... + X_5$. Then from the definition of the gamma family, we see that Y has a gamma distribution with r =

5 and $\lambda = 0.01$. Then

$$P(Y > 552) = \int_{552}^{\infty} \frac{1}{4!} 0.01^5 t^4 e^{-0.01t} \; dt \tag{7.49}$$

This integral could be evaluated via repeated integration by parts, but let's use R instead:

```
> 1 - pgamma(552,5,0.01)
[1] 0.3544101
```

Note that our parameter r is called **shape** in R, and our λ is **rate**.

Again, there are also **dgamma()**, **qgamma()** and **rgamma()**. From the R man page:

```
Usage:

    dgamma(x, shape, rate = 1, scale = 1/rate, log = FALSE)
    pgamma(q, shape, rate = 1, scale = 1/rate, lower.tail = TRUE,
           log.p = FALSE)
    qgamma(p, shape, rate = 1, scale = 1/rate, lower.tail = TRUE,
           log.p = FALSE)
    rgamma(n, shape, rate = 1, scale = 1/rate)
```

7.6.4.3   Importance in Modeling

As seen in (9.1), sums of exponentially distributed random variables often arise in applications. Such sums have gamma distributions.

You may ask what the meaning is of a gamma distribution in the case of noninteger r. There is no particular meaning, but when we have a real data set, we often wish to summarize it by fitting a parametric family to it, meaning that we try to find a member of the family that approximates our data well.

In this regard, the gamma family provides us with densities which rise near t = 0, then gradually decrease to 0 as t becomes large, so the family is useful if our data seem to look like this. Graphs of some gamma densities are shown in Figure 7.2.

As you might guess from the network performance analysis example in Section 7.6.4.2, the gamma family does arise often in the network context, and in queuing analysis in general.
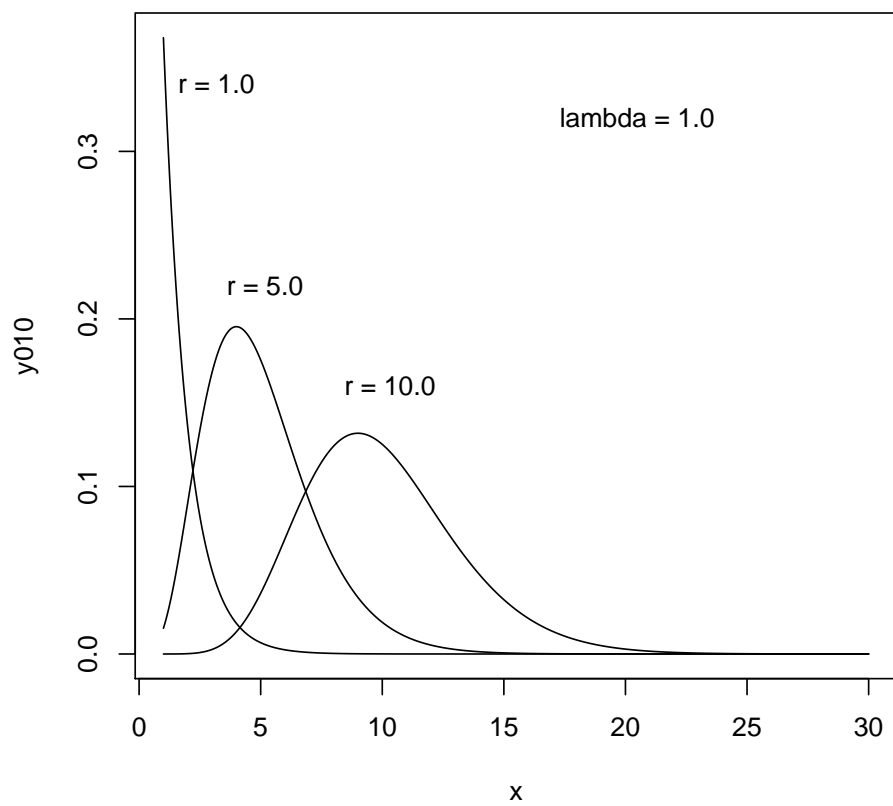
Figure 7.2: Various Gamma Densities

7.6.5   The Beta Family of Distributions

As seen in Figure 7.2, the gamma family is a good choice to consider if our data are nonnegative, with the density having a peak near 0 and then gradually tapering off to the right. What about data in the range (0,1)?

For instance, say trucking company transports many things, including furniture. Let X be the proportion of a truckload that consists of furniture. For instance, if 15% of given truckload is furniture, then X = 0.15. So here we have a distribution with support in (0,1). The beta family provides a very flexible model for this kind of setting, allowing us to model many different concave up or concave down curves.

7.6.5.1   Density Etc.

The densities of the family have the following form:

$$\frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)}t^{\alpha-1}(1-t)^{\beta-1} \tag{7.50}$$

There are two parameters, α and β. Figures 7.3 and 7.4 show two possibilities.

The mean and variance are

$$\frac{\alpha}{\alpha+\beta} \tag{7.51}$$

and

$$\frac{\alpha\beta}{(\alpha+\beta)^2(\alpha+\beta+1)} \tag{7.52}$$

Again, there are also **dbeta()**, **qbeta()** and **rbeta()**. From the R man page:

Usage:

```
dbeta(x, shape1, shape2, ncp = 0, log = FALSE)
pbeta(q, shape1, shape2, ncp = 0, lower.tail = TRUE, log.p = FALSE)
qbeta(p, shape1, shape2, ncp = 0, lower.tail = TRUE, log.p = FALSE)
rbeta(n, shape1, shape2, ncp = 0)
```

The graphs mentioned above were generated by running
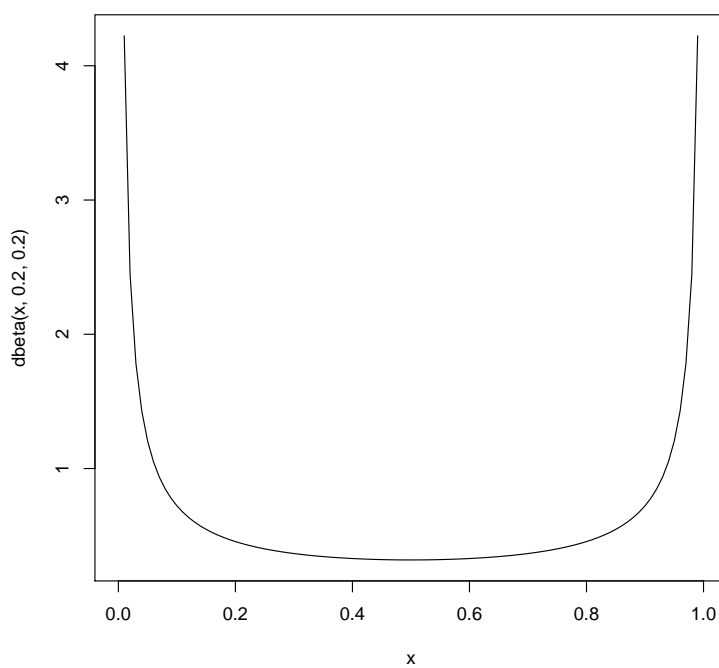
Figure 7.3: Beta Density, $\alpha = 0.2, \beta = 0.2$
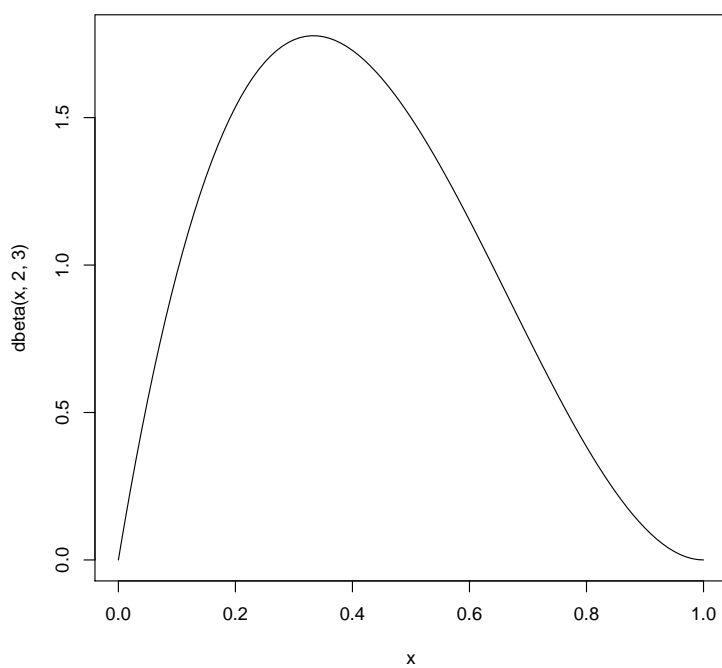
Figure 7.4: Beta Density, $\alpha = 2.0, \beta = 3.0$

```
> curve(dbeta(x,0.2,0.2))
> curve(dbeta(x,2,3)
```

7.6.5.2   Importance in Modeling

As mentioned, the beta family is a natural candidate for modeling a variable having range the interval (0,1).

This family is also popular among **Bayesian** statisticians (Section 18.4).

7.7   Choosing a Model

The parametric families presented here are often used in the real world. As indicated previously, this may be done on an empirical basis. We would collect data on a random variable X, and plot the frequencies of its values in a histogram. If for example the plot looks roughly like the curves in Figure 7.2, we could choose this as the family for our model.

Or, our choice may arise from theory. If for instance our knowledge of the setting in which we are working says that our distribution is memoryless, that forces us to use the exponential density family.

In either case, the question as to which member of the family we choose will be settled by using some kind of procedure which finds the member of the family which best fits our data. We will discuss this in detail in our chapters on statistics, especially Chapter 21.

Note that we may choose not to use a parametric family at all. We may simply find that our data does not fit any of the common parametric families (there are many others than those presented here) very well. Procedures that do not assume any parametric family are termed **nonparametric**.

7.8   Finding the Density of a Function of a Random Variable

Suppose X has, say, a uniform distribution on (1,4). Form a new random variable, $Y = X^2$. How can we find $f_Y()$? Reason as follows. For $1 < t < 16$,

$$f_Y(t) \;=\; \frac{d}{dt}F_Y(t) \quad \text{(def. of density)} \tag{7.53}$$

$$=\; \frac{d}{dt}P(Y \le t) \quad \text{(def. of cdf)} \tag{7.54}$$

$$=\; \frac{d}{dt}P(X \le \sqrt{t}) \quad \text{(def. of Y)} \tag{7.55}$$

$$=\; \frac{d}{dt}F_X[(\sqrt{t}-1)/3] \quad \text{(def. of cdf)} \tag{7.56}$$

$$=\; \frac{1}{3}f_X(\sqrt{t})\,\frac{d}{d}\sqrt{t} \quad \text{(Chain Rule)} \tag{7.57}$$

$$=\; \frac{1}{3} \cdot \frac{1}{2}t^{-0.5} \tag{7.58}$$

$$=\; \frac{1}{6}t^{-0.5} \tag{7.59}$$

Other such settings can be handled similarly. Note, though, that the above derivation relied on the fact that $X > 0$. Suppose X has a uniform distribution on (-1,1). Then the above derivation would become, for $0 < t < 1$,

$$f_Y(t) \;=\; \frac{d}{dt}P(Y \le t) \tag{7.60}$$

$$=\; \frac{d}{dt}P(-\sqrt{t} \le X \le \sqrt{t}) \tag{7.61}$$

$$=\; \frac{d}{dt}2\sqrt{t}/2 \tag{7.62}$$

$$=\; 0.5t^{-0.5} \tag{7.63}$$

7.9   Quantile Functions

First, recall the definition of the **inverse** of a function, say $h()$. The inverse of $h()$, denoted $h^{-1}()$, "brings you back to where you started." If I plug 3 into the squaring function, I get 9, and if I then plug 9 into the square root function, I get back my original 3.[5] So we say that $h(t) = t^2$ and $k(s) = \sqrt{s}$ are inverses of each other. The same relation holds between exp() and ln() and so on.

---

[5]This assumes, of course, that the domain of my squaring function consists only of the nonnegative numbers. We'll put aside this and similar situations for the time being, but will return.

For a random variable X, its **quantile function** $Q_X(s)$ is defined by

$$Q_X(s) = F_X^{-1}(s) \tag{7.64}$$

This is called the **s quantile** of X.

A well-known example is the **median** of X, the point which half the probability is above and half below. It's the 0.5 quantile of X.

The cdf tells us that cumulative probability for a particular value of X, while the quantile function does the opposite, "bringing us back." Let's make this concrete by considering the random variable Z in Section 7.3.1.

On the one hand, we can ask the question, "What is the probability that Z is at most 1?", with the answer being

$$F_Z(1) = 0.75 \tag{7.65}$$

On the other hand, one can ask conversely, "At what value of X do we have cumulative probability of 0.75?" Here the answer is

$$Q_Z(0.75) = 1 \tag{7.66}$$

It is no coincidence that the word *quantile* has that *-ile* suffix, given your familiarity with the word *percentile*. They are really the same thing.

Suppose for example that 92% of all who take the SAT Math Test have scores of at most 725. In other words, if you look at a randomly chosen test paper, and let X denote its score, then

$$F_X(725) = P(X \le 725) = 0.92 \tag{7.67}$$

On the other hand, if you are interested in the $92nd$ percentile score for the test, you are saying "Find s such that $P(X \le s) = 0.92$" you want

$$Q_X(0.92) = 725 \tag{7.68}$$

The reader should note that the R functions we've seen beginning with the letter 'q', such as **qgeom()** and **qunif()**, are quantile functions, hence the name.

However, a problem with discrete random variables is that the quantile values may not be unique. The reader should verify, for instance, in the coin-toss example above, the 0.75 quantile for Z could

be not only 1, but also 1.1, 1.288 and so on. So, one should look carefully at the documentation of quantile functions, to see what they do to impose uniqueness. But for continuous random variables there is no such problem.

7.10   Using cdf Functions to Find Probabilities

As we have seen, for many parametric families R has "d/p/q/r" functions, giving the density, cdf, quantile function and random number generator for the given family. How can we use the cdf functions to find probabilities of the form $P(a < X < b)$?

We see the answer in (7.15). We simply evaluate the cdf at b then a, and subtract.

For instance, consider the network buffer example, Section 7.6.4.2. Let's find $P(540 < Y < 562)$:

```
> pgamma(562,5,0.01) - pgamma(540,5,0.01)
[1] 0.03418264
```

Of course, we could also integrate the density,

```
> integrate(function(t) dgamma(t,5,0.01),540,562)
0.03418264 with absolute error < 3.8e-16
```

but R does that for us, there is probably little point in that second approach.

7.11   A General Method for Simulating a Random Variable

Suppose we wish to simulate a random variable X with density f_X for which there is no R function. This can be done via $F_X^{-1}(U)$, where U has a U(0,1) distribution. In other words, we call **runif()** and then plug the result into the inverse of the cdf of X.

For example, say X has the density 2t on (0,1). Then $F_X(t) = t^2$, so $F^{-1}(s) = s^{0.5}$. We can then generate an X as **sqrt(runif(1))**. Here's why:

For brevity, denote F_X^{-1} as G. Our generated random variable is then Y = G(U). Then

$$
\begin{aligned}
F_Y(t) &= P[G(U) \le t] & (7.69) \\
&= P[U \le G^{-1}(t)] & (7.70) \\
&= P[U \le F_X(t)] & (7.71) \\
&= F_X(t) & (7.72)
\end{aligned}
$$

(this last coming from the fact that U has a uniform distribution on (0,1)).

In other words, Y and X have the same cdf, i.e. the same distribution! This is exactly what we want.

Note that this method, though valid, is not necessarily practical, since computing F_X^{-1} may not be easy.

7.12 Example: Writing a Set of R Functions for a Certain Power Family

Consider the family of distributions indexed by positive values of c with densities

$$c\ t^{c-1} \tag{7.73}$$

for t in (0,1) and 0 otherwise..

The cdf is t^c, so let's call this the "tc" family.

Let's find "d", "p", "q" and "r" functions for this family, just like R has for the normal family, the gamma family and so on:

```
# density
dtc <- function(x,c) c * x^(c-1)

# cdf
ptc <- function(x,c) x^c

# quantile function
qtc <- function(q,c) q^(1/c)

# random number generator, n values generated
rtc <- function(n,c) {
   tmp <- runif(n)
   qtc(tmp,c)
}
```

Note that to get **rtc()** we simply plug U(0,1) variates into **qtc()**, according to Section 7.11.

Let's check our work. The mean for the density having c equal to 2 is 2/3 (reader should verify); let's see if a simulation will give us that:

```
> mean(rtc(10000,2))
[1] 0.6696941
```

Sure enough!

7.13   Multivariate Densities

Section 4.14 briefly introduced the notion of multivariate pmfs. Similarly, there are also multivariate densities. Probabilities are then k-fold integrals, where k is the number of random variables.

For instance, a probability involving two variables means taking a double integral of a bivariate density. Since that density can be viewed as a surface in three-dimensional space (just as a univariate density is viewed as a curve in two-dimensional space), a probability is then a volume under that surface (as opposed to area in the univariate case). Conversely, a bivariate density is the mixed partial derivative of the cdf:

$$f_{X,Y}(u,v) = \frac{\partial^2}{\partial u \partial v} F_{X,Y}(u,v) = P(X \leq u, \ Y \leq v) \tag{7.74}$$

In analogy to

$$P(B \mid A) = \frac{P(B \text{ and } A)}{P(A)} \tag{7.75}$$

we can define the conditional density of Y given X:

$$f_{Y|X}(u,v) = \frac{f_{X,Y}(u,v)}{f_X(v)} \tag{7.76}$$

The intuition behind this is that we are conditioning on X being *near* v. Actually,

$$f_{Y|X}(u,v) = \lim_{h \to 0} \ [\text{density of Y} \mid X \in (v-h, v+h)] \tag{7.77}$$

A detailed treatment is presented in Chapter 13.

7.14   Iterated Expectations

In analogy with (4.68), we have a very useful corresponding formula for the continuous case.

7.14.1 The Theorem

For any random variable W and any continuous random variable V,[6]

$$E(W) = \int_{-\infty}^{\infty} f_V(t) \; E(W \mid V = t) \; dt \tag{7.78}$$

Note that the event $V = t$ has probability 0 for coninuous V. The conditional expectation here is defined in terms of the conditional distribution of W given V; see Section 7.13.

Note too that if we have some event A, we can set W above to the indicator random variable of A (recall (3.9)), yielding

$$P(A) = \int_{-\infty}^{\infty} f_V(t) \; P(A \mid V = t) \; dt \tag{7.79}$$

7.14.2 Example: Another Coin Game

Suppose we have biased coins of various weightings, so that a randomly chosen coin's probability of heads H has density 2t on (0,1). The game has you choose a coin at random, toss it 5 times, and pays you a prize if you get 5 heads. What is your probability of winning?

First, note that the probability of winning, given H = t, is t^5. then (7.79) tells us that

$$P(\text{win}) = \int_0^1 2t \; t^5 \; dt = \frac{2}{7} \tag{7.80}$$

7.15 Continuous Random Variables Are "Useful Unicorns"

Recall our random dart example at the outset of this chapter. It must be kept in mind that this is only an idealization. D actually cannot be just any old point in (0,1). To begin with, our measuring instrument has only finite precision. Actually, then, D can only take on a finite number of values. If, say, our precision is four decimal digits, then D can only be 0.0001, 0.0002, ..., 0.9999, making it a discrete random variable after all.[7]

So this modeling of the position of the dart as continuously distributed really is just an idealization. *Indeed, in practice there are NO continuous random variables.* But the coninuous model can be an

---

[6]The treatment here will be intuitive, rather than being a mathematical definition and proof.

[7]There are also issues such as the nonzero thickness of the dart, and so on, further restricting our measurement.

excellent approximation, and the concept is extremely useful. It's like the assumption of "massless string" in physics analyses; there is no such thing, but it's a good approximation to reality.

As noted, most applications of statistics, and many of probability, are based on continuous distributions. We'll be using them heavily for the remainder of this book.

Chapter 8

The Normal Distributions

Again, these are the famous "bell-shaped curves," so called because their densities have that shape.

8.1 Density and Properties

The density for a normal distribution is

$$f_W(t) = \frac{1}{\sqrt{2\pi}\sigma} \, e^{-0.5\left(\frac{t-\mu}{\sigma}\right)^2}, \ -\infty < t < \infty \tag{8.1}$$

Again, this is a two-parameter family, indexed by the parameters μ and σ, which turn out to be the mean[1] and standard deviation μ and σ, The notation for it is $N(\mu, \sigma^2)$ (it is customary to state the variance σ^2 rather than the standard deviation).

And we write

$$X \tilde{\ } N(\mu, \sigma^2) \tag{8.2}$$

to mean that the random variable X has the distribution $N(\mu, \sigma^2)$. (The tilde is read "is distributed as.")

Note: Saying "X has a $N(\mu, \sigma^2)$ distribution" is *more* than simply saying "X has mean μ and variance σ^2." The former statement tells us not only the mean and variance of X, but *also* the fact that X has a "bell-shaped" density in the (8.1) family.

---

[1]Remember, this is a synonym for expected value.

8.1.1 Closure Under Affine Transformation

The family is closed under affine transformations:

If

$$X \tilde{\ } N(\mu, \sigma^2) \tag{8.3}$$

and we set

$$Y = cX + d \tag{8.4}$$

then

$$Y \tilde{\ } N(c\mu + d, c^2\sigma^2) \tag{8.5}$$

For instance, suppose X is the height of a randomly selection UC Davis student, measured in inches. Human heights do have approximate normal distributions; a histogram plot of the student heights would look bell-shaped. Now let Y be the student's height in centimeters. Then we have the situation above, with c = 2.54 and d = 0. The claim about affine transformations of normally distributed random variables would imply that a histogram of Y would again be bell-shaped.

Consider the above statement carefully.

> It is saying much more than simply that Y has mean $c\mu + d$ and variance $c^2\sigma^2$, which would follow from our our "mailing tubes" such as (3.49) *even if X did not have a normal distribution*. The key point is that this new variable Y is also a member of the normal family, i.e. its density is still given by (8.1), now with the new mean and variance.

Let's derive this, using the reasoning of Section 7.8.

For convenience, suppose $c > 0$. Then

$$
\begin{aligned}
F_Y(t) &= P(Y \leq t) \quad \text{(definition of } F_Y) & (8.6)\\
&= P(cX + d \leq t) \quad \text{(definition of Y)} & (8.7)\\
&= P\left(X \leq \frac{t-d}{c}\right) \quad \text{(algebra)} & (8.8)\\
&= F_X\left(\frac{t-d}{c}\right) \quad \text{(definition of } F_X) & (8.9)
\end{aligned}
$$

Therefore

$$
\begin{aligned}
f_Y(t) &= \frac{d}{dt} F_Y(t) \quad \text{(definition of } f_Y) &(8.10)\\
&= \frac{d}{dt} F_X\left(\frac{t-d}{c}\right) \quad \text{(from (8.9))} &(8.11)\\
&= f_X\left(\frac{t-d}{c}\right) \cdot \frac{d}{dt}\frac{t-d}{c} \quad \text{(definition of } f_X \text{ and the Chain Rule)} &(8.12)\\
&= \frac{1}{c} \cdot \frac{1}{\sqrt{2\pi}\sigma} e^{-0.5\left(\frac{\frac{t-d}{c}-\mu}{\sigma}\right)^2} \quad \text{(from (8.1)} &(8.13)\\
&= \frac{1}{\sqrt{2\pi}(c\sigma)} e^{-0.5\left(\frac{t-(c\mu+d)}{c\sigma}\right)^2} \quad \text{(algebra)} &(8.14)
\end{aligned}
$$

That last expression is the $N(c\mu + d, c^2\sigma^2)$ density, so we are done!

8.1.2 Closure Under Independent Summation

If X and Y are independent random variables, each having a normal distribution, then their sum S = X + Y also is normally distributed.

This is a pretty remarkable phenomenon, not true for most other parametric families. If for instance X and Y each with, say, a U(0,1) distribution, then the density of S turns out to be triangle-shaped, NOT another uniform distribution. (This can be derived using the methods of Section 13.3.2.)

Note that if X and Y are independent and normally distributed, then the two properties above imply that cX + dY will also have a normal distribution, for any constants c and d.

More generally:

For constants $a_1, ..., a_k$ and *independent* random variables $X_1, ..., X_k$, with

$$
X_i \widetilde{} N(\mu_i, \sigma_i^2 \tag{8.15}
$$

form the new random variable $Y = a_1 X_1 + ... + a_k X_k$. Then

$$
Y \widetilde{} N\left(\sum_{i=1}^{k} a_i\mu_i, \sum_{i=1}^{k} a_i^2\sigma_i^2\right) \tag{8.16}
$$

Lack of intuition:

The reader should ponder how remarkable this property of the normal family is, because there is no intuitive explanation for it.

Imagine random variables X and Y, each with a normal distribution. Say the mean and variances are 10 and 4 for X, and 18 and 6 for Y. We repeat our experiment 1000 times for our "notebook," i.e. 1000 lines with 2 columns. If we draw a histogram of the X column, we'll get a bell-shaped curve, and the same will be true for the Y column.

But now add a Z column, for $Z = X + Y$. Why in the world should a histogram of the Z column also be bell-shaped?

8.2   R Functions

```
dnorm(x, mean = 0, sd = 1)
pnorm(q, mean = 0, sd = 1)
qnorm(p, mean = 0, sd = 1)
rnorm(n, mean = 0, sd = 1)
```

Here **mean** and **sd** are of course the mean and standard deviation of the distribution. The other arguments are as in our previous examples.

8.3   The Standard Normal Distribution

Definition 13 *If $Z \sim N(0, 1)$ we say the random variable Z has a standard normal distribution.*

Note that if $X \sim N(\mu, \sigma^2)$, and if we set

$$Z = \frac{X - \mu}{\sigma} \tag{8.17}$$

then

$$Z \sim N(0, 1) \tag{8.18}$$

The above statements follow from the earlier material:

- Define $Z = \frac{X - \mu}{\sigma}$.
- Rewrite it as $Z = \frac{1}{\sigma} \cdot X + \left(\frac{-\mu}{\sigma}\right)$.

- Since E(cU + d) = c EU + d for any random variable U and constants c and d, we have

$$EZ = \frac{1}{\sigma}EX - \frac{\mu}{\sigma} = 0 \tag{8.19}$$

  and (3.56) and (3.49) imply that Var(X) = 1.

- OK, so we know that Z has mean 0 and variance 1. But does it have a normal distribution? Yes, due to our discussion above titled "Closure Under Affine Transformations."

By the way, the N(0,1) cdf is traditionally denoted by Φ.

8.4 Evaluating Normal cdfs

The function in (8.1) does not have a closed-form indefinite integral. Thus probabilities involving normal random variables must be approximated. Traditionally, this is done with a table for the cdf of N(0,1), which is included as an appendix to almost any statistics textbook; the table gives the cdf values for that distribution.

But this raises a question: There are infinitely many distributions in the normal family. Don't we need a separate table for each? That of course would not be possible, and in fact it turns out that this one table—the one for the N(0,1) distribution— is sufficient for the entire normal family. Though we of course will use R to gt such probabilities, it will be quite instructive to see how these table operations work.

Here's why one table is enough: Say X has an $N(10, 2.5^2)$ distribution. How can we get a probability like, say, $P(X < 12)$ using the N(0,1) table? Write

$$P(X < 12) = P\left(Z < \frac{12 - 10}{2.5}\right) = P(Z < 0.8) \tag{8.20}$$

Since on the right-hand side Z has a standard normal distribution, we can find that latter probably from the N(0,1) table!

As noted, traditionally it has played a central role, as one could transform any probability involving some normal distribution to an equivalent probability involving N(0,1). One would then use a table of N(0,1) to find the desired probability.

The transformation $Z = (X - \mu)/\sigma$ will play a big role in other contexts in future chapters, but for the sole purpose of simply evaluating normal probabilities, we can be much more direct. Nowadays, probabilities for any normal distribution, not just N(0,1), are easily available by computer. In the R

statistical package, the normal cdf for any mean and variance is available via the function **pnorm()**. The call form is

```
pnorm(q,mean=0,sd=1)
```

This returns the value of the cdf evaluated at **q**, for a normal distribution having the specified mean and standard deviation (default values of 0 and 1).

We can use **rnorm()** to simulate normally distributed random variables. The call is

```
rnorm(n,mean=0,sd=1)
```

which returns a vector of **n** random variates from the specified normal distribution.

There are also of course the corresponding density and quantile functions, **dnorm()** and **qnorm()**.

8.5   Example: Network Intrusion

As an example, let's look at a simple version of the network intrusion problem. Suppose we have found that in Jill's remote logins to a certain computer, the number X of disk sectors she reads or writes has an approximate normal distribution with a mean of 500 and a standard deviation of 15.

Before we continue, a comment on modeling: Since the number of sectors is discrete, it could not have an exact normal distribution. But then, no random variable in practice has an exact normal or other continuous distribution, as discussed in Section 7.15, and the distribution can indeed by approximately normal.

Now, say our network intrusion monitor finds that Jill—or someone posing as her—has logged in and has read or written 535 sectors. Should we be suspicious?

To answer this question, let's find $P(X \geq 535)$: Let $Z = (X - 500)/15$. From our discussion above, we know that Z has a N(0,1) distribution, so

$$P(X \geq 535) = P\left(Z \geq \frac{535 - 500}{15}\right) = 1 - \Phi(35/15) = 0.01 \tag{8.21}$$

Again, traditionally we would obtain that 0.01 value from a N(0,1) cdf table in a book. With R, we would just use the function **pnorm()**:

```
> 1 - pnorm(535,500,15)
[1] 0.009815329
```

Anyway, that 0.01 probability makes us suspicious. While it *could* really be Jill, this would be unusual behavior for Jill, so we start to suspect that it isn't her. It's suspicious enough for us to probe more deeply, e.g. by looking at which files she (or the impostor) accessed—were they rare for Jill too?

Now suppose there are two logins to Jill's account, accessing X and Y sectors, with X+Y = 1088. Is this rare for her, i.e. is $P(X + Y > 1088)$? small?

We'll assume X and Y are independent. We'd have to give some thought as to whether this assumption is reasonable, depending on the details of how we observed the logins, etc., but let's move ahead on this basis.

From page 153, we know that the sum S = X+Y is again normally distributed. Due to the properties in Chapter 3, we know S has mean $2 \cdot 500$ and variance $2 \cdot 15^2$. The desired probability is then found via

```
1 - pnorm(1088,1000,sqrt(450))
```

which is about 0.00002. That is indeed a small number, and we should be highly suspicious.

Note again that the normal model (or any other continuous model) can only be approximate, especially in the tails of the distribution, in this case the right-hand tail. But it is clear that S is only rarely larger than 1088, and the matter mandates further investigation.

Of course, this is very crude analysis, and real intrusion detection systems are much more complex, but you can see the main ideas here.

8.6 Example: Class Enrollment Size

After years of experience with a certain course, a university has found that online pre-enrollment in the course is approximately normally distributed, with mean 28.8 and standard deviation 3.1. Suppose that in some particular offering, pre-enrollment was capped at 25, and it hit the cap. Find the probability that the actual demand for the course was at least 30.

Note that this is a conditional probability! Evaulate it as follows. Let N be the actual demand. Then the key point is that we are given that $N \geq 25$, so

$$\begin{aligned}
P(N \geq 30 | N \geq 25) \;&=\; \frac{P(N \geq 30 \text{ and } N \geq 25)}{P(N \geq 25)} \quad ((2.7)) & (8.22) \\[2mm]
&=\; \frac{P(N \geq 30)}{P(N \geq 25)} & (8.23) \\[2mm]
&=\; \frac{1 - \Phi\left[(30 - 28.8)/3.1\right]}{1 - \Phi\left[(25 - 28.8)/3.1\right]} & (8.24) \\[2mm]
&=\; 0.39 & (8.25)
\end{aligned}$$

Sounds like it may be worth moving the class to a larger room before school starts.

Since we are approximating a discrete random variable by a continuous one, it might be more accurate here to use a **correction for continuity**, described in Section 8.14.

8.7   More on the Jill Example

Continuing the Jill example, suppose there is never an intrusion, i.e. all logins are from Jill herself. Say we've set our network intrusion monitor to notify us every time Jill logs in and accesses 535 or more disk sectors. In what proportion of all such notifications will Jill have accessed at least 545 sectors?

This is $P(X \geq 545 | X \geq 535)$. By an analysis similar to that in Section 8.6, this probability is

```
(1 - pnorm(545,500,15)) / (1 - pnorm(535,500,15))
```

8.8   Example: River Levels

Consider a certain river, and L, its level (in feet) relative to its average. There is a flood whenever $L > 8$, and it is reported that 2.5% of days have flooding. Let's assume that the level L is normally distributed; the above information implies that the mean is 0.

Suppose the standard deviation of L, σ, goes up by 10%. How much will the percentage of flooding days increase?

To solve this, let's first find σ. We have that

$$0.025 = P(L > 8) = P\left(\frac{L - 0}{\sigma} > \frac{8 - 0}{\sigma}\right) \tag{8.26}$$

Since $(L - 0)/\sigma$ has a N(0,1) distribution, we can find the 0.975 point in its cdf:

```
> qnorm(0.975,0,1)
[1] 1.959964
```

So,

$$1.96 = \frac{8 - 0}{\sigma} \tag{8.27}$$

so σ is about 4.

If it increases to 4.4, then we can evaluate $P(L > 8)$ by

```
> 1 - pnorm(8,0,4.4)
[1] 0.03451817
```

So, a 10% increase in σ would lead in this case to about a 40% increase in flood days.

8.9 Example: Upper Tail of a Light Bulb Distribution

Suppose we model light bulb lifetimes as having a normal distribution with mean and standard deviation 500 and 50 hours, respectively. Give a loop-free R expression for finding the value of d such that 30% of all bulbs have lifetime more than d.

You should develop the ability to recognize when we need **p**-series and **q**-series functions. Here we need

```
qnorm(1-0.30,500,50)
```

8.10 The Central Limit Theorem

The Central Limit Theorem (CLT) says, roughly speaking, that a random variable which is a sum of many components will have an approximate normal distribution. So, for instance, human weights are approximately normally distributed, since a person is made of many components. The same is true for SAT test scores,[2] as the total score is the sum of scores on the individual problems.

There are many versions of the CLT. The basic one requires that the summands be independent and identically distributed:[3]

---

[2]This refers to the raw scores, before scaling by the testing company.

[3]A more mathematically precise statement of the theorem is given in Section 8.20.

Theorem 14 *Suppose $X_1, X_2, ...$ are independent random variables, all having the same distribution which has mean m and variance v^2. Form the new random variable $T = X_1 + ... + X_n$. Then for large n, the distribution of T is approximately normal with mean nm and variance nv^2.*

The larger n is, the better the approximation, but typically n = 20 or even n = 10 is enough.

8.11   Example: Cumulative Roundoff Error

Suppose that computer roundoff error in computing the square roots of numbers in a certain range is distributed uniformly on (-0.5,0.5), and that we will be computing the sum of n such square roots. Suppose we compute a sum of 50 square roots. Let's find the approximate probability that the sum is more than 2.0 higher than it should be. (Assume that the error in the summing operation is negligible compared to that of the square root operation.)

Let $U_1, ..., U_{50}$ denote the errors on the individual terms in the sum. Since we are computing a sum, the errors are added too, so our total error is

$$T = U_1 + ... + U_{50} \tag{8.28}$$

By the Central Limit Theorem, since T is a sum, it has an approximately normal distribution, with mean 50 EU and variance 50 Var(U), where U is a random variable having the distribution of the U_i. From Section 7.6.1.1, we know that

$$EU = (-0.5 + 0.5)/2 = 0, \quad Var(U) = \frac{1}{12}[0.5 - (-0.5)]^2 = \frac{1}{12} \tag{8.29}$$

So, the approximate distribution of T is N(0,50/12). We can then use R to find our desired probability:

```
> 1 - pnorm(2,mean=0,sd=sqrt(50/12))
[1] 0.1635934
```

8.12   Example: R Evaluation of a Central Limit Theorem Approximation

Say $W = U_1 + ... + U_{50}$, with the U_i being independent and identically distributed (i.i.d.) with uniform distributions on (0,1). Give an R expression for the approximate value of $P(W < 23.4)$.

W has an approximate normal distribution, with mean 50×0.5 and variance $50 \times (1/12)$. So we need

```
pnorm(23.4,25,sqrt(50/12))
```

8.13 Example: Bug Counts

As an example, suppose the number of bugs per 1,000 lines of code has a Poisson distribution with mean 5.2. Let's find the probability of having more than 106 bugs in 20 sections of code, each 1,000 lines long. We'll assume the different sections act independently in terms of bugs.

Here X_i is the number of bugs in the ith section of code, and T is the total number of bugs. This is another clear candidate for using the CLT.

Since each X_i has a Poisson distribution, $m = v^2 = 5.2$. So, T, being a sum, is approximately distributed normally with mean and variance 20×5.2. So, we can find the approximate probability of having more than 106 bugs:

```
> 1 - pnorm(106,20*5.2,sqrt(20*5.2))
[1] 0.4222596
```

8.14 Example: Coin Tosses

Binomially distributed random variables, though discrete, also are approximately normally distributed. Here's why:

Say T has a binomial distribution with n trials. Then we can write T as a sum of indicator random variables (Section 3.9):

$$T = T_1 + ... + T_n \tag{8.30}$$

where T_i is 1 for a success and 0 for a failure on the ith trial. Since we have a sum of independent, identically distributed terms, the CLT applies. Thus we use the CLT if we have binomial distributions with large n.

For example, let's find the approximate probability of getting more than 12 heads in 20 tosses of a coin. X, the number of heads, has a binomial distribution with n = 20 and p = 0.5 Its mean and

variance are then np = 10 and np(1-p) = 5. So, let $Z = (X - 10)/\sqrt{5}$, and write

$$P(X > 12) = P(Z > \frac{12 - 10}{\sqrt{5}}) \approx 1 - \Phi(0.894) = 0.186 \tag{8.31}$$

Or:

```
> 1 - pnorm(12,10,sqrt(5))
[1] 0.1855467
```

The exact answer is 0.132, not too close. Why such a big error? The main reason is n here is rather small. But actually, we can still improve the approximation quite a bit, as follows.

Remember, the reason we did the above normal calculation was that X is approximately normal, from the CLT. This is an approximation of the distribution of a discrete random variable by a continuous one, which introduces additional error.

We can get better accuracy by using the **correction of continuity**, which can be motivated as follows. As an alternative to (8.31), we might write

$$P(X > 12) = P(X \geq 13) = P(Z > \frac{13 - 10}{\sqrt{5}}) \approx 1 - \Phi(1.342) = 0.090 \tag{8.32}$$

That value of 0.090 is considerably smaller than the 0.186 we got from (8.31). We could "split the difference" this way:

$$P(X > 12) = P(X \geq 12.5) = P(Z > \frac{12.5 - 10}{\sqrt{5}}) \approx 1 - \Phi(1.118) = 0.132 \tag{8.33}$$

(Think of the number 13 "owning" the region between 12.5 and 13.5, 14 owning the part between 13.5 and 14.5 and so on.) Since the exact answer to seven decimal places is 0.131588, the strategy has improved accuracy substantially.

The term *correction for continuity* alludes to the fact that we are approximately a discrete distribution by a continuous one.

8.15   Example: Normal Approximation to Gamma Family

Recall from above that the gamma distribution, or at least the Erlang, arises as a sum of independent random variables. Thus the Central Limit Theorem implies that the gamma distribution should

be approximately normal for large (integer) values of r. We see in Figure 7.2 that even with r = 10 it is rather close to normal.[4]

8.16 Example: Museum Demonstration

Many science museums have the following visual demonstration of the CLT.

There are many balls in a chute, with a triangular array of r rows of pins beneath the chute. Each ball falls through the rows of pins, bouncing left and right with probability 0.5 each, eventually being collected into one of r bins, numbered 0 to r. A ball will end up in bin i if it bounces rightward in i of the r rows of pins, i = 0,1,...,r. Key point:

> Let X denote the bin number at which a ball ends up. X is the number of rightward
> bounces ("successes") in r rows ("trials"). Therefore X has a binomial distribution with
> n = r and p = 0.5

Each bin is wide enough for only one ball, so the balls in a bin will stack up. And since there are many balls, the height of the stack in bin i will be approximately proportional to $P(X = i)$. And since the latter will be approximately given by the CLT, the stacks of balls will roughly look like the famous bell-shaped curve!

There are many online simulations of this museum demonstration, such as `http://www.mathsisfun.com/data/quincunx.html`. By collecting the balls in bins, the apparatus basically simulates a histogram for X, which will then be approximately bell-shaped.

8.17 Importance in Modeling

Needless to say, there are no random variables in the real world that are exactly normally distributed. In addition to our comments at the beginning of this chapter that no real-world random variable has a continuous distribution, there are no practical applications in which a random variable is not bounded on both ends. This contrasts with normal distributions, which extend from $-\infty$ to ∞.

Yet, many things in nature do have approximate normal distributions, so normal distributions play a key role in statistics. Most of the classical statistical procedures assume that one has sampled from a population having an approximate distribution. In addition, it will be seen later than the CLT

---

[4]It should be mentioned that technically, the CLT, which concerns convergence of cdfs, does not imply convergence of densities. However, under mild mathematical conditions, convergence of densities occurs too.

tells us in many of these cases that the quantities used for statistical estimation are approximately normal, even if the data they are calculated from are not.

Recall from above that the gamma distribution, or at least the Erlang, arises as a sum of independent random variables. Thus the Central Limit Theorem implies that the gamma distribution should be approximately normal for large (integer) values of r. We see in Figure 7.2 that even with r = 10 it is rather close to normal.

8.18   The Chi-Squared Family of Distributions

8.18.1   Density and Properties

Let $Z_1, Z_2, ..., Z_k$ be independent N(0,1) random variables. Then the distribution of

$$Y = Z_1^2 + ... + Z_k^2 \tag{8.34}$$

is called **chi-squared with k degrees of freedom**. We write such a distribution as χ_k^2. Chi-squared is a one-parameter family of distributions, and arises quite frequently in statistical applications, as will be seen in future chapters.

We can derive the mean of a chi-squared distribution as follows. First,

$$EY = E(Z_1^2 + ... + Z_k^2) = kE(Z_1^2) \tag{8.35}$$

Well, $E(Z_1^2)$ sounds somewhat like variance, suggesting that we use (3.41). This works:

$$E(Z_1^2) = Var(Z_1) + [E(Z_1)]^2 = Var(Z_1) = 1 \tag{8.36}$$

Then EY in (8.34) is k. One can also show that Var(Y) = 2k.

It turns out that chi-squared is a special case of the gamma family in Section 7.6.4 below, with r = k/2 and $\lambda = 0.5$.

The R functions **dchisq()**, **pchisq()**, **qchisq()** and **rchisq()** give us the density, cdf, quantile function and random number generator for the chi-squared family. The second argument in each case is the number of degrees of freedom. The first argument is the argument to the corresponding math function in all cases but **rchisq()**, in which it is the number of random variates to be generated.

For instance, to get the value of $f_X(5.2)$ for a chi-squared random variable having 3 degrees of freedom, we make the following call:

```
> dchisq(5.2,3)
[1] 0.06756878
```

8.18.2 Example: Error in Pin Placement

Consider a machine that places a pin in the middle of a flat, disk-shaped object. The placement is subject to error. Let X and Y be the placement errors in the horizontal and vertical directions, respectively, and let W denote the distance from the true center to the pin placement. Suppose X and Y are independent and have normal distributions with mean 0 and variance 0.04. Let's find $P(W > 0.6)$.

Since a distance is the square root of a sum of squares, this sounds like the chi-squared distribution might be relevant. So, let's first convert the problem to one involving squared distance:

$$P(W > 0.6) = P(W^2 > 0.36) \tag{8.37}$$

But $W^2 = X^2 + Y^2$, so

$$P(W > 0.6) = P(X^2 + Y^2 > 0.36) \tag{8.38}$$

This is not quite chi-squared, as that distribution involves the sum of squares of independent N(0,1) random variables. But due to the normal family's closure under affine transformations (page 152), we know that X/0.2 and Y/0.2 do have N(0,1) distributions. So write

$$P(W > 0.6) = P[(X/0.2)^2 + (Y/0.2)^2 > 0.36/0.2^2] \tag{8.39}$$

Now evaluate the right-hand side:

```
> 1 - pchisq(0.36/0.04,2)
[1] 0.01110900
```

8.18.3 Example: Generating Normal Random Numbers

How do normal random number generators such as **rnorm()** work? While in principle Section 7.11 could be used, the lack of a closed-form expression for $\Phi^{-1}()$ makes that approach infeasible.

Instead, we can exploit the relation between the normal family and exponential distribution, as follows.

Let Z_1 and Z_2 be two independent N(0,1) random variables, and define $W = Z_1^2 + Z_2^2$. By definition, W has a chi-squared distribution with 2 degrees of freedom, and from Section 8.18, we know that W has a gamma distribution with $r = 1$ and $\lambda = 0.5$.

In turn, we know from Section 7.6.4 that that distribution is actually just an exponential distribution with the same value of λ. This is quite fortuitous, since Section 7.11 *can* be used in this case.

Specifically, $f_W(t) = 0.5e^{-0.5t}$, so $F_W(t) = 1 - e^{-0.5t}$, Thus

$$F_W^{-1}(s) = -2\ln(1-s) \tag{8.40}$$

And there's more. Think of plotting the pair (Z_1, Z_2) in the X-Y plane, and define

$$\theta = \tan^{-1}\left(\frac{Z_2}{Z_1}\right) \tag{8.41}$$

One can show that θ has a uniform distribution on $(0, 2\pi)$. Also,

$$Z_1 = W\cos(\theta), \quad Z_1 = W\sin(\theta) \tag{8.42}$$

Putting all this together, we can generate a pair of independent N(0,1) random variates via the code

```
function genn01() {
   theta <- runif(1,0,2*pi)
   w <- -log(1 - runif(1))
   c(w*cos(theta),w*sin(theta))
}
```

Note by the way that we "get two for the price of one." If we need, say, 1000 random normal variates, we call the above function 500 times. If we need just variate, we use the first in the returned pair, and save the second in case we need it later in another call.

8.18.4   Importance in Modeling

This distribution family does not come up directly in application nearly so often as, say, the binomial or normal distribution family.

But the chi-squared family is used quite widely in statistical applications. As will be seen in our chapters on statistics, many statistical methods involve a sum of squared normal random variables.[5]

8.18.5   Relation to Gamma Family

One can show that the chi-square distribution with d degrees of freedom is a gamma distribution, with r = d/2 and $\lambda = 0.5$.

8.19   The Multivariate Normal Family

(Here we borrow some material from Chapter 13.)

The generalization of the normal family is the multivariate normal. Instead of being parameterized by a scalar mean and a scalar variance, the multivariate normal family has as its parameters a vector mean and a covariance matrix.

Let's look at the bivariate case first. The joint distribution of X_1 and X_2 is said to be **bivariate normal** if their density is

$$f_{X,Y}(s,t) = \frac{1}{2\pi\sigma_1\sigma_2\sqrt{1-\rho^2}}e^{-\frac{1}{2(1-\rho^2)}\left[\frac{(s-\mu_1)^2}{\sigma_1^2}+\frac{(t-\mu_2)^2}{\sigma_2^2}-\frac{2\rho(s-\mu_1)(t-\mu2)}{\sigma_1\sigma_2}\right]}, \quad -\infty < s,t < \infty \qquad (8.43)$$

This looks horrible, and it is. But don't worry, as we won't work with this directly. It's important for conceptual reasons, as follows.

First, note the parameters here: μ_1, μ_2, σ_1 and σ_2 are the means and standard deviations of X and Y, while ρ is the correlation between X and Y. So, we have a five-parameter family of distributions. The graph of the bivariate normal density looks like a 3-dimensional bell, as seen on the cover of this book.

The multivariate normal family of distributions is parameterized by one vector-valued quantity, the mean μ, and one matrix-valued quantity, the covariance matrix Σ. Specifically, suppose the random vector $X = (X_1, ..., X_k)'$ has a k-variate normal distribution.

The density has this form:

$$f_X(t) = ce^{-0.5(t-\mu)'\Sigma^{-1}(t-\mu)} \qquad (8.44)$$

---

[5]The motivation for the term *degrees of freedom* will be explained in those chapters too.

Here c is a constant, needed to make the density integrate to 1.0.

There is a Multivariate Central Limit Theorem, that says that sums of random vectors have approximately multivariate normal distributions.

In R the density, cdf and quantiles of the multivariate normal distribution are given by the functions **dmvnorm()**, **pmvnorm()** and **qmvnorm()** in the library **mvtnorm**. You can simulate a multivariate normal distribution by using **mvrnorm()** in the library **MASS**.

8.20   Optional Topic: Precise Statement of the CLT

The statement of Theorem 14 is not mathematically precise. We will fix it here, not just for mathematical niceness, but also because it leads to something called the *delta method*, a very practical tool in statistics.

8.20.1   Convergence in Distribution, and the Precisely-Stated CLT

Definition 15 *A sequence of random variables $L_1, L_2, L_3, ...$* **converges in distribution** *to a random variable M if*

$$\lim_{n \to \infty} P(L_n \leq t) = P(M \leq t), \text{ for all } t \tag{8.45}$$

Note by the way, that these random variables need not be defined on the same probability space.

The formal statement of the CLT is:

Theorem 16 *Suppose $X_1, X_2, ...$ are independent random variables, all having the same distribution which has mean m and variance v^2. Then*

$$Z = \frac{X_1 + ... + X_n - nm}{v\sqrt{n}} \tag{8.46}$$

converges in distribution to a N(0,1) random variable.

Chapter 9

The Exponential Distributions

The family of exponential distributions, Section 7.6.3, has a number of remarkable properties, which contribute to its widespread usage in probabilistic modeling. We'll discuss those here.

9.1 Connection to the Poisson Distribution Family

Suppose the lifetimes of a set of light bulbs are independent and identically distributed (**i.i.d.**), and consider the following process. At time 0, we install a light bulb, which burns an amount of time X_1. Then we install a second light bulb, with lifetime X_2. Then a third, with lifetime X_3, and so on.

Let

$$T_r = X_1 + ... + X_r \tag{9.1}$$

denote the time of the rth replacement. Also, let N(t) denote the number of replacements up to and including time t. Then it can be shown that if the common distribution of the X_i is exponentially distributed, the N(t) has a Poisson distribution with mean ηt. And the converse is true too: If the X_i are independent and identically distributed and N(t) is Poisson, then the X_i must have exponential distributions. In summary:

Theorem 17 *Suppose $X_1, X_2, ...$ are i.i.d. nonnegative continuous random variables. Define*

$$T_r = X_1 + ... + X_r \tag{9.2}$$

and

$$N(t) = \max\{k : T_k \leq t\} \tag{9.3}$$

Then the distribution of N(t) is Poisson with parameter ηt for all t if and only if the X_i have an exponential distribution with parameter η.

In other words, N(t) will have a Poisson distribution if and only if the lifetimes are exponentially distributed.

Proof

"Only if" part:

The key is to notice that the event $X_1 > t$ is exactly equivalent to $N(t) = 0$. If the first light bulb lasts longer than t, then the count of burnouts at time t is 0, and vice versa. Then

$$
\begin{aligned}
P(X_1 > t) &= P[N(t) = 0] \quad \text{(see above equiv.)} & (9.4)\\
&= \frac{(\eta t)^0}{0!} \cdot e^{-\eta t} \quad ((4.41) & (9.5)\\
&= e^{-\eta t} & (9.6)
\end{aligned}
$$

Then

$$f_{X_1}(t) = \frac{d}{dt}(1 - e^{-\eta t}) = \eta e^{-\eta t} \tag{9.7}$$

That shows that X_1 has an exponential distribution, and since the X_i are i.i.d., that implies that all of them have that distribution.

"If" part:

We need to show that if the X_i are exponentially distributed with parameter η, then for u nonnegative and each positive integer k,

$$P[N(u) = k] = \frac{(\eta u)^k e^{-\eta u}}{k!} \tag{9.8}$$

The proof for the case k = 0 just reverses (9.4) above. The general case, not shown here, notes that $N(u) \leq k$ is equivalent to $T_{k+1} > u$. The probability of the latter event can be found by integrating

(7.46) from u to infinity. One needs to perform k-1 integrations by parts, and eventually one arrives at (9.8), summed from 1 to k, as required.

■

The collection of random variables N(t) $t \geq 0$, is called a **Poisson process**.

The relation $E[N(t)] = \eta t$ says that replacements are occurring at an average rate of η per unit time. Thus η is called the **intensity parameter** of the process. It is this "rate" interpretation that makes η a natural indexing parameter in (7.40).

9.2 Memoryless Property of Exponential Distributions

One of the reasons the exponential family of distributions is so famous is that it has a property that makes many practical stochastic models mathematically tractable: The exponential distributions are **memoryless**.

9.2.1 Derivation and Intuition

What the term *memoryless* means for a random variable W is that for all positive t and u

$$P(W > t + u | W > t) = P(W > u) \tag{9.9}$$

Any exponentially distributed random variable has this property. Let's derive this:

$$
\begin{aligned}
P(W > t + u | W > t) &= \frac{P(W > t + u \text{ and } W > t)}{P(W > t)} & (9.10)\\
&= \frac{P(W > t + u)}{P(W > t)} & (9.11)\\
&= \frac{\int_{t+u}^{\infty} \lambda e^{-\lambda s}\, ds}{\int_{t}^{\infty} \lambda e^{-\lambda s}\, ds} & (9.12)\\
&= e^{-\lambda u} & (9.13)\\
&= P(W > u) & (9.14)
\end{aligned}
$$

We say that this means that "time starts over" at time t, or that W "doesn't remember" what happened before time t.

It is difficult for the beginning modeler to fully appreciate the memoryless property. Let's make it concrete. Consider the problem of waiting to cross the railroad tracks on Eighth Street in Davis, just west of J Street. One cannot see down the tracks, so we don't know whether the end of the train will come soon or not.

If we are driving, the issue at hand is whether to turn off the car's engine. If we leave it on, and the end of the train does not come for a long time, we will be wasting gasoline; if we turn it off, and the end does come soon, we will have to start the engine again, which also wastes gasoline. (Or, we may be deciding whether to stay there, or go way over to the Covell Rd. railroad overpass.)

Suppose our policy is to turn off the engine if the end of the train won't come for at least s seconds. Suppose also that we arrived at the railroad crossing just when the train first arrived, and we have already waited for r seconds. Will the end of the train come within s more seconds, so that we will keep the engine on? If the length of the train were exponentially distributed (if there are typically many cars, we can model it as continuous even though it is discrete), Equation (9.9) would say that the fact that we have waited r seconds so far is of no value at all in predicting whether the train will end within the next s seconds. The chance of it lasting at least s more seconds right now is no more and no less than the chance it had of lasting at least s seconds when it first arrived.

9.2.2   Uniquely Memoryless

By the way, the exponential distributions are the only continuous distributions which are memoryless. (Note the word *continuous*; in the discrete realm, the family of geometric distributions are also uniquely memoryless.) This too has implications for the theory. A rough proof of this uniqueness is as follows:

Suppose some continuous random variable V has the memoryless property, and let R(t) denote $1 - F_V(t)$. Then from (9.9), we would have

$$R(t + u)]/R(t) = R(u) \tag{9.15}$$

or

$$R(t + u) = R(t)R(u) \tag{9.16}$$

Differentiating both sides with respect to t, we'd have

$$R\prime(t + u) = R\prime(t)R(u) \tag{9.17}$$

Setting t to 0, this would say

$$R\prime(u) = R\prime(0)R(u) \tag{9.18}$$

This is a well-known differential equation, whose solution is

$$R(u) = e^{-cu} \tag{9.19}$$

which is exactly 1 minus the cdf for an exponentially distributed random variable.

9.2.3   Example: "Nonmemoryless" Light Bulbs

Suppose the lifetimes in years of light bulbs have the density $2t/15$ on (1,4), 0 elsewhere. Say I've been using bulb A for 2.5 years now in a certain lamp, and am continuing to use it. But at this time I put a new bulb, B, in a second lamp. I am curious as to which bulb is more likely to burn out within the next 1.2 years. Let's find the two probabilities.

For bulb A:

$$P(L > 3.7|L > 2.5) = \frac{P(L > 3.7)}{P(L > 2.5)} = 0.24 \tag{9.20}$$

For bulb B:

$$P(X > 1.2) = \int_{1.2}^{4} 2t/15 \; dt = 0.97 \tag{9.21}$$

So you can see that the bulbs do have "memory." We knew this beforehand, since the exponential distributions are the only continuous ones that have no memory.

9.3   Example: Minima of Independent Exponentially Distributed Random Variables

The memoryless property of the exponential distribution (Section 9.2 leads to other key properties. Here's a famous one:

Theorem 18 *Suppose $W_1, ..., W_k$ are independent random variables, with W_i being exponentially distributed with parameter λ_i. Let $Z = \min(W_1, ..., W_k)$. Then Z too is exponentially distributed with parameter $\lambda_1 + ... + \lambda_k$, and thus has mean equal to the reciprocal of that parameter*

Comments:

- In "notebook" terms, we would have k+1 columns, one each for the W_i and one for Z. For any given line, the value in the Z column will be the smallest of the values in the columns for $W_1, ..., W_k$; Z will be equal to one of them, but not the same one in every line. Then for instance $P(Z = W_3)$ is interpretable in notebook form as the long-run proportion of lines in which the Z column equals the W_3 column.

- It's pretty remarkable that the minimum of independent exponential random variables turns out again to be exponential. Contrast that with Section 13.3.6, where it is found that the minimum of independent uniform random variables does NOT turn out to have a uniform distribution.

- The sum $\lambda_1 + ... + \lambda_n$ in (a) should make good intuitive sense to you, for the following reasons. Recall from Section 9.1 that the parameter λ in an exponential distribution is interpretable as a "light bulb burnout rate."

  Say we have persons 1 and 2. Each has a lamp. Person i uses Brand i light bulbs, i = 1,2. Say Brand i light bulbs have exponential lifetimes with parameter λ_i. Suppose each time person i replaces a bulb, he shouts out, "New bulb!" and each time *anyone* replaces a bulb, I shout out "New bulb!" Persons 1 and 2 are shouting at a rate of λ_1 and λ_2, respectively, so I am shouting at a rate of $\lambda_1 + \lambda_2$.

Proof

$$
\begin{aligned}
F_Z(t) &= P(Z \leq t) \quad \text{(def. of cdf)} &\text{(9.22)} \\
&= 1 - P(Z > t) &\text{(9.23)} \\
&= 1 - P(W_1 > t \text{ and ... and } W_k > t) \quad (\min > t \text{ iff all } W_i > t) &\text{(9.24)} \\
&= 1 - \Pi_i \, P(W_i > t) \quad \text{(indep.)} &\text{(9.25)} \\
&= 1 - \Pi_i \, e^{-\lambda_i t} \quad \text{(expon. distr.)} &\text{(9.26)} \\
&= 1 - e^{-(\lambda_1 + ... + \lambda_n)t} &\text{(9.27)}
\end{aligned}
$$

Taking $\frac{d}{dt}$ of both sides proves the theorem.

■

Also:

Theorem 19 *Under the conditions in Theorem 18,*

$$P(W_i < W_1, ..., W_{i-1}, W_{i+1}, ..., W_k) = \frac{\lambda_i}{\lambda_1 + ... + \lambda_k} \tag{9.28}$$

(There are k terms in the denominator, not k-1.)

Equation (9.28) should be intuitively clear as well from the above "thought experiment" (in which we shouted out "New bulb!"): On average, we have one new Brand 1 bulb every $1/\lambda_1$ time, so in a long time t, we'll have about $t\lambda_1$ shouts for this brand. We'll also have about $t\lambda_2$ shouts for Brand 2. So, a proportion of about

$$\frac{t\lambda_1}{t\lambda_1 + t\lambda_2} \tag{9.29}$$

of the shots are for Brand 1. Also, at any given time, the memoryless property of exponential distributions implies that the time at which I shout next will be the *minimum* of the times at which persons 1 and 2 shout next. This intuitively implies (9.28).

Proof

Again consider the case k = 2, and then use induction.

Let $Z = \min(W_1, W_2)$ as before. Then

$$P(Z = W_1 | W_1 = t) = P(W_2 > t \mid W_1 = t) \tag{9.30}$$

(Note: We are working with continuous random variables here, so quantities like $P(W_1 = t)$ are 0 (though actually $P(Z = W_1)$ is nonzero). So, as mentioned in Section 7.79, quantities like $P(Z = W_1 \mid W_1 = t)$ really mean "the probability that $W_2 > t$ in the conditional distribution of Z given W_1.)

Since W_1 and W_2 are independent,

$$P(W_2 > t \mid W_1 = t) = P(W_2 > t) = e^{-\lambda_2 t} \tag{9.31}$$

Now use (7.79):

$$P(Z = W_1) = \int_0^\infty \lambda_1 e^{-\lambda_1 t} e^{-\lambda_2 t} \ dt = \frac{\lambda_1}{\lambda_1 + \lambda_2} \tag{9.32}$$

as claimed.

■

This property of minima of independent exponentially-distributed random variables developed in this section is key to the structure of continuous-time Markov chains, in Chapter 11.

9.3.1   Example: Computer Worm

A computer science graduate student at UCD, Senthilkumar Cheetancheri, was working on a worm alert mechanism. A simplified version of the model is that network hosts are divided into groups of size g, say on the basis of sharing the same router. Each infected host tries to infect all the others in the group. When g-1 group members are infected, an alert is sent to the outside world.

The student was studying this model via simulation, and found some surprising behavior. No matter how large he made g, the mean time until an external alert was raised seemed bounded. He asked me for advice.

I modeled the nodes as operating independently, and assumed that if node A is trying to infect node B, it takes an exponentially-distributed amount of time to do so. This is a continuous-time Markov chain. Again, this topic is much more fully developed in Chapter 11, but all we need here is the result of Section 9.3, that exponential distributions are "memoryless.".

In state i, there are i infected hosts, each trying to infect all of the g-i noninfected hosts. When the process reaches state g-1, the process ends; we call this state an **absorbing state**, i.e. one from which the process never leaves.

Scale time so that for hosts A and B above, the mean time to infection is 1.0. Since in state i there are i(g-i) such pairs, the time to the next state transition is the minimum of i(g-i) exponentially-distributed random variables with mean 1. Theorem 18 tells us that this minimum is also exponentially distributed, with parameter $i(g - i) \cdot 1$. Thus the mean time to go from state i to state i+1 is 1/[i(g-i)].

Then the mean time to go from state 1 to state g-1 is

$$\sum_{i=1}^{g-1} \frac{1}{i(g-i)} \tag{9.33}$$

Using a calculus approximation, we have

$$\int_1^{g-1} \frac{1}{x(g-x)} \, dx = \frac{1}{g} \int_1^{g-1} \left(\frac{1}{x} + \frac{1}{g-x}\right) \, dx = \frac{2}{g} \ln(g-1) \tag{9.34}$$

The latter quantity goes to zero as $g \to \infty$. This confirms that the behavior seen by the student in simulations holds in general. In other words, (9.33) remains bounded as $g \to \infty$. This is a very interesting result, since it says that the mean time to alert is bounded no matter how big our group size is.

So, even though our model here was quite simple, probably overly so, it did explain why the student was seeing the surprising behavior in his simulations.

9.3.2 Example: Electronic Components

Suppose we have three electronic parts, with independent lifetimes that are exponentially distributed with mean 2.5. They are installed simultaneously. Let's find the mean time until the last failure occurs.

Actually, we can use the same reasoning as for the computer worm example in Section 9.3.1: The mean time is simply

$$1/(3 \cdot 0.4) + 1/(2 \cdot 0.4) + 1/(1 \cdot 0.4) \tag{9.35}$$

9.4 A Cautionary Tale: the Bus Paradox

Suppose you arrive at a bus stop, at which buses arrive according to a Poisson process with intensity parameter 0.1, i.e. 0.1 arrival per minute. Recall that the means that the interarrival times have an exponential distribution with mean 10 minutes. What is the expected value of your waiting time until the next bus?

Well, our first thought might be that since the exponential distribution is memoryless, "time starts over" when we reach the bus stop. Therefore our mean wait should be 10.

On the other hand, we might think that on average we will arrive halfway between two consecutive buses. Since the mean time between buses is 10 minutes, the halfway point is at 5 minutes. Thus it would seem that our mean wait should be 5 minutes.

Which analysis is correct? Actually, the correct answer is 10 minutes. So, what is wrong with the second analysis, which concluded that the mean wait is 5 minutes? The problem is that the second analysis did not take into account the fact that although inter-bus intervals have an exponential distribution with mean 10, *the particular inter-bus interval that we encounter is special.*

9.4.1   Length-Biased Sampling

Imagine a bag full of sticks, of different lengths. We reach into the bag and choose a stick at random. The key point is that not all pieces are equally likely to be chosen; the longer pieces will have a greater chance of being selected.

Say for example there are 50 sticks in the bag, with ID numbers from 1 to 50. Let X denote the length of the stick we obtain if select a stick on an equal-probability basis, i.e. each stick having probability 1/50 of being chosen. (We select a random number I from 1 to 50, and choose the stick with ID number I.) On the other hand, let Y denote the length of the stick we choose by reaching into the bag and pulling out whichever stick we happen to touch first. Intuitively, the distribution of Y should favor the longer sticks, so that for instance $EY > EX$.

Let's look at this from a "notebook" point of view. We pull a stick out of the bag by random ID number, and record its length in the X column of the first line of the notebook. Then we replace the stick, and choose a stick out by the "first touch" method, and record its length in the Y column of the first line. Then we do all this again, recording on the second line, and so on. Again, because the "first touch" method will favor the longer sticks, the long-run average of the Y column will be larger than the one for the X column.

Another example was suggested to me by UCD grad student Shubhabrata Sengupta. Think of a large parking lot on which hundreds of buckets are placed of various diameters. We throw a ball high into the sky, and see what size bucket it lands in. Here the density would be proportional to area of the bucket, i.e. to the square of the diameter.

Similarly, the particular inter-bus interval that we hit is likely to be a longer interval. To see this, suppose we observe the comings and goings of buses for a very long time, and plot their arrivals on a time line on a wall. In some cases two successive marks on the time line are close together, sometimes far apart. If we were to stand far from the wall and throw a dart at it, we would hit the interval between some pair of consecutive marks. Intuitively we are more apt to hit a wider interval than a narrower one.

The formal name for this is **length-biased sampling**.

Once one recognizes this and carefully derives the density of that interval (see below), we discover that that interval does indeed tend to be longer—so much so that the expected value of this interval is 20 minutes! Thus the halfway point comes at 10 minutes, consistent with the analysis which appealed to the memoryless property, thus resolving the "paradox."

In other words, if we throw a dart at the wall, say, 1000 times, the mean of the 1000 intervals we would hit would be about 20. This in contrast to the mean of all of the intervals on the wall, which would be 10.

9.4.2 Probability Mass Functions and Densities in Length-Biased Sampling

Actually, we can intuitively reason out what the density is of the length of the particular inter-bus interval that we hit, as follows.

First consider the bag-of-sticks example, and suppose (somewhat artificially) that stick length X is a discrete random variable. Let Y denote the length of the stick that we pick by randomly touching a stick in the bag.

Again, note carefully that for the reasons we've been discussing here, the distributions of X and Y are different. Say we have a list of all sticks, and we choose a stick at random from the list. Then the length of that stick will be X. But if we choose by touching a stick in the bag, that length will be Y.

Now suppose that, say, stick lengths 2 and 6 each comprise 10% of the sticks in the bag, i.e.

$$p_X(2) = p_X(6) = 0.1 \tag{9.36}$$

Intuitively, one would then reason that

$$p_Y(6) = 3p_Y(2) \tag{9.37}$$

In other words, even though the sticks of length 2 are just as numerous as those of length 6, the latter are three times as long, so they should have triple the chance of being chosen. So, the chance of our choosing a stick of length j depends not only on $p_X(j)$ but also on j itself.

We could write that formally as

$$p_Y(j) \propto jp_X(j) \tag{9.38}$$

where $\propto$ is the "is proportional to" symbol. Thus

$$p_Y(j) = cjp_X(j) \qquad (9.39)$$

for some constant of proportionality c.

But a probability mass function must sum to 1. So, summing over all possible values of j (whatever they are), we have

$$1 = \sum_j p_Y(j) = \sum_j cjp_X(j) \qquad (9.40)$$

That last term is c E(X)! So, c = 1/EX, and

$$p_Y(j) = \frac{1}{EX} \cdot jp_X(j) \qquad (9.41)$$

The continuous analog of (9.41) is

$$f_Y(t) = \frac{1}{EX} \cdot tf_X(t) \qquad (9.42)$$

So, for our bus example, in which $f_X(t) = 0.1e^{-0.1t}$, $t > 0$ and EX = 10,

$$f_Y(t) = 0.01te^{-0.1t} \qquad (9.43)$$

You may recognize this as an Erlang density with r = 2 and $\lambda = 0.1$. That distribution does indeed have mean 20, consistent with the discussion at the end of Section 9.4.1.

Chapter 10

Stop and Review: Probability Structures

There's quite a lot of material in the preceding chapters, but it's crucial that you have a good command of it before proceeding, as the coming chapters will continue to build on it.

With that aim, here are the highlights of what we've covered so far, with links to the places at which they were covered:

- **expected value** (Section 3.5):

  Consider random variables X and Y (not assumed independent), and constants c_1 and c_2. We have:

  $$E(X + Y) = EX + EY \tag{10.1}$$

  $$E(c_1 X) = c_1 EX \tag{10.2}$$

  $$E(c_1 X + c_2 Y) = c_1 EX + c_2 EY \tag{10.3}$$

  By induction,

  $$E(a_1 U_1 + ... + a_k U_k) = a_1 EX_1 + ... + a_k EX_k \tag{10.4}$$

  for random variables U_i and constants a_i.

- **variance** (Section 3.6):

  For any variable W,

$$Var(W) = E[(W - EW)^2] = E(W^2) - (EW)^2 \tag{10.5}$$

  Consider random variables X and Y (now assumed independent), and constants c_1 and c_2. We have:

$$Var(X + Y) = Var(X) + Var(Y) \tag{10.6}$$

$$Var(c_1 X) = c_1^2 Var(X) \tag{10.7}$$

  By induction,

$$Var(a_1 U_1 + ... + a_k U_k) = a_1^2 Var(U_1) + ... + a_k^2 Var(U_k) \tag{10.8}$$

  for independent random variables U_i and constants a_i.

- **indicator random variables** (Section 3.9):

  Equal 1 or 0, depending on whether a specified event A occurs.

  If T is an indicator random variable for the event A, then

$$ET = P(A), \quad Var(T) = P(A)[1 - P(A)] \tag{10.9}$$

- **distributions**:

  - **cdfs**(Section 7.3):

    For any random variable X,

$$F_X(t) = P(X \le t), \quad -\infty < t < \infty \tag{10.10}$$

  - **pmfs** (Section 3.13):

    For a discrete random variable X,

$$p_X(k) = P(X = k) \tag{10.11}$$

– **density functions** (Section 3.13):

For a continuous random variable X,

$$f_X(t) = \frac{d}{dt} F_X(t), \quad -\infty < t < \infty \tag{10.12}$$

and

$$P(X \text{ in } A) = \int_A f_X(s) \, ds \tag{10.13}$$

• **famous parametric families of distributions**:

Just like one can have a family of curves, say $sin(2\pi n\theta(t)$ (different curve for each n and θ), certain families of distributions have been found useful. They're called *parametric families*, because they are indexed by one or more parameters, anlagously to n and θ above.

discrete:

– **geometric** (Section 4.2)

Number of i.i.d. trials until first success. For success probability p:

$$p_N(k) = (1-p)^k p \tag{10.14}$$

$$EN = 1/p, \quad Var(N) = \frac{1-p}{p^2} \tag{10.15}$$

– **binomial** (Section 4.3):

Number of successes in n i.i.d. trials, probability p of success per trial:

$$p_N(k) = \binom{n}{k} p^k (1-p)^{n-k} \tag{10.16}$$

$$EN = np, \quad Var(N) = np(1-p) \tag{10.17}$$

– **Poisson** (Section 4.5):

Has often been found to be a good model for counts over time periods.

One parameter, often called λ. Then

$$p_N(k) = \frac{e^{-\lambda}\lambda^k}{k!}, k = 0, 1, 2, ... \tag{10.18}$$

$$EN = Var(N) = \lambda \tag{10.19}$$

– **negative binomial** (Section 4.4):

Number of i.i.d. trials until rth success. For success probability p:

$$p_N(k) = \binom{k-1}{r-1}(1-p)^{k-r}p^r, k = r, r+1, ...$$

(10.20)

$$E(N) = r \cdot \frac{1}{p}, \quad Var(N) = r \cdot \frac{1-p}{p^2}$$

(10.21)

continuous:

– **uniform** (Section 7.6.1.1):

All points "equally likely." If the interval is (q,r),

$$f_X(t) = \frac{1}{r-q}, \quad q < t < r$$

(10.22)

$$EX == \frac{q+r}{2}, \quad Var(D) = \frac{1}{12}(r-q)^2$$

(10.23)

– **normal (Gaussian)** (Section 7.6.2):

"Bell-shaped curves." Useful due to Central Limit Theorem (Section 8.10. (Thus good approximation to binomial distribution.)

Closed under affine transformations (Section 8.1.1)!

Parameterized by mean and variance, μ and σ^2:

$$f_X(t) = \frac{1}{\sqrt{2\pi}\sigma} \, e^{-0.5\left(\frac{t-\mu}{\sigma}\right)^2}, -\infty < t < \infty$$

(10.24)

exponential (Section 7.6.3):

– Memoryless! One parameter, usually called λ. Connectedt to Poisson family.

$$f_X(t) == \lambda e^{-\lambda t}, 0 < t < \infty$$

(10.25)

$$EX = 1/\lambda, \quad Var(X) = 1/\lambda^2$$

(10.26)

– **gamma** (Section 7.6.4):

Special case, Erlang family, arises as the distribution of the sum of i.i.d. exponential random variables.

$$f_X(t) = \frac{1}{\Gamma(r)}\lambda^r t^{r-1}e^{-\lambda t}, \; t > 0$$

(10.27)

- **iterated expected values:**

  - For discrete U (4.68),

  $$E(V) = \sum_c P(U = c) \ E(V \mid U = c) \qquad (10.28)$$

  - For continuous V (7.78),

  $$E(W) = \int_{-\infty}^{\infty} f_V(t) \ E(W \mid V = t) \ dt \qquad (10.29)$$

Chapter 11

Introduction to Continuous-Time Markov Chains

In the Markov chains we analyzed in Chapter 6, events occur only at integer times. However, many Markov chain models are of the **continuous-time** type, in which events can occur at any times. Here the **holding time**, i.e. the time the system spends in one state before changing to another state, is a continuous random variable.

11.1   Continuous-Time Markov Chains

The state of a Markov chain at any time now has a continuous subscript. Instead of the chain consisting of the random variables X_n, $n = 1, 2, 3, ...$, it now consists of $\{X_t : t \in [0, \infty)\}$. The Markov property is now

$$P(X_{t+u} = k | X_s \text{ for all } 0 \le s \le t) = P(X_{t+u} = k | X_t) \text{ for all } t, u \ge 0 \qquad (11.1)$$

11.2   Holding-Time Distribution

In order for the Markov property to hold, the distribution of holding time at a given state needs to be "memoryless." Recall that exponentially distributed random variables have this property. In other words, if a random variable W has density

$$f(t) = \lambda e^{-\lambda t} \qquad (11.2)$$

for some λ then

$$P(W > r + s | W > r) = P(W > s) \tag{11.3}$$

for all positive r and s. And exponential distributions are the only continuous distributions which have this property. Therefore, *holding times in Markov chains must be exponentially distributed.*

Because it is central to the Markov property, the exponential distribution is assumed for all basic activities in Markov models. In queuing models, for instance, both the interarrival time and service time are assumed to be exponentially distributed (though of course with different values of λ). In reliability modeling, the lifetime of a component is assumed to have an exponential distribution.

Such assumptions have in many cases been verified empirically. If you go to a bank, for example, and record data on when customers arrive at the door, you will find the exponential model to work well (though you may have to restrict yourself to a given time of day, to account for nonrandom effects such as heavy traffic at the noon hour). In a study of time to failure for airplane air conditioners, the distribution was also found to be well fitted by an exponential density. On the other hand, in many cases the distribution is not close to exponential, and purely Markovian models cannot be used for anything more than a rough approximation.

11.2.1   The Notion of "Rates"

A key point is that the parameter λ in (11.2) has the interpretation of a rate, in the sense discussed in Theorem 18. To review, first, recall that $1/\lambda$ is the mean. Say light bulb lifetimes have an exponential distribution with mean 100 hours, so $\lambda = 0.01$. In our lamp, whenever its bulb burns out, we immediately replace it with a new on. Imagine watching this lamp for, say, 100,000 hours. During that time, we will have done approximately $100000/100 = 1000$ replacements. That would be using 1000 light bulbs in 100000 hours, so we are using bulbs at the rate of 0.01 bulb per hour. For a general λ, we would use light bulbs at the rate of λ bulbs per hour. This concept is crucial to what follows.

11.3   Stationary Distribution

In analogy to (6.4), we again define π_i to be the long-run proportion of time the system is in state i, where now N_{it} is the proportion of the time spent in state i, during (0,t). We again will derive a system of linear equations to solve for these proportions, using a flow out = flow in argument.

11.3.1 Intuitive Derivation

To this end, let λ_i denote the parameter in the holding-time distribution at state i, and define the quantities

$$\rho_{rs} = \lambda_r p_{rs} \tag{11.4}$$

where p_{rs} is that probably that, when a jump out of state r occurs, the jump is to state s.

The equations has the following interpretation. Note:

- λ_r is the rate of jumps out of state r, so

- $\lambda_r p_{rs}$ is the rate of jumps from state r to state s, and since

- π_r is the long-run proportion of the time we are in state r, then

- $\pi_r \lambda_r p_{rs}$ is the rate of jumps from r to s

Then, equating the rate of transitions into i and the rate out of i, we have

$$\pi_i \lambda_i = \sum_{j \neq i} \pi_j \lambda_j p_{ji} \tag{11.5}$$

These equations can then be solved for the π_i.

11.3.2 Computation

Motivated by (11.5), define the matrix Q by

$$q_{ij} = \begin{cases} \lambda_j p_{ji}, & \text{if } i \neq j \\ -\lambda_i, & \text{if } i = j \end{cases} \tag{11.6}$$

Q is called the **infinitesimal generator** of the system, so named because it is the basis of the system of differential equations that can be used to find the finite-time probabilistic behavior of X_t.

Then (11.5) is stated in matrix form as

$$Q\pi = 0 \tag{11.7}$$

But the π_i must sum to 1, so the above equation is subject to

$$1'\pi = 1 \tag{11.8}$$

where 1 denotes a (column) vector of n 1s, where n is the number of states.

In view of (11.8), the system (11.7) is redundant; there are n equations for n-1 unknowns, So, replace the last equation in (11.7) by (11.8).

Here is R code to solve the system:

```
findpicontin <- function(q) {
   n <- nrow(q)
   q[n,] <- rep(1,n)
   rhs <- c(rep(0,n-1),1)
   pivec <- solve(q,rhs)
   return(pivec)
}
```

To formulate the equations (11.5), we'll need a property of exponential distributions derived in Section 9.3, copied here for convenience:

Theorem 20 *Suppose $W_1, ..., W_k$ are independent random variables, with W_i being exponentially distributed with parameter λ_i. Let $Z = \min(W_1, ..., W_k)$. Then*

(a) *Z is exponentially distributed with parameter $\lambda_1 + ... + \lambda_k$*

(b) *$P(Z = W_i) = \frac{\lambda_i}{\lambda_1 + ... + \lambda_k}$*

11.4   Example: Machine Repair

Suppose the operations in a factory require the use of a certain kind of machine. The manager has installed two of these machines. This is known as a **gracefully degrading system**: When both machines are working, the fact that there are two of them, instead of one, leads to a shorter wait time for access to a machine. When one machine has failed, the wait is longer, but at least the factory operations may continue. Of course, if both machines fail, the factory must shut down until at least one machine is repaired.

Suppose the time until failure of a single machine, carrying the full load of the factory, has an exponential distribution with mean 20.0, but the mean is 25.0 when the other machine is working, since it is not so loaded. Repair time is exponentially distributed with mean 8.0.

We can take as our state space $\{0,1,2\}$, where the state is the number of working machines. Now, let us find the parameters λ_i and p_{ji} for this system. For example, what about λ_2? The holding time in state 2 is the <u>minimum</u> of the two lifetimes of the machines, and thus from the results of Section 9.3, has parameter $\frac{1}{25.0} + \frac{1}{25.0} = 0.08$.

For λ_1, a transition out of state 1 will be either to state 2 (the down machine is repaired) or to state 0 (the up machine fails). The time until transition will be the minimum of the lifetime of the up machine and the repair time of the down machine, and thus will have parameter $\frac{1}{20.0} + \frac{1}{8.0} = 0.175$. Similarly, $\lambda_0 = \frac{1}{8.0} + \frac{1}{8.0} = 0.25$.

It is important to understand how the Markov property is being used here. Suppose we are in state 1, and the down machine is repaired, sending us into state 2. Remember, the machine which had already been up has "lived" for some time now. But the memoryless property of the exponential distribution implies that this machine is now "born again."

What about the parameters p_{ji}? Well, p_{21} is certainly easy to find; since the transition $2 \to 1$ is the *only* transition possible out of state 2, $p_{21} = 1$.

For p_{12}, recall that transitions out of state 1 are to states 0 and 2, with rates $1/20.0$ and $1/8.0$, respectively. So,

$$p_{12} = \frac{1/8.0}{1/20.0 + 1/8.0} = 0.72 \tag{11.9}$$

Working in this manner, we finally arrive at the complete system of equations (11.5):

$$\pi_2(0.08) = \pi_1(0.125) \tag{11.10}$$
$$\pi_1(0.175) = \pi_2(0.08) + \pi_0(0.25) \tag{11.11}$$
$$\pi_0(0.25) = \pi_1(0.05) \tag{11.12}$$

In matrix terms:

$$\begin{pmatrix} -0.25 & 0.05 & 0 \\ 0.25 & -0.175 & 0.08 \\ 0 & 0.125 & -0.08 \end{pmatrix} \pi = 0 \tag{11.13}$$

Let's find the solution:

```
> q
       [,1]    [,2]    [,3]
[1,] -0.25   0.050   0.00
```

```
[2,]   0.25 -0.175   0.08
[3,]   0.00   0.125 -0.08
> findpicontin(q)
[1] 0.07239819 0.36199095 0.56561086
```

So,

$$\pi = (0.072, 0.362, 0.566) \tag{11.14}$$

Thus for example, during 7.2% of the time, there will be no machine available at all.

Several variations of this problem could be analyzed. We could compare the two-machine system with a one-machine version. It turns out that the proportion of down time (i.e. time when no machine is available) increases to 28.6%. Or we could analyze the case in which only one repair person is employed by this factory, so that only one machine can be repaired at a time, compared to the situation above, in which we (tacitly) assumed that if both machines are down, they can be repaired in parallel. We leave these variations as exercises for the reader.

11.5   Example: Migration in a Social Network

The following is a simplified version of research in online social networks.

There is a town with two social groups, with each person being in exactly one group. People arrive from outside town, with exponentially distributed interarrival times at rate α, and join one of the groups with probability 0.5 each. Each person will occasionally switch groups, with one possible "switch" being to leave town entirely. A person's time before switching is exponentially distributed with rate σ; the switch will either be to the other group or to the outside world, with probabilities q and 1-q, respectively. Let the state of the system be (i,j), where i and j are the number of current members in groups 1 and 2, respectively.

Let's find a typical balance equation, say for the state (8,8):

$$\pi_{(8,8)}(\alpha + 16 \cdot \sigma) = (\pi_{(9,8)} + \pi_{(8,9)}) \cdot 9\sigma(1 - q) + (\pi_{(9,7)} + \pi_{(7,9)}) \cdot 9\sigma q + (\pi_{(8,7)} + \pi_{(7,8)}) \cdot 0.5\alpha \tag{11.15}$$

The reasoning is straightforward. How can we move out of state (8,8)? Well, there could be an arrival (rate α), or any one of the 16 people could switch groups (rate 16σ), etc.

Now, in a "going beyond finding the π" vein, let's find the long-run fraction of transfers into group 1 that come from group 2, as opposed to from the outside.

The rate of transitions into that group from outside is 0.5α. When the system is in state (i,j), the rate of transitions into group 1 from group 2 is $j\sigma q$, so the overall rate is $\sum_{i,j} \pi_{(i,j)} j\sigma q$. Thus the fraction of new members coming in to group 1 from transfers is

$$\frac{\sum_{i,j} \pi_{(i,j)} j\sigma q}{0.5\alpha + \sum_{i,j} \pi_{(i,j)} j\sigma q} \tag{11.16}$$

The above reasoning is very common, quite applicable in many situations. By the way, note that $\sum_{i,j} \pi_{(i,j)} j\sigma q = \sigma q E N$, where N is the number of members of group 2.

11.6 Birth/Death Processes

We noted earlier that the system of equations for the π_i may not be easy to solve. In many cases, for instance, the state space is infinite and thus the system of equations is infinite too. However, there is a rich class of Markov chains for which closed-form solutions have been found, called **birth/death processes**.[1]

Here the state space consists of (or has been mapped to) the set of nonnegative integers, and p_{ji} is nonzero only in cases in which $|i - j| = 1$. (The name "birth/death" has its origin in Markov models of biological populations, in which the state is the current population size.) Note for instance that the example of the gracefully degrading system above has this form. An M/M/1 queue—one server, "Markov" (i.e. exponential) interarrival times and Markov service times—is also a birth/death process, with the state being the number of jobs in the system.

Because the p_{ji} have such a simple structure, there is hope that we can find a closed-form solution for the π_i, and it turns out we can. Let $u_i = \rho_{i,i+1}$ and $d_i = \rho_{i,i-1}$ ('u' for "up," 'd' for "down"). Then (**??**) is

$$\pi_{i+1}d_{i+1} + \pi_{i-1}u_{i-1} = \pi_i \lambda_i = \pi_i(u_i + d_i), \ i \geq 1 \tag{11.17}$$

$$\pi_1 d_1 = \pi_0 \lambda_0 = \pi_0 u_0 \tag{11.18}$$

In other words,

$$\pi_{i+1}d_{i+1} - \pi_i u_i = \pi_i d_i - \pi_{i-1}u_{i-1}, \ i \geq 1 \tag{11.19}$$

---

[1]Though we treat the continuous-time case here, there is also a discrete-time analog.

$$\pi_1 d_1 - \pi_0 u_0 = 0 \tag{11.20}$$

Applying (11.19) recursively to the base (11.20), we see that

$$\pi_i d_i - \pi_{i-1} u_{i-1} = 0, \ i \geq 1 \tag{11.21}$$

so that

$$\pi_i = \pi_{i-1} \frac{u_{i-1}}{d_i} \ i \geq 1 \tag{11.22}$$

and thus

$$\pi_i = \pi_0 r_i \tag{11.23}$$

where

$$r_i = \Pi_{k=1}^i \frac{u_{k-1}}{d_k} \tag{11.24}$$

where $r_i = 0$ for $i > m$ if the chain has no states past m.

Then since the π_i must sum to 1, we have that

$$\pi_0 = \frac{1}{1 + \sum_{i=1}^{\infty} r_i} \tag{11.25}$$

and the other π_i are then found via (11.23).

Note that the chain might be finite, i.e. have $u_i = 0$ for some i. In that case it is still a birth/death chain, and the formulas above for π still apply.

11.7   Cell Communications Model

Let's consider a more modern example of this sort, involving cellular phone systems. (This is an extension of the example treated in K.S. Trivedi, *Probability and Statistics, with Reliability and Computer Science Applications* (second edition), Wiley, 2002, Sec. 8.2.3.2, which is in turn is based on two papers in the *IEEE Transactions on Vehicular Technology*.)

We consider one particular cell in the system. Mobile phone users drift in and out of the cell as they move around the city. A call can either be a **new call**, i.e. a call which someone has just dialed, or a **handoff call**, i.e. a call which had already been in progress in a neighboring cell but now has moved to this cell.

Each call in a cell needs a **channel**.[2] There are n channels available in the cell. We wish to give handoff calls priority over new calls.[3] This is accomplished as follows.

The system always reserves g channels for handoff calls. When a request for a new call (i.e. a non-handoff call) arrives, the system looks at X_t, the current number of calls in the cell. If that number is less than n-g, so that there are more than g idle channels available, the new call is accepted; otherwise it is rejected.

We assume that new calls originate from within the cells according to a Poisson process with rate λ_1, while handoff calls drift in from neighboring cells at rate λ_2. Meanwhile, call durations are exponential with rate μ_1, while the time that a call remains within the cell is exponential with rate μ_2.

11.7.1 Stationary Distribution

We again have a birth/death process, though a bit more complicated than our earlier ones. Let $\lambda = \lambda_1 + \lambda_2$ and $\mu = \mu_1 + \mu_2$. Then here is a sample balance equation, focused on transitions into (left-hand side in the equation) and out of (right-hand side) state 1:

$$\pi_0 \lambda + \pi_2 2\mu = \pi_1 (\lambda + \mu) \tag{11.26}$$

Here's why: How can we enter state 1? Well, we could do so from state 0, where there are no calls; this occurs if we get a new call (rate λ_1) or a handoff call (rate λ_2. In state 2, we enter state 1 if one of the two calls ends (rate μ_1) or one of the two calls leaves the cell (rate μ_2). The same kind of reasoning shows that we leave state 1 at rate $\lambda + \mu$.

As another example, here is the equation for state n-g:

$$\pi_{n-g}[\lambda_2 + (n-g)\mu] = \pi_{n-g+1} \cdot (n-g+1)\mu + \pi_{n-g-1}\lambda \tag{11.27}$$

Note the term λ_2 in (11.27), rather than λ as in (11.26).

---

[2]This could be a certain frequency or a certain time slot position.

[3]We would rather give the caller of a new call a polite rejection message, e.g. "No lines available at this time, than suddenly terminate an existing conversation.

Using our birth/death formula for the π_i, we find that

$$
\pi_k = \begin{cases} \pi_0 \frac{A^k}{k!}, & \text{k} \le \text{n-g} \\ \pi_0 \frac{A^{n-g}}{k!} A_1^{k-(n-g)}, & \text{k} \ge \text{n-g} \end{cases}
\tag{11.28}
$$

where $A = \lambda/\mu$, $A_1 = \lambda_2/\mu$ and

$$
\pi_0 = \left[\sum_{k=0}^{n-g-1} \frac{A^k}{k!} + \sum_{k=n-g}^{n} \frac{A^{n-g}}{k!} A_1^{k-(n-g)} \right]^{-1}
\tag{11.29}
$$

11.7.2   Going Beyond Finding the π

One can calculate a number of interesting quantities from the π_i:

- The probability of a handoff call being rejected is π_n.

- The probability of a new call being blocked is

$$
\sum_{k=n-g}^{n} \pi_k
\tag{11.30}
$$

- Since the per-channel utilization in state i is i/n, the overall long-run per-channel utilization is

$$
\sum_{i=0}^{n} \pi_i \frac{i}{n}
\tag{11.31}
$$

- The long-run proportion of accepted calls which are handoff calls is the rate at which handoff calls are accepted, divided by the rate at which calls are accepted:

$$
\frac{\lambda_2 \sum_{i=0}^{n-1} \pi_i}{\lambda_1 \sum_{i=0}^{n-g-1} \pi_i + \lambda_2 \sum_{i=0}^{n-1} \pi_i}
\tag{11.32}
$$

Chapter 12

Covariance and Random Vectors

Most applications of probability and statistics involve the <u>interaction</u> between variables. For instance, when you buy a book at Amazon.com, the software will likely inform you of other books that people bought in conjunction with the one you selected. Amazon is relying on the fact that sales of certain pairs or groups of books are correlated.

Thus we need the notion of distributions that describe how two or more variables vary together. This chapter develops that notion, **which forms the very core of statistics**.

12.1 Measuring Co-variation of Random Variables

12.1.1 Covariance

Definition 21 *The* **covariance** *between random variables X and Y is defined as*

$$Cov(X, Y) = E[(X - EX)(Y - EY)] \tag{12.1}$$

Suppose that typically when X is larger than its mean, Y is also larger than its mean, and vice versa for below-mean values. Then (12.1) will likely be positive. In other words, if X and Y are positively correlated (a term we will define formally later but keep intuitive for now), then their covariance is positive. Similarly, if X is often smaller than its mean whenever Y is larger than its mean, the covariance and correlation between them will be negative. All of this is roughly speaking, of course, since it depends on *how much* and *how often* X is larger or smaller than its mean, etc.

Linearity in both arguments:

$$Cov(aX + bY, cU + dV) = acCov(X,U) + adCov(X,V) + bcCov(Y,U) + bdCov(Y,V) \quad (12.2)$$

for any constants a, b, c and d.

Insensitivity to additive constants:

$$Cov(X, Y + q) = Cov(X, Y) \quad (12.3)$$

for any constant q and so on.

Covariance of a random variable with itself:

$$Cov(X, X) = Var(X) \quad (12.4)$$

for any X with finite variance.

Shortcut calculation of covariance:

$$Cov(X, Y) = E(XY) - EX \cdot EY \quad (12.5)$$

The proof will help you review some important issues, namely (a) E(U+V) = EU + EV, (b) E(cU) = c EU and Ec = c for any constant c, and (c) EX and EY are constants in (12.5).

$$
\begin{aligned}
Cov(X,Y) &= E[(X - EX)(Y - EY)] \quad \text{(definition)} & (12.6) \\
&= E\left[XY - EX \cdot Y - EY \cdot X + EX \cdot EY\right] \quad \text{(algebra)} & (12.7) \\
&= E(XY) + E[-EX \cdot Y] + E[-EY \cdot X] + E[EX \cdot EY] \quad \text{(E[U+V]=EU+EV)} & (12.8) \\
&= E(XY) - EX \cdot EY \quad \text{(E[cU] = cEU, Ec = c)} & (12.9)
\end{aligned}
$$

Variance of sums:

$$Var(X + Y) = Var(X) + Var(Y) + 2Cov(X, Y) \quad (12.10)$$

This comes from (12.5), the relation $Var(X) = E(X^2) - EX^2$ and the corresponding one for Y. Just substitute and do the algebra.

By induction, (12.10) generalizes for more than two variables:

$$Var(W_1 + ... + W_r) = \sum_{i=1}^{r} Var(W_i) + 2 \sum_{1 \leq j < i \leq r} Cov(W_i, W_j) \tag{12.11}$$

12.1.2 Example: Variance of Sum of Nonindependent Variables

Consider random variables X_1 and X_2, for which $Var(X_i) = 1.0$ for i = 1,2, and $Cov(X_1, X_2) = 0.5$. Let's find $Var(X_1 + X_2)$.

This is quite straightforward, from (12.10):

$$Var(X_1 + X_2) = Var(X_1) + Var(X_2) + 2Cov(X_1, X_2) = 3 \tag{12.12}$$

12.1.3 Example: the Committee Example Again

Let's find Var(M) in the committee example of Section 3.9.3. In (3.97), we wrote M as a sum of indicator random variables:

$$M = G_1 + G_2 + G_3 + G_4 \tag{12.13}$$

and found that

$$P(G_i = 1) = \frac{2}{3} \tag{12.14}$$

for all i.

You should review why this value is the same for all i, as this reasoning will be used again below. Also review Section 3.9.

Applying (12.11) to (12.13), we have

$$Var(M) = 4Var(G_1) + 12Cov(G_1.G_2) \tag{12.15}$$

Finding that first term is easy, from (3.79):

$$Var(G_1) = \frac{2}{3} \cdot \left(1 - \frac{2}{3}\right) = \frac{2}{9} \tag{12.16}$$

Now, what about $Cov(G_1.G_2)$? Equation (12.5) will be handy here:

$$Cov(G_1.G_2) = E(G_1G_2) - E(G_1)E(G_2) \tag{12.17}$$

That first term in (12.17) is

$$
\begin{aligned}
E(G_1G_2) &= P(G_1 = 1 \text{ and } G_2 = 1) && \text{(12.18)} \\
&= P(\text{choose a man on both the first and second pick}) && \text{(12.19)} \\
&= \frac{6}{9} \cdot \frac{5}{8} && \text{(12.20)} \\
&= \frac{5}{12} && \text{(12.21)}
\end{aligned}
$$

That second term in (12.17) is, again from Section 3.9,

$$\left(\frac{2}{3}\right)^2 = \frac{4}{9} \tag{12.22}$$

All that's left is to put this together in (12.15), left to the reader.

12.2 Correlation

Covariance does measure how much or little X and Y vary together, but it is hard to decide whether a given value of covariance is "large" or not. For instance, if we are measuring lengths in feet and change to inches, then (12.2) shows that the covariance will increase by $12^2 = 144$. Thus it makes sense to scale covariance according to the variables' standard deviations. Accordingly, the *correlation* between two random variables X and Y is defined by

$$\rho(X,Y) = \frac{Cov(X,Y)}{\sqrt{Var(X)}\sqrt{Var(Y)}} \tag{12.23}$$

So, correlation is unitless, i.e. does not involve units like feet, pounds, etc.

It is shown later in this chapter that

- $-1 \leq \rho(X,Y) \leq 1$

- $|\rho(X,Y)| = 1$ if and only if X and Y are exact linear functions of each other, i.e. Y = cX + d for some constants c and d

So the scaling of covariance not only gave a dimensionless (i.e. unitless) quantity, but also one contained within [-1,1]. This helps us recognize what a "large" correlation is, vs. a small one.

12.2.1 Example: a Catchup Game

Consider the following simple game. There are two players, who take turns playing. One's position after k turns is the sum of one's winnings in those turns. Basically, a turn consists of generating a random U(0,1) variable, with one difference—if that player is currently losing, he gets a bonus of 0.2 to help him catch up.

Let X and Y be the total winnings of the two players after 10 turns. Intuitively, X and Y should be positively correlated, due to the 0.2 bonus which brings them closer together. Let's see if this is true.

Though very simply stated, this problem is far too tough to solve mathematically in an elementary course (or even an advanced one). So, we will use simulation. In addition to finding the correlation between X and Y, we'll also find $F_{X,Y}(5.8, 5.2) = P(X \le 5.8 \text{ and } Y \le 5.2)$.

```
1   taketurn <- function(a,b) {
2      win <- runif(1)
3      if (a >= b) return(win)
4      else return(win+0.2)
5   }
6
7   nturns <- 10
8   xyvals <- matrix(nrow=nreps,ncol=2)
9   for (rep in 1:nreps) {
10      x <- 0
11      y <- 0
12      for (turn in 1:nturns) {
13         # x's turn
14         x <- x + taketurn(x,y)
15         # y's turn
16         y <- y + taketurn(y,x)
17      }
18      xyvals[rep,] <- c(x,y)
19   }
20   print(cor(xyvals[,1],xyvals[,2]))
```

The output is 0.65. So, X and Y are indeed positively correlated as we had surmised.

Note the use of R's built-in function **cor()** to compute correlation, a shortcut that allows us to avoid summing all the products **xy** and so on, from (12.5). The reader should make sure he/she understands how this would be done.

12.3   Sets of Independent Random Variables

Recall from Section 3.3:

Definition 22 *Random variables X and Y are said to be* **independent** *if for any sets I and J, the events {X is in I} and {Y is in J} are independent, i.e. P(X is in I and Y is in J) = P(X is in I) P(Y is in J).*

Intuitively, though, it simply means that knowledge of the value of X tells us nothing about the value of Y, and vice versa.

Great mathematical tractability can be achieved by assuming that the X_i in a random vector $X = (X_1, ..., X_k)$ are independent. In many applications, this is a reasonable assumption.

12.3.1   Properties

In the next few sections, we will look at some commonly-used properties of sets of independent random variables. For simplicity, consider the case k = 2, with X and Y being independent (scalar) random variables.

12.3.1.1   Expected Values Factor

If X and Y are independent, then

$$E(XY) = E(X)E(Y) \tag{12.24}$$

12.3.1.2   Covariance Is 0

If X and Y are independent, we have

$$Cov(X, Y) = 0 \tag{12.25}$$

and thus

$\rho(X, Y) = 0$ as well.

This follows from (12.24) and (12.5).

However, the converse is false. A counterexample is the random pair (X, Y) that is uniformly distributed on the unit disk, $\{(s, t) : s^2 + t^2 \leq 1\}$. Clearly $0 = E(XY) = EX = EY$ due to the symmetry of the distribution about (0,0), so Cov(X,Y) = 0 by (12.5).

But X and Y just as clearly are not independent. If for example we know that $X > 0.8$, say, then $Y^2 < 1 - 0.8^2$ and thus $|Y| < 0.6$. If X and Y were independent, knowledge of X should not tell us anything about Y, which is not the case here, and thus they are not independent. If we also know that X and Y are bivariate normally distributed (Section 13.5.2.1), then zero covariance does imply independence.

12.3.1.3 Variances Add

If X and Y are independent, then we have

$$Var(X + Y) = Var(X) + Var(Y). \tag{12.26}$$

This follows from (12.10) and (12.24).

12.3.2   Examples Involving Sets of Independent Random Variables

12.3.2.1   Example: Dice

In Section 12.2.1, we speculated that the correlation between X, the number on the blue die, and S, the total of the two dice, was positive. Let's compute it.

Write S = X + Y, where Y is the number on the yellow die. Then using the properties of covariance presented above, we have that

$$
\begin{aligned}
Cov(X, S) &= Cov(X, X + Y) \ \ (\text{def. of S}) & (12.27) \\
&= Cov(X, X) + Cov(X, Y) \ \ (\text{from (12.2)}) & (12.28) \\
&= Var(X) + 0 \ \ (\text{from (12.4), (12.25)}) & (12.29)
\end{aligned}
$$

Also, from (12.26),

$$Var(S) = Var(X + Y) = Var(X) + Var(Y) \tag{12.30}$$

But Var(Y) = Var(X). So the correlation between X and S is

$$\rho(X, S) = \frac{Var(X)}{\sqrt{Var(X)}\sqrt{2Var(X)}} = 0.707 \tag{12.31}$$

Since correlation is at most 1 in absolute value, 0.707 is considered a fairly high correlation. Of course, we did expect X and S to be highly correlated.

12.3.2.2   Example: Variance of a Product

Suppose X_1 and X_2 are independent random variables with $EX_i = \mu_i$ and $Var(X_i) = \sigma_i^2$, i = 1,2. Let's find an expression for $Var(X_1 X_2)$.

$$
\begin{aligned}
Var(X_1 X_2) &= E(X_1^2 X_2^2) - [E(X_1 X_2)]^2 \quad (3.41) & (12.32)\\
&= E(X_1^2) \cdot E(X_2^2) - \mu_1^2 \mu_2^2 \quad (12.24) & (12.33)\\
&= (\sigma_1^2 + \mu_1^2)(\sigma_2^2 + \mu_2^2) - \mu_1^2 \mu_2^2 & (12.34)\\
&= \sigma_1^2 \sigma_2^2 + \mu_1^2 \sigma_2^2 + \mu_2^2 \sigma_1^2 & (12.35)
\end{aligned}
$$

Note that $E(X_1^2) = \sigma_1^2 + \mu_1^2$ by virtue of (3.41).

12.3.2.3   Example: Ratio of Independent Geometric Random Variables

Suppose X and Y are independent geometrically distributed random variables with success probability p. Let Z = X/Y. We are interested in EZ and F_Z.

First, by (12.24), we have

$$
\begin{aligned}
EZ &= E(X \cdot \frac{1}{Y}) & (12.36)\\
&= EX \cdot E(\frac{1}{Y}) \quad (12.24) & (12.37)\\
&= \frac{1}{p} \cdot E(\frac{1}{Y}) \quad \text{(mean of geom is 1/p)} & (12.38)
\end{aligned}
$$

So we need to find E(1/Y). Using (3.36), we have

$$E(\frac{1}{Y}) = \sum_{i=1}^{\infty} \frac{1}{i}(1-p)^{i-1}p \qquad (12.39)$$

Unfortunately, no further simplification seems possible.

Now let's find $F_Z(m)$ for a positive integer m.

$$
\begin{aligned}
F_Z(m) &= P\left(\frac{X}{Y} \le m\right) & (12.40) \\
&= P(X \le mY) & (12.41) \\
&= \sum_{i=1}^{\infty} P(Y=i)\ P(X \le mY | Y=i) & (12.42) \\
&= \sum_{i=1}^{\infty} (1-p)^{i-1}p\ P(X \le mi) & (12.43) \\
&= \sum_{i=1}^{\infty} (1-p)^{i-1}p\ [1-(1-p)^{mi}] & (12.44)
\end{aligned}
$$

this last step coming from (4.16).

We can actually reduce (12.44) to closed form, by writing

$$(1-p)^{i-1}(1-p)^{mi} = (1-p)^{mi+i-1} = \frac{1}{1-p}\left[(1-p)^{m+1}\right]^{i} \qquad (12.45)$$

and then using (4.6). Details are left to the reader.

12.4 Matrix Formulations

(Note that there is a review of matrix algebra in Appendix B.)

In your first course in matrices and linear algebra, your instructor probably motivated the notion of a matrix by using an example involving linear equations, as follows.

Suppose we have a system of equations

$$a_{i1}x_1 + ... + a_{in}x_n = b_i, \quad i = 1, ..., n, \tag{12.46}$$

where the x_i are the unknowns to be solved for.

This system can be represented compactly as

$$AX = B, \tag{12.47}$$

where A is nxn and X and B are nx1.

That compactness coming from the matrix formulation applies to statistics too, though in different ways, as we will see. (Linear algebra in general is used widely in statistics—matrices, rank and subspace, eigenvalues, even determinants.)

When dealing with multivariate distributions, some very messy equations can be greatly compactified through the use of matrix algebra. We will introduce this here.

Throughout this section, consider a random vector $W = (W_1, ..., W_k)'$ where $'$ denotes matrix transpose, and a vector written horizontally like this without a $'$ means a row vector.

12.4.1   Properties of Mean Vectors

In statistics, we frequently need to find covariance matrices of linear combinations of random vectors.

Definition 23 *The expected value of W is defined to be the vector*

$$EW = (EW_1, ..., EW_k)' \tag{12.48}$$

The linearity of the components implies that of the vectors:

For any scalar constants c and d, and any random vectors V and W, we have

$$E(cV + dW) = cEV + dEW \tag{12.49}$$

where the multiplication and equality is now in the vector sense.

Also, multiplication by a constant matrix factors:

If A is a nonrandom matrix having k columns, then

$$E(AW) = AEW \tag{12.50}$$

12.4.2 Covariance Matrices

In moving from random variables, which we dealt with before, to random vectors, we now see that expected value carries over as before. What about variance? The proper extension is the following.

Definition 24 *The covariance matrix Cov(W) of $W = (W_1, ..., W_k)'$ is the k x k matrix whose $(i, j)^{th}$ element is $Cov(W_i, W_j)$.*

Note that that implies that the diagonal elements of the matrix are the variances of the W_i, and that the matrix is symmetric.

As you can see, in the statistics world, the Cov() notation is "overloaded." If it has two arguments, it is ordinary covariance, between two variables. If it has one argument, it is the covariance matrix, consisting of the covariances of all pairs of components in the argument. When people mean the matrix form, they always say so, i.e. they say "covariance MATRIX" instead of just "covariance."

The covariance matrix is just a way to compactly do operations on ordinary covariances. Here are some important properties:

Say c is a constant scalar. Then cW is a k-component random vector like W, and

$$Cov(cW) = c^2 Cov(W) \tag{12.51}$$

Suppose V and W are independent random vectors, meaning that each component in V is independent of each component of W. (But this does NOT mean that the components within V are independent of each other, and similarly for W.) Then

$$Cov(V + W) = Cov(V) + Cov(W) \tag{12.52}$$

Of course, this is also true for sums of any (nonrandom) number of independent random vectors.

In analogy with (3.41), for any random vector Q,

$$Cov(Q) = E(QQ') - EQ \ (EQ)' \tag{12.53}$$

12.4.3   Covariance Matrices Linear Combinations of Random Vectors

Suppose A is an r x k but nonrandom matrix. Then AW is an r-component random vector, with its ith element being a linear combination of the elements of W. Then one can show that

$$Cov(AW) = A\ Cov(W)\ A' \tag{12.54}$$

An important special case is that in which A consists of just one row. In this case AW is a vector of length 1—a scalar! And its covariance matrix, which is of size 1×1, is thus simply the variance of that scalar. In other words:

> Suppose we have a random vector $U = (U_1, ..., U_k)'$ and are interested in the variance of a linear combination of the elements of U,
>
> $$Y = c_1 U_1 + ... + c_k U_k \tag{12.55}$$
>
> for a vector of constants $c = (c_1, ..., c_k)'$.
>
> Then
>
> $$Var(Y) = c' Cov(U) c \tag{12.56}$$

Here are the details: (12.55) is, in matrix terms, AU, where A is the one-row matrix consisting of c'. Thus (12.54) gives us the right-hand side of (12.55) What about the left-hand side?

In this context, Y is the one-element vector (Y_1). So, its covariance matrix is of size 1×1, and it sole element is, according to Definition 24, $Cov(Y_1, Y_1)$. But that is $Cov(Y, Y) = Var(X)$.

12.4.4   Example: (X,S) Dice Example Again

Recall Sec. 12.3.2.1. We rolled two dice, getting X and Y dots, and set S to X+Y. We then found $\rho(X, S)$. Let's find $\rho(X, S)$ using matrix methods.

The key is finding a proper choice for A in (12.54). A little thought shows that

$$\begin{pmatrix} X \\ S \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} X \\ Y \end{pmatrix} \tag{12.57}$$

Thus the covariance matrix of (X,S)' is

$$
Cov[(X,S)'] = \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} Var(X) & 0 \\ 0 & Var(Y) \end{pmatrix} \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} \tag{12.58}
$$

$$
= \begin{pmatrix} Var(X) & 0 \\ Var(X) & Var(Y) \end{pmatrix} \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} \tag{12.59}
$$

$$
= \begin{pmatrix} Var(X) & Var(X) \\ Var(X) & Var(X) + Var(Y) \end{pmatrix} \tag{12.60}
$$

since X and Y are independent. We would then proceed as before.

This matches what we found earlier, as it should, but shows how matrix methods can be used. This example was fairly simple, so those methods did not produce a large amount of streamlining, but in other examples later in the book, the matrix approach will be key.

12.4.5   Example: Easy Sum Again

Let's redo the example in Section 12.1.2 again, this time using matrix methods.

First note that

$$
X_1 + X_2 = (1,1) \begin{pmatrix} X_1 \\ X_2 \end{pmatrix} \tag{12.61}
$$

i.e. it is of the form (12.55). So, (12.56) gives us

$$
Var(X_1 + X_2) = (1,1) \begin{pmatrix} 1 & 0.5 \\ 0.5 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \end{pmatrix} = 3 \tag{12.62}
$$

Of course using the matrix formulation didn't save us much time here, but for complex problems it's invaluable. We will frequently have need for finding the variance of a linear combination of the elements of a vector, exactly what we did above.

12.5   The Multivariate Normal Family of Distributions

This is a generalization of the normal distribution. It is covered in detail in Section 13.5.2, but here is the overview:

- Just as the univariate normal family is parameterized by the mean and variance, the multivariate normal family has as its parameters the mean *vector* and the covariance *matrix*.

- In the bivariate case, the density looks like a three-dimensional bell, as on the cover of this book.

- If a random vector W has a multivariate normal distribution, and A is a constant matrix, then the new random vector AW is also multivariate normally distributed.

- The multivariate version of the Central Limit Theorem holds, i.e. the sum of i.i.d. random vectors has an approximate multivariate normal distribution.

12.5.1   R Functions

R provides functions that compute probabilities involving this family of distributions, in the library **mvtnorm**. In particular the R function **pmvnorm()**, which computes probabilities of "rectangular" regions for multivariate normally distributed random vectors W. The arguments we'll use for this function here are:

- **mean**: the mean vector

- **sigma**: the covariance matrix

- **lower, upper**: bounds for a multidimensional "rectangular" region of interest

Since a multivariate normal distribution is characterized by its mean vector and covariance matrix, the first two arguments above shouldn't suprise you. But what about the other two?

The function finds the probability of our random vector falling into a multidimensional rectangular region that we specify, through the arguments are **lower** and **upper**. For example, suppose we have a trivariate normally distributed random vector $(U, V, W)'$, and we want to find

$$P(1.2 < U < 5 \text{ and } -2.2 < V < 3 \text{ and } 1 < W < 10) \tag{12.63}$$

Then **lower** would be (1.2,-2.2,1) and **upper** would be (5,3,10).

Note that these will typically be specified via R's **c()** function, but default values are recycled versions of **-Inf** and **Inf**, built-in R constants for $-\infty$ and ∞.

An important special case is that in which we specify **upper** but allow **lower** to be the default values, thus computing a probability of the form

$$P(W_1 \leq c_1, ..., W_r \leq c_r) \tag{12.64}$$

12.5.2 Special Case: New Variable Is a Single Linear Combination of a Random Vector

Suppose the vector $U = (U_1, ..., U_k)'$ has an approximately k-variate normal distribution, and we form the scalar

$$Y = c_1 U_1 + ... + c_k U_k \tag{12.65}$$

Then Y is approximately univate normal, and its (exact) variance is given by (12.56). It's mean is obtained via (12.50).

We can then use the R functions for the univariate normal distribution, e.g. **pnorm()**.

12.6 Indicator Random Vectors

Let's extend the notion of indicator random variables in Section 3.9 to vectors.

Say one of events $A_1, ..., A_k$ must occur, and they are disjoint. So, their probabilities sum to 1. Define the k-component random vector I to consist of k-1 0s and one 1, where the position of the 1 is itself random; if A_i occurs, then I_i is 1.

For example, say U has a U(0,1) distribution, and say A_1, A_2 and A_3 are the events corresponding to $U < 0.2$, $0.2 \leq U \leq 0.7$ and $U > 0.7$, respectively. Then the random vector I would be $(1, 0, 0)'$ in the first case, and so on.

Let $p_i = P(A_i)$. The analogs of (3.78) and (3.79) can easily be shown to be as follows:

- The mean vector is $E(I) = (p_1, ..., p_k)'$.

- Cov(I) has $p_i(1 - p_i)$ as its ith element, and for $i \neq j$, element (i,j) is $-p_i p_j$.

The see the latter property, note that

$$
\begin{aligned}
Cov(I_i, I_j) &= E(I_i I_j) - EI_i \cdot EI_j & (12.66) \\
&= 0 - p_i p_j \quad (I_i \text{ and } I_j \text{ cannot be simultaneously 1}) & (12.67) \\
&= -p_i p_j & (12.68)
\end{aligned}
$$

12.7   Example: Dice Game

This example will really illustrate the value of using matrices.

12.7.1   Problem Statement

Suppose we roll a die 50 times. Let X denote the number of rolls in which we get one dot, and let Y be the number of times we get either two or three dots. For convenience, let's also define Z to be the number of times we get four or more dots. Suppose also that we win \$5 for each roll of a one, and \$2 for each roll of a two or three.

Let's find the approximate values of the following:

- $P(X \leq 12 \text{ and } Y \leq 16)$

- P(win more than \$90)

- $P(X > Y > Z)$

12.7.2   Exact Distribution

First, the distribution of $(X, Y, Z)'$ belongs to yet another parametric family, the **multinomial** family, a generalization of the binomial. Think about it: The "bi" in "binomial" alludes to the fact that each trial has two outcomes, e.g. heads or tails. Here in the dice example, we have three outcomes — one dot, two or three dots, and four or more dots. So we have a *trinomial* distribution.

The pmf for the multinomial family is easy to derive. Recall the intuition behind (4.25),

$$p_X(k) = P(X = k) = \binom{n}{k} p^k (1 - p)^{n-k} \tag{12.69}$$

There are $\binom{n}{k}$ different ways to get k successes out of n trials, with each such pattern having probability $p^k (1 - p)^{n-k}$. In the multinomial case, the number of patterns is given in Section 2.15.5. So, for an m-category multinomial random vector $(W_1, ..., W_m)'$, we have for any set of s_i summing to n, the number of trials,

$$P(W_1 = s_1, ..., W_m = s_m) = \frac{n!}{s_1! ... s_m!} \, p_1^{s_1} ... p_m^{s_m}, \quad s_1 + ... + s_m = n \tag{12.70}$$

Here in our die example, $m = 3$, and the p_i are 1/6, 2/6 and 3/6. So for instance,

$$P(X = 12, Y = 22, Z = 16) = \frac{50!}{12!22!16!} \ (\frac{1}{6})^{12}(\frac{2}{6})^{22}(\frac{3}{6})^{16} \tag{12.71}$$

Thus the exact probabilities listed in Section 12.7.1 above could be calculated in this way. But that would be cumbersome, too many possibilities. But we can get approximate answers by noting that the triple (X,Y,Z) has an approximate multivariate (trivariate) normal distribution.

12.7.3 Multinomial-Family Random Vectors Are Approximately Multivariate Normally Distributed

To see the connection, recall from (4.26) that a binomial random variable can be expressed as a sum of indicator random variables. Similarly, a multinomial random *vector* such as $(X, Y, Z)'$ can be expressed as a sum or random indicator *vectors*, as in Section 12.6. And there is a multivariate version of the Central Limit Theorem, which says sums are approximately normal. Result:

The multinomial random vector W is approximately multivariate normally distributed. By (12.49), (12.52) and Section 12.6, we have

$$EW = n(p_1, ..., p_m)' \tag{12.72}$$

$$[Cov(W)]_{ij} = \begin{cases} np_i(1 - p_i), & i = j, \\ -np_ip_j, & i \neq j \end{cases} \tag{12.73}$$

12.7.4 Finding the Die Game Probabilities

Now that we know that $(X, Y, Z)'$ is approximately trivariate normal, we need merely call the R function **pmvnorm()**. It requires the mean vector and covariance matrix, which from (12.72) and (12.73) we know to be

$$E[(X, Y, Z)] = (50/6, 50/3, 50/2) \tag{12.74}$$

and

$$Cov[(X, Y, Z)] = 50 \begin{pmatrix} 5/36 & -1/18 & -1/12 \\ -1/18 & 2/9 & -1/6 \\ -1/12 & -1/6 & 1/4 \end{pmatrix} \tag{12.75}$$

To account for the integer nature of X and Y, we call the function with upper limits of 12.5 and 16.5, rather than 12 and 16, which is often used to get a better approximation. (Recall the "correction for continuity," Section 8.14.) Our code is

```
1   p1 <- 1/6
2   p23 <- 1/3
3   meanvec <- 50*c(p1,p23)
4   var1 <- 50*p1*(1-p1)
5   var23 <- 50*p23*(1-p23)
6   covar123 <- -50*p1*p23
7   covarmat <- matrix(c(var1,covar123,covar123,var23),nrow=2)
8   print(pmvnorm(upper=c(12.5,16.5),mean=meanvec,sigma=covarmat))
```

(Note: The above code is exact, not a simulation.[1])

We find that

$$P(X \leq 12 \text{ and } Y \leq 16) \approx 0.43 \tag{12.76}$$

Now, let's find the probability that our total winnings, T, is over \$90. We know that T = 5X + 2Y, and Section 12.5.2 above applies. We simply choose the vector **c** to be

$$c = (5, 2, 0)' \tag{12.77}$$

since

$$(5, 2, 0) \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = 5X + 2Y \tag{12.78}$$

Then Section 12.5.2 tells us that 5X + 2Y also has an approximate univariate normal distribution. Excellent—we can now use **pnorm()**. We thus need the mean and variance of T, again using Section 12.5.2:

$$ET = E(5X + 2Y) = 5EX + 2EY = 250/6 + 100/3 = 75 \tag{12.79}$$

$$Var(T) = c' \; Cov \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} c = (5, 2, 0) \; 50 \begin{pmatrix} 5/36 & -1/18 & -1/12 \\ -1/18 & 2/9 & -1/6 \\ -1/12 & -1/6 & 1/4 \end{pmatrix} \begin{pmatrix} 5 \\ 2 \\ 0 \end{pmatrix} = 162.5 \tag{12.80}$$

---

[1]Granted, the function **pmvnorm()** does numerical integration, and thus gives an approximate value, but it is not a simulation, which generates random numbers.

So, we have our answer:

```
> 1 - pnorm(90,75,sqrt(162.5))
[1] 0.1196583
```

Now to find $P(X > Y > Z)$, we need to work with $(U, V)' = (X - Y, Y - Z)$. U and V are both linear functions of X, Y and Z, so let's write the matrix equation:

We need to have

$$\begin{pmatrix} X - Y \\ Y - Z \end{pmatrix} = A \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} \tag{12.81}$$

so set

$$A = \begin{pmatrix} 1 & -1 & 0 \\ 0 & 1 & -1 \end{pmatrix} \tag{12.82}$$

and then proceed as before to find $P(U > 0, V > 0)$. Now we take **lower** to be (0,0), and **upper** to be the default values, ∞ in **pmvnorm()**.

Chapter 13

Multivariate PMFs and Densities

Individual pmfs p_X and densities f_X don't describe correlations betwen variables. We need something more. We need ways to describe multivariate distributions.

13.1   Multivariate Probability Mass Functions

Recall that for a single discrete random variable X, the distribution of X was defined to be a list of all the values of X, together with the probabilities of those values. The same is done for a pair of discrete random variables U and V, as follows.

Suppose we have a bag containing two yellow marbles, three blue ones and four green ones. We choose four marbles from the bag at random, without replacement. Let Y and B denote the number of yellow and blue marbles that we get. Then define the *two-dimensional* pmf of Y and B to be

$$p_{Y,B}(i,j) = P(Y = i \text{ and } B = j) = \frac{\binom{2}{i}\binom{3}{j}\binom{4}{4-i-j}}{\binom{9}{4}} \tag{13.1}$$

Here is a table displaying all the values of P(Y = i and B = j):

i ↓, j →	0	1	2	3
0	0.0079	0.0952	0.1429	0.0317
1	0.0635	0.2857	0.1905	0.1587
2	0.0476	0.0952	0.0238	0.000

So this table is the distribution of the pair (Y,B).

Recall further that in the discrete case, we introduced a symbolic notation for the distribution of

a random variable X, defined as $p_X(i) = P(X = i)$, where i ranged over all values that X takes on. We do the same thing for a pair of random variables:

Definition 25 *For discrete random variables U and V, their probability mass function is defined to be*

$$p_{U,V}(i, j) = P(U = i \ and \ V = j) \tag{13.2}$$

where (i,j) ranges over all values taken on by (U,V). Higher-dimensional pmfs are defined similarly, e.g.

$$p_{U,V,W}(i, j, k) = P(U = i \ and \ V = j \ and \ W = k) \tag{13.3}$$

So in our marble example above, $p_{Y,B}(1, 2) = 0.048$, $p_{Y,B}(2, 0) = 0.012$ and so on.

Just as in the case of a single discrete random variable X we have

$$P(X \in A) = \sum_{i \in A} p_X(i) \tag{13.4}$$

for any subset A of the range of X, for a discrete pair (U,V) and any subset A of the pair's range, we have

$$P[(U, V) \in A] = \sum_{(i,j) \in A} \sum p_{U,V}(i, j) \tag{13.5}$$

Again, consider our marble example. Suppose we want to find $P(Y < B)$. Doing this "by hand," we would simply sum the relevant probabilities in the table above, which are marked in bold face below:

i ↓, j →	0	1	2	3
0	0.002	**0.024**	**0.036**	**0.008**
1	0.162	0.073	**0.048**	**0.004**
2	0.012	0.024	0.006	**0.000**

The desired probability would then be 0.024+0.036+0.008+0.048+0.004 = 0.12.

Writing it in the more formal way using (13.5), we would set

$$A = \{(i, j) : i < j\} \tag{13.6}$$

and then

$$P(Y < B) = P[(Y, B) \in A] = \sum_{i=0}^{2} \sum_{j=i+1}^{3} p_{Y,B}(i,j) \tag{13.7}$$

Note that the lower bound in the inner sum is j = i+1. This reflects the common-sense point that in the event $Y < B$, B must be at least equal to Y+1.

Of course, this sum still works out to 0.12 as before, but it's important to be able to express this as a double sum of $p_{Y,B}()$, as above. We will rely on this to motivate the continuous case in the next section.

Expected values are calculated in the analogous manner. Recall that for a function g() of X

$$E[g(X)] = \sum_{i} g(i) p_X(i) \tag{13.8}$$

So, for any function g() of two discrete random variables U and V, define

$$E[g(U,V)] = \sum_{i} \sum_{j} g(i,j) p_{U,V}(i,j) \tag{13.9}$$

For instance, if for some bizarre reason we wish to find the expected value of the product of the numbers of yellow and blue marbles above,[1] the calculation would be

$$E(YB) = \sum_{i=0}^{2} \sum_{j=0}^{3} ij \ p_{Y,B}(i,j) = 0.255 \tag{13.10}$$

The univariate pmfs, called *marginal pmfs*, can of course be recovered from the multivariate pmf:

$$p_U(i) = P(U = i) = \sum_{j} P(U = i, V = j) = \sum_{j} p_{U,V}(i,j) \tag{13.11}$$

For example, look at the table following (13.5). Evaluating (13.11) for i = 1, say, with U = Y and V = B, would give us 0.012 + 0.024 + 0.006 + 0.000 = 0.042. Then all that (13.11) tells us is the P(Y = 1) = 0.042, which is obvious from the table; (13.11) simply is an application of our old principle, "Break big events down into small events."

---

[1]Not so bizarre, we'll find in Section 12.1.1.

Needless to say, we can recover the marginal distribution of V similarly to (13.11):

$$p_V(j) = P(V = j) = \sum_i P(U = i, V = j) = \sum_i p_{U,V}(i, j) \tag{13.12}$$

13.2 Multivariate Densities

13.2.1 Motivation and Definition

Extending our previous definition of cdf for a single variable, we define the two-dimensional cdf for a pair of random variables X and Y (discrete or continuous) as

$$F_{X,Y}(u, v) = P(X \leq u \text{ and } Y \leq v) \tag{13.13}$$

If X and Y were discrete, we would evaluate that cdf via a double sum of their bivariate pmf. You may have guessed by now that the analog for continuous random variables would be a double integral, and it is. The integrand is the bivariate density:

$$f_{X,Y}(u, v) = \frac{\partial^2}{\partial u \partial v} F_{X,Y}(u, v) \tag{13.14}$$

Densities in higher dimensions are defined similarly.[2]

As in the univariate case, a bivariate density shows which regions of the X-Y plane occur more frequently, and which occur less frequently.

13.2.2 Use of Multivariate Densities in Finding Probabilities and Expected Values

Again by analogy, for any region A in the X-Y plane,

$$P[(X, Y)\epsilon A] = \iint\limits_A f_{X,Y}(u, v) \; du \; dv \tag{13.15}$$

---

[2]Just as we noted in Section **??** that some random variables are neither discrete nor continuous, there are some pairs of continuous random variables whose cdfs do not have the requisite derivatives. We will not pursue such cases here.

So, just as probabilities involving a single variable X are found by integrating f_X over the region in question, for probabilities involving X and Y, we take the double integral of $f_{X,Y}$ over that region.

Also, for any function g(X,Y),

$$E[g(X,Y)] = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} g(u,v) f_{X,Y}(u,v) \ du \ dv \tag{13.16}$$

where it must be kept in mind that $f_{X,Y}(u,v)$ may be 0 in some regions of the U-V plane. Note that there is no set A here as in (13.15). See (13.20) below for an example.

Finding marginal densities is also analogous to the discrete case, e.g.

$$f_X(s) = \int_t f_{X,Y}(s,t) \ dt \tag{13.17}$$

Other properties and calculations are analogous as well. For instance, the double integral of the density is equal to 1, and so on.

13.2.3   Example: a Triangular Distribution

Suppose (X,Y) has the density

$$f_{X,Y}(s,t) = 8st, \ 0 < t < s < 1 \tag{13.18}$$

The density is 0 outside the region $0 < t < s < 1$.

First, think about what this means, say in our notebook context. We do the experiment many times. Each line of the notebook records the values of X and Y. Each of these (X,Y) pairs is a point in the triangular region $0 < t < s < 1$. Since the density is highest near the point (1,1) and lowest near (0,0), (X,Y) will be observed near (1,1) much more often than near (0,0), with points near, say, (1,0.5) occurring with middling frequencies.

Let's find $P(X + Y > 1)$. This calculation will involve a double integral. The region A in (13.15) is $\{(s,t) : s + t > 1, 0 < t < s < 1\}$. We have a choice of integrating in the order ds dt or dt ds. The latter will turn out to be more convenient.

To see how the limits in the double integral are obtained, first review (13.7). We use the same reasoning here, changing from sums to integrals and applying the current density, as shown in this figure:
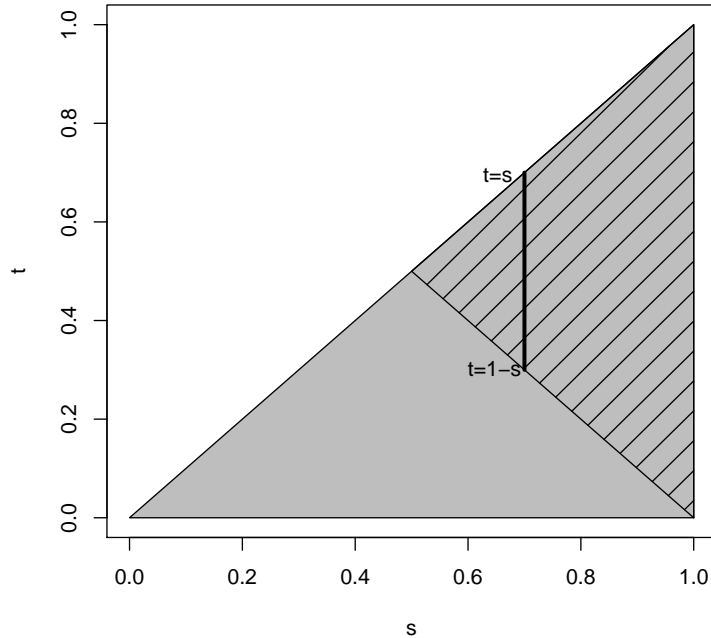
Here s represents X and t represents Y. The gray area is the region in which (X,Y) ranges. The subregion A in (13.15), corresponding to the event X+Y > 1, is shown in the striped area in the figure.

The dark vertical line shows all the points (s,t) in the striped region for a typical value of s in the integration process. Since s is the variable in the outer integral, considered it fixed for the time being and ask where t will range *for that s*. We see that for X = s, Y will range from 1-s to s; thus we set the inner integral's limits to 1-s and s. Finally, we then ask where s can range, and see from the picture that it ranges from 0.5 to 1. Thus those are the limits for the outer integral.

$$P(X + Y > 1) = \int_{0.5}^{1} \int_{1-s}^{s} 8st \ dt \ ds = \int_{0.5}^{1} 8s \cdot (s - 0.5) \ ds = \frac{5}{6} \tag{13.19}$$

Following (13.16),

$$E[\sqrt{X + Y}] = \int_{0}^{1} \int_{0}^{s} \sqrt{s + t} \ 8st \ dt \ ds \tag{13.20}$$

Let's find the marginal density $f_Y(t)$. Just as we "summed out" in (13.11), in the continuous case

we must "integrate out" the s in (13.18):

$$f_Y(t) = \int_t^1 8st \ ds = 4t - 4t^3 \tag{13.21}$$

for $0 < t < 1$, 0 elsewhere.

Let's find the correlation between X and Y for this density.

$$
\begin{aligned}
E(XY) &= \int_0^1 \int_0^s st \cdot 8st \ dt \ ds & \text{(13.22)} \\
&= \int_0^1 8s^2 \cdot s^3/3 \ ds & \text{(13.23)} \\
&= \frac{4}{9} & \text{(13.24)}
\end{aligned}
$$

$$
\begin{aligned}
f_X(s) &= \int_0^s 8st \ dt & \text{(13.25)} \\
&= 4st^2 \Big|_0^s & \text{(13.26)} \\
&= 4s^3 & \text{(13.27)}
\end{aligned}
$$

$$
\begin{aligned}
f_Y(t) &= \int_t^1 8st \ ds & \text{(13.28)} \\
&= 4t \cdot s^2 \Big|_t^1 & \text{(13.29)} \\
&= 4t(1 - t^2) & \text{(13.30)}
\end{aligned}
$$

$$EX = \int_0^1 s \cdot 4s^3 \ ds = \frac{4}{5} \tag{13.31}$$

$$E(X^2) = \int_0^1 s^2 \cdot 4s^3 \ ds = \frac{2}{3} \tag{13.32}$$

$$Var(X) = \frac{2}{3} - \left(\frac{4}{5}\right)^2 = 0.027 \tag{13.33}$$

$$EY = \int_0^1 t \cdot (4t - 4t^3) \ ds = \frac{4}{3} - \frac{4}{5} = \frac{8}{15} \tag{13.34}$$

$$E(Y^2) = \int_0^1 t^2 \cdot (4t - 4t^3) \ dt = 1 - \frac{4}{6} = \frac{1}{3} \tag{13.35}$$

$$Var(Y) = \frac{1}{3} - \left(\frac{8}{15}\right)^2 = 0.049 \tag{13.36}$$

$$Cov(X, Y) = \frac{4}{9} - \frac{4}{5} \cdot \frac{8}{15} = 0.018 \tag{13.37}$$

$$\rho(X, Y) = \frac{0.018}{\sqrt{0.027 \cdot 0.049}} = 0.49 \tag{13.38}$$

13.2.4   Example: Train Rendezvouz

Train lines A and B intersect at a certain transfer point, with the schedule stating that trains from both lines will arrive there at 3:00 p.m. However, they are often late, by amounts X and Y, measured in hours, for the two trains. The bivariate density is

$$f_{X,Y}(s, t) = 2 - s - t, \ 0 < s, t < 1 \tag{13.39}$$

Two friends agree to meet at the transfer point, one taking line A and the other B. Let W denote the time in minutes the person arriving on line B must wait for the friend. Let's find $P(W > 6)$.

First, convert this to a problem involving X and Y, since they are the random variables for which we have a density, and then use (13.15):

$$
\begin{aligned}
P(W > 0.1) \ &= \ P(Y + 0.1 < X) \tag{13.40} \\
&= \ \int_{0.1}^1 \int_0^{s-0.1} (2 - s - t) \ dt ds \tag{13.41}
\end{aligned}
$$

13.3 More on Sets of Independent Random Variables

13.3.1 Probability Mass Functions and Densities Factor in the Independent Case

If X and Y are independent, then

$$p_{X,Y} = p_X p_Y \tag{13.42}$$

in the discrete case, and

$$f_{X,Y} = f_X f_Y \tag{13.43}$$

in the continuous case. In other words, the joint pmf/density is the product of the marginal ones. This is easily seen in the discrete case:

$$
\begin{aligned}
p_{X,Y}(i,j) &= P(X = i \text{ and } Y = j) \quad \text{(definition)} & (13.44)\\
&= P(X = i)P(Y = j) \quad \text{(independence)} & (13.45)\\
&= p_X(i)p_Y(j) \quad \text{(definition))} & (13.46)
\end{aligned}
$$

Here is the proof for the continuous case;

$$
\begin{aligned}
f_{X,Y}(u,v) &= \frac{\partial^2}{\partial u \partial v} F_{X,Y}(u,v) & (13.47)\\
&= \frac{\partial^2}{\partial u \partial v} P(X \leq u \text{ and } Y \leq v) & (13.48)\\
&= \frac{\partial^2}{\partial u \partial v} [P(X \leq u) \cdot P(Y \leq v)] & (13.49)\\
&= \frac{\partial^2}{\partial u \partial v} F_X(u) \cdot F_Y(v) & (13.50)\\
&= f_X(u) f_Y(v) & (13.51)
\end{aligned}
$$

13.3.2   Convolution

Definition 26 *Suppose g and h are densities of continuous random variables X and Y, respectively. The* **convolution** *of g and h, denoted g*h,[3] is another density, defined to be that of the random variable X+Y. In other words, convolution is a binary operation on the set of all densities.*

If X and Y are nonnegative and independent, then the convolution reduces to

$$f_Z(t) = \int_0^t g(s)h(t - s) \ ds \tag{13.52}$$

You can get intuition on this by considering the discrete case. Say U and V are nonnegative integer-valued random variables, and set W = U+V. Let's find p_W;

$$
\begin{aligned}
p_W(k) &= P(W = k) \quad \text{(by definition)} & (13.53)\\
&= P(U + V = k) \quad \text{(substitution)} & (13.54)\\
&= \sum_{i=0}^{k} P(U = i \text{ and } V = k - i) \quad \text{(``In what ways can it happen?'')} & (13.55)\\
&= \sum_{i=0}^{k} p_{U,V}(i, k - i) \quad \text{(by definition)} & (13.56)\\
&= \sum_{i=0}^{k} p_U(i)p_V(k - i) \quad \text{(from Section 13.3.1)} & (13.57)
\end{aligned}
$$

Review the analogy between densities and pmfs in our unit on continuous random variables, Section 7.3.2, and then see how (13.52) is analogous to (13.53) through (13.57):

- k in (13.53) is analogous to t in (13.52)

- the limits 0 to k in (13.57) are analogous to the limits 0 to t in (13.52)

- the expression k-i in (13.57) is analogous to t-s in (13.52)

- and so on

---

[3]The reason for the asterisk, suggesting a product, will become clear in Section 14.3.

13.3.3 Example: Ethernet

Consider this network, essentially Ethernet. Here nodes can send at any time. Transmission time is 0.1 seconds. Nodes can also "hear" each other; one node will not start transmitting if it hears that another has a transmission in progress, and even when that transmission ends, the node that had been waiting will wait an additional random time, to reduce the possibility of colliding with some other node that had been waiting.

Suppose two nodes hear a third transmitting, and thus refrain from sending. Let X and Y be their random backoff times, i.e. the random times they wait before trying to send. (In this model, assume that they do not do "listen before talk" after a backoff.) Let's find the probability that they clash, which is $P(|X - Y| \leq 0.1)$.

Assume that X and Y are independent and exponentially distributed with mean 0.2, i.e. they each have density $5e^{-5u}$ on $(0, \infty)$. Then from (13.43), we know that their joint density is the product of their marginal densities,

$$f_{X,Y}(s,t) = 25e^{-5(s+t)}, s, t > 0 \tag{13.58}$$

Now

$$P(|X - Y| \leq 0.1) = 1 - P(|X - Y| > 0.1) = 1 - P(X > Y + 0.1) - P(Y > X + 0.1) \tag{13.59}$$

Look at that first probability. Applying (13.15) with $A = \{(s,t) : s > t + 0.1, 0 < s, t\}$, we have

$$P(X > Y + 0.1) = \int_0^\infty \int_{t+0.1}^\infty 25e^{-5(s+t)} \, ds \, dt = 0.303 \tag{13.60}$$

By symmetry, $P(Y > X + 0.1)$ is the same. So, the probability of a clash is 0.394, rather high. We may wish to increase our mean backoff time, though a more detailed analysis is needed.

13.3.4 Example: Analysis of Seek Time

This will be an analysis of seek time on a disk. Suppose we have mapped the innermost track to 0 and the outermost one to 1, and assume that (a) the number of tracks is large enough to treat the position H of the read/write head the interval [0,1] to be a continous random variable, and (b) the track number requested has a uniform distribution on that interval.

Consider two consecutive service requests for the disk, denoting their track numbers by X and Y. In the simplest model, we assume that X and Y are independent, so that the joint distribution of X and Y is the product of their marginals, and is thus is equal to 1 on the square $0 \le X, Y \le 1$.

The seek distance will be $|X - Y|$. Its mean value is found by taking g(s,t) in (13.16) to be $|s - t|$.

$$\int_0^1 \int_0^1 |s - t| \cdot 1 \ ds \ dt = \frac{1}{3} \tag{13.61}$$

Let's find the density of the seek time $S = |X - Y|$:

$$
\begin{aligned}
F_S(v) &= P(|X - Y| \le v) & (13.62) \\
&= P(-v \le X - Y \le v) & (13.63) \\
&= 1 - P(X - Y < -v) - P(X - Y > v) & (13.64) \\
&= 1 - (1 - v)^2 & (13.65)
\end{aligned}
$$

where for instance $P(X - Y > v)$ the integral of 1 on the triangle with vertices (v,0), (1,0) and (1,1-v), thus equal to the area of that triangle, $0.5(1 - v)^2$.

Then

$$f_S(v) = \frac{d}{dt} F_S(v) = 2(1 - v) \tag{13.66}$$

By the way, what about the assumptions here? The independence would be a good assumption, for instance, for a heavily-used file server accessed by many different machines. Two successive requests are likely to be from different machines, thus independent. In fact, even within the same machine, if we have a lot of users at this time, successive requests can be assumed independent. On the other hand, successive requests from a particular user probably can't be modeled this way.

As mentioned in our unit on continuous random variables, page 132, if it's been a while since we've done a defragmenting operation, the assumption of a uniform distribution for requests is probably good.

Once again, this is just scratching the surface. Much more sophisticated models are used for more detailed work.

13.3.5 Example: Backup Battery

Suppose we have a portable machine that has compartments for two batteries. The main battery has lifetime X with mean 2.0 hours, and the backup's lifetime Y has mean life 1 hours. One replaces the first by the second as soon as the first fails. The lifetimes of the batteries are exponentially distributed and independent. Let's find the density of W, the time that the system is operational (i.e. the sum of the lifetimes of the two batteries).

Recall that if the two batteries had the same mean lifetimes, W would have a gamma distribution. But that's not the case here. But we notice that the distribution of W is a convolution of two exponential densities, as it is the sum of two nonnegative independent random variables. Using (13.3.2), we have

$$f_W(t) = \int_0^t f_X(s)f_Y(t-s) \, ds = \int_0^t 0.5e^{-0.5s}e^{-(t-s)} \, ds = e^{-0.5t} - e^{-t}, \; 0 < t < \infty \qquad (13.67)$$

13.3.6 Example: Minima of Uniformly Distributed Random Variables

Suppose X and Y be independent and each have a uniform distribution on the interval (0,1). Let Z = min(X,Y). Find f_Z:

$$
\begin{aligned}
F_Z(t) \;&=\; P(Z \le t) \quad \text{(def. of cdf)} & (13.68)\\
&=\; 1 - P(Z > t) & (13.69)\\
&=\; 1 - P(X > t \text{ and } Y > t) \quad (\min(u,v) > t \text{ iff both u,v} > t) & (13.70)\\
&=\; 1 - P(X > t)P(Y > t) \quad \text{(indep.)} & (13.71)\\
&=\; 1 - (1-t)^2 \quad \text{(indep.)} & (13.72)\\
& & (13.73)
\end{aligned}
$$

The density of Z is then the derivative of that last expression:

$$f_Z(t) = 2(1-t), \quad 0 < t < 1 \qquad (13.74)$$

13.3.7 Example: Ethernet Again

In the Ethernet example in Section 13.3.3, we assumed that transmission time was a constant, 0.1. Now let's account for messages of varying sizes, by assuming that transmission time T for a message is random, exponentially distributed with mean 0.1. Let's find $P(X < Y$ and there is no collision$)$.

That probability is equal to $P(X + T < Y)$. Well, this sounds like we're going to have to deal with triple integrals, but actually not. The derivation in Section 13.3.5 shows that the density of S = X+T is

$$f_S(t) = e^{-0.1t} - e^{-0.2t}, \ 0 < t < \infty \tag{13.75}$$

Thus the joint density of S and Y is

$$f_{S,Y}(u, v) = (e^{-0.1u} - e^{-0.2u})0.2e^{-0.2v}, \ \ 0 < u, v, < \infty \tag{13.76}$$

We can then evaluate $P(S < Y)$ as a double integral, along the same lines as we did for instance in (13.19).

13.4 Example: Finding the Distribution of the Sum of Nonindependent Random Variables

In Section 13.3.2, we found a general formula for the distribution of the sum of two independent random variables. What about the nonindependent case?

Suppose for instance $f_{X,Y}(s, t) = 2$ on $0 < t < s < 1$, 0 elsewhere. Let's find $f_{X+Y}(w)$ for the case $0 < w < 1$.

Since X and Y are not independent, we cannot use convolution. But:

$$
\begin{aligned}
F_{X+Y}(w) \ &= \ P(X + Y) \leq w &\tag{13.77} \\
&= \ \int_0^{w/2} \int_t^{w-t} 2 \ ds \ dt &\tag{13.78} \\
&= \ w^2/2 &\tag{13.79}
\end{aligned}
$$

So $f_{X+Y}(w) = w$.

The case $1 < w < 2$ is similar.

13.5 Parametric Families of Multivariate Distributions

Since there are so many ways in which random variables can correlate with each other, there are rather few parametric families commonly used to model multivariate distributions (other than

those arising from sets of independent random variables have a distribution in a common parametric univariate family). We will discuss two here.

13.5.1 The Multinomial Family of Distributions

13.5.1.1 Probability Mass Function

This is a generalization of the binomial family.

Suppose one rolls a die 8 times. What is the probability that the results consist of two 1s, one 2, one 4, three 5s and one 6? Well, if the rolls occur in that order, i.e. the two 1s come first, then the 2, etc., then the probability is

$$\left(\frac{1}{6}\right)^2 \left(\frac{1}{6}\right)^1 \left(\frac{1}{6}\right)^0 \left(\frac{1}{6}\right)^1 \left(\frac{1}{6}\right)^3 \left(\frac{1}{6}\right)^1 \tag{13.80}$$

But there are many different orderings, in fact

$$\frac{8!}{2!1!0!1!3!1!} \tag{13.81}$$

of them, from Section 2.15.5, and thus

$$P(\text{two 1s, one 2, no 3s, one 4, three 5s, one 6}) = \frac{8!}{2!1!0!1!3!1!} \left(\frac{1}{6}\right)^2 \left(\frac{1}{6}\right)^1 \left(\frac{1}{6}\right)^0 \left(\frac{1}{6}\right)^1 \left(\frac{1}{6}\right)^3 \left(\frac{1}{6}\right)^1 \tag{13.82}$$

From this, we can more generally see the following. Suppose:

- we have n trials, each of which has r possible outcomes or categories

- the trials are independent

- the ith outcome has probability p_i

Let X_i denote the number of trials with outcome i, i $= 1,...,$r. In the die example above, for instance, r $= 6$ for the six possible outcomes of one trial, i.e. one roll of the die, and X_1 is the number of times we got one dot, in our n $= 8$ rolls.

Then we say that the vector $X = (X_1, ..., X_r)$ have a **multinomial distribution**. Since the X_i are discrete random variables, they have a joint pmf $p_{X_1,...,X_r}()$. Taking the above die example for illustration again, the probability of interest there is $p_X(2, 1, 0, 1, 3, 1)$. We then have in general,

$$p_{X_1,...,X_r}(j_1, ..., j_r) = \frac{n!}{j_1!...j_r!}p_1^{j_1}...p_r^{j_r} \tag{13.83}$$

Note that this family of distributions has r+1 parameters: n and the p_i. Of course, you might count it as only r, since the p_i sum to one and thus are not free of each other.

R has the function **dmultinom()** for the multinomial pmf. The call **dmultinom(x,n,prob)** evaluates (13.83), where **x** is the vector $(j_1, ..., j_r)$ and **prob** is $(p_1, ..., p_r)$.

We can simulate multinomial random vectors in R using the **sample()** function:

```
1   # n is the number of trials, p the vector of probabilities of the r
2   # categories
3   multinom <- function(n,p) {
4      r <- length(p)
5      outcome <- sample(x=1:r,size=n,replace=T,prob=p)
6      counts <- vector(length=r)  # counts of the various categories
7      # tabulate the counts (could be done more efficiently)
8      for (i in 1:n) {
9          j <- outcome[i]
10         counts[j] <- counts[j] + 1
11     }
12     return(counts)
13  }
```

13.5.1.2   Example: Component Lifetimes

Say the lifetimes of some electronic component, say a disk drive, are exponentially distributed with mean 4.5 years. If we have six of them, what is the probability that two fail before 1 year, two last between 1 and 2 years, and the remaining two last more than 2 years?

Let (X,Y,Z) be the number that last in the three time intervals. Then this vector has a multinomial distribution, with n = 6 trials, and

$$p_1 = \int_0^1 \frac{1}{4.5}e^{-t/4.5} \ dt = 0.20 \tag{13.84}$$

$$p_2 = \int_1^2 \frac{1}{4.5}e^{-t/4.5} \ dt = 0.16 \tag{13.85}$$

$$p_3 = \int_2^\infty \frac{1}{4.5} e^{-t/4.5} \, dt = 0.64 \tag{13.86}$$

We then use (13.83) to find the specified probability, which is:

$$\frac{6!}{2!2!2!} \, 0.20^2 0.16^2 0.64^2 \tag{13.87}$$

13.5.1.3 Mean Vectors and Covariance Matrices in the Multinomial Family

Consider a multinomially distributed random vector $X = (X_1, ..., X_r)'$, with n trials and category probabilities p_i. Let's find its mean vector and covariance matrix.

First, note that the marginal distributions of the X_i are binomial! So,

$$EX_i = np_i \text{ and } Var(X_i) = np_i(1 - p_i) \tag{13.88}$$

So we know EX now:

$$EX = \begin{pmatrix} np_1 \\ ... \\ np_r \end{pmatrix} \tag{13.89}$$

We also know the diagonal elements of Cov(X)—$np_i(1 - p_i)$ is the ith diagonal element, i = 1,...,r.

But what about the rest? The derivation will follow in the footsteps of those of (4.30), but now in a vector context. Prepare to use your indicator random variable, random vector and covariance matrix skills! Also, this derivation will really build up your "probabilistic stamina level." So, it's good for you! But **now is the time to review (4.30), Section 3.9 and Section 12.4, before continuing.**

We'll continue the notation of the last section. In order to keep on eye on the concrete, we'll often illustrate the notation with the die example above; there we rolled a die 8 times, and defined 6 categories (one dot, two dots, etc.). We were interested in probabilities involving the number of trials that result in each of the 6 categories.

Define the random vector T_i to be the outcome of the ith trial. It is a vector of indicator random variables, one for each of the r categories. In the die example, for instance, consider the second

roll, which is recorded in T_2. If that roll turns out to be, say, 5, then

$$T_2 = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{pmatrix} \tag{13.90}$$

Here is the key observation:

$$\begin{pmatrix} X_1 \\ ... \\ X_r \end{pmatrix} = \sum_{i=1}^{n} T_i \tag{13.91}$$

Keep in mind, (13.91) is a vector equation. In the die example, the first element of the left-hand side,X_1, is the number of times the 1-dot face turns up, and on the right-hand side, the first element of T_i is 1 or 0, according to whether the 1-dot face turns up on the ith roll. Make sure you believe this equation before continuing.

Since the trials are independent, (12.52) and (13.91) now tell us that

$$Cov[(X_1, ..., X_r)'| = \sum_{i=1}^{n} Cov(T_i) \tag{13.92}$$

But the trials are not only independent, but also identically distributed. (The die, for instance, has the same probabilities on each trial.) So the last equation becomes

$$Cov \left[\begin{pmatrix} X_1 \\ ... \\ X_r \end{pmatrix} \right] = nCov(T_1) \tag{13.93}$$

One more step to go. Remember, T_1 is a vector, recording what happens on the first trial, e.g. the first roll of the die. Write it as

$$T_1 = \begin{pmatrix} U_1 \\ ... \\ U_r \end{pmatrix} \tag{13.94}$$

Then the covariance matrix of T_1 consists of elements of the form

$$Cov(U_i, U_j) \tag{13.95}$$

Let's evaluate them.

Case 1: $i = j$

$$
\begin{aligned}
Cov(U_i, U_j) &= Var(U_i) \quad (12.4) & (13.96) \\
&= p_i(1 - p_i) \quad (3.79) & (13.97)
\end{aligned}
$$

Case 2: $i \neq j$

$$
\begin{aligned}
Cov(U_i, U_j) &= E(U_i U_j) - EU_i EU_j \quad (12.5) & (13.98) \\
&= E(U_i U_j) - p_i p_j \quad (3.78) & (13.99) \\
&= -p_i p_j & (13.100)
\end{aligned}
$$

with that last step coming from the fact that U_i and U_j can never both be 1 (e.g. never on the same line of the our "notebook"). Thus the product $U_i U_j$ is always 0, and thus so is is expected value. In the die example, for instance, if our roll resulted in the 2-dot face turned upward, then the 5-dot face definitely did NOT turn upward, so $U_2 = 1$ while $U_5 = 0$.

So, we've now found $Cov(T_1)$, and using this in (13.93), we see that

$$
Cov\left[\begin{pmatrix} X_1 \\ \dots \\ X_r \end{pmatrix}\right] = n \begin{pmatrix} p_1(1-p_1) & -p_1 p_2 & \dots & -p_1 p_r \\ -p_1 p_2 & p_2(1-p_2) & \dots & -p_2 p_r \\ \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & p_r(1-p_r) \end{pmatrix} \tag{13.101}
$$

Note too that if we define R = X/n, so that R is the vector of proportions in the various categories (e.g. X_1/n is the fraction of trials that resulted in category 1), then from (13.101) and (12.51), we have

$$
Cov(R) = \frac{1}{n} \begin{pmatrix} p_1(1-p_1) & -p_1 p_2 & \dots & -p_1 p_r \\ -p_1 p_2 & p_2(1-p_2) & \dots & -p_2 p_r \\ \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & p_r(1-p_r) \end{pmatrix} \tag{13.102}
$$

Whew! That was a workout, but these formulas will become very useful later on, both in this chapter and subsequent ones.

13.5.1.4   Application: Text Mining

One of the branches of computer science in which the multinomial family plays a prominent role is in text mining. One goal is automatic document classification. We want to write software that will make reasonably accurate guesses as to whether a document is about sports, the stock market, elections etc., based on the frequencies of various key words the program finds in the document.

Many of the simpler methods for this use the **bag of words model**. We have r key words we've decided are useful for the classification process, and the model assumes that statistically the frequencies of those words in a given document category, say sports, follow a multinomial distribution. Each category has its own set of probabilities $p_1, ..., p_r$. For instance, if "Barry Bonds" is considered one word, its probability will be much higher in the sports category than in the elections category, say. So, the observed frequencies of the words in a particular document will hopefully enable our software to make a fairly good guess as to the category the document belongs to.

Once again, this is a very simple model here, designed to just introduce the topic to you. Clearly the multinomial assumption of independence between trials is grossly incorrect here, most models are much more complex than this.

13.5.2   The Multivariate Normal Family of Distributions

Note to the reader: This is a more difficult section, but worth putting extra effort into, as so many statistical applications in computer science make use of it. It will seem hard at times, but in the end won't be too bad.

13.5.2.1   Densities

Intuitively, this family has densities which are shaped like multidimensional bells, just like the univariate normal has the famous one-dimensional bell shape.

Let's look at the bivariate case first. The joint distribution of X_1 and X_2 is said to be **bivariate normal** if their density is

$$f_{X,Y}(s,t) = \frac{1}{2\pi\sigma_1\sigma_2\sqrt{1-\rho^2}} e^{-\frac{1}{2(1-\rho^2)}\left[\frac{(s-\mu_1)^2}{\sigma_1^2} + \frac{(t-\mu_2)^2}{\sigma_2^2} - \frac{2\rho(s-\mu_1)(t-\mu 2)}{\sigma_1\sigma_2}\right]}, \quad -\infty < s,t < \infty \quad (13.103)$$

This looks horrible, and it is. But don't worry, as we won't work with this directly. It's important for conceptual reasons, as follows.

First, note the parameters here: μ_1, μ_2, σ_1 and σ_2 are the means and standard deviations of X and Y, while ρ is the correlation between X and Y. So, we have a five-parameter family of distributions.

The multivariate normal family of distributions is parameterized by one vector-valued quantity, the mean μ, and one matrix-valued quantity, the covariance matrix Σ. Specifically, suppose the random vector $X = (X_1, ..., X_k)'$ has a k-variate normal distribution.

The density has this form:

$$f_X(t) = ce^{-0.5(t-\mu)'\Sigma^{-1}(t-\mu)} \tag{13.104}$$

Here c is a constant, needed to make the density integrate to 1.0. It turns out that

$$c = \frac{1}{(2\pi)^{k/2}\sqrt{det(\Sigma)}} \tag{13.105}$$

but we'll never use this fact.

Here again $'$ denotes matrix transpose, -1 denotes matrix inversion and det() means determinant. Again, note that t is a kx1 vector.

Since the matrix is symmetric, there are k(k+1)/2 distinct parameters there, and k parameters in the mean vector, for a total of k(k+3)/2 parameters for this family of distributions.

13.5.2.2 Geometric Interpretation

Now, let's look at some pictures, generated by R code which I've adapted from one of the entries in the R Graph Gallery, `http://addictedtor.free.fr/graphiques/graphcode.php?graph=42`.[4] Both are graphs of bivariate normal densities, with $EX_1 = EX_2 = 0$, $Var(X_1) = 10$, $Var(X_2) = 15$ and a varying value of the correlation ρ between X_1 and X_2. Figure 13.1 is for the case $\rho = 0.2$.

The surface is bell-shaped, though now in two dimensions instead of one. Again, the height of the surface at any (s,t) point the relative likelihood of X_1 being near s and X_2 being near t. Say for instance that X_1 is height and X_2 is weight. If the surface is high near, say, (70,150) (for height of 70 inches and weight of 150 pounds), it mean that there are a lot of people whose height and weight are near those values. If the surface is rather low there, then there are rather few people whose height and weight are near those values.

---

[4]There appears to be an error in their definition of the function **f()**; the assignment to **term5** should not have a negative sign at the beginning.
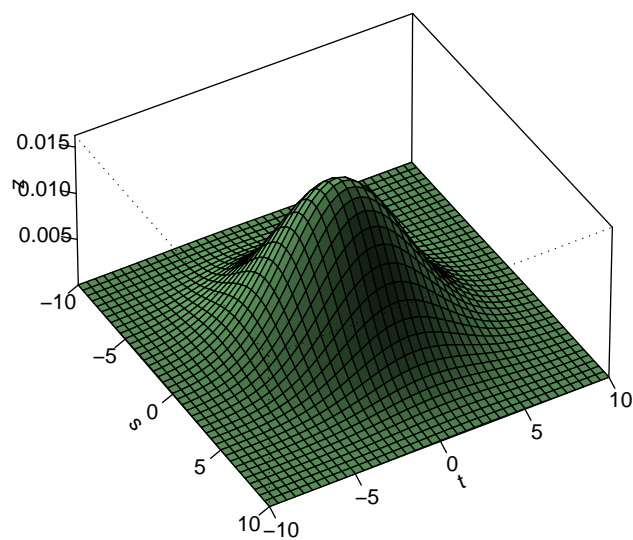
Figure 13.1: Bivariate Normal Density, $\rho = 0.2$
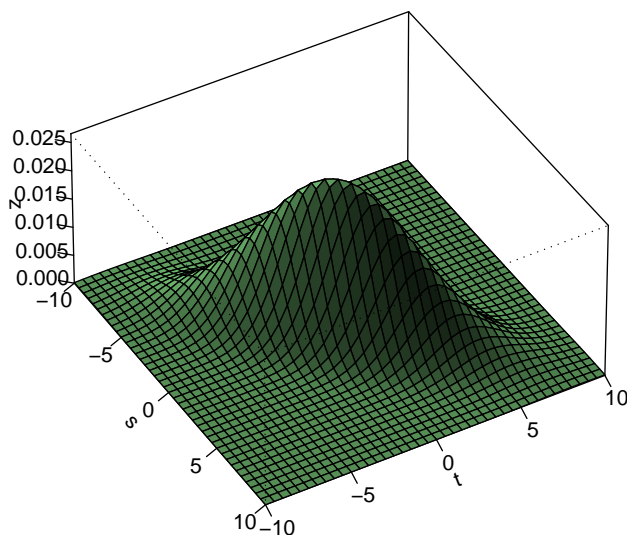
Figure 13.2: Bivariate Normal Density, $\rho = 0.8$

Now compare that picture to Figure 13.2, with $\rho = 0.8$.

Again we see a bell shape, but in this case "narrower." In fact, you can see that when X_1 (s) is large, X_2 (t) tends to be large too, and the same for "large" replaced by small. By contrast, the surface near (5,5) is much higher than near (5,-5), showing that the random vector (X_1, X_2) is near (5,5) much more often than (5,-5).

All of this reflects the high correlation (0.8) between the two variables. If we were to continue to increase ρ toward 1.0, we would see the bell become narrower and narrower, with X_1 and X_2 coming closer and closer to a linear relationship, one which can be shown to be

$$X_1 - \mu_1 = \frac{\sigma_1}{\sigma_2}(X_2 - \mu_2) \tag{13.106}$$

In this case, that would be

$$X_1 = \sqrt{\frac{10}{15}} X_2 = 0.82 X_2 \tag{13.107}$$

13.5.2.3   Properties of Multivariate Normal Distributions

Theorem 27 *Suppose $X = (X_1, ..., X_k)$ has a multivariate normal distribution with mean vector μ and covariance matrix Σ. Then:*

(a) *The contours of f_X are k-dimensional ellipsoids. In the case $k = 2$ for instance, where we can visualize the density of X as a three-dimensional surface, the contours for points at which the bell has the same height (think of a topographical map) are elliptical in shape. The larger the correlation (in absolute value) between X_1 and X_2, the more elongated the ellipse. When the absolute correlation reaches 1, the ellipse degenerates into a straight line.*

(b) *Let A be a constant (i.e. nonrandom) matrix with k columns. Then the random vector $Y = AX$ also has a multivariate normal distribution.*[5]

The parameters of this new normal distribution must be $EY = A\mu$ and $Cov(Y) = A\Sigma A'$, by (12.50) and (12.54).

(c) *If $U_1, ..., U_m$ are each univariate normal and they are independent, then they jointly have a multivariate normal distribution. (In general, though, having a normal distribution for each U_i does not imply that they are jointly multivariate normal.)*

(d) *Suppose W has a multivariate normal distribution. The conditional distribution of some components of W, given other components, is again multivariate normal.*

Part [(b)] has some important implications:

(i) The lower-dimensional marginal distributions are also multivariate normal. For example, if k = 3, the pair $(X_1, X_3)'$ has a bivariate normal distribution, as can be seen by setting

$$A = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{13.108}$$

in (b) above.

---

[5]Note that this is a generalization of the material on affine transformations on page 152.

(ii) Scalar linear combinations of X are normal. In other words, for constant scalars $a_1, ..., a_k$, set $a = (a_1, ..., a_k)'$. Then the quantity $Y = a_1 X_1 + ... + a_k X_k$ has a univariate normal distribution with mean $a'\mu$ and variance $a'\Sigma a$.

(iii) Vector linear combinations are multivariate normal. Again using the case k = 3 as our example, consider $(U, V)' = (X_1 - X_3, X_2 - X_3)$. Then set

$$A = \left(\begin{array}{ccc} 1 & 0 & -1 \\ 0 & 1 & -1 \end{array} \right) \tag{13.109}$$

(iv) The r-component random vector X has a multivariate normal distribution if and only if $c'X$ has a univariate normal distribution for all constant r-component vectors c.

In R the density, cdf and quantiles of the multivariate normal distribution are given by the functions **dmvnorm()**, **pmvnorm()** and **qmvnorm()** in the library **mvtnorm**. You can simulate a multivariate normal distribution by using **mvrnorm()** in the library **MASS**.

13.5.2.4 The Multivariate Central Limit Theorem

The multidimensional version of the Central Limit Theorem holds. A sum of independent identically distributed (*iid*) random vectors has an approximate multivariate normal distribution. Here is the theorem:

Theorem 28 *Suppose $X_1, X_2, ...$ are independent random vectors, all having the same distribution which has mean vector μ and covariance matrix Σ. Form the new random vector $T = X_1 + ... + X_n$. Then for large n, the distribution of T is approximately normal with mean $n\mu$ and covariance matrix $n\Sigma$.*

For example, since a person's body consists of many different components, the CLT (a non-independent, non-identically version of it) explains intuitively why heights and weights are approximately bivariate normal. Histograms of heights will look approximately bell-shaped, and the same is true for weights. The multivariate CLT says that three-dimensional histograms—plotting frequency along the "Z" axis against height and weight along the "X" and "Y" axes—will be approximately three-dimensional bell-shaped.

The proof of the multivariate CLT is easy, from Property (iv) above. Say we have a sum of iid random vectors:

$$S = X_1 + ... + X_n \tag{13.110}$$

Then

$$c'S = c'X_1 + ... + c'X_n \tag{13.111}$$

Now on the right side we have a sum of iid *scalars*, not vectors, so the univariate CLT applies! We thus know the right-hand side is a approximately normal for all c, which means $c'S$ is also approximately normal for all c, which then by (iv) above means that S itself is approximately multivariate normal.

13.5.2.5   Example: Finishing the Loose Ends from the Dice Game

Recall the game example in Section 12.7:

Suppose we roll a die 50 times. Let X denote the number of rolls in which we get one dot, and let Y be the number of times we get either two or three dots. For convenience, let's also define Z to be the number of times we get four or more dots, though our focus will be on X and Y. Suppose also that we win \$5 for each roll of a one, and \$2 for each roll of a two or three.

Our analysis relied on the vector (X,Y,Z)' having an approximate multivariate normal distribution. Where does that come from? Well, first note that the exact distribution of (X,Y,Z)' is multinomial. Then recall (13.91). The latter makes (X,Y,Z)' a sum of iid vectors, so that the multivariate CLT applies.

13.5.2.6   Application: Data Mining

The multivariate normal family plays a central role in multivariate statistical methods.

For instance, a major issue in data mining is **dimension reduction**, which means trying to reduce what may be hundreds or thousands of variables down to a manageable level. One of the tools for this, called **principle components analysis** (PCA), is based on multivariate normal distributions. Google uses this kind of thing quite heavily. We'll discuss PCA in Section 25.1.

To see a bit of how this works, note that in Figure 13.2, X_1 and X_2 had nearly a linear relationship with each other. That means that one of them is nearly redundant, which is good if we are trying to reduce the number of variables we must work with.

In general, the method of principle components takes r original variables, in the vector X and forms r new ones in a vector Y, each of which is some linear combination of the original ones. These new ones are independent. In other words, there is a square matrix A such that the components of Y = AX are independent. (The matrix A consists of the eigenvectors of Cov(X); more on this in Section 25.1 of our unit on statistical relations.

We then discard the Y_i with small variance, as that means they are nearly constant and thus do not carry much information. That leaves us with a smaller set of variables that still captures most of the information of the original ones.

Many analyses in bioinformatics involve data that can be modeled well by multivariate normal distributions. For example, in automated cell analysis, two important variables are forward light scatter (FSC) and sideward light scatter (SSC). The joint distribution of the two is approximately bivariate normal.[6]

---

[6]See *Bioinformatics and Computational Biology Solutions Using R and Bioconductor*, edited by Robert Gentleman, Wolfgang Huber, Vincent J. Carey, Rafael A. Irizarry and Sandrine Dudoit, Springer, 2005.

Chapter 14

Transform Methods

We often use the idea of **transform** functions. For example, you may have seen **Laplace transforms** in a math or engineering course. The functions we will see here differ from this by just a change of variable.

Though in the form used here they involve only univariate distributions, their applications are often multivariate, as will be the case here.

14.1 Generating Functions

Let's start with the **generating function**. For any nonnegative-integer valued random variable V, its generating function is defined by

$$g_V(s) = E(s^V) = \sum_{i=0}^{\infty} s^i p_V(i), \ 0 \le s \le 1 \tag{14.1}$$

For instance, suppose N has a geometric distribution with parameter p, so that $p_N(i) = (1-p)p^{i-1}$, i = 1,2,... Then

$$g_N(s) = \sum_{i=1}^{\infty} s^i \cdot (1-p)p^{i-1} = \frac{1-p}{p} \sum_{i=1}^{\infty} s^i \cdot p^i = \frac{1-p}{p} \frac{ps}{1-ps} = \frac{(1-p)s}{1-ps} \tag{14.2}$$

Why restrict s to the interval [0,1]? The answer is that for $s > 1$ the series in (14.1) may not converge. for $0 \le s \le 1$, the series does converge. To see this, note that if s = 1, we just get the

245

sum of all probabilities, which is 1.0. If a nonnegative s is less than 1, then s^i will also be less than 1, so we still have convergence.

One use of the generating function is, as its name implies, to generate the probabilities of values for the random variable in question. In other words, if you have the generating function but not the probabilities, you can obtain the probabilities from the function. Here's why: For clarify, write (14.1) as

$$g_V(s) = P(V = 0) + sP(V = 1) + s^2 P(V = 2) + \dots \tag{14.3}$$

From this we see that

$$g_V(0) = P(V = 0) \tag{14.4}$$

So, we can obtain P(V = 0) from the generating function. Now differentiating (14.1) with respect to s, we have

$$
\begin{aligned}
g'_V(s) &= \frac{d}{ds} \left[P(V = 0) + sP(V = 1) + s^2 P(V = 2) + \dots \right] \\
&= P(V = 1) + 2sP(V = 2) + \dots
\end{aligned}
\tag{14.5}
$$

So, we can obtain P(V = 2) from $g'_V(0)$, and in a similar manner can calculate the other probabilities from the higher derivatives.

14.2   Moment Generating Functions

The generating function is handy, but it is limited to discrete random variables. More generally, we can use the **moment generating function**, defined for any random variable X as

$$m_X(t) = E[e^{tX}] \tag{14.6}$$

for any t for which the expected value exists.

That last restriction is anathema to mathematicians, so they use the characteristic function,

$$\phi_X(t) = E[e^{itX}] \tag{14.7}$$

which exists for any t. However, it makes use of pesky complex numbers, so we'll stay clear of it here.

Differentiating (14.6) with respect to t, we have

$$m'_X(t) = E[Xe^{tX}] \tag{14.8}$$

We see then that

$$m'_X(0) = EX \tag{14.9}$$

So, if we just know the moment-generating function of X, we can obtain EX from it. Also,

$$m''_X(t) = E(X^2 e^{tX}) \tag{14.10}$$

so

$$m''_X(0) = E(X^2) \tag{14.11}$$

In this manner, we can for various k obtain $E(X^k)$, the **k**th **moment** of X, hence the name.

14.3   Transforms of Sums of Independent Random Variables

Suppose X and Y are independent and their moment generating functions are defined. Let Z = X+Y. then

$$m_Z(t) = E[e^{t(X+Y)}] = E[e^{tX}e^{tY}] = E(e^{tX}) \cdot E(e^{tY}) = m_X(t)m_Y(t) \tag{14.12}$$

In other words, the mgf of the sum is the product of the mgfs! This is true for other transforms, by the same reasoning.

Similarly, it's clear that the mgf of a sum of three independent variables is again the product of their mgfs, and so on.

14.4   Example: Network Packets

As an example, suppose say the number of packets N received on a network link in a given time period has a Poisson distribution with mean μ, i.e.

$$P(N = k) = \frac{e^{-\mu}\mu^k}{k!}, k = 0, 1, 2, 3, ... \tag{14.13}$$

14.4.1   Poisson Generating Function

Let's first find its generating function.

$$g_N(t) = \sum_{k=0}^{\infty} t^k \frac{e^{-\mu}\mu^k}{k!} = e^{-\mu} \sum_{k=0}^{\infty} \frac{(\mu t)^k}{k!} = e^{-\mu+\mu t} \tag{14.14}$$

where we made use of the Taylor series from calculus,

$$e^u = \sum_{k=0}^{\infty} u^k/k! \tag{14.15}$$

14.4.2   Sums of Independent Poisson Random Variables Are Poisson Distributed

Supposed packets come in to a network node from two independent links, with counts N_1 and N_2, Poisson distributed with means μ_1 and μ_2. Let's find the distribution of $N = N_1 + N_2$, using a transform approach.

From Section 14.3:

$$g_N(t) = g_{N_1}(t)g_{N_2}(t) = e^{-\nu+\nu t} \tag{14.16}$$

where $\nu = \mu_1 + \mu_2$.

But the last expression in (14.16) is the generating function for a Poisson distribution too! And since there is a one-to-one correspondence between distributions and transforms, we can conclude that N has a Poisson distribution with parameter ν. We of course knew that N would have mean ν but did not know that N would have a Poisson distribution.

So: A sum of two independent Poisson variables itself has a Poisson distribution. By induction, this is also true for sums of k independent Poisson variables.

14.5 Other Uses of Transforms

Transform techniques are used heavily in queuing analysis, including for models of computer networks. The techniques are also used extensively in modeling of hardware and software reliability.

Transforms also play key roles in much of theoretical probability, the Central Limit Theorems[1] being a good example. Here's an outline of the proof of the basic CLT, assuming the notation of Section 8.20:

First rewrite Z as

$$Z = \sum_{i=1}^{n} \frac{X_i - m}{v\sqrt{n}} \tag{14.17}$$

Then work with the characteristic function of Z:

$$
\begin{aligned}
c_Z(t) &= E(e^{itZ}) \quad \text{(def.)} & (14.18)\\
&= \Pi_{i=1}^{n} E[e^{it(X_i-m)/(v\sqrt{n})}] \quad \text{(indep.)} & (14.19)\\
&= \Pi_{i=1}^{n} E[e^{it(X_1-m)/(v\sqrt{n})}] \quad \text{(ident. distr.)} & (14.20)\\
&= [g(\frac{it}{\sqrt{n}})]^n & (14.21)
\end{aligned}
$$

where g(s) is the characteristic function of $(X_1 - m)/v$, i.e.

$$g(s) = E[e^{is\cdot \frac{X_1-m}{v}}] \tag{14.22}$$

Now expand (14.21) in a Taylor series around 0, and use the fact that $g'(0)$ is the expected value of $(X_1 - m)/v$, which is 0:

$$
\begin{aligned}
[g(\frac{t}{\sqrt{n}})]^n &= \left[1 - \frac{t^2}{2n} + o(\frac{t^2}{n})\right]^n & (14.23)\\
&\rightarrow e^{-t^2/2} \text{ as } n \rightarrow \infty & (14.24)
\end{aligned}
$$

where we've also used the famous fact that $(1 - s/n)^n$ converges to e^{-s} as $n \rightarrow \infty$.

But (14.24) is the density of N(0,1), so we have proved the Central Limit Theorem.

---

[1] The plural is used here because there are many different versions, which for instance relax the condition that the summands be independent and identically distributed.

Chapter 15

Statistics: Prologue

There are three kinds of lies: lies, damned lies and statistics—variously attributed to Benjamin Disraeli, Mark Twain etc.

Consider the following problems:

- Suppose you buy a ticket for a raffle, and get ticket number 68. Two of your friends bought tickets too, getting numbers 46 and 79. Let c be the total number of tickets sold. You don't know the value of c, but hope it's small, so you have a better chance of winning. How can you estimate the value of c, from the data, 68, 46 and 79?

- It's presidential election time. A poll says that 56% of the voters polled support candidate X, with a margin of error of 2%. The poll was based on a sample of 1200 people. How can a sample of 1200 people out of more than 100 million voters have a margin of error that small? And what does the term *margin of error* really mean, anyway?

- A satellite detects a bright spot in a forest. Is it a fire? How can we design the software on the satellite to estimate the probability that this is a fire?

If you think that statistics is nothing more than adding up columns of numbers and plugging into formulas, you are badly mistaken. Actually, statistics is an application of probability theory. We employ probabilistic models for the behavior of our sample data, and *infer* from the data accordingly—hence the name, **statistical inference**.

Arguably the most powerful use of statistics is prediction, as known these days as *machine learning*.

15.1 Sampling Distributions

We first will set up some infrastructure, which will be used heavily throughout the next few chapters.

15.1.1   Random Samples

Definition 29 *Random variables $X_1, X_2, X_3, ...$ are said to be* **i.i.d.** *if they are independent and identically distributed. The latter term means that p_{X_i} or f_{X_i} is the same for all i.*

Note the following carefully:

> For i.i.d. $X_1, X_2, X_3, ...$, we often use X to represent a generic random variable having the common distribution of the X_i.

Definition 30 *We say that $X_1, X_2, X_3, ..., X_n$ is a* **random sample** *of size n from a population if the X_i are i.i.d. and their common distribution is that of the population.*

(**Please note:** Those numbers $X_1, X_2, X_3, ..., X_n$ collectively form <u>one</u> sample; you should not say anything like "We have n samples.")

If the sampled population is finite,[1] then a random sample must be drawn in this manner. Say there are k entities in the population, e.g. k people, with values $v_1, ..., v_k$. If we are interested in people's heights, for instance, then $v_1, ..., v_k$ would be the heights of all people in our population. Then a random sample is drawn this way:

  (a) The sampling is done with replacement.

  (b) Each X_i is drawn from $v_1, ..., v_k$, with each v_j having probability $\frac{1}{k}$ of being drawn.

Condition (a) makes the X_i independent, while (b) makes them identically distributed.

If sampling is done without replacement, we call the data a **simple random sample**. Note how this implies lack of independence of the X_i. If for instance $X_1 = v_3$, then we know that no other X_i has that value, contradicting independence; if the X_i were independent, knowledge of one should not give us knowledge concerning others.

---

[1] You might wonder how it could be infinite. Say for example we model the distribution of X as a continuous random variable, then there are infinitely many values X can take on, though it must be kept in mind that, as noted in Section 7.15, continuous random variables are only idealizations.

But we usually assume true random sampling, i.e. with replacement, and will mean that unless explicitly stating otherwise. In most cases, the population is so large, even infinite, that there is no practical distinction, as we are extremely unlikely to sample the same person (or other unit) twice.

Keep this very important point in mind:

> Note most carefully that *each X_i has the same distribution as the population.* If for instance a third of the population, i.e. a third of the v_j, are less than 28, then $P(X_i < 28)$ will be 1/3.
>
> If the mean value of X in the population is, say, 51.4, then EX will be 51.4, and so on.
>
> These points areeasy to see, but keep them in mind at all times, as they will arise again and again.

15.2 The Sample Mean—a Random Variable

A large part of this chapter will concern the **sample mean**,

$$\overline{X} = \frac{X_1 + X_2 + X_3 + ... + X_n}{n} \tag{15.1}$$

A simple yet crucial concept is point that $\overline{X}$ is a random variable. Since $X_1, X_2, X_3, ..., X_n$ are random variables, $\overline{X}$ is a random variable too.

Make absolutely sure to distinguish between the sample mean and the population mean.

15.2.1 Toy Population Example

Let's illustrate it with a tiny example. Suppose we have a population of three people, with heights 69, 72 and 70, and we draw a random sample of size 2. As noted, $\overline{X}$ is a random variable. Its support consists of six values:

$$\frac{69 + 69}{2} = 69, \frac{69 + 72}{2} = 70.5, \frac{69 + 70}{2} = 69.5, \frac{70 + 70}{2} = 70, \frac{70 + 72}{2} = 71, \frac{72 + 72}{2} = 72 \tag{15.2}$$

So $\overline{X}$ is a discrete random variable, and its pmf is given by 1/9, 2/9, 2/9, 1/9, 2/9 and 1/9, respectively. So,

$$p_{\overline{X}}(69) = \frac{1}{9}, \ \ p_{\overline{X}}(70.5) = \frac{2}{9}, \ \ p_{\overline{X}}(69.5) = \frac{2}{9}, \ \ p_{\overline{X}}(70) = \frac{1}{9}, \ \ p_{\overline{X}}(71) = \frac{2}{9}, \ \ p_{\overline{X}}(72) = \frac{1}{9} \tag{15.3}$$

Since $\overline{X}$ is a random variable, it also has a cdf. For instance, the reader should verify that $F_{\overline{X}}(69.9) = 3/9$.

Viewing it in "notebook" terms, we might have, in the first three lines:

notebook line	X_1	X_2	$\overline{X}$
1	70	70	70
2	69	70	69.5
3	72	70	71

Again, the point is that all of X_1, X_2 and $\overline{X}$ are random variables.

15.2.2   Expected and Variance of $\overline{X}$

Now, returning to the case of general n and our sample $X_1, ..., X_n$, since $\overline{X}$ is a random variable, we can ask about its expected value and variance. Note that in notebook terms, these are the long-run mean and variance of the values in the $\overline{X}$ column above.

Let μ denote the population mean. Remember, each X_i is distributed as is the population, so $EX_i = \mu$. Again in notebook terms, this says that the long-run average in the X_1 column will be μ. (The same will be true for the X_2 column.)

This then implies that the expected value of $\overline{X}$ is also μ. Here's why:

$$
\begin{aligned}
E(\overline{X}) \;&=\; E\left[\frac{1}{n}\sum_{i=1}^{n} X_i\right] \quad \text{(def. of } \overline{X}\text{)} & (15.4)\\[2mm]
&=\; \frac{1}{n}E\left(\sum_{i=1}^{n} X_i\right) \quad \text{(for const. c, } E(cU) = cEU) & (15.5)\\[2mm]
&=\; \frac{1}{n}\sum_{i=1}^{n} EX_i \quad (E[U+V] = EU + EV) & (15.6)\\[2mm]
&=\; \frac{1}{n}n\mu \quad (EX_i = \mu) & (15.7)\\[2mm]
&=\; \mu & (15.8)
\end{aligned}
$$

Moreover, the variance of $\overline{X}$ is 1/n times the population variance:

$$Var(\overline{X}) \;=\; Var\left[\frac{1}{n}\sum_{i=1}^{n}X_i\right] \tag{15.9}$$

$$=\; \frac{1}{n^2}Var\left(\sum_{i=1}^{n}X_i\right) \quad \text{(for const. c, , } Var[cU] = c^2 Var[U]) \tag{15.10}$$

$$=\; \frac{1}{n^2}\sum_{i=1}^{n}Var(X_i) \quad \text{(for U,V indep., , } Var[U+V] = Var[U] + Var[V]) \tag{15.11}$$

$$=\; \frac{1}{n^2}n\sigma^2 \tag{15.12}$$

$$=\; \frac{1}{n}\sigma^2 \tag{15.13}$$

This derivation plays a crucial role in statistics, and you in turn can see that the independence of the X_i played a crucial role in the derivation. This is why we assume sampling with replacement

15.2.3 Toy Population Example Again

Let's verify (15.8) and (15.13) for toy population in Section 15.2.1. The population mean is

$$\mu = (69 + 70 + 72)/3 = 211/3 \tag{15.14}$$

Using (3.13) and (15.3), we have

$$E\overline{X} = \sum_{c} c\, p_{\overline{X}}(c) = 69 \cdot \frac{1}{9} + 69.5 \cdot \frac{2}{9} + 70 \cdot \frac{1}{9} + 70.5 \cdot \frac{2}{9} + 71 \cdot \frac{2}{9} + 72 \cdot \frac{1}{9} = 211/3 \tag{15.15}$$

So, (15.8) is confirmed. What about (15.13)?

First, the population variance is

$$\sigma^2 = \frac{1}{3} \cdot (69^2 + 70^2 + 72^2) - (\frac{211}{3})^2 = \frac{14}{9} \tag{15.16}$$

The variance of $\overline{X}$ is

$$Var(\overline{X}) \;=\; E(\overline{X}^2) - (E\overline{X})^2 \tag{15.17}$$

$$=\; E(\overline{X}^2) - (\frac{211}{3})^2 \tag{15.18}$$

Using (3.36) and (15.3), we have

$$E(\overline{X}^2) = \sum_c c^2 \; p_{\overline{X}}(c) = 69^2 \cdot \frac{1}{9} + 69.5^2 \cdot \frac{2}{9} + 70^2 \cdot \frac{1}{9} + 70.5^2 \cdot \frac{2}{9} + 71^2 \cdot \frac{2}{9} + 72^2 \cdot \frac{1}{9} \qquad (15.19)$$

The reader should now wrap things up and confirm that (15.17) does work out to (14/9) / 2 = 7/9, as claimed by (15.13) and (15.16).

15.2.4   Interpretation

Now, let's step back and consider the significance of the above findings (15.8) and (15.13):

(a) Equation (15.8) tells us that although some samples give us an $\overline{X}$ that is too high, i.e. that overestimates μ, while other samples give us an $\overline{X}$ that is too low, on average $\overline{X}$ is "just right."

(b) Equation (15.13) tells us that for large samples, i.e. large n, $\overline{X}$ doesn't vary much from sample to sample.

If you put (a) and (b) together, it says that for large n, $\overline{X}$ is probably pretty accurate, i.e. pretty close to the population mean μ. (You may wish to view this in terms of Section 3.63.) So, the story of statistics often boils down to asking, "Is the variance of our estimator small enough?" You'll see this in the coming chapters.

15.2.5   Simple Random Sample Case

What if we sample without replacement? The reader should make sure to understand that (15.8) still holds completely.b2 The additivity of E() holds even if the summands are not independent. And the distribution of the X_i is still the population distribution as in the with-replacement case. (The reader may recall the similar issue in Section 3.9.3.)

What does change is the derivation (15.13). The summands are no longer independent, so variance is no longer additive. That means that covariance terms must be brought in, as in (12.10), and though one might proceed as before in a somewhat messier set of equations, for general statistical procedures this is not possible. So, the independence assumption is ubiquitous.

Actually, simple random sampling does yield smaller variances for $\overline{X}$. This is good, and makes intuitive sense — we potentially sample a greater number of different people. So in our toy example above, the variance will be smaller than the 14/9 value obtained there. The reader should verify this.

The reader should

15.3 Sample Means Are Approximately Normal—No Matter What the Population Distribution Is

The Central Limit Theorem tells us that the numerator in (15.1) has an approximate normal distribution. That means that affine transformations of that numerator are also approximately normally distributed (page 152). So:

Approximate distribution of (centered and scaled) $\overline{X}$:

The quantity

$$Z = \frac{\overline{X} - \mu}{\sigma/\sqrt{n}} \qquad (15.20)$$

has an approximately N(0,1) distribution, where σ^2 is the population variance.

Remember, we don't know either μ or σ; the whole point of taking the random sample is to estimate them. Nevertheless, their values do exist, and thus the fraction Z does exist. And by the CLT, Z will have an approximate N(0,1) distribution. This simple observation is at the core of statistics. The theorem is not called Central for nothing!

Make sure you understand why it is the "N" that is approximate here, not the 0 or 1.

So even if the population distribution is very skewed, multimodal and so on, the sample mean will still have an approximate normal distribution. This will turn out to be the core of statistics; they don't call the theorem the *Central* Limit Theorem for nothing.

15.3.1 The Sample Variance—Another Random Variable

Later we will be using the sample mean $\overline{X}$, a function of the X_i, to estimate the population mean μ. What other function of the X_i can we use to estimate the population variance σ^2?

Let X denote a generic random variable having the distribution of the X_i, which, note again, is the distribution of the population. Because of that property, we have

$$Var(X) = \sigma^2 \quad (\sigma^2 \text{ is the population variance}) \qquad (15.21)$$

pop. entity	samp. entity
EX	$\overline{X}$
X	X_i
E[]	$\frac{1}{n}\sum_{i=1}^{n}$

Table 15.1: Population and Sample Analogs

Recall that by definition

$$Var(X) = E[(X - EX)^2] \tag{15.22}$$

15.3.1.1   Intuitive Estimation of σ^2

Let's estimate $Var(X) = \sigma^2$ by taking sample analogs in (15.22). The correspondences are shown in Table 15.1.

The sample analog of μ is $\overline{X}$. What about the sample analog of the "E()"? Well, since E() averaging over the whole population of Xs, the sample analog is to average over the sample. So, our sample analog of (15.22) is

$$s^2 = \frac{1}{n}\sum_{i=1}^{n}(X_i - \overline{X})^2 \tag{15.23}$$

In other words, just as it is natural to estimate the population mean of X by its sample mean, the same holds for Var(X):

> The population variance of X is the mean squared distance from X to its population mean, as X ranges over all of the population. Therefore it is natural to estimate Var(X) by the average squared distance of X from its sample mean, among our sample values X_i, shown in (15.23).

We use s^2 as our symbol for this estimate of population variance.[2]

---

[2]Though I try to stick to the convention of using only capital letters to denote random variables, it is conventional to use lower case in this instance.

15.3.1.2   Easier Computation

By the way, it can be shown that (15.23) is equal to

$$\frac{1}{n}\sum_{i=1}^{n} X_i^2 - \overline{X}^2 \qquad (15.24)$$

This is a handy way to calculate s^2, though it is subject to more roundoff error. Note that (15.24) is a sample analog of (3.41).

15.3.1.3   To Divide by n or n-1?

It should be noted that it is common to divide by n-1 instead of by n in (15.23). In fact, almost all textbooks divide by n-1 instead of n. Clearly, unless n is very small, the difference will be minuscule; such a small difference is not going to affect any analyst's decisionmaking. But there are a couple of important conceptual questions here:

- Why do most people (and R, in its **var()** function) divide by n-1?

- Why do I choose to use n?

The answer to the first question is that (15.23) is what is called **biased downwards**, meaning that it can be shown (Section 18.2.2) that

$$E(s^2) = \frac{n-1}{n}\sigma^2 \qquad (15.25)$$

In notebook terms, if we were to take many, many samples, one per line in the notebook, in the long run the average of all of our s^2 values would be slightly smaller than σ^2. This bothered the early pioneers of statistics, so they decided to divide by n-1 to make the sample variance an **unbiased** estimator of σ^2. Their definition of s^2 is

$$s^2 = \frac{1}{n-1}\sum_{i=1}^{n}(X_i - \overline{X})^2 \qquad (15.26)$$

This is why W. Gossett defined his now-famous Student-t distribution using (15.26), with a factor of n-1 instead of n. But he could have just as easily defined it as (15.23).

Moreover, even though s^2 is unbiased under their definition, their s itself is still biased downward (Section 18.2.2.1). And since s itself is what we (this book and all others) use in forming confidence intervals, one can see that insisting on unbiasedness is a losing game.

I choose to use (15.23), dividing by n, because of Table 15.1; it's very important that students understand this idea of sample analogs. Another virtue of this approach is that it is in a certain sense more consistent, as we'll see in Section 16.4.2.

15.4   Observational Studies

The above formulation of sampling is rather idealized. It assumes a well-defined population, from which each unit is equally likely to be sampled. In real life, though, things are often not so clearcut.

In Section 16.9, for instance, we analyze data on major league baseball players, and apply statistical inference procedures based on the material in the current chapter. The player data is for a particular year, and our population is the set of all major league players, past, present and future. But here, no physical sampling occurred; we are implicitly assuming that our data *act like* a random sample from that population.

That in turn means that there was nothing special about our particular year. A player in our year, for instance, is just as likely to weigh more than 220 pounds than in previous years. This is probably a safe assumption, but at the same time it probably means we should restrict our (conceptual) population to recent years; back in the 1920s, players probably were not as big as those we see today.

The term usually used for settings like that of the baseball player data is *observational studies*. We passively *observe* our data rather than obtaining it by physically sampling from the population. The careful analyst must consider whether his/her data are representative of the conceptual population, versus subject to some bias.

15.5   A Good Time to Stop and Review!

The material we've discussed since page 252, is absolutely key, forming the very basis of statistics. It will be used constantly, throughout all our chapters here on statistics. It would be highly worthwhile for the reader to review this chapter before continuing.

Chapter 16

Introduction to Confidence Intervals

The idea of a confidence interval is central to statistical inference. But actually, you already know about it—from the term *margin of error* in news reports about opinion polls.

16.1 The "Margin of Error" and Confidence Intervals

To explain the idea of margin of error, let's begin with a problem that has gone unanswered so far:

In our simulations in previous units, it was never quite clear how long the simulation should be run, i.e. what value to set for **nreps** in Section 2.14.7. Now we will finally address this issue.

As our example, consider the Bus Paradox, which presented in Section 9.4: Buses arrive at a certain bus stop at random times, with interarrival times being independent exponentially distributed random variables with mean 10 minutes. You arrive at the bus stop every day at a certain time, say four hours (240 minutes) after the buses start their morning run. What is your mean wait μ for the next bus?

We found mathematically that, due to the memoryless property of the exponential distribution, our wait is again exponentially distributed with mean 10. But suppose we didn't know that, and we wished to find the answer via simulation. (Note to reader: Keep in mind throughout this example that we will be pretending that we don't know the mean wait is actually 10. Reminders of this will be brought up occasionally.)

We could write a program to do this:

```
1  doexpt <- function(opt) {
2      lastarrival <- 0.0
3      while (lastarrival < opt)
```

```
4          lastarrival <- lastarrival + rexp(1,0.1)
5       return(lastarrival-opt)
6    }
7
8    observationpt <- 240
9    nreps <- 1000
10   waits <- vector(length=nreps)
11   for (rep in 1:nreps) waits[rep] <- doexpt(observationpt)
12   cat("approx. mean wait = ",mean(waits),"\n")
```

Running the program yields

```
approx. mean wait = 9.653743
```

Note that μ is a population mean, where our "population" here is the set of all possible bus wait times (some more frequent than others). Our simulation, then, drew a sample of size 1000 from that population. The expression **mean(waits)** was our sample mean.

Now, was 1000 iterations enough? How close is this value 9.653743 to the true expected value of waiting time?[1]

What we would like to do is something like what the pollsters do during presidential elections, when they say "Ms. X is supported by 62% of the voters, with a margin of error of 4%." In other words, we want to be able to attach a margin of error to that figure of 9.653743 above. We do this in the next section.

16.2   Confidence Intervals for Means

We are now set to make use of the infrastructure that we've built up in the preceding sections of this chapter. Everything will hinge on understanding that the sample mean is a random variable, with a known approximate distribution.

The goal of this section (and several that follow) is to develop a notion of margin of error, just as you see in the election campaign polls. This raises two questions:

(a) What do we mean by "margin of error"?

(b) How can we calculate it?

---

[1]Of course, continue to ignore the fact that we know that this value is 10.0. What we're trying to do here is figure out how to answer "how close is it" questions in general, when we don't know the true mean.

16.2.1 Basic Formulation

So, suppose we have a random sample $W_1, ..., W_n$ from some population with mean μ and variance σ^2.

Recall that (15.20) has an approximate N(0,1) distribution. We will be interested in the central 95% of the distribution N(0,1). Due to symmetry, that distribution has 2.5% of its area in the left tail and 2.5% in the right one. Through the R call **qnorm(0.025)**, or by consulting a N(0,1) cdf table in a book, we find that the cutoff points are at -1.96 and 1.96. In other words, if some random variable T has a N(0,1) distribution, then $P(-1.96 < T < 1.96) = 0.95$.

Thus

$$0.95 \approx P\left(-1.96 < \frac{\overline{W} - \mu}{\sigma/\sqrt{n}} < 1.96\right) \tag{16.1}$$

(Note the approximation sign.) Doing a bit of algebra on the inequalities yields

$$0.95 \approx P\left(\overline{W} - 1.96\frac{\sigma}{\sqrt{n}} < \mu < \overline{W} + 1.96\frac{\sigma}{\sqrt{n}}\right) \tag{16.2}$$

Now remember, not only do we not know μ, we also don't know σ. But we can estimate it, as we saw, via (15.23). One can show (the details will be given in Section **??**) that (16.2) is still valid if we substitute s for σ, i.e.

$$0.95 \approx P\left(\overline{W} - 1.96\frac{s}{\sqrt{n}} < \mu < \overline{W} + 1.96\frac{s}{\sqrt{n}}\right) \tag{16.3}$$

In other words, we are about 95% sure that the interval

$$(\overline{W} - 1.96\frac{s}{\sqrt{n}}, \overline{W} + 1.96\frac{s}{\sqrt{n}}) \tag{16.4}$$

contains μ. This is called a 95% **confidence interval** for μ. The quantity $1.96\frac{s}{\sqrt{n}}$ is the margin of error.

16.2.2 Example: Simulation Output

We could add this feature to our program in Section 16.1:

```
1   doexpt <- function(opt) {
2      lastarrival <- 0.0
3      while (lastarrival < opt)
4         lastarrival <- lastarrival + rexp(1,0.1)
5      return(lastarrival-opt)
6   }
7
8   observationpt <- 240
9   nreps <- 10000
10  waits <- vector(length=nreps)
11  for (rep in 1:nreps) waits[rep] <- doexpt(observationpt)
12  wbar <- mean(waits)
13  cat("approx. mean wait =",wbar,"\n")
14  s2 <- mean(waits^2) - wbar^2
15  s <- sqrt(s2)
16  radius <- 1.96*s/sqrt(nreps)
17  cat("approx. CI for EW =",wbar-radius,"to",wbar+radius,"\n")
```

When I ran this, I got 10.02565 for the estimate of EW, and got an interval of (9.382715, 10.66859). Note that the margin of error is the radius of that interval, about 1.29/2. We would then say, "We are about 95% confident that the true mean wait time is between 9.38 and 10.67."

What does this really mean? This question is of the utmost importance. We will devote an entire section to it, Section 16.3.

Note that our analysis here is approximate, based on the Central Limit Theorem, which was applicable because $\overline{W}$ involves a sum. We are making no assumption about the density of the population from which the W_i are drawn. However, if that population density itself is normal, then an exact confidence interval can be constructed. This will be discussed in Section 16.7.

16.3   Meaning of Confidence Intervals

16.3.1   A Weight Survey in Davis

Consider the question of estimating the mean weight, denoted by μ, of all adults in the city of Davis. Say we sample 1000 people at random, and record their weights, with W_i being the weight of the ith person in our sample.[2]

Now remember, we don't know the true value of that population mean, μ—again, that's why we are collecting the sample data, to estimate μ! Our estimate will be our sample mean, $\overline{W}$. But we don't know how accurate that estimate might be. That's

---

[2]Do you like our statistical pun here? Typically an example like this would concern people's heights, not weights. But it would be nice to use the same letter for random variables as in Section 16.2, i.e. the letter W, so we'll have our example involve people's weights instead of heights. It works out neatly, because the word *weight* has the same sound as *wait*.

the reason we form the confidence interval, as a gauge of the accuracy of $\overline{\mathbf{W}}$ as an estimate of μ.

Say our interval (16.4) turns out to be (142.6,158.8). We say that we are about 95% confident that the mean weight μ of all adults in Davis is contained in this interval. **What does this mean?**

Say we were to perform this experiment many, many times, recording the results in a notebook: We'd sample 1000 people at random, then record our interval $(\overline{W} - 1.96\frac{s}{\sqrt{n}}, \overline{W} + 1.96\frac{s}{\sqrt{n}})$ on the first line of the notebook. Then we'd sample another 1000 people at random, and record what interval we got that time on the second line of the notebook. This would be a different set of 1000 people (though possibly with some overlap), so we would get a different value of $\overline{W}$ and so, thus a different interval; it would have a different center and a different radius. Then we'd do this a third time, a fourth, a fifth and so on.

Again, each line of the notebook would contain the information for a different random sample of 1000 people. There would be two columns for the interval, one each for the lower and upper bounds. And though it's not immediately important here, note that there would also be columns for W_1 through W_{1000}, the weights of our 1000 people, and columns for $\overline{W}$ and s.

Now here is the point: Approximately 95% of all those intervals would contain μ, the mean weight in the entire adult population of Davis. The value of μ would be unknown to us—once again, that's why we'd be sampling 1000 people in the first place—but it does exist, and it would be contained in approximately 95% of the intervals.

As a variation on the notebook idea, think of what would happen if you and 99 friends each do this experiment. Each of you would sample 1000 people and form a confidence interval. Since each of you would get a different sample of people, you would each get a different confidence interval. What we mean when we say the confidence level is 95% is that of the 100 intervals formed—by you and 99 friends—about 95 of them will contain the true population mean weight. Of course, you hope you yourself will be one of the 95 lucky ones! But remember, you'll never know whose intervals are correct and whose aren't.

Now remember, in practice we only take *one* sample of 1000 people. Our notebook idea here is merely for the purpose of understanding what we mean when we say that we are about 95% confident that one interval we form does contain the true value of μ.

16.3.2   More About Interpretation

Some statistics instructors give students the odd warning, "You can't say that the probability is 95% that μ is IN the interval; you can only say that the probability is 95% confident that the interval CONTAINS μ." This of course is nonsense. As any fool can see, the following two statements are

equivalent:

- "μ is in the interval"

- "the interval contains μ"

So it is ridiculous to say that the first is incorrect. Yet many instructors of statistics say so.

Where did this craziness come from? Well, way back in the early days of statistics, some instructor was afraid that a statement like "The probability is 95% that μ is in the interval" would make it sound like μ is a random variable. Granted, that was a legitimate fear, because μ is not a random variable, and without proper warning, some learners of statistics might think incorrectly. The random entity is the interval (both its center and radius), not μ; $\overline{W}$ and s in (16.4) vary from sample to sample, so the interval is indeed the random object here, not μ.

So, it was reasonable for teachers to warn students not to think μ is a random variable. But later on, some misguided instructor must have then decided that it is incorrect to say "μ is in the interval," and others then followed suit. They continue to this day, sadly.

A variant on that silliness involves saying that one can't say "The probability is 95% that μ is in the interval," because μ is either in the interval or not, so that "probability" is either 1 or 0! That is equally mushy thinking.

Suppose, for example, that I go into the next room and toss a coin, letting it land on the floor. I return to you, and tell you the coin is lying on the floor in the next room. I know the outcome but you don't. What is the probability that the coin came up heads? To me that is 1 or 0, yes, but to you it is 50%, in any practical sense.

It is also true in the "notebook" sense. If I do this experiment many times—go to the next room, toss the coin, come back to you, go to the next room, toss the coin, come back to you, etc., one line of the notebook per toss—then in the long run 50% of the lines of the notebook have Heads in the Outcome column.

The same is true for confidence intervals. Say we conduct many, many samplings, one per line of the notebook, with a column labeled Interval Contains Mu. Unfortunately, we ourselves don't get to see that column, but it exists, and in the long run 95% of the entries in the column will be Yes.

Finally, there are those who make a distinction between saying "There is a 95% probability that..." and "We are 95% confident that..." That's silly too. What else could "95% confident" mean if not 95% probability?

Consider the experiment of tossing two fair dice. The probability is 34/36, or about 94%, that we get a total that is different from 2 or 12. As we toss the dice, what possible distinction could be made between saying, "The probability is 94% that we will get a total between 3 and 11"

and saying, "We are 94% confident that we will get a total between 3 and 11"? The notebook interpretation supports both phrasings, really. The words *probability* and *confident* should not be given much weight here; remember the quote at the beginning of our Chapter 1:

> *I learned very early the difference between knowing the name of something and knowing something*—Richard Feynman, Nobel laureate in physics

16.4 Confidence Intervals for Proportions

So we know how to find confidence intervals for means. How about proportions?

16.4.1 Derivation

For example, in an election opinion poll, we might be interested in the proportion p of people in the entire population who plan to vote for candidate A.

We will estimate p by taking a random sample of n voters, and finding the *sample* proportion of voters who plan to vote for A. The latter is usually denoted $\widehat{p}$, pronounced "p-hat." (The symbolˆ is often used in statistics to mean "estimate of.")

Assign to each voter in the population a value of Y, 1 if he/she plans to vote for A, 0 otherwise. Let Y_i be the value of Y for the ith person in our sample. Then

$$\widehat{p} = \overline{Y} \tag{16.5}$$

where $\overline{Y}$ is the sample mean among the Y_i, and p is the population mean of Y.

So we are really working with means after all, and thus in order to get a confidence interval for p from $\widehat{p}$, we can use (16.4)! We have that an approximate 95% confidence interval for p is

$$\left(\widehat{p} - 1.96s/\sqrt{n}, \widehat{p} + 1.96s/\sqrt{n}\right) \tag{16.6}$$

where as before s^2 is the sample variance among the Y_i, defined in 15.23.

But there's more, because we can exploit the fact that in this special case, each Y_i is either 1 or 0, in order to save ourselves a bit of computation, as follows:

Recalling the convenient form of s^2, (15.24), we have

$$s^2 \;=\; \frac{1}{n}\sum_{i=1}^{n} Y_i^2 - \overline{Y}^2 \tag{16.7}$$

$$=\; \frac{1}{n}\sum_{i=1}^{n} Y_i - \overline{Y}^2 \tag{16.8}$$

$$=\; \overline{Y} - \overline{Y}^2 \tag{16.9}$$

$$=\; \widehat{p} - \widehat{p}^2 \tag{16.10}$$

Then (16.6) simplifies to

$$\left(\widehat{p} - 1.96\sqrt{\widehat{p}(1-\widehat{p})/n},\, \widehat{p} + 1.96\sqrt{\widehat{p}(1-\widehat{p})/n}\right) \tag{16.11}$$

16.4.2   That n vs. n-1 Thing Again

Recall Section 15.3.1.3, in which it was noted that this book's definition of the sample variance, (15.23), is a little at odds with the the way most books define it, (15.26). The above derivation sheds a bit more light on this topic.

In the way I've defined things here, I was consistent: I divided by n both in (15.23) and in (16.7). Yet most books divide by n-1 in the former case but by n in the latter case! Their version of (16.11) is exactly the same as mine, yet they use a different s in (16.4)—even though they too observe that the proportions case is just a special case of estimating means (as in (16.5)). So, another reason to divide by n in (15.23) is to be consistent.

Again, the difference is usually minuscule anyway, but conceptually it's important to understand. As noted earlier, the n-1 divisor is really just a historical accident.

16.4.3   Simulation Example Again

In our bus example above, suppose we also want our simulation to print out the (estimated) probability that one must wait longer than 6.4 minutes. As before, we'd also like a margin of error for the output.

We incorporate (16.11) into our program:

```
1   doexpt <- function(opt) {
2      lastarrival <- 0.0
3      while (lastarrival < opt)
```

```
4        lastarrival <- lastarrival + rexp(1,0.1)
5      return(lastarrival-opt)
6  }
7
8  observationpt <- 240
9  nreps <- 1000
10 waits <- vector(length=nreps)
11 for (rep in 1:nreps) waits[rep] <- doexpt(observationpt)
12 wbar <- mean(waits)
13 cat("approx. mean wait =",wbar,"\n")
14 s2 <- (mean(waits^2) - mean(wbar)^2)
15 s <- sqrt(s2)
16 radius <- 1.96*s/sqrt(nreps)
17 cat("approx. CI for EW =",wbar-radius,"to",wbar+radius,"\n")
18 prop <- length(waits[waits > 6.4]) / nreps
19 s2 <- prop*(1-prop)
20 s <- sqrt(s2)
21 radius <- 1.96*s/sqrt(nreps)
22 cat("approx. P(W > 6.4) =",prop,", with a margin of error of",radius,"\n")
```

When I ran this, the value printed out for $\widehat{p}$ was 0.54, with a margin of error of 0.03, thus an interval of (0.51,0.57). We would say, "We don't know the exact value of $P(W > 6.4)$, so we ran a simulation. The latter estimates this probability to be 0.54, with a 95% margin of error of 0.03."

16.4.4   Example: Davis Weights

Note again that this uses the same principles as our Davis weights example. Suppose we were interested in estimating the proportion of adults in Davis who weigh more than 150 pounds. Suppose that proportion is 0.45 in our sample of 1000 people. This would be our estimate $\widehat{p}$ for the population proportion p, and an approximate 95% confidence interval (16.11) for the population proportion would be (0.42,0.48). We would then say, "We are 95% confident that the true population proportion p of people who weigh over 150 pounds is between 0.42 and 0.48."

Note also that although we've used the word *proportion* in the Davis weights example instead of *probability*, they are the same. If I choose an adult at random from the population, the probability that his/her weight is more than 150 is equal to the proportion of adults in the population who have weights of more than 150.

And the same principles are used in opinion polls during presidential elections. Here p is the population proportion of people who plan to vote for the given candidate. This is an unknown quantity, which is exactly the point of polling a sample of people—to estimate that unknown quantity p. Our estimate is $\widehat{p}$, the proportion of people in our sample who plan to vote for the given candidate, and n is the number of people that we poll. We again use (16.11).

16.4.5   Interpretation

The same interpretation holds as before. Consider the examples in the last section:

- If each of you and 99 friends were to run the R program at the beginning of Section 16.4.4, you 100 people would get 100 confidence intervals for $P(W > 6.4)$. About 95 of you would have intervals that do contain that number.

- If each of you and 99 friends were to sample 1000 people in Davis and come up with confidence intervals for the true population proportion of people who weight more than 150 pounds, about 95 of you would have intervals that do contain that true population proportion.

- If each of you and 99 friends were to sample 1200 people in an election campaign, to estimate the true population proportion of people who will vote for candidate X, about 95 of you will have intervals that do contain this population proportion.

Of course, this is just a "thought experiment," whose goal is to understand what the term "95% confident" really means. In practice, we have just one sample and thus compute just one interval. But we say that the interval we compute has a 95% chance of containing the population value, since 95% of all intervals will contain it.

16.4.6   (Non-)Effect of the Population Size

Note that in both the Davis and election examples, it doesn't matter what the size of the population is. The approximate distribution of $\widehat{p}$ is N(p,p(1-p)/n), so the accuracy of $\widehat{p}$, depends only on p and n. So when people ask, "How a presidential election poll can get by with sampling only 1200 people, when there are more than 100,000,000 voters in the U.S.?" now you know the answer. (We'll discuss the question "Why 1200?" below.)

Another way to see this is to think of a situation in which we wish to estimate the probability p of heads for a certain coin. We toss the coin n times, and use $\widehat{p}$ as our estimate of p. Here our "population"—the population of all coin tosses—is infinite, yet it is still the case that 1200 tosses would be enough to get a good estimate of p.

16.4.7   Inferring the Number Polled

A news report tells us that in a poll, 54% of those polled supported Candidate A, with a 2.2% margin of error. Assuming that the methods here were used, with a 95% level of confidence, let's

find the approximate number polled.

$$0.022 = 1.96 \times \sqrt{0.54 \cdot 0.46/n} \tag{16.12}$$

Solving, we find that n is approximately 1972.

16.4.8   Planning Ahead

Now, why do the pollsters often sample 1200 people?

First, note that the maximum possible value of $\widehat{p}(1 - \widehat{p})$ is 0.25.[3] Then the pollsters know that their margin of error with n = 1200 will be at most $1.96 \times 0.5/\sqrt{1200}$, or about 3%, even before they poll anyone. They consider 3% to be sufficiently accurate for their purposes, so 1200 is the n they choose.

16.5   General Formation of Confidence Intervals from Approximately Normal Estimators

We would now like to move on to constructing confidence intervals for other settings than the case handled so far, estimation of a single population mean or proportion.

16.5.1   The Notion of a Standard Error

Suppose we are estimating some population quantity θ based on sample data $Y_1, ..., Y_n$. So far, our only examples have had θ as a population mean μ, a population proportion p. But we'll see other examples as things unfold in this and subsequent chapters.

Consider an estimate for θ, $\widehat{\theta}$, and suppose that the estimator is composed of some sum for which the Central Limit Theorem applies,[4] so that the approximate distribution of $\widehat{\theta}$ is normal with mean θ and some variance.

Ponder this sequence of points:

- $\widehat{\theta}$ is a random variable.

- Thus $\widehat{\theta}$ has a variance.

---

[3]Use calculus to find the maximum value of f(x) = x(1-x).
[4]Using more advanced tools (Section **??**), one can show approximate normality even in many nonlinear cases.

- Thus $\widehat{\theta}$ has a standard deviation η.

- Unfortunately, η is an unknown population quantity.

- But we may be able to estimate η from our sample data. Call that estimate $\widehat{\eta}$.

- We refer to $\widehat{\eta}$ as the *standard error* of $\widehat{\theta}$, or s.e.$\widehat{\theta}$.

Back in Section 16.2.1, we found a standard error for $\overline{W}$, the sample mean, using the following train of thought:

- $Var(\overline{W}) = \frac{\sigma^2}{n}$

- $\widehat{Var}(\overline{W}) = \frac{s^2}{n}$

- s.e.$(\overline{W}) = \frac{s}{\sqrt{n}}$

In many cases, deriving a standard error is more involved than the above, But the point is this:

Suppose $\widehat{\theta}$ is a sample-based estimator of a population quantity θ, and that, due to being composed of sums or some other reason, $\widehat{\theta}$ is approximately normally distributed with mean θ, and some (possibly unknown) variance. Then the quantity

$$\frac{\widehat{\theta} - \theta}{\text{s.e.}(\widehat{\theta})} \tag{16.13}$$

has an approximate N(0,1) distribution.

16.5.2   Forming General Confidence Intervals

That means we can mimic the derivation that led to (16.4). As with (16.1), write

$$0.95 \approx P\left(-1.96 < \frac{\widehat{\theta} - \theta}{\text{s.e.}(\widehat{\theta})} < 1.96\right) \tag{16.14}$$

After going through steps analogous to those following (16.1), we find that an approximate 95% confidence interval for θ is

$$\widehat{\theta} \pm 1.96 \cdot \text{s.e.}(\widehat{\theta}) \tag{16.15}$$

In other words, the margin of error is 1.96 s.e.$(\widehat{\theta})$.

The standard error of the estimate is one of the most commonly-used quantities in statistical applications. You will encounter it frequently in the output of R, for instance, and in the subsequent portions of this book. Make sure you understand what it means and how it is used.

And note again that $\sqrt{\widehat{p}(1-\widehat{p})/n}$ is the standard error of $\widehat{p}$.

16.5.3   Standard Errors of Combined Estimators

Here is further chance to exercise your skills in the mailing tubes regarding variance.

Suppose we have two population values to estimate, ω and γ, and that we are also interested in the quantity $\omega + 2\gamma$. We'll estimate the latter with $\widehat{\omega} + 2\widehat{\gamma}$. Suppose the standard errors of $\widehat{\omega}$ and $\widehat{\gamma}$ turn out to be 3.2 and 8.8, respectively, and that the two estimators are independent and approximately normal.[5] Let's find the standard error of $\widehat{\omega} + 2\widehat{\gamma}$.

We know from the material surrounding (8.16) that $\widehat{\omega} + 2\widehat{\gamma}$ has an approximately normal distribution with variance

$$Var(\widehat{\omega}) + 2^2 Var(\widehat{\gamma}) \tag{16.16}$$

Thus the standard error of $\widehat{\omega} + 2\widehat{\gamma}$ is

$$\sqrt{\cdot 3.2^2 + 2^2 \cdot 8.8^2} \tag{16.17}$$

Now that we know the standard error of $\widehat{\omega} + 2\widehat{\gamma}$, we can use it in (16.15). We add and subtract 1.96 times (16.17) to $\widehat{\omega} + 2\widehat{\gamma}$, and that is our interval.

In general, for constants a and b, an approximate 95% confidence interval for the population quantity $a\omega + b\gamma$ is

$$a\widehat{\omega} + b\widehat{\gamma} \pm 1.96 \ \sqrt{a^2 s.e.^2(\widehat{\omega}) + b^2 s.e.^2(\widehat{\gamma})} \tag{16.18}$$

We can go even further. If $\widehat{\omega}$ and $\widehat{\gamma}$ are not independent but have known covariance, we can use the methods of Chapter 12 to obtain a standard error for any linear combination of these two estimators.

---

[5]Technically, the term *standard error* is only used for approximately normal estimators anyway.

16.6 Confidence Intervals for Differences of Means or Proportions

16.6.1 Independent Samples

Suppose in our sampling of people in Davis we are mainly interested in the difference in weights between men and women. Let $\overline{X}$ and n_1 denote the sample mean and sample size for men, and let $\overline{Y}$ and n_2 for the women. Denote the population means and variances by μ_i and σ_i^2, i = 1,2. We wish to find a confidence interval for $\mu_1 - \mu_2$. The natural estimator for that quantity is $\overline{X} - \overline{Y}$.

So, how can we form a confidence interval for $\mu_1 - \mu_2$ using $\overline{X} - \overline{Y}$? Since the latter quantity is composed of sums, we can use (16.15) and (16.18). Here:

- $a = 1,\ b = -1$

- $\omega = \mu_1,\ \gamma = \mu_2$

- $\hat{\omega} = \overline{X},\ \hat{\gamma} = \overline{Y}$

But we know from before that $s.e.(\overline{X} = s_1/\sqrt{n}$, where s_1^2 is the sample variance for the men,

$$s_1^2 = \frac{1}{n_1} \sum_{i=1}^{n_1} (X_i - \overline{X})^2 \tag{16.19}$$

and similarly for $\overline{Y}$ and the women. So, we have

$$\text{s.e.}(\overline{X} - \overline{Y}) = \sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}} \tag{16.20}$$

Thus (16.15) tells us that an approximate 95% confidence interval for $\mu_1 - \mu_2$ is

$$\left(\overline{X} - \overline{Y} - 1.96\sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}}, \overline{X} - \overline{Y} + 1.96\sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}} \right) \tag{16.21}$$

What about confidence intervals for the difference in two population proportions $p_1 - p_2$? Recalling that in Section 16.4 we noted that proportions are special cases of means, we see that finding a confidence interval for the difference in two proportions is covered by (16.21). Here

- $\overline{X}$ reduces to $\widehat{p}_1$

- $\overline{Y}$ reduces to $\widehat{p}_2$

- s_1^2 reduces to $\widehat{p}_1(1 - \widehat{p}_1)$

- s_2^2 reduces to $\widehat{p}_2(1 - \widehat{p}_2)$

So, (16.21) reduces to

$$\widehat{p}_1 - \widehat{p}_2 \pm R \tag{16.22}$$

where the radius R is

$$1.96 \ \sqrt{\frac{\widehat{p}_1(1 - \widehat{p}_1)}{n_1} + \frac{\widehat{p}_2(1 - \widehat{p}_2)}{n_2}} \tag{16.23}$$

16.6.2 Example: Network Security Application

In a network security application, C. Mano *et al*[6] compare round-trip travel time for packets involved in the same application in certain wired and wireless networks. The data was as follows:

sample	sample mean	sample s.d.	sample size
wired	2.000	6.299	436
wireless	11.520	9.939	344

We had observed quite a difference, 11.52 versus 2.00, but could it be due to sampling variation? Maybe we have unusual samples? This calls for a confidence interval!

Then a 95% confidence interval for the difference between wireless and wired networks is

$$11.520 - 2.000 \pm 1.96\sqrt{\frac{9.939^2}{344} + \frac{6.299^2}{436}} = 9.52 \pm 1.22 \tag{16.24}$$

So you can see that there is a big difference between the two networks, even after allowing for sampling variation.

---

[6]RIPPS: Rogue Identifying Packet Payload Slicer Detecting Unauthorized Wireless Hosts Through Network Traffic Conditioning, C. Mano and a ton of other authors, ACM TRANSACTIONS ON INFORMATION SYSTEMS AND SECURITY, May 2007.

16.6.3   Dependent Samples

Note carefully, though, that a key point above was the independence of the two samples. By contrast, suppose we wish, for instance, to find a confidence interval for $\nu_1 - \nu_2$, the difference in mean heights in Davis of 15-year-old and 10-year-old children, and suppose our data consist of pairs of height measurements at the two ages on *the same children.* In other words, we have a sample of n children, and for the ith child we have his/her height U_i at age 15 and V_i at age 10. Let $\overline{U}$ and $\overline{V}$ denote the sample means.

The problem is that the two sample means are not independent. If a child is taller than his/her peers at age 15, he/she was probably taller than them when they were all age 10. In other words, for each i, V_i and U_i are positively correlated, and thus the same is true for $\overline{V}$ and $\overline{U}$. Thus we cannot use (16.21).

As always, it is instructive to consider this in "notebook" terms. Suppose on one particular sample at age 10—one line of the notebook—we just happen to have a lot of big kids. Then $\overline{V}$ is large. Well, if we look at the same kids later at age 15, they're liable to be bigger than the average 15-year-old too. In other words, among the notebook lines in which $\overline{V}$ is large, many of them will have $\overline{U}$ large too.

Since $\overline{U}$ is approximately normally distributed with mean ν_1, about half of the notebook lines will have $\overline{U} > \nu_1$. Similarly, about half of the notebook lines will have $\overline{V} > \nu_2$. But the nonindependence will be reflected in MORE than one-fourth of the lines having both $\overline{U} > \nu_1$ and $\overline{V} > \nu_2$. (If the two sample means were 100% correlated, that fraction would be 1.0.)

Contrast that with a sample scheme in which we sample some 10-year-olds and some 15-year-olds, say at the same time. Now *there are different kids in each of the two samples.* So, if by happenstance we get some big kids in the first sample, that has no impact on which kids we get in the second sample. In other words, $\overline{V}$ and $\overline{U}$ will be independent. In this case, one-fourth of the lines will have both $\overline{U} > \nu_1$ and $\overline{V} > \nu_2$.

So, we cannot get a confidence interval for $\nu_1 - \nu_2$ from (16.21), since the latter assumes that the two sample means are independent. What to do?

The key to the resolution of this problem is that the random variables $T_i = V_i - U_i$, i = 1,2,...,n are still independent. Thus we can use (16.4) on these values, so that our approximate 95% confidence interval is

$$\left(\overline{T} - 1.96\frac{s}{\sqrt{n}}, \overline{T} + 1.96\frac{s}{\sqrt{n}}\right) \tag{16.25}$$

where $\overline{T}$ and s^2 are the sample mean and sample variance of the T_i.

A common situation in which we have dependent samples is that in which we are comparing two

dependent proportions. Suppose for example that there are three candidates running for a political office, A, B and C. We poll 1,000 voters and ask whom they plan to vote for. Let p_A, p_B and p_C be the three population proportions of people planning to vote for the various candidates, and let $\widehat{p}_A$, $\widehat{p}_B$ and $\widehat{p}_C$ be the corresponding sample proportions.

Suppose we wish to form a confidence interval for $p_A - p_B$. Clearly, the two sample proportions are not independent random variables, since for instance if $\widehat{p}_A = 1$ then we know for sure that $\widehat{p}_B$ is 0.

Or to put it another way, define the indicator variables U_i and V_i as above, with for example U_i being 1 or 0, according to whether the ith person in our sample plans to vote for A or not, with V_i being defined similarly for B. Since U_i and V_i are "measurements" on *the same person*, they are not independent, and thus $\widehat{p}_A$ and $\widehat{p}_B$ are not independent either.

Note by the way that while the two sample means in our kids' height example above were positively correlated, in this voter poll example, the two sample proportions are negatively correlated.

So, we cannot form a confidence interval for $p_A - p_B$ by using (16.22). What can we do instead?

We'll use the fact that the vector $(N_A, N_B, N_C)^T$ has a multinomial distribution, where N_A, N_B and N_C denote the numbers of people in our sample who state they will vote for the various candidates (so that for instance $\widehat{p}_A = N_A/1000$).

Now to compute $Var(\widehat{p}_A - \widehat{p}_B)$, we make use of (12.10):

$$Var(\widehat{p}_A - \widehat{p}_B) = Var(\widehat{p}_A) + Var(\widehat{p}_B) - 2Cov(\widehat{p}_A, \widehat{p}_B) \tag{16.26}$$

Or, we could have taken a matrix approach, using (12.54) with A equal to the row vector (1,-1,0).

So, using (13.102), the standard error of $\widehat{p}_A - \widehat{p}_B$ is

$$\sqrt{0.001\widehat{p}_A(1 - \widehat{p}_A) + 0.001\widehat{p}_B(1 - \widehat{p}_B) + 0.002\widehat{p}_A\widehat{p}_B} \tag{16.27}$$

16.6.4 Example: Machine Classification of Forest Covers

Remote sensing is machine classification of type from variables observed aerially, typically by satellite. The application we'll consider here involves forest cover type for a given location; there are seven different types. (See Blackard, Jock A. and Denis J. Dean, 2000, "Comparative Accuracies of Artificial Neural Networks and Discriminant Analysis in Predicting Forest Cover Types from Cartographic Variables," *Computers and Electronics in Agriculture*, 24(3):131-151.) Direct observation of the cover type is either too expensive or may suffer from land access permission issues. So, we wish to guess cover type from other variables that we can more easily obtain.

One of the variables was the amount of hillside shade at noon, which we'll call HS12. *Here's our goal:* Let μ_1 and μ_2 be the population mean HS12 among sites having cover types 1 and 2, respectively. If $\mu_1 - \mu_2$ is large, then HS12 would be a good predictor of whether the cover type is 1 or 2.

So, we wish to estimate $\mu_1 - \mu_2$ from our data, in which we do know cover type. There were over 50,000 observations, but for simplicity we'll just use the first 1,000 here. Let's find an approximate 95% confidence interval for $\mu_1 - \mu_2$. The two sample means were 223.8 and 226.3, with s values of 15.3 and 14.3, and the sample sizes were 226 and 585.

Using (16.21), we have that the interval is

$$223.8 - 226.3 \pm 1.96\sqrt{\frac{15.3^2}{226} + \frac{14.3^2}{585}} = -2.5 \pm 2.3 = (-4.8, -0.3) \qquad (16.28)$$

Given that HS12 values are in the 200 range (see the sample means), this difference between them actually is not very large. This is a great illustration of an important principle, it will turn out in Section 17.11.

As another illustration of confidence intervals, let's find one for the difference in population proportions of sites that have cover types 1 and 2. Our sample estimate is

$$\widehat{p_1} - \widehat{p_2} = 0.226 - 0.585 = -0.359 \qquad (16.29)$$

The standard error of this quantity, from (16.27), is

$$\sqrt{0.001 \cdot 0.226 \cdot 0.774\, 0.001 \cdot 0.585 \cdot 0.415 + 002 \cdot 0.226 \cdot 0.585} = 0.019 \qquad (16.30)$$

That gives us a confidence interval of

$$-0.359 \pm 1.96 \cdot 0.019 = (-0.397, -0.321) \qquad (16.31)$$

16.7   And What About the Student-t Distribution?

Far better an approximate answer to the right question, which is often vague, than an exact answer to the wrong question, which can always be made precise—John Tukey, pioneering statistician at Bell Labs

Another thing we are not doing here is to use the **Student t-distribution**. That is the name of the distribution of the quantity

$$T = \frac{\overline{W} - \mu}{\tilde{s}/\sqrt{n-1}} \tag{16.32}$$

where $\tilde{s}^2$ is the version of the sample variance in which we divide by n-1 instead of by n, i.e. (15.24).

Note carefully that we are assuming that the W_i themselves—not just $\overline{W}$—have a normal distribution. In other words, if we are studying human weight, say, then the assumption is that weight follows an exact bell-shaped curve. The exact distribution of T is called the **Student t-distribution with n-1 degrees of freedom**. These distributions thus form a one-parameter family, with the degrees of freedom being the parameter.

The general definition of the Student-t family is distribution of ratios $U/\sqrt{V/k}$, where

- U has a N(0,1) distribution

- V has a chi-squared distribution with k degrees of freedom

- U and V are independent

It can be shown that in (16.32), if the sampled population has a normal distribution, then $(\overline{W}-\mu)/\sigma$ and $\tilde{s}^2/\sigma^2$ actually do satisfy the above conditions on U and V, respectively, with k = n-1. (If we are forming a confidence interval for the difference of two means, the calculation of degrees of freedom becomes more complicated, but it is not important here.)

This distribution has been tabulated. In R, for instance, the functions **dt()**, **pt()** and so on play the same roles as **dnorm()**, **pnorm()** etc. do for the normal family. The call **qt(0.975,9)** returns 2.26. This enables us to get a confidence interval for μ from a sample of size 10, at EXACTLY a 95% confidence level, rather than being at an APPROXIMATE 95% level as we have had here, as follows.

We start with (16.1), replacing 1.96 by 2.26, $(\bar{W} - \mu)/(\sigma/\sqrt{n})$ by T, and $\approx$ by $=$. Doing the same algebra, we find the following confidence interval for μ:

$$(\overline{W} - 2.26\frac{\tilde{s}}{\sqrt{10}}, \overline{W} + 2.26\frac{\tilde{s}}{\sqrt{10}}) \tag{16.33}$$

Of course, for general n, replace 2.26 by $t_{0.975,n-1}$, the 0.975 quantile of the t-distribution with n-1 degrees of freedom. The distribution is tabulated by the R functions **dt()**, **pt()** and so on.

I do not use the t-distribution here because:

- It depends on the parent population having an exact normal distribution, which is never really true. In the Davis case, for instance, people's weights are approximately normally distributed, but definitely not exactly so. For that to be exactly the case, some people would have to have weights of say, a billion pounds, or negative weights, since any normal distribution takes on all values from $-\infty$ to ∞.

- For large n, the difference between the t-distribution and N(0,1) is negligible anyway. That wasn't true in the case n = 10 above, where our confidence interval multiplied the standard error by 2.26 instead of 1.96 as we'd seen earlier. But for n = 50, the 2.26 already shrinks to 2.01, and for n = 100, it is 1.98.

16.8   R Computation

The R function **t.test()** forms confidence intervals for a single mean or for the difference of two means. In the latter case, the two samples must be independent; otherwise, do the single-mean CI on differences, as in Section 16.6.3.

This function uses the Student-t distribution, rather than the normal, but as discussed in Section 16.7, the difference is negligible except in small samples.

Thus you can conveniently use **t.test()** to form a confidence interval for a single mean, instead of computing (16.4) yourself (or writing the R code yourself).

It's slightly more complicated in the case of forming a confidence interval for the difference of two means. The **t.test()** function will do that for you too, but will make the assmption that we have $\sigma_1^2 = \sigma_2^2$ in Section 16.6.1. Unless you believe there is a huge difference between the two population variances, this approximation is not bad.

16.9   Example: Pro Baseball Data

The SOCR data repository at the UCLA Statistics Department includes a data set on major league baseball players, at `http://wiki.stat.ucla.edu/socr/index.php/SOCR_Data_MLB_HeightsWeights`. There are 1035 players in the sample, with the variables Name, Team, Position, Height, Weight and Age. I downloaded it and placed it into a file **Baseball.dat**

16.9.1   R Code

First we read in the data:

```
> players <- read.table("Baseball.dat",header=T)
Error in scan(file, what, nmax, sep, dec, quote, skip, nlines, na.strings,
:
  line 641 did not have 6 elements
```

Oops! The entry for one player, Kirk Saarloos, did not have a weight figure. So I edited the file by hand, placing the string "NA" there for weight; this is R's code for missing data. I then tried again:

```
> players <- read.table("Baseball.dat",header=T)
> head(players)
            Name Team        Position Height Weight   Age
1    Adam_Donachie  BAL         Catcher     74    180 22.99
2        Paul_Bako  BAL         Catcher     74    215 34.69
3 Ramon_Hernandez  BAL         Catcher     72    210 30.78
4    Kevin_Millar  BAL   First_Baseman     72    210 35.43
5     Chris_Gomez  BAL   First_Baseman     73    188 35.71
6   Brian_Roberts  BAL  Second_Baseman     69    176 29.39
```

I read in the file **Baseball.dat**, whose first line consisted of a header giving the names of the variables. I assigned the result to **players**, whose type will be that of an R **data frame**. I then called R's **head()** function, to take a look at the results to make sure things are OK.

We could then query various items in the object **players**, say the mean weight (not conditioned on height), via `players[,5]` or `players$Weight`.

16.9.2 Analysis

Let's find an approximate 95% confidence interval for the population mean weight of catchers.[7]

```
> catch <- players[players$Position == "Catcher",]
> t.test(catch$Weight)

        One Sample t-test

data:  catch$Weight
t = 113.1467, df = 75, p-value < 2.2e-16
alternative hypothesis: true mean is not equal to 0
95 percent confidence interval:
```

---

[7]Note that this is an observational study, as discussed in Section 15.4, with attendant care needed for the conclusions we derive from this data.

```
 200.7315 207.9264
sample estimates:
mean of x
 204.3289
```

Our CI is (200.7,207.9).

(There is material in the above output on significance testing, which we will cover in the next chapter.)

How about a comparison in population mean weights between catchers and first basemen?

```
> firstb <- players[players$Position == "First_Baseman",]
> t.test(catch$Weight,firstb$Weight)

        Welch Two Sample t-test

data:  catch$Weight and firstb$Weight
t = -2.7985, df = 102.626, p-value = 0.006133
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -15.002763  -2.557524
sample estimates:
mean of x mean of y
```

We might be interested in inference concerning the population proportion of catchers older than 32:

```
> old <- (catch$Age > 32)
> head(old)
[1] FALSE  TRUE FALSE FALSE FALSE FALSE
> old <- as.integer(old)
> head(old)
[1] 0 1 0 0 0 0
> t.test(old)

        One Sample t-test

data:  old
t = 5.705, df = 75, p-value = 2.189e-07
alternative hypothesis: true mean is not equal to 0
95 percent confidence interval:
 0.1969573 0.4083058
```

```
sample estimates:
mean of x
0.3026316
```

Note that the intervals, especially the last one, are rather wide. We just don't have enough catchers to get much accuracy. How many are there?[8]

```
> nrow(catch)
[1] 76
```

16.10   Example: UCI Bank Marketing Dataset

This data set was obtained from the UC Irvine Machine Learning Data Repository, `http://archive.ics.uci.edu/ml/about.html`. A bank in Portugal had developed a new type of account, and they were interested in studying what types of customers would be more likely to switch to the new account.

```
> bank <- read.table("bank-full.csv",header=T,sep=";")
> head(bank)
  age           job marital education default balance housing loan  day
1  58    management married  tertiary      no    2143     yes   no
5
2  44    technician  single secondary      no      29     yes   no
5
3  33  entrepreneur married secondary      no       2     yes  yes
5
4  47   blue-collar married   unknown      no    1506     yes   no
5
5  33       unknown  single   unknown      no       1      no   no
5
6  35    management married  tertiary      no     231     yes   no
5
  month duration campaign pdays previous poutcome   y
1   may      261        1    -1        0  unknown  no
2   may      151        1    -1        0  unknown  no
3   may       76        1    -1        0  unknown  no
4   may       92        1    -1        0  unknown  no
```

---

[8]In order to do any of this, we are tacitly assuming that our players are a sample of some general population of players, say, past, present and future., even if we have all the present ones. This is very common in applied statistical analysis.

```
5    may        198       1    -1       0   unknown no
6    may        139       1    -1       0   unknown no
```

(The variable **contact** has been omitted here, to fit the display on the page.)

There are many variables here, explained in more detail at the UCI site. We'll come back to this example, but let's do one quick confidence interval. Here we will compare the success rates of the marketing campaign for married and unmarried people:

```
> marrd <- bank[bank$marital == "married",]
> unmarrd <- bank[bank$marital != "married",]
> t.test(marrd$success,unmarrd$success)


        Welch Two Sample t-test


data:  marrd$success and unmarrd$success
t = -12.471, df = 34676.26, p-value < 2.2e-16
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -0.04578514 -0.03334804
sample estimates:
mean of x mean of y
0.1012347 0.1408012
```

So, we are 95% confident that the population success rate is between 3.3% and 4.6% less for married people.

Note by the way that there are more than 46,000 people in this sample. So, the Student-t and N(0,1) distributions are now indistinguishable.

16.11   Example: Amazon Links

This example involves the Amazon product co-purchasing network, March 2 2003. The data set is large but simple. It stores a directed graph of what links to what: If a record show i then j, it means that i is often co-purchased with j (though not necessarily vice versa). Let's find a confidence interval for the mean number of inlinks, i.e. links into a node.

Actually, even the R manipulations are not so trivial, so here is the complete code (`http://snap.stanford.edu/data/amazon0302.html`):

```
1  mzn <- read.table("amazon0302.txt",header=F)
2  # cut down the data set for convenience
```

```
3  mzn1000 <- mzn[mzn[1.] <= 1000 & mzn[,2] <= 1000,]
4  # make an R list, one element per value of j
5  degrees1000 <- split(mzn1000,mzn1000[,2])
6  # by finding the number of rows in each matrix, we get the numbers of
7  # inlinks
8  indegrees1000 <- sapply(degrees1000,nrow)
```

Now run **t.test()**:

```
> t.test(indegrees1000)

        One Sample t-test

data:  indegrees1000
t = 35.0279, df = 1000, p-value < 2.2e-16
alternative hypothesis: true mean is not equal to 0
95 percent confidence interval:
 3.728759 4.171340
sample estimates:
mean of x
  3.95005
```

So, in our sample data, the mean number of inlinks was 3.95, and we are 95% confident that the true population mean is between 3.73 and 4.17.

16.12   Example: Master's Degrees in CS/EE

In an analysis of the National Survey of College Graduates, I looked at workers in CS or EE, who have CS or EE degrees.[9] I had them in R data frames named **cs** and **ee**, each of which had an indicator variable **ms** signifying that the worker has a Master's degree (but not a PhD). Let's see the difference between CS and EE on this variable:

```
> t.test(cs$ms,ee$ms)

        Welch Two Sample t-test

data:  cs$ms and ee$ms
t = 2.4895, df = 1878.108, p-value = 0.01288
alternative hypothesis: true difference in means is not equal to 0
```

---

[9]There were various other restrictions, which I will not go into here.

```
95 percent confidence interval:
 0.01073580 0.09045689
sample estimates:
mean of x mean of y
0.3560551 0.3054588
```

So, in our sample, 35.6% and 30.5% of the two groups had Master's degrees, and we are 95% confident that the true population difference in proportions of Master's degrees in the two groups is between 0.01 and 0.09.

16.13   Other Confidence Levels

We have been using 95% as our confidence level. This is common, but of course not unique. We can for instance use 90%, which gives us a narrower interval (in (16.4), we multiply by 1.65 instead of by 1.96. (The reader should check this, using the **qnorm()** function.) Narrower is better, of course, but it comes at the expense of lower confidence.

A confidence interval's error rate is usually denoted by $1-\alpha$, so a 95% confidence level has $\alpha = 0.05$.

16.14   One More Time: Why Do We Use Confidence Intervals?

After all the variations on a theme in the very long Section 16.1, it is easy to lose sight of the goal, so let's review:

Almost everyone is familiar with the term "margin of error," given in every TV news report during elections. The report will say something like, "In our poll, 62% stated that they plan to vote for Ms. X. The margin of error is 3%." Those two numbers, 62% and 3%, form the essence of confidence intervals:

- The 62% figure is our estimate of p, the true population fraction of people who plan to vote for Ms. X.

- Recognizing that that 62% figure is only a sample estimate of p, we wish to have a measure of how accurate the figure is—our margin of error. Though the poll reports don't say this, what they are actually saying is that we are 95% sure that the true population value p is in the range 0.62 ± 0.03.

So, a confidence interval is nothing more than the concept of the $a \pm b$ range that we are so familiar with.

Chapter 17

Introduction to Significance Tests

Suppose (just for fun, but with the same pattern as in more serious examples) you have a coin that will be flipped at the Super Bowl to see who gets the first kickoff. (We'll assume slightly different rules here. The coin is not "called." Instead, it is agreed beforehand that if the coin comes up heads, Team A will get the kickoff, and otherwise it will be Team B.) You want to assess for "fairness." Let p be the probability of heads for the coin.

You could toss the coin, say, 100 times, and then form a confidence interval for p using (16.11). The width of the interval would tell you the margin of error, i.e. it tells you whether 100 tosses were enough for the accuracy you want, and the location of the interval would tell you whether the coin is "fair" enough.

For instance, if your interval were (0.49,0.54), you might feel satisfied that this coin is reasonably fair. In fact, **note carefully that even if the interval were, say, (0.502,0.506), you would still consider the coin to be reasonably fair**; the fact that the interval did not contain 0.5 is irrelevant, as the entire interval would be reasonably near 0.5.

However, this process would not be the way it's traditionally done. Most users of statistics would use the toss data to test the **null hypothesis**

$$H_0 : p = 0.5 \tag{17.1}$$

against the **alternate hypothesis**

$$H_A : p \neq 0.5 \tag{17.2}$$

For reasons that will be explained below, this procedure is called **significance testing**. It forms

the very core of statistical inference as practiced today. This, however, is unfortunate, as there are some serious problems that have been recognized with this procedure. We will first discuss the mechanics of the procedure, and then look closely at the problems with it in Section 17.11.

17.1   The Basics

Here's how significance testing works.

The approach is to consider H_0 "innocent until proven guilty," meaning that we assume H_0 is true unless the data give strong evidence to the contrary. **KEEP THIS IN MIND!**—we are continually asking, "What if...?"

The basic plan of attack is this:

> We will toss the coin n times. Then we will believe that the coin is fair unless the number of heads is "suspiciously" extreme, i.e. much less than n/2 or much more than n/2.

Let p denote the true probability of heads for our coin. As in Section 16.4.1, let $\widehat{p}$ denote the proportion of heads in our sample of n tosses. We observed in that section that $\widehat{p}$ is a special case of a sample mean (it's a mean of 1s and 0s). We also found that the standard deviation of $\widehat{p}$ is $\sqrt{p(1-p)/n}$, so that the standard error (Section 16.5.1) is $\sqrt{\widehat{p}(1-\widehat{p})/n}$.

In other words, the material surrounding (16.13) tells us that

$$\frac{\widehat{p} - p}{\sqrt{\frac{1}{n} \cdot \widehat{p}(1 - \widehat{p})}} \tag{17.3}$$

has an approximate N(0,1) distribution.

But remember, we are going to assume H_0 for now, until and unless we find strong evidence to the contrary. So, in the numerator of (17.3), p = 0.5. And in the denominator, why use the standard error when we "know" (under our provisional assumption) that p = 0.5? The exact standard deviation of $\widehat{p}$ is $\sqrt{p(1-p)/n}$. So, replace (17.3) by

$$Z = \frac{\widehat{p} - 0.5}{\sqrt{\frac{1}{n} \cdot 0.5(1 - 0.5)}} \tag{17.4}$$

then Z has an approximate N(0,1) distribution under the assumption that H_0 is true.

Now recall from the derivation of (16.4) that -1.96 and 1.96 are the lower- and upper-2.5% points of the N(0,1) distribution. Thus,

$$P(Z < -1.96 \text{ or } Z > 1.96) \approx 0.05 \tag{17.5}$$

Now here is the point: After we collect our data, in this case by tossing the coin n times, we compute $\widehat{p}$ from that data, and then compute Z from (17.4). If Z is smaller than -1.96 or larger than 1.96, we reason as follows:

> Hmmm, Z would stray that far from 0 only 5% of the time. So, either I have to believe that a rare event has occurred, or I must abandon my assumption that H_0 is true.

For instance, say n = 100 and we get 62 heads in our sample. That gives us Z = 2.4, in that "rare" range. We then **reject** H_0, and announce to the world that this is an unfair coin. We say, "The value of p is significantly different from 0.5."

The 5% "suspicion criterion" used above is called the **significance level**, typically denoted α. One common statement is "We rejected H_0 at the 5% level."

On the other hand, suppose we get 47 heads in our sample. Then Z = -0.60. Again, taking 5% as our significance level, this value of Z would not be deemed suspicious, as it occurs frequently. We would then say "We accept H_0 at the 5% level," or "We find that p is not significantly different from 0.5."

The word *significant* is misleading. It should NOT be confused with *important*. It simply is saying we don't believe the observed value of Z is a rare event, which it would be under H_0; we have instead decided to abandon our believe that H_0 is true.

Note by the way that Z values of -1.96 and 1.96 correspond getting $50 - 1.96 \cdot 0.5 \cdot \sqrt{100}$ or $50 + 1.96 \cdot 0.5 \cdot \sqrt{100}$ heads, i.e. roughly 40 or 60. In other words, we can describe our rejection rule to be "Reject if we get fewer than 40 or more than 60 heads, out of our 100 tosses."

17.2   General Testing Based on Normally Distributed Estimators

In Section 16.5, we developed a method of constructing confidence intervals for general approximately normally distributed estimators. Now we do the same for significance testing.

Suppose $\widehat{\theta}$ is an approximately normally distributed estimator of some population value θ. Then

to test $H_0 : \theta = c$, form the test statistic

$$Z = \frac{\widehat{\theta} - c}{s.e.(\widehat{\theta})} \tag{17.6}$$

where $s.e.(\widehat{\theta})$ is the standard error of $\widehat{\theta}$,[1] and proceed as before:

Reject $H_0 : \theta = c$ at the significance level of $\alpha = 0.05$ if $|Z| \geq 1.96$.

17.3   Example: Network Security

Let's look at the network security example in Section 16.6.2 again. Here $\widehat{\theta} = \overline{X} - \overline{Y}$, and c is presumably 0 (depending on the goals of Mano *et al*). From 16.20, the standard error works out to 0.61. So, our test statistic (17.6) is

$$Z = \frac{\overline{X} - \overline{Y} - 0}{0.61} = \frac{11.52 - 2.00}{0.61} = 15.61 \tag{17.7}$$

This is definitely larger in absolute value than 1.96, so we reject H_0, and conclude that the population mean round-trip times are different in the wired and wireless cases.

17.4   The Notion of "p-Values"

Recall the coin example in Section 17.1, in which we got 62 heads, i.e. Z = 2.4. Since 2.4 is considerably larger than 1.96, our cutoff for rejection, we might say that in some sense we not only rejected H_0, we actually <u>strongly</u> rejected it.

To quantify that notion, we compute something called the **observed significance level**, more often called the **p-value**.

We ask,

> We rejected H_0 at the 5% level. Clearly, we would have rejected it even at some small—thus more stringent—levels. What is the smallest such level? Call this the p-value of the test.

---

[1]See Section 16.5. Or, if we know the exact standard deviation of $\widehat{\theta}$ under H_0, which was the case in our coin example above, we could use that, for a better normal approximation.

By checking a table of the N(0,1) distribution, or by calling **pnorm(2.40)** in R, we would find that the N(0,1) distribution has area 0.008 to the right of 2.40, and of course by symmetry there is an equal area to the left of -2.40. That's a total area of 0.016. In other words, we would have been able to reject H_0 even at the much more stringent significance level of 0.016 (the 1.6% level) instead of 0.05. So, Z = 2.40 would be considered even more significant than Z = 1.96. In the research community it is customary to say, "The p-value was 0.016." [2] The smaller the p-value, the more significant the results are considered.

In our network security example above in which Z was 15.61, the value is literally "off the chart"; **pnorm(15.61)** returns a value of 1. Of course, it's a tiny bit less than 1, but it is so far out in the right tail of the N(0,1) distribution that the area to the right is essentially 0. So the p-value would be essentially 0, and the result would be treated as very, very highly significant.

In computer output or research reports, we often see small p-values being denoted by asterisks. There is generally one asterisk for p under 0.05, two for p less than 0.01, three for 0.001, etc. The more asterisks, the more significant the data is supposed to be. See for instance the R regression output on page 392.

17.5 Example: Bank Data

Consider again the bank marketing data in Section 16.10. Our comparison was between marketing campaign success rates for married and unmarried customers. The p-value was quite tiny, 2.2×10^{-16}, but be careful interpreting this.

First, don't take that p-value as exact by any means. Though our sample sizes are certainly large enough for the Central Limit Theorem to work well, that is in the heart of the distribution, not the far tails. So, just take the p-value as "tiny," and leave it at that.

Second, although the standard description for a test with such a small p-value is "very highly significant," keep in mind that the difference between the two groups was not that large. The confidence interval we are 95% confident that the population success rate is between 3.3% and 4.6% less for married people. That is an interesting difference and possibly of some use to the marketing people, but it is NOT large.

---

[2]The 'p' in "p-value" of course stands for "probability," meaning the probably that a N(0,1) random variable would stray as far, or further, from 0 as our observed Z here. By the way, be careful not to confuse this with the quantity p in our coin example, the probability of heads.

17.6 One-Sided H_A

Suppose that—somehow—we are sure that our coin in the example above is either fair or it is more heavily weighted towards heads. Then we would take our alternate hypothesis to be

$$H_A : p > 0.5 \tag{17.8}$$

A "rare event" which could make us abandon our belief in H_0 would now be if Z in (17.4) is very large in the positive direction. So, with $\alpha = 0.05$, we call **qnorm(0.95)**, and find that our rule would now be to reject H_0 if $Z > 1.65$.

One-sided tests are not common, as their assumptions are often difficult to justify.

17.7 Exact Tests

Remember, the tests we've seen so far are all approximate. In (17.4), for instance, $\widehat{p}$ had an approximate normal distribution, so that the distribution of Z was approximately N(0,1). Thus the significance level α was approximate, as were the p-values and so on.[3]

But the only reason our tests were approximate is that we only had the *approximate* distribution of our test statistic Z, or equivalently, we only had the approximate distribution of our estimator, e.g. $\widehat{p}$. If we have an *exact* distribution to work with, then we can perform an exact test.

17.7.1 Example: Test for Biased Coin

Let's consider the coin example again, with the one-sided alternative (17.8) . To keep things simple, let's suppose we toss the coin 10 times. We will make our decision based on X, the number of heads out of 10 tosses. Suppose we set our threshhold for "strong evidence" again H_0 to be 8 heads, i.e. we will reject H_0 if $X \geq 8$. What will α be?

$$\alpha = \sum_{i=8}^{10} P(X = i) = \sum_{i=8}^{10} \binom{10}{i} \left(\frac{1}{2}\right)^{10} = 0.055 \tag{17.9}$$

That's not the usual 0.05. Clearly we cannot get an exact significance level of 0.05,[4] but our α is

---

[3]Another class of probabilities which would be approximate would be the **power** values. These are the probabilities of rejecting H_0 if the latter is not true. We would speak, for instance, of the power of our test at p = 0.55, meaning the chances that we would reject the null hypothesis if the true population value of p were 0.55.

[4]Actually, it could be done by introducing some randomization to our test.

exactly 0.055, so this is an exact test.

So, we will believe that this coin is perfectly balanced, unless we get eight or more heads in our 10 tosses. The latter event would be very unlikely (probability only 5.5%) if H_0 were true, so we decide not to believe that H_0 is true.

17.7.2 Example: Improved Light Bulbs

Suppose lifetimes of lightbulbs are exponentially distributed with mean μ. In the past, $\mu = 1000$, but there is a claim that the new light bulbs are improved and $\mu > 1000$. To test that claim, we will sample 10 lightbulbs, getting lifetimes $X_1, ..., X_{10}$, and compute the sample mean $\overline{X}$. We will then perform a significance test of

$$H_0 : \mu = 1000 \tag{17.10}$$

vs.

$$H_A : \mu > 1000 \tag{17.11}$$

It is natural to have our test take the form in which we reject H_0 if

$$\overline{X} > w \tag{17.12}$$

for some constant w chosen so that

$$P(\overline{X} > w) = 0.05 \tag{17.13}$$

under H_0. Suppose we want an exact test, not one based on a normal approximation.

Remember, we are making our calculations under the assumption that H_0 is true. Now recall (Section 7.6.4.1) that $10\overline{X}$, the sum of the X_i, has a gamma distribution, with r = 10 and $\lambda = 0.001$. So, we can find the w for which $P(\overline{X} > w) = 0.05$ by using R's **qgamma()**:

```
> qgamma(0.95,10,0.001)
[1] 15705.22
```

So, we reject H_0 if our sample mean is larger than 1570.5.

Now suppose it turns out that $\overline{X} = 1624.2$. Under H_0 there was only a 0.05 chance that $\overline{X}$ would exceed 1570.5, so we would reject H_0 with $\alpha = 0.05$. But what if we had set w to 1624.2? We didn't do so, of course, but what if? The computation

```
> 1 - pgamma(16242,10,0.001)
[1] 0.03840629
```

shows that we would have rejected H_0 even if we had originally set α to the more stringent criterion of 0.038 instead of 0.05. So we report that the p-value was 0.038.

The idea of a p-value is to indicate in our report "how strongly" we rejected H_0. Arguably there is a bit of game-playing in p-values, as there is with significance testing in general. This will be pursued in Section 17.11.

17.7.3   Example: Test Based on Range Data

Suppose lifetimes of some electronic component formerly had an exponential distribution with mean 100.0. However, it's claimed that now the mean has increased. (Suppose we are somehow sure it has not decreased.) Someone has tested 50 of these new components, and has recorded their lifetimes, $X_1, ..., X_{50}$. Unfortunately, they only reported to us the range of the data, $R = \max_i X_i - \min_i X_i$, not the individual X_i. We will need to do a significance test with this limited data, at the 0.05 level.

Recall that the variance of an exponential random variable is the square of its mean. Intutively, then, the larger this population mean of X, the larger the mean of the range R. In other words, the form of the test should be to reject H_0 if R is greater than some cutoff value c. So, we need to find the value of c to make α equal to 0.05.

Unfortunately, we can't do this analytically, i.e. mathematically, as the distribution of R is far too complex. This we'll have to resort to simulation.[5] Here is code to do that:

```
1  # code to determine the cutoff point for significance
2  # at 0.05 level
3
4  nreps <- 200000
5  n <- 50
6
7  rvec <- vector(length=nreps)
8  for (i in 1:nreps) {
9      x <- rexp(n,0.01)
```

---

[5]I am still referring to the following as an exact test, as we are not using any statistical approximation, such as the Central Limit Theorm.

```
10      rng <- range(x)
11      rvec[i] <- rng[2] - rng[1]
12  }
13
14  rvec <- sort(rvec)
15  cutoff <- rvec[ceiling(0.95*nreps)]
16  cat("reject␣H0␣if␣R␣>",rvec[cutoff],"\n")
```

Here we general **nreps** samples of size 50 from an exponential distribution having mean 100. Note that since we are setting α, a probability defined in the setting in which H_0 is true, we assume the mean is 100. For each of the **nreps** samples we find the value of R, recording it in **rvec**. We then take the 95th percentile of those values, which is the c for which $P(R > c) = 0.05$.[6]

The value of c output by the code was 220.4991. A second run yielded, 220.9304, and a third 220.7099. The fact that these values varied little among themselves indicates that our value of **nreps**, 200000, was sufficiently large.

17.7.4   Exact Tests under a Normal Distribution Assumption

If you are willing to assume that you are sampling from a normally-distributed population, then the Student-t test is nominally exact. The R function **t.test()** performs this operation, with the argument **alternative** set to be either **"less"** or **"greater"**.

17.8   Don't Speak of "the Probability That H_0 Is True"

It is very important to understand that throughout this chapter, we cannot speak of "the probability that H_0 is true," because we have no probablistic structure on H_0.

Consider the fair-coin problem at the beginning of this chapter. Suppose we hope to make a statement like, say, "Given that we got 62 heads out of 100 tosses, we find that the probability that this is a fair coin is 0.04." What kind of derivation would need to go into this? It would go along the following lines:

---

[6]Of course, this is approximate. The greater the value of **nreps**, the better the approximation.

$$P(H_0 \text{ is true } | \text{ our data}) = \frac{P(H_0 \text{ is true and our data})}{P(\text{our data})} \tag{17.14}$$

$$= \frac{P(H_0 \text{ is true and our data})}{P(H_0 \text{ is true and our data}) + P(H_0 \text{ is false and our data})}$$

$$= \frac{P(H_0 \text{ true}) \, P(\text{our data} | H_0 \text{ true})}{P(H_0 \text{ true}) \, P(\text{our data } | H_0 \text{ true}) + P(H_0 \text{ false}) \, P(\text{our data} | H_0 \text{ false})}$$

Through our modeling process, e.g. the discussion surrounding (17.4), we can calculate $P(\text{our data} | H_0 \text{ is true})$. (The false case would be more complicated, since there are many different kinds of false cases here, for different values of p, but could be handled similarly.) But what we don't have is $P(H_0 \text{ is true})$.

We could certainly try to model that latter quantity, say by taking a sample of all possible pennies (if our coin is a penny), doing very extensive testing of them;[7] the proportion found to be fair would then be $P(H_0 \text{ is true})$. But lacking that, we have no probablistic structure for $P(H_0 \text{ is true})$, and thus cannot use language like "the probability that H_0 is true,"

17.9   R Computation

The R function **t.test()**, discussed in Section 16.8, does both confidence intervals and tests, including p-values in the latter case.

17.10   The Power of a Test

In addition to the significance level of a test, we may also be interested in its **power** (or its many power values, as will be seen).

17.10.1   Example: Coin Fairness

For example, consider our first example in this chapter, in which we were testing a coin for fairness (Section 17.1). Our rule for a test at a 0.05 significance level turned out to be that we reject H_0 if we get fewer than 40 or more than 60 heads out of our 100 tosses. We might ask the question, say:

> Suppose the true heads probability is 0.65. We don't know, of course, but what if that were the case. That's a pretty substantial departure from H_0, so hopefully we would reject. Well, what is the probability that we would indeed reject?

---

[7]Say, 100,000 tosses per coin.

We could calculate this. Let N denote the number of heads. Then the desired probability is $P(N < 40 \text{ or } N > 60) = P(N < 40) + P(N > 60)$. Let's find the latter.[8]

Once again, since N has a binomial distribution, it is approximately normal, in this case with mean $np = 100 \times 0.65 = 65$ and variance $np(1 - p) = 100 \times 0.65 \times 0.35 = 22.75$. Then $P(N > 60)$ is about

```
1 - pnorm(60,65,sqrt(22.75))
```

or about 0.85. So we would be quite likely to decide this is an unfair coin if (unknown to us) the true probability of heads is 0.65.

We say that the power of this test *at* p = 0.65 is 0.85. There is a different power for each p.

17.10.2   Example: Improved Light Bulbs

Let's find the power of the test in Section 17.7.2, at $\mu = 1250$. Recall that we reject H_0 if $\overline{X} > 1570.522$. Thus our power is

```
1 - pgamma(15705.22,10,1/1250)
```

This turns out to be about 0.197. So, if (remember, this is just a "what if?") the true new mean were 1250, we'd only have about a 20% chance of discovering that the new bulbs are improved.

17.11   What's Wrong with Significance Testing—and What to Do Instead

The first principle is that you must not fool yourself—and you are the easiest person to fool. So you have to be very careful about that. After you've not fooled yourself, it's easy not to fool other scientists.—Richard Feynman, Nobel laureate in physics

"Sir Ronald [Fisher] has befuddled us, mesmerized us, and led us down the primrose path"—Paul Meehl, professor of psychology and the philosophy of science

Significance testing is a time-honored approach, used by tens of thousands of people every day. But it is "wrong." I use the quotation marks here because, although significance testing is mathematically correct, it is at best noninformative and at worst seriously misleading.

---

[8]The former would be found similarly, but would come out quite small.

17.11.1   History of Significance Testing, and Where We Are Today

We'll see why significance testing has serious problems shortly, but first a bit of history.

When the concept of significance testing, especially the 5% value for α, was developed in the 1920s by Sir Ronald Fisher, many prominent statisticians opposed the idea—for good reason, as we'll see below. But Fisher was so influential that he prevailed, and thus significance testing became the core operation of statistics.

So, significance testing became entrenched in the field, in spite of being widely recognized as faulty, to this day. Most modern statisticians understand this, even if many continue to engage in the practice.[9] Here are a few places you can read criticism of testing:

- There is an entire book on the subject, *The Cult of Statistical Significance*, by S. Ziliak and D. McCloskey. Interestingly, on page 2, they note the prominent people who have criticized testing. Their list is a virtual "who's who" of statistics, as well as physics Nobel laureate Richard Feynman and economics Nobelists Kenneth Arrow and Milton Friedman.

- See `http://www.indiana.edu/~stigtsts/quotsagn.html` for a nice collection of quotes from famous statisticians on this point.

- There is an entire chapter devoted to this issue in one of the best-selling elementary statistics textbooks in the nation.[10]

- The Federal Judicial Center, which is the educational and research arm of the federal court system, commissioned two prominent academics, one a statistics professor and the other a law professor, to write a guide to statistics for judges: *Reference Guide on Statistics*. David H. Kaye. David A. Freedman, at

  `http://www.fjc.gov/public/pdf.nsf/lookup/sciman02.pdf/$file/sciman02.pdf`

  There is quite a bit here on the problems of significance testing, and especially p.129.

17.11.2   The Basic Fallacy

To begin with, **it's absurd to test H_0 in the first place**, because we know *a priori* that H_0 is false.

---

[9]Many are forced to do so, e.g. to comply with government standards in pharmaceutical testing. My own approach in such situations is to quote the test results but then point out the problems, and present confidence intervals as well.

[10]*Statistics*, third edition, by David Freedman, Robert Pisani, Roger Purves, pub. by W.W. Norton, 1997.

Consider the coin example, for instance. No coin is absolutely perfectly balanced, and yet that is the question that significance testing is asking:

$$H_0 : p = 0.500000000000000000000000000... \tag{17.15}$$

We know before even collecting any data that the hypothesis we are testing is false, and thus it's nonsense to test it.

But much worse is this word "significant." Say our coin actually has p = 0.502. From anyone's point of view, that's a fair coin! But look what happens in (17.4) as the sample size n grows. If we have a large enough sample, eventually the denominator in (17.4) will be small enough, and $\hat{p}$ will be close enough to 0.502, that Z will be larger than 1.96 and we will declare that p is "significantly" different from 0.5. But it isn't! Yes, 0.502 is different from 0.5, but NOT in any significant sense in terms of our deciding whether to use this coin in the Super Bowl.

The same is true for government testing of new pharmaceuticals. We might be comparing a new drug to an old drug. Suppose the new drug works only, say, 0.4% (i.e. 0.004) better than the old one. Do we want to say that the new one is "signficantly" better? This wouldn't be right, especially if the new drug has much worse side effects and costs a lot more (a given, for a new drug).

Note that in our analysis above, in which we considered what would happen in (17.4) as the sample size increases, we found that eventually *everything* becomes "signficiant"—even if there is no practical difference. This is especially a problem in computer science applications of statistics, because they often use very large data sets. A data mining application, for instance, may consist of hundreds of thousands of retail purchases. The same is true for data on visits to a Web site, network traffic data and so on. In all of these, the standard use of significance testing can result in our pouncing on very small differences that are quite insignificant to us, yet will be declared "significant" by the test.

Conversely, if our sample is too small, we can miss a difference that actually *is* significant—i.e. important to us—and we would declare that p is NOT significantly different from 0.5. In the example of the new drug, this would mean that it would be declared as "not significantly better" than the old drug, even if the new one is much better but our sample size wasn't large enough to show it.

In summary, the basic problems with significance testing are

- H_0 is improperly specified. What we are really interested in here is whether p is *near* 0.5, not whether it is *exactly* 0.5 (which we know is not the case anyway).

- Use of the word *significant* is grossly improper (or, if you wish, grossly misinterpreted).

Significance testing forms the very core usage of statistics, yet you can now see that it is, as I said

above, "at best noninformative and at worst seriously misleading." This is widely recognized by thinking statisticians and prominent scientists, as noted above. But the practice of significance testing is too deeply entrenched for things to have any prospect of changing.

17.11.3   You Be the Judge!

This book has been written from the point of view that every educated person should understand statistics. It impacts many vital aspects of our daily lives, and many people with technical degrees find a need for it at some point in their careers.

In other words, statistics is something to be *used*, not just learned for a course. You should think about it critically, especially this material here on the problems of significance testing. You yourself should decide whether the latter's widespread usage is justified.

17.11.4   What to Do Instead

Note carefully that I am not saying that we should not make a decision. We *do* have to decide, e.g. decide whether a new hypertension drug is safe or in this case decide whether this coin is "fair" enough for practical purposes, say for determining which team gets the kickoff in the Super Bowl. But it should be an informed decision, and even testing the modified H_0 above would be much less informative than a confidence interval.

In fact, the real problem with significance tests is that they **take the decision out of our hands**. They make our decision mechanically for us, not allowing us to interject issues of importance to us, such possible side effects in the drug case.

So, what can we do instead?

In the coin example, we could set limits of fairness, say require that p be no more than 0.01 from 0.5 in order to consider it fair. We could then test the hypothesis

$$H_0 : 0.49 \leq p \leq 0.51 \tag{17.16}$$

Such an approach is almost never used in practice, as it is somewhat difficult to use and explain. But even more importantly, what if the true value of p were, say, 0.51001? Would we still really want to reject the coin in such a scenario?

Forming a confidence interval is the far superior approach. The width of the interval shows us whether n is large enough for $\widehat{p}$ to be reasonably accurate, and the location of the interval tells us whether the coin is fair enough for our purposes.

Note that in making such a decision, we do NOT simply check whether 0.5 is in the interval. That would make the confidence interval reduce to a significance test, which is what we are trying to avoid. If for example the interval is (0.502,0.505), we would probably be quite satisfied that the coin is fair enough for our purposes, even though 0.5 is not in the interval.

On the other hand, say the interval comparing the new drug to the old one is quite wide and more or less equal positive and negative territory. Then the interval is telling us that the sample size just isn't large enough to say much at all.

Significance testing is also used for model building, such as for predictor variable selection in regression analysis (a method to be covered in Chapter 22). The problem is even worse there, because there is no reason to use $\alpha = 0.05$ as the cutoff point for selecting a variable. In fact, even if one uses significance testing for this purpose—again, very questionable—some studies have found that the best values of α for this kind of application are in the range 0.25 to 0.40, far outside the range people use in testing.

In model building, we still can and should use confidence intervals. However, it does take more work to do so. We will return to this point in our unit on modeling, Chapter 21.

17.11.5 Decide on the Basis of "the Preponderance of Evidence"

I was in search of a one-armed economist, so that the guy could never make a statement and then say: "on the other hand"—President Harry S Truman

If all economists were laid end to end, they would not reach a conclusion—Irish writer George Bernard Shaw

In the movies, you see stories of murder trials in which the accused must be "proven guilty beyond the shadow of a doubt." But in most noncriminal trials, the standard of proof is considerably lighter, **preponderance of evidence**. This is the standard you must use when making decisions based on statistical data. Such data cannot "prove" anything in a mathematical sense. Instead, it should be taken merely as evidence. The width of the confidence interval tells us the likely accuracy of that evidence. We must then weigh that evidence against other information we have about the subject being studied, and then ultimately make a decision on the basis of the preponderance of all the evidence.

Yes, juries must make a decision. But they don't base their verdict on some formula. Similarly, you the data analyst should not base your decision on the blind application of a method that is usually of little relevance to the problem at hand—significance testing.

17.11.6   Example: the Forest Cover Data

In Section 16.6.4, we found that an approximate 95% confidence interval for $\mu_1 - \mu_2$ was

$$223.8 - 226.3 \pm 2.3 = (-4.8, -0.3) \tag{17.17}$$

Clearly, the difference in HS12 between cover types 1 and 2 is tiny when compared to the general size of HS12, in the 200s. Thus HS12 is not going to help us guess which cover type exists at a given location. Yet with the same data, we would reject the hypothesis

$$H_0 : \mu_1 = \mu_2 \tag{17.18}$$

and say that the two means are "significantly" different, which sounds like there is an important difference—which there is not.

17.11.7   Example: Assessing Your Candidate's Chances for Election

Imagine an election between Ms. Smith and Mr. Jones, with you serving as campaign manager for Smith. You've just gotten the results of a very small voter poll, and the confidence interval for p, the fraction of voters who say they'll vote for Smith, is (0.45,0.85). Most of the points in this interval are greater than 0.5, so you would be highly encouraged! You are certainly not sure of the final election result, as a small part of the interval is below 0.5, and anyway voters might change their minds between now and the election. But the results would be highly encouraging.

Yet a significance test would say "There is no significant difference between the two candidates. It's a dead heat." Clearly that is not telling the whole story. The point, once again, is that **the confidence interval is giving you much more information than is the significance test.**

Chapter 18

General Statistical Estimation and Inference

Earlier, we often referred to certain estimators as being "natural." For example, if we are estimating a population mean, an obvious choice of estimator would be the sample mean. But in many applications, it is less clear what a "natural" estimate for a population quantity of interest would be. We will present general methods for estimation in this section.

We will also discuss advanced methods of inference.

18.1 General Methods of Parametric Estimation

Let's begin with a simple motivating example.

18.1.1 Example: Guessing the Number of Raffle Tickets Sold

You've just bought a raffle ticket, and find that you have ticket number 68. You check with a couple of friends, and find that their numbers are 46 and 79. Let c be the total number of tickets. How should we estimate c, using our data 68, 46 and 79?

Let $X_1, X_2, ..., X_n$ denote the ticket numbers we are aware of. In the above case, $n = 3$, with $X_1 = 68$ etc. We will treat the X_i as a random sample from 1,2,...,c. This assumption should be carefully considered. Recall that this means that $X_1, X_2, ..., X_n$ are i.i.d. with a (discrete) uniform distribution on 1,2....,c.

If we know, say, that the three friends bought their tickets right after they became available.

This might mean the X_i are nearer to 1 than to c, in which case the assumption of the uniform distribution may not be very good. Or, suppose we know that the three friends were standing close to each other in line. Then the independence assumption might be dubious.

And even if not, and even if the X_i are considered a sample with the uniformity assumption satisfied, it would technically be what is called a *simple* random sample (Section 15.1.1), since the "sampling" is done without replacement; no two people get the same ticket.

So, in practice one would need to consider how good our assumption is that we have a random sample. If $n << c$, the various probabilities are virtually identical for sampling with and without replacement, so that may not be an issue, so we would just worry about the uniformity.

But let's suppose that has been resolved, and then look at how we can estimate c.

18.1.2 Method of Moments

One approach, a very intuitive one, is the Method of Momentsi (MM). It is less commonly used than the other method we'll cover, Maximum Likelihood, but is much easier to explain. Also, in recent years, MM has become more popular in certain fields, such as random graph models (Section 2.13).

18.1.2.1 Example: Lottery Model

Let X be distributed the same as the X_i, i.e. it is uniformly distributed on 1,2,...,c.

Note first that

$$E(X) = \frac{c+1}{2} \tag{18.1}$$

Let's solve for c:

$$c = 2EX - 1 \tag{18.2}$$

We know that we can use

$$\overline{X} = \frac{1}{n} \sum_{i=1}^{n} X_i \tag{18.3}$$

to estimate EX, so by (18.2), $2\overline{X} - 1$ is an intuitive estimate of c. Thus we take our estimator for c to be

$$\widehat{c} = 2\overline{X} - 1 \tag{18.4}$$

This estimator is called the Method of Moments estimator of c.

18.1.2.2   General Method

Say our sample is $X_1, ..., X_n; a, and$ we have k parameters to estimate, $\theta_1, ..., \theta_k$. Our goal is to come up with estimators $\widehat{\theta}_1, ..., \widehat{\theta}_k$.

Here is the procedure:

- For each $i = 1, ..., k$, write $E(X^k)$ as a function of the θ_j,

$$E(X^i) = g_i(\theta_1, ..., \theta_k) \tag{18.5}$$

- On the left side of (18.5), replace $E(X^i)$ by the corresponding sample moment

$$\frac{1}{n} \sum_{r=1}^{n} X_r^i \tag{18.6}$$

- On the right side of (18.5), replace each θ_s by $\widehat{\theta}_s$.
- Solve for the $\widehat{\theta}_s$

In the raffle example, we had k = 1, $\theta_1 = c$, $g_1(c) = (c+1)/2$ and so on.

18.1.3   Method of Maximum Likelihood

Another method, much more commonly used, is called the **Method of Maximum Likelihood**.

18.1.3.1   Example: Raffle Model

In our example above, it means asking the question, "What value of c would have made our data—68, 46, 79—most likely to happen?" Well, let's find what is called the **likelihood**, i.e. the

probability of our particular data values occurring:

$$L = P(X_1 = 68, X_2 = 46, X_3 = 79) = \begin{cases} (\frac{1}{c})^3, & \text{if } c \geq 79 \\ 0, & \text{otherwise} \end{cases} \tag{18.7}$$

Now keep in mind that c is a fixed, though unknown constant. It is not a random variable. What we are doing here is just asking "What if" questions, e.g. "If c were 85, how likely would our data be? What about c = 91?"

Well then, what value of c maximizes (18.7)? Clearly, it is c = 79. Any smaller value of c gives us a likelihood of 0. And for c larger than 79, the larger c is, the smaller (18.7) is. So, our maximum likelihood estimator (MLE) is 79. In general, if our sample size in this problem were n, our MLE for c would be

$$\check{c} = \max_i X_i \tag{18.8}$$

Note that in our earlier discussion of whether our data can be treated as a random sample, we raised the issue of the independence assumption. Note that this assumption in fact *was* used for $\check{c}$, while it was not for $\hat{c}$.

18.1.3.2 General Procedure

Say again our random sample is $X_1, ..., X_n$ and we have k parameters to estimate, $\theta_1, ..., \theta_k$. Our goal is to come up with estimators $\hat{\theta}_1, ..., \hat{\theta}_k$.

Denote the pmf or density of our X_i as $f(t; \theta_1, ..., \theta_k)$. Then our estimators $\hat{\theta}_i$ are the values that maximize the likelihood

$$\Pi_{i=1}^n f(X_i; \theta_1, ..., \theta_k) \tag{18.9}$$

with respect to the θ_j.

For "smooth" problems, i.e. ones in which the likelihood is a differentiable function of the parameters, maximization involves taking the derivatives with respect to the θ_j, setting them to 0, then solving for the θ_j; the solutions are the $\hat{\theta}_j$.

Since it is easier to take derivatives of sums than products, typically one maximizes the log likelihood

$$\sum_{i=1}^n \ln[f(X_i; \theta_1, ..., \theta_k)] \tag{18.10}$$

18.1.4   Example: Estimation of the Parameters of a Gamma Distribution

As another example, suppose we have a random sample $X_1, ..., X_n$ from a gamma distribution.

$$f_X(t) = \frac{1}{\Gamma(c)} \lambda^c t^{c-1} e^{-\lambda t}, \ t > 0 \tag{18.11}$$

for some unknown c and λ. How do we estimate c and λ from the X_i?

18.1.4.1   Method of Moments

Let's try the Method of Moments, as follows. We have two population parameters to estimate, c and λ, so we need to involve two moments of X. That could be EX and $E(X^2)$, but here it would more conveniently be EX and Var(X). We know from our previous unit on continuous random variables, Chapter 7, that

$$EX = \frac{c}{\lambda} \tag{18.12}$$

$$Var(X) = \frac{c}{\lambda^2} \tag{18.13}$$

In our earlier notation, this would be r = 2, $\theta_1 = c$, $\theta_2 = \lambda$ and $g_1(c, \lambda) = c/\lambda$ and $g_2(c, \lambda) = c/\lambda^2$.

Switching to sample analogs and estimates, we have

$$\frac{\widehat{c}}{\widehat{\lambda}} = \overline{X} \tag{18.14}$$

$$\frac{\widehat{c}}{\widehat{\lambda}^2} = s^2 \tag{18.15}$$

Dividing the two quantities yields

$$\widehat{\lambda} = \frac{\overline{X}}{s^2} \tag{18.16}$$

which then gives

$$\widehat{c} = \frac{\overline{X}^2}{s^2} \tag{18.17}$$

18.1.4.2   MLEs

What about the MLEs of c and λ? Remember, the X_i are continuous random variables, so the likelihood function, i.e. the analog of (18.7), is the product of the density values:

$$
\begin{aligned}
L &= \Pi_{i=1}^{n} \left[\frac{1}{\Gamma(c)} \lambda^c X_i^{c-1} e^{-\lambda X_i} \right] & (18.18) \\
&= [\lambda^c / \Gamma(c)]^n \, (\Pi_{i=1}^{n} X_i)^{c-1} \, e^{-\lambda \sum_{i=1}^{n} X_i} & (18.19)
\end{aligned}
$$

In general, it is usually easier to maximize the log likelihood (and maximizing this is the same as maximizing the original likelihood):

$$
l = (c-1) \sum_{i=1}^{n} \ln(X_i) - \lambda \sum_{i=1}^{n} X_i + nc \ln(\lambda) - n \ln(\Gamma(c)) \qquad (18.20)
$$

One then takes the partial derivatives of (18.20) with respect to c and λ, and sets the derivatives to zero. The solution values, $\check{c}$ and $\check{\lambda}$, are then the MLEs of c and λ. Unfortunately, in this case, these equations do not have closed-form solutions. So the equations must be solved numerically. (In fact, numerical methods are needed even more in this case, because finding the derivative of $\Gamma(c)$ is not easy.)

18.1.5   R's mle() Function

R provides a function, **mle()**, for finding MLEs in mathematically intractable situations such as the one in the last section.

Note: The function is in the **stats4** library, so run

```
> library (stats4)
```

first.

Here's an example in that context. We'll simulate some data from a gamma distribution with given parameter values, then pretend we don't know those, and find the MLEs from the data:

```
> n <- 1000
> x <- rgamma(n,shape=2,rate=1)  # Erlang, r = 2, lambda is 1

# function to compute negative log likelihood
ll <- function(c,lambda) {
```

```
   loglik <- (c-1) * sum(log(x)) - sum(x)*lambda + n*c*log(lambda) -
      n*log(gamma(c))
   -loglik
}

summary(mle(minuslogl=ll,start=list(c=1.5,lambda=2)))
Maximum likelihood estimation

Call:
mle(minuslogl = ll, start = list(c = 1, lambda = 1))

Coefficients:
      Estimate Std. Error
c       2.147605 0.08958823
lambda 1.095344 0.05144393

-2 log L: 3072.181
```

How did this work? The main task we have is to write a function that calculates the negative log likelihood, with that function's arguments will be the parameters to be estimated. We defined our function and named it **ll()**.[1]

Fortunately for us, **mle()** calculates the derivatives numerically too, so we didn't need to specify them in the log likelihood function. (Needless to say, this function thus cannot be used in a problem in which derivatives cannot be used, such as the lottery example above.)

We also need to supply **mle()** with initial guesses for the parameters. That's done in the **start** argument. I more or less arbitrarily chose these values. You may have to experiment, though, as some sets of initial values may not result in convergence.

The standard errors of the estimated parameters are also printed out, enabling the formation of confidence intervals and significance tests. (See Section 16.5 for the case of confidence intervals.) In fact, you can get the estimated covariance matrix for the vector of estimated parameters. In our case here:

```
> mleout <- mle(minuslogl=ll,start=list(c=1.5,lambda=2))
> solve(mleout@details$hessian)
                c        lambda
c       0.008026052 0.004093526
lambda 0.004093526 0.002646478
```

By the way, there were also some warning messages, due to the fact that during the iterative maximization process, some iterations generated guesses for $\check{\lambda}$ were 0 or near it, causing problems with **log()**.

Maximizing the likelihood meant minimizing the negative log likelihood. Why did the authors of R

---

[1]Note that in R, **log()** calculates the natural logarithm by default.

write it this way, and in fact report -2 times the log likelihood? The answer is that this is related to a significance testing procedure, the *likelihood ratio test*, a matter beyond the scope of this book.

Note that if you are using **mle()** with a parametric family for which R supplies a density function, it's quite convenient to simply use that in our likelihood function code:

```
> ll <- function(c,lambda) -sum(dgamma(x,shape=c,rate=lambda,log=TRUE))
> summary(mle(minuslogl=ll,start=list(c=1.5,lambda=2)))
Maximum likelihood estimation

Call:
mle(minuslogl = ll, start = list(c = 1.5, lambda = 2))

Coefficients:
       Estimate Std. Error
c      2.147605 0.08958823
lambda 1.095344 0.05144393

-2 log L: 3072.181
```

R made things so convenient that we didn't even need to take the logs ourselves.

18.1.6   R's gmm() Function

A function to do Method of Moments estimation is also available, in R's **gmm** package on CRAN. The package actually does more, something called Generalized Method of Moments, but we'll see how to use it in elementary form here.

The work is done by the eponymous function **gmm()**. In our usage here, the call form is

```
gmm(data,momentftn{,start)}
```

where **data** is our data in matrix or vector form, **momentftn** specifies the moments, and **start** is our initial guess for the iterative solution process.

The function **momentftn()** specifies how to compute the differences between the estimated values of the population moments (in terms of the parameter estimates) and the corresponding sample powers. During the computation, **gmm()** will set the means of these differences to 0, iterating until convergence.

18.1.6.1 Example: Bodyfat Data

Our data set will be **bodyfat**, which is included in the **mfp** package, with measurements on 252 men. The first column of this data frame, **brozek**, is the percentage of body fat, which when converted to a proportion is in (0,1). That makes it a candidate for fitting a beta distribution (Section 7.6.5).

Here is our **momentftn()**:

```
g <- function(th,x) {
   t1 <- th[1]
   t2 <- th[2]
   t12 <- t1 + t2
   meanb <- t1 / t12
   m1 <- meanb - x
   m2 <- t1*t2 / (t12^2 * (t12+1)) - (x - meanb)^2
   f <- cbind(m1,m2)
   return(f)
}
```

Here the argument **th** (theta) will be the MM estimates (at any given iteration) of the population parameters, in this case of α and β.

Now look at the line

```
m1 <- meanb - x
```

The value **meanb** is our estimate in the currently iteration of the mean of the beta distribution, while **x** is our data. The **gmm()** function, in calling our **g()**, will calculate the average of **m1** over all our data, and then in setting that average to 0, we are equating our parametric estimate of the population mean to our sample mean — exactly what MM is supposed to do.

Let's try it:

```
> library(mfp)
> data(bodyfat)
> gmm(g,x,c(alpha=0.1,beta=0.1))
Method
 twoStep

Objective function value: 2.559645e-10

 alpha beta
 4.6714 19.9969
```

Histogram of bodyfat$brozek/100
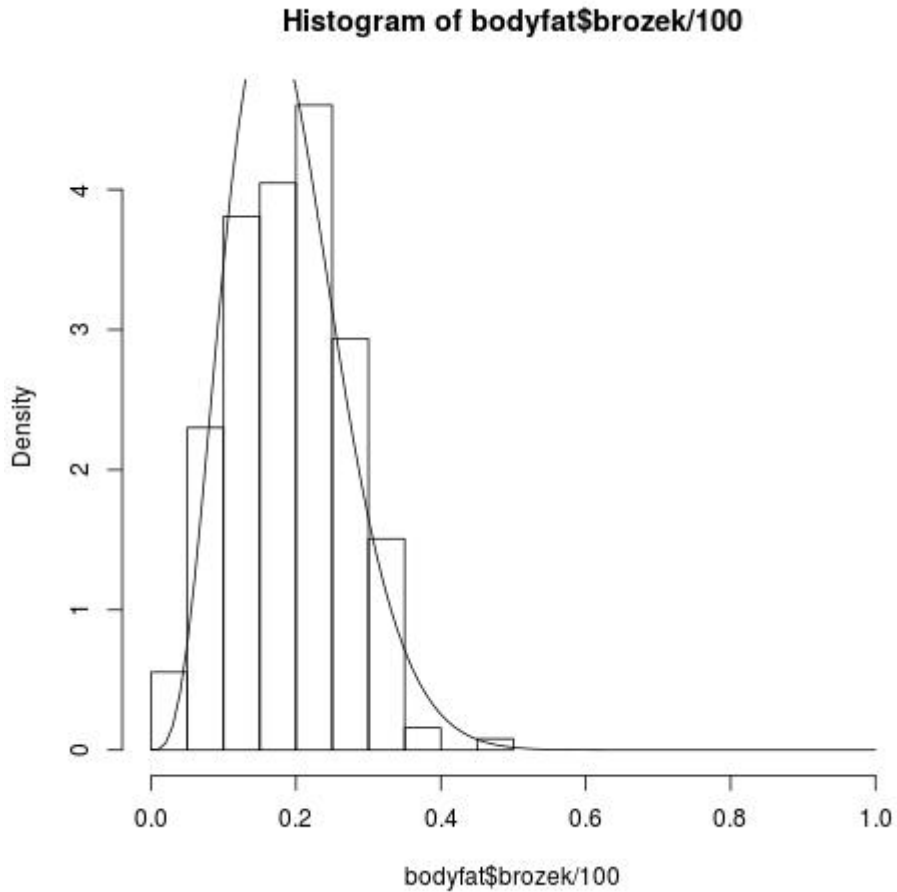
Figure 18.1: Fitting a Beta Density

```
Convergence code = 0
> hist(bodyfat$brozek/100,xlim=c(0,1),
    probability=TRUE)
> curve(dbeta(x,4.67,20.00),add=TRUE)
```

The result is seen in Figure 18.1. At least visually, the fit seems pretty good.

18.1.7 More Examples

Suppose $f_W(t) = ct^{c-1}$ for t in (0,1), with the density being 0 elsewhere, for some unknown $c > 0$. We have a random sample $W_1, ..., W_n$ from this density.

Let's find the Method of Moments estimator.

$$EW = \int_0^1 tct^{c-1} \, dt = \frac{c}{c+1} \tag{18.21}$$

So, set

$$\overline{W} = \frac{\widehat{c}}{\widehat{c}+1} \tag{18.22}$$

yielding

$$\widehat{c} = \frac{\overline{W}}{1 - \overline{W}} \tag{18.23}$$

What about the MLE?

$$L = \Pi_{i=1}^n cW_i^{c-1} \tag{18.24}$$

so

$$l = n \ln c + (c-1) \sum_{i=1}^n \ln W_i \tag{18.25}$$

Then set

$$0 = \frac{n}{\widehat{c}} + \sum_{i=1}^n \ln W_i \tag{18.26}$$

and thus

$$\widehat{c} = -\frac{1}{\frac{1}{n} \sum_{i=1}^n \ln W_i} \tag{18.27}$$

As in Section 18.1.3, not every MLE can be determined by taking derivatives. Consider a continuous analog of the example in that section, with $f_W(t) = \frac{1}{c}$ on (0,c), 0 elsewhere, for some $c > 0$.

The likelihood is

$$\left(\frac{1}{c}\right)^n \tag{18.28}$$

as long as

$$c \geq \max_i W_i \tag{18.29}$$

and is 0 otherwise. So,

$$\widehat{c} = \max_i W_i \tag{18.30}$$

as before.

Now consider a different problem. Suppose the random variable X is equal to 1, 2 and 3, with probabilities c, c and 1-2c. The value c is thus a population parameter. We have a random sample $X_1, ..., X_n$ from this population. Let's find the Method of Moments Estimator of c, and its bias.

First,

$$EX = c \cdot 1 + c \cdot 2 + (1 - 2c) \cdot 3 = 3 - 3c \tag{18.31}$$

Thus

$$c = (3 - EX)/3 \tag{18.32}$$

and so set

$$\hat{c} = (3 - \overline{X})/3 \tag{18.33}$$

Next,

$$E\hat{c} = E[(3 - \overline{X})/3] \tag{18.34}$$

$$= \frac{1}{3} \cdot (3 - E\overline{X}) \tag{18.35}$$

$$= \frac{1}{3}[3 - EX] \tag{18.36}$$

$$= \frac{1}{3}[3 - (3 - 3c)] \tag{18.37}$$

$$= c \tag{18.38}$$

On average, not too high and not too low; we say the *bias* is 0.

18.1.8 Asymptotic Properties

Most of the inference methods in this book use large-sample, i.e. *asymptotic* techniques, based on the Central Limit Theorem. Let's look at the asymptotic properties of Method of Moments and Maximum Likelihood estimators.

18.1.8.1 Consistency

We say that an estimator $\widehat{\nu}$ of some parameter ν is **consistent** if, with probability 1,

$$\lim_{n\to\infty} \widehat{\nu} = \nu \tag{18.39}$$

where n is the sample size. In other words, as the sample size grows, the estimator eventually converges to the true population value. Clearly, in most situations consistency is a minimal property we want our estimators to have.

Of course here $\overline{X}$ is a consistent estimator of EX, and (18.6) is a consistent estimator of $E(X^i)$. So, as long as the $g_i()$ in (18.6) are continous etc., we see that the Method of Moments gives us consistent estimators.

Showing consistency for MLEs is a little more involved. To start, recall (18.10). Its derivative with respect to θ_j is

$$\sum_{i=1}^{n} \frac{f'(X_i; \theta_1, ..., \theta_k)}{f(X_i; \theta_1, ..., \theta_k)} \tag{18.40}$$

For each value of $\theta = (\theta_1, ..., \theta_k)'$, this will converge to

$$E\left(\frac{f'(X;\theta)}{f(X;\theta)}\right) \tag{18.41}$$

Let η denote the true population value of θ, with $\widehat{\theta}$ being the MLE. The latter is the root of (18.40),

$$0 = \sum_{i=1}^{n} \frac{f'(X_i;\widehat{\theta})}{f(X_i;\widehat{\theta})} \tag{18.42}$$

Moreover, in for instance the continuous case (we'll be implicitly assuming some mathematical conditions here, e.g. that roots are uniqe),

$$\begin{aligned}
E\left(\frac{f'(X;\eta)}{f(X;\eta)}\right) &= \int_{-\infty}^{\infty} f'(t;\eta)\, dt \tag{18.43}\\
&= \frac{d}{dt}\int_{-\infty}^{\infty} f(t;\eta)\, dt \tag{18.44}\\
&= \frac{d}{dt}1 \tag{18.45}\\
&= 0 \tag{18.46}
\end{aligned}$$

Due to the uniqeness of the roots, we see that $\widehat{theta}$, the root of (18.40) must converge to η, the root of (18.41).

18.1.8.2   Approximate Confidence Intervals

Usually we are not satisfied with simply forming estimates (called **point estimates**). We also want some indication of how accurate these estimates are, in the form of confidence intervals (**interval estimates**).

In many special cases, finding confidence intervals can be done easily on an *ad hoc* basis. Look, for instance, at the Method of Moments Estimator in Section 18.1.2. Our estimator (18.4) is a linear function of $\overline{X}$, so we easily obtain a confidence interval for c from one for EX.

Another example is (18.27). Taking the limit as $n \to \infty$ the equation shows us (and we could verify) that

$$c = \frac{1}{E[\ln W]} \tag{18.47}$$

Defining $X_i = \ln W_i$ and $\overline{X} = (X_1 + ... + X_n)/$, we can obtain a confidence interval for EX in the usual way. We then see from (18.47) that we can form a confidence interval for c by simply taking the reciprocal of each endpoint of the interval, and swapping the left and right endpoints.

What about in general? For the Method of Moments case, our estimators are functions of the sample moments, and since the latter are formed from sums and thus are asymptotically normal, the delta method (Section **??**) can be used to show that our estimators are asymptotically normal and to obtain asymptotic variances for them.

There is a well-developed asymptotic theory for MLEs, which under certain conditions shows asymptotic normality with a certain asymptotic variance, thus enabling confidence intervals. The theory also establishes that MLEs are in a certain sense optimal among all estimators. We will not pursue this here, but will note that **mle()** does give standard errors for the estimates, thus enabling the formation of confidence intervals.

18.2 Bias and Variance

The notions of **bias** and **variance** play central roles in the evaluation of goodness of estimators.

18.2.1 Bias

This bowl of porridge is not too big, not too small, but just right—from the children's story, *Goldilocks* (paraphrased)

Definition 31 *Suppose $\hat{\theta}$ is an estimator of θ. Then the* **bias** *of $\hat{\theta}$ is*

$$bias = E(\hat{\theta}) - \theta \tag{18.48}$$

If the bias is 0, we say that the estimator is **unbiased***.*

So, if $\hat{\theta}$ is an unbiased estimator of θ, then its average value over all possible samples is not too high, not too low, but just right.

At first that would seem to be a "must have" property for any estimator. But it's very important to note that, in spite of the pejorative-sounding name, bias is not an inherently bad property for an estimator to have. Indeed, most good estimators are at least slightly biased.[2] We'll explore this

---

[2]Typically, though, the amount of bias will go to 0 as the sample size goes to infinity. That is the case for most consistent estimators (Sec. 18.1.2, though technically it is not implied by consistency; if a sequence of random variables converges to a limit, their expected values do not necessarily converge to that limit, or converge at all).

in the next section.

18.2.2   Why Divide by n-1 in s^2?

It should be noted that it is customary in (15.23) to divide by n-1 instead of n, for reasons that are largely historical. Here's the issue:

If we divide by n, as we have been doing, then it turns out that s^2 is biased.

$$E(s^2) = \frac{n-1}{n} \cdot \sigma^2 \tag{18.49}$$

Think about this in the Davis people example, once again in the notebook context. Remember, here n is 1000, and each line of the notebook represents our taking a different random sample of 1000 people. Within each line, there will be entries for W_1 through W_{1000}, the weights of our 1000 people, and for $\overline{W}$ and s. For convenience, let's suppose we record that last column as s^2 instead of s.

Now, say we want to estimate the population variance σ^2. As discussed earlier, the natural estimator for it would be the sample variance, s^2. What (18.49) says is that after looking at an infinite number of lines in the notebook, the average value of s^2 would be just...a...little...bit...too...small. All the s^2 values would average out to $0.999\sigma^2$, rather than to σ^2. We might say that s^2 has a little bit more tendency to underestimate σ^2 than to overestimate it.

So, (18.49) implies that s^2 is a biased estimator of the population variance σ^2, with the amount of bias being

$$\frac{n-1}{n} \cdot \sigma^2 - \sigma^2 = -\frac{1}{n} \cdot \sigma^2 \tag{18.50}$$

Let's prove (18.49). As before, let $W_1, ..., W_n$ be a random sample from some population. So, $EW_i = \mu$ and $Var(W_i) = \sigma^2$, where again μ and σ^2 are the population mean and variance.

It will be more convenient to work with ns^2 than s^2, since it will avoid a lot of dividing by n. So, write

$$ns^2 \;=\; \sum_{i=1}^{n} (W_i - \overline{W})^2 \quad \text{(def.)} \tag{18.51}$$

$$=\; \sum_{i=1}^{n} \left[(W_i - \mu) + (\mu - \overline{W}) \right]^2 \quad \text{(alg.)} \tag{18.52}$$

$$=\; \sum_{i=1}^{n} (W_i - \mu)^2 + 2(\mu - \overline{W}) \sum_{i=1}^{n} (W_i - \mu) + n(\mu - \overline{W})^2 \quad \text{(alg.)} \tag{18.53}$$

But that middle sum is

$$\sum_{i=1}^{n} (W_i - \mu) = \sum_{i=1}^{n} W_i - n\mu = n\overline{W} - n\mu \tag{18.54}$$

So,

$$ns^2 = \sum_{i=1}^{n} (W_i - \mu)^2 - n(\overline{W} - \mu)^2 \tag{18.55}$$

Now let's take the expected value of (18.55). First,

$$E\left(\sum_{i=1}^{n} (W_i - \mu)^2 \right) \;=\; \sum_{i=1}^{n} E[(W_i - \mu)^2] \quad \text{(E is lin.)} \tag{18.56}$$

$$=\; \sum_{i=1}^{n} E[(W_i - EW_i)^2] \quad (W_i \text{ distr. as pop.}) \tag{18.57}$$

$$=\; \sum_{i=1}^{n} Var(W_i) \quad \text{(def. of Var())} \tag{18.58}$$

$$=\; \sum_{i=1}^{n} \sigma^2 \quad (W_i \text{ distr. as pop.}) \tag{18.59}$$

$$=\; n\sigma^2 \tag{18.60}$$

Also,

$$
\begin{aligned}
E[(\overline{W} - \mu)^2] &= E[(\overline{W} - E\overline{W})^2] \quad ((15.8)) & (18.61) \\
&= Var(\overline{W}) \quad \text{(def. of Var())} & (18.62) \\
&= \frac{\sigma^2}{n} \quad (15.13) & (18.63)
\end{aligned}
$$

Applying these last two findings to (18.55), we get (18.49).

$$
E(s^2) = \frac{n-1}{n}\sigma^2 \tag{18.64}
$$

The earlier developers of statistics were bothered by this bias, so they introduced a "fudge factor" by dividing by n-1 instead of n in (15.23). We will call that $\tilde{s}^2$:

$$
\tilde{s}^2 = \frac{1}{n-1}\sum_{i=1}^{n}(W_i - \overline{W})^2 \tag{18.65}
$$

This is the "classical" definition of sample variance, in which we divide by n-1 instead of n.

The R functions **var()** and **sd()** calculate the versions of s^2 and s, respectively, that have a divisor of n-1. In other words, **var()** calculates (18.65), and **sd()** computes its square root.

18.2.2.1   But in This Book, We Divide by n, not n-1 Anyway

But we will use n. After all, when n is large—which is what we are assuming by using the Central Limit Theorem in most of the inference machinery here—it doesn't make any appreciable difference. Clearly it is not important in our Davis example, or our bus simulation example.

Moreover, speaking generally now rather than necessarily for the case of s^2 there is no particular reason to insist that an estimator be unbiased anyway. An alternative estimator may have a little bias but much smaller variance, and thus might be preferable.

And anyway, even though the classical version of s^2, i.e. $\tilde{s}^2$, is an unbiased estimator for σ^2, $\tilde{s}$ is not an unbiased estimator for σ, the population standard deviation (see below). Since we typically have use for $\tilde{s}$ rather than for $\tilde{s}^2$—in (16.4), for example—you can see that unbiasedness is not such an important property after all.

Let's show that $\tilde{s}$ is biased. Recalling the shortcut formula $Var(U) = E(U^2) - (EU)^2$, we have

$$0 \quad < \quad Var(\tilde{s}) \tag{18.66}$$
$$= \quad E[\tilde{s}^2] - [E\tilde{s}]^2 \tag{18.67}$$
$$= \quad \sigma^2 - [E\tilde{s}]^2 \tag{18.68}$$

since $\tilde{s}^2$ is an unbiased estimator of σ^2. So,

$$E\tilde{s} < \sigma \tag{18.69}$$

and $\tilde{s}$ is biased downward.[3]

So, $\tilde{s}$, the standard estimator of σ, is indeed biased, as are many other standard estimators of various quantities. It would be futile to insist on unbiasedness as a criterion of the goodness of an estimator.

18.2.3 Example of Bias Calculation: Max from U(0,c)

Let's find the bias of the estimator (18.30).

The bias is $E\hat{c} - c$. To get $E\hat{c}$ we need the density of that estimator, which we get as follows:

$$P(\hat{c} \leq t) \quad = \quad P(\text{all } W_i \leq t) \quad \text{(definition)} \tag{18.70}$$
$$= \quad \left(\frac{t}{c}\right)^n \quad \text{(density of } W_i) \tag{18.71}$$

So,

$$f_{\hat{c}}(t) = \frac{n}{c^n} t^{n-1} \tag{18.72}$$

Integrating against t, we find that

$$E\hat{c} = \frac{n}{n+1} c \tag{18.73}$$

---

[3]The reader may wonder why we have strict inequality in (18.66). But although it tis rue that Var(U) can be 0, you'll recall that that occurs only when U is constant. Here it would mean that $\tilde{s}$ is constant. This in turn would mean that all the W_i in (18.65) are identical, with probability 1.0—which would mean the population random variable W is constant, e.g. everyone in Davis has the same weight. So, other that in that absurd situation, the inequality in (18.66) will indeed be strict.

So the bias is c/(n+1), not bad at all.

18.2.4   Example of Bias Calculation: Gamma Family

Let us find via simulation, the bias of the Method of Moments Estimator of the parameter λ for the family of gamma distributions. (The estimator was derived in Section 18.1.4.)

```
lambbias <- function(r,lamb,n,nreps) {
   lambhat <- vector(length=nreps)
   unfudge <- (n-1) / n
   for (i in 1:nreps) {
      x <- rgamma(n,shape=r,rate=lamb)
      xbar <- mean(x)
      s2 <- var(x) * unfudge
      lambhat[i] <- xbar / s2
   }
   mean(lambhat) - lamb
}
```

18.2.5   Tradeoff Between Variance and Bias

Consider a general estimator Q of some population value b. Then a common measure of the quality (of course there are many others) of the estimator Q is the **mean squared error** (MSE),

$$E[(Q - b)^2] \tag{18.74}$$

Of course, the smaller the MSE, the better.

One can break (18.74) down into variance and (squared) bias components, as follows:[4]

$$
\begin{aligned}
MSE(Q) &= E[(Q - b)^2] \text{ (definition)} & (18.75) \\
&= E[\{(Q - EQ) + (EQ - b)\}^2] \text{ (algebra)} & (18.76) \\
&= E[(Q - EQ)^2] + 2E\left[(Q - EQ)(EQ - b)\right] + E[(EQ - b)^2] \text{ (E props.)} & (18.77) \\
&= E[(Q - EQ)^2] + E[(EQ - b)^2] \text{ (factor out constant EQ-b)} & (18.78) \\
&= Var(Q) + (EQ - b)^2 \text{ (def. of Var(), fact that EQ-b is const.)} & (18.79) \\
&= \text{variance + squared bias} & (18.80)
\end{aligned}
$$

---

[4]In reading the following derivation, keep in mind that EQ and b are constants.

In other words, in discussing the accuracy of an estimator—especially in comparing two or more candidates to use for our estimator—the average squared error has two main components, one for variance and one for bias. In building a model, these two components are often at odds with each other; we may be able to find an estimator with smaller bias but more variance, or vice versa.

We also see from (18.80) that a little bias in an estimator may be quite tolerable, as long as the variance is low. This is good, because as mentioned earlier, most estimators are in fact biased.

These point will become central in Chapters 21 and 22.

18.3 Simultaneous Inference Methods

Events of small probability happen all the time, because there are so many of them—Jim Sutton, old Cal Poly economics professor

Suppose in our study of heights, weights and so on of people in Davis, we are interested in estimating a number of different quantities, with our forming a confidence interval for each one. Though our confidence level for each one of them will be 95%, our *overall* confidence level will be less than that. In other words, we cannot say we are 95% confident that all the intervals contain their respective population values.

In some cases we may wish to construct confidence intervals in such a way that we can say we are 95% confident that all the intervals are correct. This branch of statistics is known as **simultaneous inference** or **multiple inference**.

(The same issues apply to significance testing, but we will focus on confidence intervals here.)

In this age of Big Data, simultaneous inference is a major issue. We may have hundreds of variables, so the chances of getting spurious results are quite high.

Usually this kind of methodology is used in the comparison of several **treatments**. This term originated in the life sciences, e.g. comparing the effectiveness of several different medications for controlling hypertension, it can be applied in any context. For instance, we might be interested in comparing how well programmers do in several different programming languages, say Python, Ruby and Perl. We'd form three groups of programmers, one for each language, with say 20 programmers per group. Then we would have them write code for a given application. Our measurement could be the length of time T that it takes for them to develop the program to the point at which it runs correctly on a suite of test cases.

Let T_{ij} be the value of T for the jth programmer in the ith group, i = 1,2,3, j = 1,2,...,20. We would then wish to compare the three "treatments," i.e. programming languages, by estimating $\mu_i = ET_{i1}$, i = 1,2,3. Our estimators would be $U_i = \sum_{j=1}^{20} T_{ij}/20$, i = 1,2,3. Since we are comparing the three population means, we may not be satisfied with simply forming ordinary 95% confidence

intervals for each mean. We may wish to form confidence intervals which *jointly* have confidence level 95%.[5]

Note very, very carefully what this means. As usual, think of our notebook idea. Each line of the notebook would contain the 60 observations; different lines would involve different sets of 60 people. So, there would be 60 columns for the raw data, three columns for the U_i. We would also have six more columns for the confidence intervals (lower and upper bounds) for the μ_i. Finally, imagine three more columns, one for each confidence interval, with the entry for each being either Right or Wrong. A confidence interval is labeled Right if it really does contain its target population value, and otherwise is labeled Wrong.

Now, if we construct individual 95% confidence intervals, that means that in a given Right/Wrong column, in the long run 95% of the entries will say Right. But for simultaneous intervals, we hope that within a line we see <u>three</u> Rights, and 95% of all lines will have that property.

In our context here, if we set up our three intervals to have individual confidence levels of 95%, their simultaneous level will be $0.95^3 = 0.86$, since the three confidence intervals are independent. Conversely, if we want a simultaneous level of 0.95, we could take each one at a 98.3% level, since $0.95^{\frac{1}{3}} \approx 0.983$.

However, in general the intervals we wish to form will not be independent, so the above "cube root method" would not work. Here we will give a short introduction to more general procedures.

Note that "nothing in life is free." If we want simultaneous confidence intervals, they will be wider.

Another reason to form simultaneous confidence intervals is that it gives you "license to browse," i.e. to rummage through the data looking for interesting nuggets.

18.3.1   The Bonferonni Method

One simple approach is **Bonferonni's Inequality**:

Lemma 32 *Suppose $A_1, ..., A_g$ are events. Then*

$$P(A_1 \text{ or } ... \text{ or } A_g) \leq \sum_{i=1}^{g} P(A_i) \tag{18.81}$$

---

[5]The word *may* is important here. It really is a matter of philosophy as to whether one uses simultaneous inference procedures.

You can easily see this for g = 2:

$$P(A_1 \text{ or } A_2) = P(A_1) + P(A_2) - P(A_1 \text{ and } A_2) \leq P(A_1) + P(A_2) \tag{18.82}$$

One can then prove the general case by mathematical induction.

Now to apply this to forming simultaneous confidence intervals, take A_i to be the event that the ith confidence interval is incorrect, i.e. fails to include the population quantity being estimated. Then (18.81) says that if, say, we form two confidence intervals, each having individual confidence level (100-5/2)%, i.e. 97.5%, then the overall collective confidence level for those two intervals is at least 95%. Here's why: Let A_1 be the event that the first interval is wrong, and A_2 is the corresponding event for the second interval. Then

$$
\begin{aligned}
\text{overall conf. level} \quad &= \quad P(\text{not } A_1 \text{ and not } A_2) & (18.83) \\
&= \quad 1 - P(A_1 \text{ or } A_2) & (18.84) \\
&\geq \quad 1 - P(A_1) - P(A_2) & (18.85) \\
&= \quad 1 - 0.025 - 0.025 & (18.86) \\
&= \quad 0.95 & (18.87)
\end{aligned}
$$

18.3.2 Scheffe's Method

The Bonferonni method is unsuitable for more than a few intervals; each one would have to have such a high individual confidence level that the intervals would be very wide. Many alternatives exist, a famous one being **Scheffe's method**.[6] The large-sample version we'll use here is:

Theorem 33 *Suppose $R = (R_1, ..., R_k)'$ has an approximately multivariate normal distribution, with mean vector $\mu = (\mu_i)$ and covariance matrix $\Sigma = (\sigma_{ij})$. Let $\widehat{\Sigma}$ be a* **consistent** *estimator of Σ, meaning that it converges in probability to Σ as the sample size goes to infinity.*

For any constants $c_1, ..., c_k$, consider linear combinations of the R_i,

$$\sum_{i=1}^{k} c_i R_i \tag{18.88}$$

---

[6]The name is pronounced "sheh-FAY."

which estimate

$$\sum_{i=1}^{k} c_i \mu_i \tag{18.89}$$

Form the confidence intervals

$$\sum_{i=1}^{k} c_i R_i \pm \sqrt{\chi^2_{\alpha;k}} s(c_1, ..., c_k) \tag{18.90}$$

where

$$[s(c_1, ..., c_k)]^2 = (c_1, ..., c_k)^T \widehat{\Sigma}(c_1, ..., c_k) \tag{18.91}$$

and where $\chi^2_{\alpha;k}$ is the upper-α percentile of a chi-square distribution with k degrees of freedom.[7]

Then all of these intervals (for infinitely many values of the c_i!) have simultaneous confidence level $1 - \alpha$.

By the way, if we are interested in only constructing confidence intervals for **contrasts**, i.e. c_i having the property that $\Sigma_i c_i = 0$, we the number of degrees of freedom reduces to k-1, thus producing narrower intervals.

Just as in Section 18.2.2 we avoided the t-distribution, here we have avoided the F distribution, which is used instead of ch-square in the "exact" form of Scheffe's method.

18.3.3   Example

For example, again consider the Davis heights example in Section 16.6. Suppose we want to find approximate 95% confidence intervals for two population quantities, μ_1 and μ_2. These correspond to values of c_1, c_2 of (1,0) and (0,1). Since the two samples are independent, $\sigma_{12} = 0$. The chi-square value is 5.99,[8] so the square root in (18.90) is 3.46. So, we would compute (16.4) for $\overline{X}$ and then for $\overline{Y}$, but would use 3.46 instead of 1.96.

This actually is not as good as Bonferonni in this case. For Bonferonni, we would find two 97.5% confidence intervals, which would use 2.24 instead of 1.96.

---

[7]Recall that the distribution of the sum of squares of g independent N(0,1) random variables is called **chi-square with g degrees of freedom**. It is tabulated in the R statistical package's function **qchisq()**.

[8]Obtained from R via **qchisq(0.95,2)**.

Scheffe's method is too conservative if we just are forming a small number of intervals, but it is great if we form a lot of them. Moreover, it is very general, usable whenever we have a set of approximately normal estimators.

18.3.4 Other Methods for Simultaneous Inference

There are many other methods for simultaneous inference. It should be noted, though, that many of them are limited in scope, and quite a few of them are oriented only toward significance testing, rather than confidence intervals. In any event, they are beyond the scope of this book.

18.4 Bayesian Methods

Everyone is entitled to his own opinion, but not his own facts—Daniel Patrick Moynihan, senator from New York, 1976-2000

Black cat, white cat, it doesn't matter as long as it catches mice—Deng Xiaoping, when asked about his plans to give private industry a greater role in China's economy

Whiskey's for drinkin' and water's for fightin' over—Mark Twain, on California water jurisdiction battles

The most controversial topic in statistics by far is that of **Bayesian** methods, the "California water" of the statistics world. In fact, it is so controversial that a strident Bayesian colleague of mine even took issue with my calling it "controversial"!

The name stems from Bayes' Rule (Section 2.12),

$$P(A|B) = \frac{P(A)P(B|A)}{P(A)P(B|A) + P(\text{not } A)P(B|\text{not } A)} \tag{18.92}$$

No one questions the validity of Bayes' Rule, and thus there is no controversy regarding statistical procedures that make use of probability calculations based on that rule. But the key word is *probability*. As long as the various terms in (18.92) are real probabilities—that is, based on actual data—there is no controversy.

But instead, the debate stems from the cases in which Bayesians replace some of the probabilities in the theorem with "feelings," i.e. non-probabilities, arising from what they call **subjective prior distributions**. The key word is then *subjective*. Our section here will concern the controversy over

the use of subjective priors.[9]

Say we wish to estimate a population mean. Here the Bayesian analyst, before even collecting data, says, "Well, I think the population mean could be 1.2, with probability, oh, let's say 0.28, but on the other hand, it might also be 0.88, with probability, well, I'll put it at 0.49..." etc. This is the analyst's subjective prior distribution for the population mean. The analyst does this before even collecting any data. Note carefully that he is NOT claiming these are real probabilities; he's just trying to quantify his hunches. The analyst then collects the data, and uses some mathematical procedure that combines these "feelings" with the actual data, and which then outputs an estimate of the population mean or other quantity of interest.

The Bayesians justify this by saying one should use all available information, even if it is just a hunch. "The analyst is typically an expert in the field under study. You wouldn't want to throw away his/her expertise, would you?" Moreover, they cite theoretical analyses that show that Bayes estimator doing very well in terms of criteria such as mean squared error, even if the priors are not "valid."

The non-Bayesians, known as **frequentists**, on the other hand dismiss this as unscientific and lacking in impartiality. "In research on a controversial health issue, say, you wouldn't want the researcher to incorporate his/her personal political biases into the number crunching, would you?" So, the frequentists' view is reminiscent of the Moynihan quoted above.

On the other hand, in the computer science world Bayesian estimation seems to be much less of a controversy. Computer scientists, being engineers, tend to be interested in whether a method seems to work, with the reasons being less important. This is the "black cat, white cat" approach.

By the way, the frequentists also point out that in the real world one must typically perform inference (confidence intervals or significance tests), not just compute point estimates; Bayesian methods are not really suited for inference.

Note carefully the key role of *data*. One might ask, for instance, "Why this sharp distinction between the Bayesians and the frequentists over the subjectivity issue? Don't the frequentists make subjective decisions too?" Consider an analysis of disk drive lifetime data, for instance. Some frequentist statistician might use a normal model, instead of, say, a gamma model. Isn't that subjectivity? The answer is no, because the statistician can *use the data* to assess the validity of her model, employing the methods of Section 21.2.

---

[9]By contrast, there is no controversy if the prior makes use of real data. I will explain this in Section 18.4.1.1 below, but in the mean time, note that my use of the term *Bayesian* refers only to subjective priors.

18.4.1 How It Works

To introduce the idea, consider again the example of estimating p, the probability of heads for a certain penny. Suppose we were to say—before tossing the penny even once—"I think p could be any number, but more likely near 0.5, something like a normal distribution with mean 0.5 and standard deviation, oh, let's say 0.1."

Of course, the true value of p is between 0 and 1, while the normal distribution extends from $-\infty$ to ∞. As noted in Section 8.17, the use of normal distributions is common for modeling bounded quantities like this one. Actually, many Bayesians prefer to use a beta distribution for the prior in this kind of setting, as the math works out more cleanly (derivations not shown here). But let's stick with the normal prior here for illustration.

The prior distribution is then $N(0.5, 0.1^2)$. But again, note that the Bayesians do not consider it to be a distribution in the sense of probability. It just quantifies our "gut feeling" here, our "hunch."

Nevertheless, in terms of the mathematics involved, it's as if the Bayesians are treating p as random, with p's distribution being whatever the analyst specifies as the prior. Under this "random p" assumption, the Maximum Likelihood Estimate (MLE), for instance, would change. Just as in the frequentist approach, the data here is X, the number of heads we get from n tosses of the penny. But in contrast to the frequentist approach, in which the likelihood would be

$$L = \left(\begin{array}{c} n \\ X \end{array} \right) p^X (1-p)^{n-X} \tag{18.93}$$

it now becomes

$$L = \frac{1}{\sqrt{2\pi}\,0.1} \, \exp -0.5[(p-0.5)/0.1]^2 \left(\begin{array}{c} n \\ X \end{array} \right) p^X (1-p)^{n-X} \tag{18.94}$$

This is basically P(A and B) = P(A) P(B|A), though using a density rather than a probability mass function. We would then find the value of p which maximizes L, and take that as our estimate.

A Bayesian would use Bayes' Rule to compute the "distribution" of p given X, called the **posterior distribution**. The analog of (18.92) would be (18.94) divided by the integral of (18.94) as p ranges from 0 to 1, with the resulting quotient then being treated as a density. The (conditional) MLE would then be the **mode**, i.e. the point of maximal density of the posterior distribution.

But we could use any measure of central tendency, and in fact typically the mean is used, rather than the mode. In other words:

> To estimate a population value θ, the Bayesian constructs a prior "distribution" for
> θ (again, the quotation marks indicate that it is just a quantified gut feeling, rather

than a real probability distribution). Then she uses the prior together with the actual observed data to construct the posterior distribution. Finally, she takes her estimate $\hat{\theta}$ to be the mean of the posterior distribution.

Note how this procedure achieves a kind of balance between what our hunch says and what our data say. In (18.94), suppose the mean of p is 0.5 but n = 20 and X = 12 Then the frequentist estimator would be X/n = 0.6, while the Bayes estimator would be about 0.56. (Computation not shown here.) So our Bayesian approach "pulled" our estimate away from the frequentist estimate, toward our hunch that p is at or very near 0.5. This pulling effect would be stronger for smaller n or for a smaller standard deviation of the prior "distribution."

18.4.1.1   Empirical Bayes Methods

Note carefully that if the prior distribution in our model is not subjective, but is a real distribution verifiable from data, the above analysis on p would not be controversial at all. Say p does vary a substantial amount from one penny to another, so that there is a physical distribution involved. Suppose we have a sample of many pennies, tossing each one n times. If n is very large, we'll get a pretty accurate estimate of the value of p for each coin, and we can then plot these values in a histogram and compare it to the $N(0.5, 0.1^2)$ density, to check whether our prior is reasonable. This is called an **empirical Bayes** model, because we can empirically estimate our prior distribution, and check its validity. In spite of the name, frequentists would not consider this to be "Bayesian" analysis. Note that we could also assume that p has a general $N(\mu, \sigma^2)$ distribution, and estimate μ and σ from the data.

18.4.2   Extent of Usage of Subjective Priors

Though many statisticians, especially academics, are staunch, often militantly proselytizing, Bayesians, only a small minority of statisticians use the Bayesian approach **in practice**.

One way to see that Bayesian methodology is not mainstream is through the R programming language. For example, as of December 2010, only about 65 of the more than 3000 packages on CRAN, the R repository, involve Bayesian techniques. (See `http://cran.r-project.org/web/packages/tgp/index.html`.) There is actually a book on the topic, *Bayesian Computation with R*, by Jim Albert, Springer, 2007, and among those who use Bayesian techniques, many use R for that purpose. However, almost all general-purpose books on R do not cover Bayesian methodology at all.

Significantly, even among Bayesian academics, many use frequentist methods when they work on real, practical problems. Choose a Bayesian academic statistician at random, and you'll likely find on the Web that he/she does not use Bayesian methods when working on real applications.

On the other hand, use of subjective priors has become very common in the computer science research community. Papers using Bayesian methods appear frequently (no pun intended) in the CS research literature, and "seldom is heard a discouraging word."

18.4.3 Arguments Against Use of Subjective Priors

As noted, most professional statisticians are frequentists. In this section we will look at the arguments they have against the subjective Bayesian approach.

First, it's vital to reiterate a point made earlier:

> Frequentists have no objection at all to use of prior distributions based on actual data. They only object to use of subjective priors.

So, what is it about subjective priors that frequentists don't like?

The first point is that ultimately, the use of any statistical analysis is to make a decision about something. This could be a very formal decision, such as occurs when the Food and Drug Administration (FDA) decides whether to approve a new drug, or it could be informal, for instance when an ordinary citizen reads a newspaper article reporting on a study analyzing data on traffic accidents, and she decides what to conclude from the study.

There is nothing wrong using one's gut feelings to make a final decision, but it should not be part of the mathematical analysis of the data. One's hunches can play a role in deciding the "preponderance of evidence," as discussed in Section 17.11.5, but that should be kept separate from our data analysis.

If for example the FDA's data shows the new drug to be effective, but at the same time the FDA scientists still have their doubts, they may decide to delay approval of the drug pending further study. So they can certainly act on their hunch, or on non-data information they have, concerning approval of the drug. But the FDA, as a public agency, has a responsibility to the citizenry to state what the data say, i.e. to report the frequentist estimate, rather than merely reporting a number—the Bayesian estimate—that mixes fact and hunch.

In many if not most applications of statistics, there is a need for impartial estimates. As noted above, even if the FDA acts on a hunch to delay approval of a drug in spite of favorable data, the FDA owes the public (and the pharmaceutal firm) an impartial report of what the data say. Bayesian estimation is by definition not impartial. One Bayesian statistician friend put it very well, saying "I believe my own subjective priors, but I don't believe those of other people."

Furthermore, in practice we are typically interested in inference, i.e. confidence intervals and significance tests, rather than just point estimation. We are sampling from populations, and want

to be able to legitimately make inferences about those populations. For instance, though one can derive a Bayesian 95% confidence interval for p for our coin, it really has very little meaning, and again is certainly not impartial.

Some Bayesians justify their approach by pointing to the problems of p-values. Yet most frequentists also dislike p-values, and certainly use confidence intervals instead. The Bayesians who say one should uses subjective priors instead of p-values are simply setting up a sraw man, the critics say.

A common Bayesian approach concerns lower and upper bounds. The analyt feels sure that the true population value θ is, say, between r and s. He then chooses a prior distribution on (r,s), say uniform. Putting aside the question of the meaning of the results that ensure from the particular choice of prior, critics may ask, "What if the frequentist estimate turns out to be less than r? It could be a sampling artificat, but wouldn't you want to know about it, rather than automatically preventing such a situation?" This leads to the next section.

18.4.4   What Would You Do? A Possible Resolution

Consider the following scenario. Steven is running for president. Leo, his campaign manager, has commissioned Lynn to conduct a poll to assess Steven's current support among the voters. Lynn takes her poll, and finds that 57% of those polled support Steven. But her own gut feeling as an expert in politics, is that Steven's support is only 48%. She then combines these two numbers in some Bayesian fashion, and comes up with 50.2% as her estimate of Steven's support.

So, here the frequentist estimate is 57%, while Lynn's Bayesian estimate is 50.2%.

Lynn then gives Steven only the 50.2% figure, not reporting the value 57% number to him. Leo asks Lynn how she arrived at that number, and she explains that she combined her prior distribution with the data.

If you were Leo, what would you do? Consider two choices as to instructions you might give Lynn:

(a) You could say, "Lynn, I trust your judgment, so as the election campaign progresses, always give me only your Bayesian estimate."

(b) You might say, "Lynn, I trust your judgment, but as the election campaign progresses, always give me both your Bayesian estimate and what the impartial data actually say."

I believe that choice (b) is something that both the Bayesian and frequentist camps would generally agree upon.

18.4.5   The Markov Chain Monte Carlo Method

The computation of posterior distributions can involve complex multiple integrals. One could use numerical integration techniques, but again this can get complicated.

The *Markov Chain Monte Carlo* method approaches this problem by defining a certain Markov chain through the integration space, and then simulating that chain. The details are beyond the scope of this book, but here is yet another application of Markov chains.

18.4.6   Further Reading

Two UCD professors, the first current and the second former, have written interesting books about the Bayesian approach:

- *A Comparison of the Bayesian and Frequentist Approaches to Estimation*, Frank Samaniego, Springer, 2010.

- *Bayesian Ideas and Data Analysis: An Introduction for Scientists and Statisticians*, Chapman & Hall, 2010.

Chapter 19

Mixture Models

Mixture distributions pop up in lots of places in probabilistic modeling, so we devote a chapter to them here.

19.1 The Old Trick Coin Example, Updated

Recall the trick coin example in Section 5.1. Let's look at a somewhat different version.

Suppose we have a large box of coins of two types. One type has probability p of heads, the other probability q of heads. A proportion r of the coins is of the first type. We reach into the box, choose a coin at random, and toss that coin 10 times, resulting in N heads. What is the distribution of N?

Since N is discrete, the distribution is described by its pmf, which is

$$p_N(k) = r\binom{10}{k}p^k(1-p)^{10-k} + (1-r)\binom{10}{k}q^k(1-q)^{10-k} \tag{19.1}$$

You can see why this is called a *mixture* model. The above pmf is a mixture of two binomial pmfs, with mixing proportions r and 1-r.

19.2 General Mixture Distributions

The general definition is rather abstract. Let's first restrict it to discrete outputs:

Definition 34 *Let $\{g_t\}_{t \in A}$ be a collection of pmfs, and let M be a random variable taking values*

in A. Consider discrete and continous cases for M, as follows. The **mixture distribution** *of the* g_t *with weights given by the distribution of M is defined as follows:*

- *M is discrete: The mixture distribution is the pmf*[1]

$$h = \sum_{i \in A} p_M(i) \; g_i \tag{19.2}$$

- *M is continuous: The mixture distribution has the pmf*

$$h(u) = \int_{t \in A} f_M(t) \; g_t(u) \; dt \tag{19.3}$$

For continuous outcomes, replace "pmf" by "density" everywhere above.

Actually, the definition isn't so abstract after all. A random variable Y having pmf or density h arises first of the random variable M, and then if M = i, getting Y randomly from the distribution g_i. A "notebook" analysis would have columns for both M and Y.

In other words, the pmf of Y is a weighted average of the g_i, with weights being the pmf of M. Note that this implies that the weights most be nonnegative and sum to 1.[2]

As noted, the above definition is pretty abstract, but it can be readily grasped by considering the above trick coin example. Here,

- $Y = N$

- A = {0,1}

- $p_M(0) = p_M(1) = 0.5$

- $g_0 =$ the pmf for B(i,0.1)

- $g_1 =$ the pmf for B(i,0.9)

---

[1]The reader should verify that H does qualify as a pmf, i.e. it consists of numbers in [0,1] that sum to 1.
[2]Some readers may recognize this as a **convex combination**.

Mixture distributions occur in a surprisingly wide variety of applications, and thus this material should be mastered. It is somewhat difficult at first, but if you always keep concrete examples such as the trick coin problem in mind, it is not bad at all.

By the way, there is nothing in the above definition that limits the G_i and H to be for scalar random variables. They could all be bivariate, for instance (though of course they all have to be of the same dimension.)

19.3   Generating Random Variates from a Mixture Distribution

One way to get additional insight on mixture distributions is to think about how to simulate them. For instance, consider the trick coin problem. The following R code simulates X_i:

```
rxi <- function(i) {
   m <- sample(0:1,1)
   p <- if (M ==0) 0.1 else 0.9
   rbinom(1,i,p)
}
```

In general:

- We first generate M.

- Then we generate a random variable from the pmf G_M.

Various examples of mixture distributions will be presented in this chapter. But first, we need to develop some machinery.

19.4   A Useful Tool: the Law of Total Expectation

Let's put the mixture distribution issue aside for a moment, to discuss a tool that is useful both in mixture models and elsewhere.

The notion of iterated expectation, which was introduced in Sections 4.15 and 7.78, is an extremely important tool for the analysis of mixture distributions. We first need to extend our earlier definitions somewhat.

Again, this will seem rather abstract at first, but you'll see that it is a great convenience.

19.4.1 Conditional Expected Value As a Random Variable

For random variables Y and W, and an event, say W = t, the quantity E(Y|W = t) is the long-run average of Y, among the times when W = t occurs.

Note several things about the expression E(Y|W = t):

- The item to the left of the | symbol is a *random variable* (Y).

- The item on the right of the | symbol is an *event* (W = t).

- The overall expression evaluates to a constant.

By contrast, for the quantity E(Y|W) to be defined shortly, it is the case that:

- The item to the left of the | symbol is a random variable (Y).

- The item to the right of the | symbol is a random variable (W).

- The overall expression itself is a random variable, not a constant.

It will be very important to keep these differences in mind.

Consider the function g(t) defined as[3]

$$g(t) = E(Y|W = t) \tag{19.4}$$

In this case, the item to the right of the | is an event, and thus g(t) is a constant (for each value of t), not a random variable.

Definition 35 *Define g() as in (19.4). Form the new random variable Q = g(W). Then the quantity E(Y|W) is defined to be Q.*

(Before reading any further, re-read the two sets of bulleted items above, and make sure you understand the difference between E(Y|W=t) and E(Y|W).)

One can view E(Y|W) as a projection in an abstract vector space. This is very elegant, and actually aids the intuition. If (and only if) you are mathematically adventurous, read the details in Section 19.10.2.

---

[3]Of course, the t is just a placeholder, and any other letter could be used.

19.4.2 Famous Formula: Theorem of Total Expectation

An extremely useful formula, given only scant or no mention in most undergraduate probability courses, is

$$E(Y) = E[E(Y|W)] \tag{19.5}$$

for any random variables Y and W (for which the expectations are defined).

The RHS of (19.5) looks odd at first, but it's merely E[g(W)]; since Q = E(Y|W) is a random variable, we can certainly ask what its expected value is.

Equation (19.5) is a bit abstract. It's a very useful abstraction, enabling streamlined writing and thinking about the probabilistic structures at hand. Be sure to review the intuitive explanation following (4.68) before continuing.

19.4.3 Properties of Conditional Expectation and Variance

Conditional expected value is still expected value, and thus still has the usual properties of expected value. In particular,

$$E(aU + bV \mid W) = aE(U \mid W) + bE(V \mid W) \tag{19.6}$$

for any constants a and b. The same is true for variance, e.g.

$$Var(aU|W) = a^2 Var(U \mid W) \tag{19.7}$$

for any constants a, and

$$Var(U + V \mid W) = Var(U \mid W) + Var(V \mid W) \tag{19.8}$$

if U and V are *conditionally independent*, given W.

The reason for all this is simple. Consider the discrete case, for convenience. As seen in (3.13), E(Z) is a weighted average of the values of Z, with the weights being the values of p_Z:

$$EZ = \sum_c c p_Z(c) \tag{19.9}$$

But $E(Z|T)$ is simply another weighted average, with the weights being the conditional distribution of Z given T:

$$E(Z|T) = \sum_c c\, p_{Z|T}(c) \tag{19.10}$$

Note that (continuing the discrete case)

$$p_{Z|T=r}(c) = \frac{P(Z = c, T = r)}{P(T = r)} \tag{19.11}$$

All this extends to variance, since it too is defined in terms of expected value, in (3.38). So,

$$Var(Z|T) = E\{[Z - E(Z|T)]^2\} \tag{19.12}$$

And of course, if the variables above are continuous, just replace pmfs for densities, and sums by integrals.

19.4.4   Example: More on Flipping Coins with Bonuses

Recall the situation of Section 4.12: A game involves flipping a coin k times. Each time you get a head, you get a bonus flip, not counted among the k. (But if you get a head from a bonus flip, that does not give you its own bonus flip.) Let Y denote the number of heads you get among all flips, bonus or not. We'll compute EY.

Let

$$
\begin{aligned}
W &= \quad \text{number of heads from nonbonus flips} & (19.13) \\
&= \quad \text{number of bonus flips} & (19.14)
\end{aligned}
$$

This is a natural situation in which to try the Theorem of Total Expectation, conditioning on Y. Reason as follows:

Y-W is the number of heads obtained from the bonus flips. Given W, the random variable Y-W has a binomial distribution with parameters W and 0.5, so

$$E(Y - W \mid W) = 0.5W \tag{19.15}$$

Then from (19.5),

$$
\begin{aligned}
EY &= E[E(Y \mid W)] && (19.16)\\
&= E\left[E(\{Y - W\} + W \mid W)\right] && (19.17)\\
&= E\left[E(Y - W|W) + E(W \mid W)\right] && (19.18)\\
&= E\left[0.5W + W\right] && (19.19)\\
&= 1.5EW && (19.20)\\
&= 0.75k, && (19.21)
\end{aligned}
$$

that last result coming from the fact that W itself has a binomial distribution with k trials and success probability 0.5.

Now here is the point: We could have used (4.68) above. But our use of (19.5) really streamlined things. If we had invoked (4.68), we would have had a messy sum, with binomial pmf expressions and so on. By using (19.5), we avoided that, a big win.

19.4.5 Example: Trapped Miner

This rather whimsical example is adapted from *Stochastic Processes,* by Sheldon Ross, Wiley, 1996.

A miner is trapped in a mine, and has a choice of three doors. Though he doesn't realize it, if he chooses to exit the first door, it will take him to safety after 2 hours of travel. If he chooses the second one, it will lead back to the mine after 3 hours of travel. The third one leads back to the mine after 5 hours of travel. Suppose the doors look identical, and if he returns to the mine he does not remember which door(s) he tried earlier. What is the expected time until he reaches safety?

The answer turns out to be 10. This is no accident, as 2+3+5 = 10. This was discovered on an intuitive basis (shown below) by UCD grad student Ahmed Ahmedin. Here is how we can derive the answer, using (19.5):

Let Y be the time needed to reach safety, let N denote the total number of attempts the miner makes before escaping (including the successful attempt at the end), and let U_i denote the time spent traveling during the ith attempt, i = 1,...,N. Then

$$
\begin{aligned}
EY &= E(U_1 + ..., +U_N) && (19.22)\\
&= E\left[E(U_1 + ... + U_N|N)\right] && (19.23)
\end{aligned}
$$

Note carefully that the expression $U_1 + ... + U_N$ not only has random terms, but also the *number* of

terms, N, is random. So, technically, we cannot use (3.21) or its extension to multiple terms. But by conditioning on N, we are back in the situation of a fixed number of terms, and can proceed.

Given N, each of $U_1, ..., U_{N-1}$ takes on the values 3 and 5, with probability 0.5 each, while U_N is the constant 2. Thus

$$E(U_1 + ..., +U_N|N) = (N - 1)\frac{3 + 5}{2} + 2 = 4N - 2 \tag{19.24}$$

N has a geometric distribution with success probability p = 1/3, so EN = 3. Putting all this together, we have

$$
\begin{aligned}
EY &= E(U_1 + ..., +U_N) & (19.25) \\
&= E\left[E(U_1 + ..., +U_N|N)\right] & (19.26) \\
&= E(4N - 2) = 10 & (19.27)
\end{aligned}
$$

If 2, 3 and 5 were replaced by *a*, *b* and *c*, the result would turn out to be a+b+c. Intuitively: It takes an average of 3 attempts to escape, with each attempt having mean time of (a+b+c)/3, so the mean time overall is a+b+c.

Here is a different derivation (the one in Ross' book):

Let W denote the number of the door chosen (1, 2 or 3) on the first try. Then let us consider what values E(Y|W) can have. If W = 1, then Y = 2, so

$$E(Y|W = 1) = 2 \tag{19.28}$$

If W = 2, things are a bit more complicated. The miner will go on a 3-hour excursion, and then be back in his original situation, and thus have a further expected wait of EY, since "time starts over." In other words,

$$E(Y|W = 2) = 3 + EY \tag{19.29}$$

Similarly,

$$E(Y|W = 3) = 5 + EY \tag{19.30}$$

In summary, now considering the *random variable* E(Y|W), we have

$$Q = E(Y|W) = \begin{cases} 2, & w.p.\frac{1}{3} \\ 3 + EY, & w.p.\frac{1}{3} \\ 5 + EY, & w.p.\frac{1}{3} \end{cases} \tag{19.31}$$

where "w.p." means "with probability." So, using (19.5) we have

$$EY = EQ = 2 \times \frac{1}{3} + (3 + EY) \times \frac{1}{3} + (5 + EY) \times \frac{1}{3} = \frac{10}{3} + \frac{2}{3}EY \tag{19.32}$$

Equating the extreme left and extreme right ends of this series of equations, we can solve for EY, which we find to be 10.

It is left to the reader to see how this would change if we assume that the miner remembers which doors he has already hit.

19.4.6 Example: Analysis of Hash Tables

(Famous example, adapted from various sources.)

Consider a database table consisting of m cells, only some of which are currently occupied. Each time a new key must be inserted, it is used in a hash function to find an unoccupied cell. Since multiple keys can map to the same table cell, we may have to probe multiple times before finding an unoccupied cell.

We wish to find E(Y), where Y is the number of probes needed to insert a new key. One approach to doing so would be to condition on W, the number of currently occupied cells at the time we do a search. After finding E(Y|W), we can use the Theorem of Total Expectation to find EY. We will make two assumptions (to be discussed later):

(a) Given that W = k, each probe will collide with an existing cell with probability k/m, with successive probes being independent.

(b) W is uniformly distributed on the set 1,2,...,m, i.e. P(W = k) = 1/m for each k.

To calculate E(Y|W=k), we note that given W = k, then Y is the number of independent trials until a "success" is reached, where "success" means that our probe turns out to be to an unoccupied cell. This is a **geometric** distribution, i.e.

$$P(Y = r|W = k) = \left(\frac{k}{m}\right)^{r-1}\left(1 - \frac{k}{m}\right) \tag{19.33}$$

The mean of this geometric distribution is, from (4.4),

$$\frac{1}{1 - \frac{k}{m}} \tag{19.34}$$

Then

$$
\begin{aligned}
EY &= E[E(Y|W)] & \text{(19.35)}\\
&= \sum_{k=1}^{m-1} \frac{1}{m} E(Y|W = k) & \text{(19.36)}\\
&= \sum_{k=1}^{m-1} \frac{1}{m-k} & \text{(19.37)}\\
&= 1 + \frac{1}{2} + \frac{1}{3} + ... + \frac{1}{m-1} & \text{(19.38)}\\
&\approx \int_1^m \frac{1}{u} du & \text{(19.39)}\\
&= \ln(m) & \text{(19.40)}
\end{aligned}
$$

where the approximation is something you might remember from calculus (you can picture it by drawing rectangles to approximate the area under the curve.).

Now, what about our assumptions, (a) and (b)? The assumption in (a) of each cell having probability k/m should be reasonably accurate if k is much smaller than m, because hash functions tend to distribute probes uniformly, and the assumption of independence of successive probes is all right too, since it is very unlikely that we would hit the same cell twice. However, if k is not much smaller than m, the accuracy will suffer.

Assumption (b) is more subtle, with differing interpretations. For example, the model may concern one specific database, in which case the assumption may be questionable. Presumably W grows over time, in which case the assumption would make no sense—it doesn't even *have* a distribution. We could instead think of a database which grows and shrinks as time progresses. However, even here, it would seem that W would probably oscillate around some value like m/2, rather than being uniformly distributed as assumed here. Thus, this model is probably not very realistic. However, even idealized models can sometimes provide important insights.

19.4.7 What About the Variance?

By the way, one might guess that the analog of the Theorem of Total Expectation for variance is

$$Var(Y) = E[Var(Y|W)] \tag{19.41}$$

But this is false. Think for example of the extreme case in which Y = W. Then Var(Y|W) would be 0, but Var(Y) should be nonzero.

The correct formula, called the Law of Total Variance, is

$$Var(Y) = E[Var(Y|W)] + Var[E(Y|W)] \tag{19.42}$$

Deriving this formula is easy, by simply evaluating both sides of (19.42), and using the relation $Var(X) = E(X^2) - (EX)^2$. This exercise is left to the reader. See also Section 19.10.2.

19.5 The EM Algorithm

Now returning to our main topic of mixture models, let's discuss a very common tool for fitting such models.

Consider again the example in Section 19.1. Now suppose p, q and r are all unknown, and we wish to estimate them from data. Say we will perform the above experiment 50 times, resulting in our data $N_1, ..., N_{50}$ We will estimate p, q and r by applying some method, say the Method of Moments, Maximum Likelihood or some ad hoc method of our own, to our data. Each N_i has the distribution seen in p_N above. This is a parametric family, with 3 parameters. (If, say, r is known, we only have a two-parameter family, and things are easier.)

So, how can we estimate those three parameters? In Chapter 18, we found some general methods for parameter estimation. Let's look at one of them, Maximum Likelihood Estimators (MLEs).

In (19.1), the MLE vector $(\widehat{p}, \widehat{q}, \widehat{r})'$ would be found by maximizing

$$\Pi_{i=1}^n \left[r \binom{10}{N_i} p^{N_i} (1-p)^{10-N_i} + (1-r) \binom{10}{N_i} q^{N_i} (1-q)^{10-N_i} \right] \tag{19.43}$$

After taking derivatives, etc., one would end up with a messy, nonlinear set of equations to solve. One could try **mle()**, but there is a better way to get the MLE: the Expection/Maximization (EM) algorithm. The derivation and ultimate formulas can get quite complex. Fortunately, R libraries

exist, such as **mixtools**, so you can avoid knowing all the details, as long as you understand the basic notion of a mixture model.

19.5.1   Overall Idea

In something like (19.2), for instance, one would make initial guesses for the $p_M(i)$ and then estimate the parameters of the g_i. In the next step, we'd do the opposite—take our current guesses for the latter parameters as known, and estimate the $p_M(i)$. Keep going until convergence.

To make things concrete, recall the trick coin example Section 19.1. But change it a little, so that the probabilities of heads for the two coins are unknown; call them p_0 (heads-light coin) and p_1 (heads-heavy coin). And also suppose that the two coins are not equally likely to be chosen, so that $p_M()$ is not known; denote P(M = 1) by q.

Suppose we have sample data, consisting of doing this experiment multiple times, say by reaching into the box n times and then doing m flips each time. We then wish to estimate 3 quantities—q and the two p_i—using our sample data.

We do so using the following iterative process. (The account here is not exactly EM, but captures the spirit of it.) We set up initial guesses, and iterate until convergence:

- **E step:** Update guess for q (complicated Bayes Rule equations).

- **M step:** Using the new guess for q, update the gueses for the two p_i.

The details are beyond the scope of this book.[4]

19.5.2   The mixtools Package

R's CRAN repository of contributed software includes the **mixtools** library. Let's see how to use one of the functions in that library, **normalmixEM()**, which assumes that the densities in (19.3) are from the normal distribution family.

Let's suppose we are modeling the data as a mixture of 2 normals. So, with our observed data set, our goal is to estimate 5 parameters from our data:

- μ_1, the mean of the first normal distribution

- μ_2, the mean of the second normal distribution

---

[4] "M" in the M step refers to the Maximum Likelihood method, a special case of the material in Section 18.1.3.

- σ_1, the mean of the second normal distribution

- σ_2, the mean of the second normal distribution

- q = P(M = 1) = 1 - P(M = 2)

The basic form of the call is

```
normalmixEM(x,lambda=NULL,mu=NULL,sigma=NULL,k=2)
```

where

- **x** is the data, a vector

- **lambda** is a vector of our initial guesses for the quantities P(M = i), of length **k** (recycling will be used if it is shorter); note that these must be probabilities summing to 1

- **mu** is a vector of our initial guesses for the μ_i

- **sigma** is a vector of our initial guesses for the σ_i

- **k** is the number of values M can take on, e.g. 2 for a mixture of 2 normals

One can leave the initial guesses as the default NULL values, but reasonable guesses may help convergence.

The return value will be an R list, whose components include **lambda**, **mu** and **sigma**, the final estimates of those values.

19.5.3 Example: Old Faithful Geyser

This example is taken from the online documentation in **mixtools**. The data concern the Old Faithful Geyser, a built-in data set in R.

The data here consist of waiting times between eruptions. A histogram, obtained via

```
> hist(faithful$waiting)
```

and shown in Figure 19.1, seems to suggest that the waiting time distribution is a mixture of 2 normals.

I tried initial guesses of means and standard deviations from the appearance of the histogram, and used equal weights for my initial guesses in that aspect:
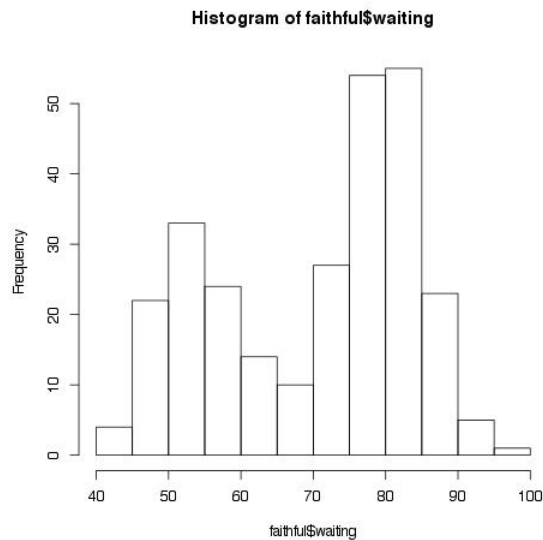
Figure 19.1: Old Faithful waiting times

```
> mixout <- normalmixEM(faithful$waiting,lambda=0.5,mu=c(55,80),sigma=10,k=2)
number of iterations= 9
> str(mixout)
List of 9
 $ x         : num [1:272] 79 54 74 62 85 55 88 85 51 85 ...
 $ lambda    : num [1:2] 0.361 0.639
 $ mu        : num [1:2] 54.6 80.1
 $ sigma     : num [1:2] 5.87 5.87
 $ loglik    : num -1034
 $ posterior : num [1:272, 1:2] 1.02e-04 1.00 4.12e-03 9.67e-01 1.21e-06
...
  ..- attr(*, "dimnames")=List of 2
  .. ..$ : NULL
  .. ..$ : chr [1:2] "comp.1" "comp.2"
 $ all.loglik: num [1:10] -1085 -1051 -1037 -1034 -1034 ...
 $ restarts  : num 0
 $ ft        : chr "normalmixEM"
 - attr(*, "class")= chr "mixEM"
```

So, the estimate from EM is that about 36% of the eruptions are of Type 1, etc. Interesting, when
I tried it without my own initial guesses,

```
mixout <- normalmixEM(faithful$waiting,k=2)
```

the results were the same, so it seems to be a fairly stable model here.

By the way, since we are working with a hidden variable M here—in fact, we are merely postulating that it exists—how do we check this assumption? We'll return to this general idea of model fitting in Chapter 21.

19.6 Mean and Variance of Random Variables Having Mixture Distributions

Think of the random variables M and Y in the discussion following (19.3). Then EY is easy to find using the Law of Total Expectation:

$$EY = E[E(Y|M)] \tag{19.44}$$

Of course, evaluating this would require being able to compute E(Y | M), which is easy in some cases, not so easy in others.

Also, using the Law of Total Variance, we have that

$$Var(Y) = E[Var(Y|M)] + Var[E(Y|M)] \tag{19.45}$$

19.7 Example: Two Kinds of Batteries

Say a factory produces two kinds of batteries. One kind has lifetime that is exponentially distributed with mean 200 and the other's distribution is exponential with mean 500. Suppose 60% of the factory's production is of the former type, with 40% being of the latter type. Let's find the mean and variance of the lifetime Y of a randomly chosen battery.

Note that the distribution of Y is a mixture, in the sense of Section 19.2, in particular (19.3). Y here is a continuous random variable, referred to in that section as the "continuous outcome" case. In the notation there, let M be the battery type, with M being 0 or 1, for the 200-hour and 500-hour batteries, respectively. The conditional density of Y given M = 0 is exponential with mean 200, while given M = 1 it is exponential with mean 500. The unconditional density of Y is the mixture of these two exponential densities, as (19.3).

We want to find the <u>un</u>conditional mean and variance of Y.

Then

$$E(Y|M) = \begin{cases} 200, & w.p.\ 0.60 \\ 500, & w.p.\ 0.40 \end{cases} \tag{19.46}$$

and

$$Var(Y|M) = \begin{cases} 200^2, & w.p.\ 0.60 \\ 500^2, & w.p.\ 0.40 \end{cases} \tag{19.47}$$

(recalling that in the exponential family, variance is the square of the mean).

We can now use the formulas in Section 19.6. Let $Q_1 = E(Y|M)$ and $Q_2 = Var(Y|M)$. Then

$$EY = EQ_1 = 0.60 \times 200 + 0.40 \times 500 \tag{19.48}$$

and

$$
\begin{aligned}
Var(Y) &= E(Q_2) + Var(Q_1) & (19.49)\\
&= (0.60 \times 200^2 + 0.40 \times 500^2) + Var(Q_1) & (19.50)\\
&= (0.60 \times 200^2 + 0.40 \times 500^2) + E(Q_1^2) - (EQ_1)^2 & (19.51)\\
&= (0.60 \times 200^2 + 0.40 \times 500^2) + (0.60 \times 200^2 + 0.40 \times 500^2) & (19.52)\\
&\quad -(0.60 \times 200 + 0.40 \times 500)^2 & (19.53)\\
& & (19.54)
\end{aligned}
$$

19.8   Example: Overdispersion Models

A common model used in practice is that of **overdispersion**, in connection with Poisson models.

Recall the following about the Poisson distribution family:

(a) This family is often used to model counts.

(b) For any Poisson distribution, the variance equals the mean.

In some cases in which we are modeling count data, condition (b) is too constraining. We want a "Poisson-ish" distribution in which the variance is greater than the mean, called **overdispersion**.

One may then try to fit a mixture of several Poisson distributions, instead of a single one. This does induce overdispersion, as we will now see.

Suppose M can equal 1,2,...,k, with probabilities $p_1, ..., p_k$ that sum to 1. Say the distribution of Y given M = i is Poisson with parameter λ_i. Then Y has a mixture distribution. Our goal here will be to show that Y is overdispersed, i.e. has a large variance than mean.

By the Law of Total Expectation,

$$
\begin{aligned}
EY &= E[E(Y|M)] & (19.55)\\
&= E(\lambda_M) & (19.56)\\
&= \sum_{i=1}^{k} p_i \lambda_i & (19.57)
\end{aligned}
$$

Note that in the above, the expression λ_M is a random variable, since its subscript M is random. Indeed, it is a function of M, so Equation (3.36) then applies, yielding the final equation. The random variable λ_M takes on the values $\lambda_1, ..., \lambda_k$ with probabilities $p_1, ..., p_k$, hence that final sum.

The corresponding formula for variance, (19.42), can be used to derive Var(Y).

$$
\begin{aligned}
Var(Y) &= E[Var(Y|M)] + Var[E(Y|M)] & (19.58)\\
&= E(\lambda_M) + Var(\lambda_M) & (19.59)\\
&= EY + Var(\lambda_M) & (19.60)\\
& & (19.61)
\end{aligned}
$$

Did you notice that this last equation achieves our goal of showing overdispersed? Since

$$Var(\lambda_M) > 0 \qquad (19.62)$$

we have that

$$Var(Y) > E(\lambda_M) = EY \qquad (19.63)$$

by (19.56).

But let's see just how much greater the variance is than the mean. The second term in (19.60) is evaluated the same way as in (19.56): This is the variance of a random variable that takes on the

values $\lambda_1, ..., \lambda_k$ with probabilities $p_1, ..., p_k$, which is

$$\sum_{i=1}^{k} p_i(\lambda_i - \overline{\lambda})^2 \tag{19.64}$$

where

$$\overline{\lambda} = E\lambda_M = \sum_{i=1}^{k} p_i\lambda_i \tag{19.65}$$

Thus

$$EY = \overline{\lambda} \tag{19.66}$$

and

$$Var(Y) = \overline{\lambda} + \sum_{i=1}^{k} p_i(\lambda_i - \overline{\lambda})^2 \tag{19.67}$$

So, as long as the λ_i are not equal, we have

$$Var(Y) > EY \tag{19.68}$$

in this Poisson mixture model, in contrast to the single-Poisson case in which Var(Y) = EY. You can now see why the Poisson mixture model is called an overdispersion model.

So, if one has count data in which the variance is greater than the mean, one might try using this model.

In mixing the Poissons, there is no need to restrict to discrete M. In fact, it is not hard to derive the fact that if X has a gamma distribution with parameters r and p/(1-p) for some $0 < p < 1$, and Y given X has a Poisson distribution with mean X, then the resulting Y neatly turns out to have a negative binomial distribution.

19.9   Example: Hidden Markov Models

The Hidden Markov Model to be introduced in Section **??** fits our definition of a mixture distribution, with M being the mixing variable.

According, the machinery of mixing models applies. For instance, we can use the EM algorithm to calculate various quantities.

19.10 Vector Space Interpretations (for the mathematically adventurous only)

The abstract vector space notion in linear algebra has many applications to statistics. We develop some of that material in this section.

Let $\mathcal{V}$ be the set of all such random variables having finite variance and mean 0. We can set up $\mathcal{V}$ as a vector space. For that, we need to define a sum and a scalar product. Define the sum of any two vectors X and Y to be the random variable X+Y. For any constant c, the vector cX is the random variable cX. Note that $\mathcal{V}$ is closed under these operations, as it must be: If X and Y both have mean 0, then X+Y does too, and so on.

Define an inner product on this space:

$$(X, Y) = Cov(X, Y) = E(XY) \tag{19.69}$$

(Recall that Cov(X,Y) = E(XY) - EX EY, and that we are working with random variables that have mean 0.) Thus the norm of a vector X is

$$||X|| = (X, X)^{0.5} = \sqrt{E(X^2)} = \sqrt{Var(X)} \tag{19.70}$$

again since E(X) = 0.

19.10.1 Properties of Correlation

The famous Cauchy-Schwarz Inequality for inner products says,

$$|(X, Y)| \leq ||X|| \, ||Y|| \tag{19.71}$$

Applying that here, we have

$$|\rho(X, Y)| \leq 1 \tag{19.72}$$

So, vector space theory tells us that correlations are bounded between -1 and 1.

Also, the Cauchy-Schwarz Inequality yields equality if and only if one vector is a scalar multiple of the other, i.e. Y = cX for some c. When we then translate this to random variables of nonzero means, we get Y = cX + d.

In other words, the correlation between two random variables is between -1 and 1, with equality if and only if one is an exact linear function of the other.

19.10.2   Conditional Expectation As a Projection

For a random variable X in $\mathcal{V}$, let $\mathcal{W}$ denote the subspace of $\mathcal{V}$ consisting of all functions h(X) with mean 0 and finite variance. (Again, note that this subspace is indeed closed under vector addition and scalar multiplication.)

Now consider any Y in $\mathcal{V}$. Recall that the *projection* of Y onto $\mathcal{W}$ is the closest vector T in $\mathcal{W}$ to Y, i.e. T minimizes $||Y - T||$. That latter quantity is

$$\left(E[(Y - T)^2] \right)^{0.5} \tag{19.73}$$

To find the minimizing T, consider first the minimization of

$$E[(S - c)^2] \tag{19.74}$$

with respect to constant c for some random variable S. We already solved this problem back in Section 3.66. The minimizing value is c = ES.

Getting back to (19.73), use the Law of Total Expectation to write

$$E[(Y - T)^2] = E\left(E[(Y - T)^2 | X] \right) \tag{19.75}$$

From what we learned with (19.74), applied to the conditional (i.e. inner) expectation in (19.75), we see that the T which minimizes (19.75) is T = E(Y|X).

In other words, the conditional mean is a projection! Nice, but is this useful in any way? The answer is yes, in the sense that it guides the intuition. All this is related to issues of statistical prediction—here we would be predicting Y from X—and the geometry here can really guide our insight. This is not very evident without getting deeply into the prediction issue, but let's explore some of the implications of the geometry.

For example, a projection is perpendicular to the line connecting the projection to the original

vector. So

$$0 = (E(Y|X), Y - E(Y|X)) = Cov[E(Y|X), \ Y - E(Y|X)] \tag{19.76}$$

This says that the prediction E(Y|X) is uncorrelated with the prediction error, Y-E(Y|X). This in turn has statistical importance. Of course, (19.76) could have been derived directly, but the geometry of the vector space intepretation is what suggested we look at the quantity in the first place. Again, the point is that the vector space view can guide our intuition.

Simlarly, the Pythagorean Theorem holds, so

$$||Y||^2 = ||E(Y|X)||^2 + ||Y - E(Y|X)||^2 \tag{19.77}$$

which means that

$$Var(Y) = Var[E(Y|X)] + Var[Y - E(Y|X)] \tag{19.78}$$

Equation (19.78) is a common theme in linear models in statistics, the decomposition of variance.

There is an equivalent form that is useful as well, derived as follows from the second term in (19.78). Since

$$E[Y - E(Y|X)] = EY - E[E(Y|X)] = EY - EY = 0 \tag{19.79}$$

we have

$$
\begin{aligned}
Var[Y - E(Y|X)] &= E\left[(Y - E(Y|X))^2\right] \tag{19.80} \\
&= E\left[Y^2 - 2YE(Y|X) + (E(Y|X))^2\right] \tag{19.81}
\end{aligned}
$$

Now consider the middle term, $E[-2YE(Y|X)]$. Conditioning on X and using the Law of Total Expectation, we have

$$E[-2YE(Y|X)] = -2E\left[(E(Y|X))^2\right] \tag{19.82}$$

Then (19.80) becomes

$$
\begin{aligned}
Var[Y - E(Y|X)] &= E(Y^2) - E\left[(E(Y|X))^2\right] & (19.83)\\
&= E\left[E(Y^2|X)\right] - E\left[(E(Y|X))^2\right] & (19.84)\\
&= E\left(E(Y^2|X) - (E(Y|X))^2\right) & (19.85)\\
&= E\left[Var(Y|X)\right] & (19.86)
\end{aligned}
$$

the latter coming from our old friend, $Var(U) = E(U^2) - (EU)^2$, with U being Y here, under conditioning by X.

In other words, we have just derived another famous formula:

$$
Var(Y) = E[Var(Y|X)] + Var[E(Y|X)] \tag{19.87}
$$

19.11   Proof of the Law of Total Expectation

Let's prove (19.5) for the case in which W and Y take values only in the set $\{1,2,3,...\}$. Recall that if T is an integer-value random variable and we have some function h(), then L = h(T) is another random variable, and its expected value can be calculated as[5]

$$
E(L) = \sum_k h(k)P(T = k) \tag{19.88}
$$

In our case here, Q is a function of W, so we find its expectation from the distribution of W:

---

[5]This is sometimes called The Law of the Unconscious Statistician, by nasty probability theorists who look down on statisticians. Their point is that technically $EL = \sum_k kP(L = k)$, and that (19.88) must be proven, whereas the statisticians supposedly think it's a definition.

$$
\begin{aligned}
E(Q) &= \sum_{i=1}^{\infty} g(i) P(W = i) \\
&= \sum_{i=1}^{\infty} E(Y|W = i) P(W = i) \\
&= \sum_{i=1}^{\infty} \left[\sum_{j=1}^{\infty} j P(Y = j | W = i) \right] P(W = i) \\
&= \sum_{j=1}^{\infty} j \sum_{i=1}^{\infty} P(Y = j | W = i) P(W = i) \\
&= \sum_{j=1}^{\infty} j P(Y = j) \\
&= E(Y)
\end{aligned}
$$

In other words,

$$
E(Y) = E[E(Y|W)] \tag{19.89}
$$

Chapter 20

Histograms and Beyond: Nonparametric Density Estimation

Here we will be concerned with estimating density functions in settings in which we do not assume our distribution belongs to some parametric model.

Why is this important? Actually, you've been seeing density estimates for years—except that they've been called *histograms*—and hopefully you are convinced that histograms are indeed useful tools for data visualization. Simply reporting the (estimated) mean and variance of a distribution may not capture the nuances.

Moreover, density estimates can be used in *clasification* problems, in which we are trying to get the class of something, based on other variables. In a medical context, for instance, we may be trying predict whether a patient will develop a certain disease, knowing her current test results, family history and so on. We'll cover such problems in Chapter 23, but for now what is important to know is that the clasification problem can be expressed in terms of ratios of densities.

But guess what! Histograms are actually density estimates, as we will see. And we can do better than histograms, with more sophisticated density estimates.

20.1   Example: Baseball Player Data

In the data in Section 16.9, suppose we are looking at player's weights W, and we are interested in $f_W(182)$, the population density for weights, evaluated at 182. Our data is considered to be a sample from the populatin of all major league players, past, present and future,[1] and we will use

---

[1] Recall Section 15.4.

this data to develop an estimate,

$$\widehat{f}_W(142) \tag{20.1}$$

for the population quantity

$$f_W(182) \tag{20.2}$$

But how can we do this?

20.2   Basic Ideas in Density Estimation

Suppose we have a random sample $R_1, ..., R_n$ from a distribution F_R. How can we estimate f_R from the R_i?

Recall from the first day of your calculus course that for a function $g()$

$$g'(x) = \lim_{h \to 0} \frac{g(x+h) - g(x)}{h} \tag{20.3}$$

and thus for small h,

$$g'(x) \approx \frac{g(x+h) - g(x)}{h} \tag{20.4}$$

Now take $g()$ to be $F_R()$ for our random variable R above, so that

$$f_R(t) \approx \frac{F_R(t+h) - F_R(t)}{h} \tag{20.5}$$

We can alter this a little: Instead of looking at the interval $(t, t+h)$, we can use $(t-h, t+h)$:

$$f_R(t) \approx \frac{F_R(t+h) - F_R(t-h)}{2h} \tag{20.6}$$

That means that

$$f_R(t) \approx \frac{F_R(t+h) - F_R(t-h)}{2h} \tag{20.7}$$

$$= \frac{P(R \le t+h) - P(R \le t-h)}{2h} \tag{20.8}$$

$$= \frac{P(t-h < R \le t+h)}{2h} \tag{20.9}$$

if h is small. We can then form an estimate $\widehat{f}_R(t)$ by plugging in sample analogs in the right-hand side of (20.7):

$$\widehat{f}_R(t) = \frac{\#(t-h, t+h)/n}{2h} \tag{20.10}$$

$$= \frac{\#(t-h, t+h)}{2hn} \tag{20.11}$$

where the notation $\#(a, b)$ means the number of R_i in the interval (a,b).

There is an important issue of how to choose the value of h here, but let's postpone that for now. For the moment, let's take

$$h = \frac{\max_i R_i - \min_i R_i}{100} \tag{20.12}$$

i.e. take h to be 0.01 of the range of our data.

At this point, we'd then compute (20.11) at lots of different points t. Although it would seem that theoretically we must compute (20.11) at infinitely many such points, the graph of the function is actually a step function. Imagine t moving to the right, starting at $\min_i R_i$. The interval $(t-h, t+h)$ moves along with it. Whenever the interval moves enough to the right to either pick up a new R_i or lose one that it had had, (20.11) will change value, but not at any other time. So, we only need to evaluate the function at about $2n$ values of t.

20.3 Histograms

If for some reason we really want to save on computation, let's again say that we first break the interval $(\min_i R_i, \max_i R_i)$ into 100 subintervals of size h given by (20.12). We then compute (20.11) only at the midpoints of those intervals, and assume that the graph of $\widehat{f}_R(t)$ is approximately constant within each subinterval (true for small h). Do you know what we get from that? A

histogram! Yes, a histogram is a form of density estimation. (Usually a histogram merely displays counts. We do so here too, but we have scaled things so that the total area under the curve is 1, a property of densities.)

20.4   Kernel-Based Density Estimation

No matter what the interval width is, the histogram will consist of a bunch of rectangles, rather than a curve. We can get a smoother result if we used more sophisticated methods, one of which is called **kernel-based** density estimation. In base R, this is handled by the function **density()**.

For any particular value of t, $\widehat{f_R(t)}$ above depends only on the R_i that fall into that interval. If for instance some R_i is just barely outside the interval, it won't count. We'll now change that.

We need a set of weights, more precisely a weight function k, called the **kernel**. Any nonnegative function which integrates to 1—i.e. a density function in its own right—will work. Our estimator is then

$$\widehat{f_R}(t) = \frac{1}{nh} \sum_{i=1}^{n} k\left(\frac{t - R_i}{h}\right) \tag{20.13}$$

To make this idea concrete, take k to be the uniform density on (-1,1), which has the value 0.5 on (-1,1) and 0 elsewhere. Then (20.13) reduces to (20.11). Note how the parameter h, called the **bandwidth**, continues to control how far away from to t we wish to go for data points.

But as mentioned, what we really want is to include all data points, so we typically use a kernel with support on all of $(-\infty, \infty)$. In R's **density()** function, the default kernel is that of the N(0,1) density.

The bandwidth h controls how much smoothing we do; smaller values of h place heavier weights on data points near t and much lighter weights on the distant points.

There is no surefire way to choose a good bandwidth. A commonly used rule of thumb is

$$h = 1.06 \ s \ n^{-1/5} \tag{20.14}$$

where s is the sample standard deviation.

The default bandwidth in R is taken to the the standard deviation of k.

20.5   Example: Baseball Player Data

Some figures are plotted below for the baseball data, introduced in Section 16.9, for player weights, using functions in **ggplot2**:

- Figure 20.1 shows a histogram using the default number of bins, 30, programmed as follows

```
p <- ggplot(baseball)
p + geom_histogram(data=baseball,aes(x=Weight,y=..density..))
```

  As conceded in the documentation for **geom_histogram()**, the default tends to be not very good. This was the case here, with a very choppy figure.

- I then tried a binwidth of 10 pounds,

```
p + geom_histogram(data=baseball,aes(x=Weight,y=..density..),binwidth=10)
```

  This gave the much smoother plot in Figure 20.2.

- I then tried a kernel density estimate with the default bandwidth:

```
p + geom_density(aes(x=Weight))
```

  The result was similar to the histogram, but smoother, which is the goal.

- Finally, I superimposed on that last plot a plot for the catchers only (the latter in red):

```
p + geom_density(aes(x=Weight)) +
    geom_density(data=catch,aes(x=Weight,colour="red"))
```

  As seen in Figure 20.4, the catchers tend to be a bit heavier, and have less variation than the players in general.

20.6   More on Density Estimation in ggplot2

See Section C.6.

20.7   Bias, Variance and Aliasing

Nonparametric density estimation gives us an opportunity to apply the principles of bias from Chapter 18.
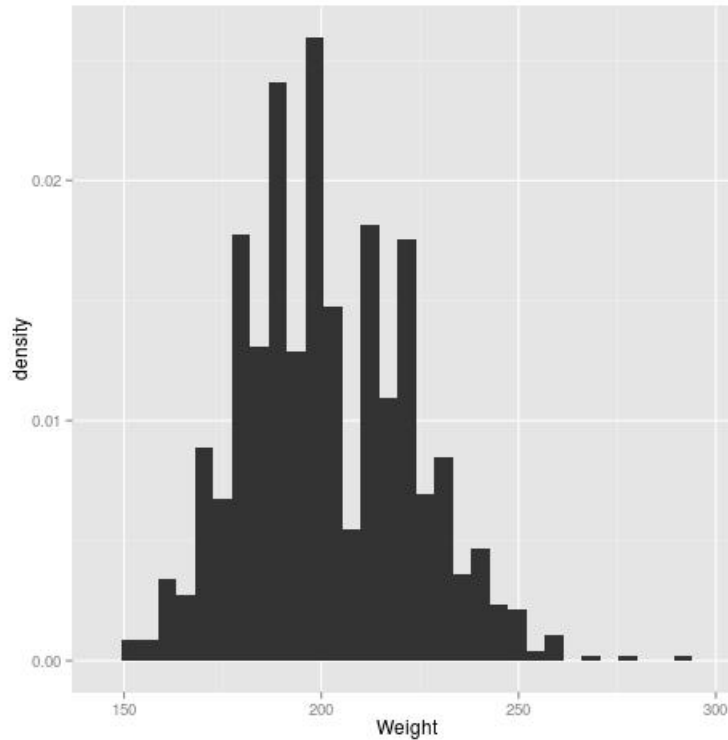
Figure 20.1: Histogram estimate, default binwidth

20.7.1   Bias vs. Variance

Recall from Section 18.2.5 that for an estimator $\widehat{\theta}$ of a population quantity θ we have that an overall measure of the accuracy of the estimator is

$$E[(\widehat{\theta} - \theta)^2] = bias(\widehat{\theta})^2 + Var(\widehat{\theta}) \tag{20.15}$$

In many cases there is a tradeoff between the bias and variance components here. We can have a smaller bias, for instance, but at the expense of increased variance. This is certainly the case with nonparametric density estimation.

As an illustration, suppose the true population density is $f_R(t) = 4t^3$ for t in (0,1), 0 elsewhere. Let's use (20.11):

$$\widehat{f}_R(t) = \frac{\#(t - h, t + h))}{2hn} \tag{20.16}$$
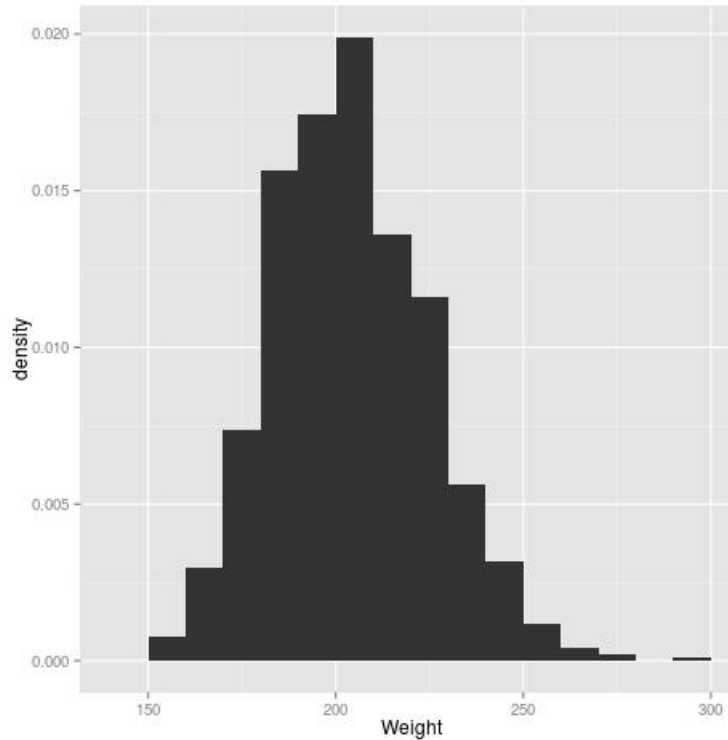
Figure 20.2: Histogram estimate, binwidth = 10

What is the bias? The numerator has a binomial distribution with n trials and success probability[2]

$$p = P(t - h < R < t + h) = \int_{t-h}^{t+h} 4u^3 \ du = (t + h)^4 - (t - h)^4 = 8t^3h + 8th^3 \tag{20.17}$$

By the binomial property, the numerator of (20.16) has expected value np, and thus

$$E[\widehat{f_R}(t)] = \frac{np}{2nh} = 4t^3 + 4th^2 \tag{20.18}$$

Subtracting $f_R(t)$, we have

$$bias[\widehat{f_R}(t)] = 4th^2 \tag{20.19}$$

---

[2]Note in the calculation here that it doesn't matter whether we write $\leq t + h$ or $< t + h$, since R is a continuous random variable.
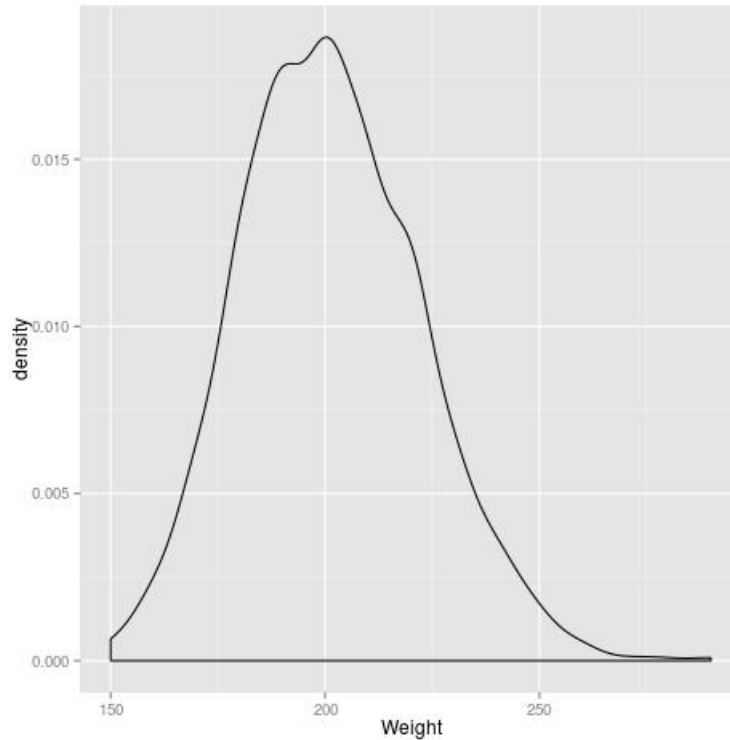
Figure 20.3: Kernel estimate, all players, default bandwidth

So, the smaller we set h, the smaller the bias, consistent with intuition.

How about the variance? Again using the binomial property, the variance of the numerator of (20.16) is np(1-p), so that

$$Var[[\widehat{f_R}(t)] = \frac{np(1-p)}{(2nh)^2} = \frac{np}{2nh} \cdot \frac{1-p}{2nh} = (4t^3 + 4th^2) \cdot \frac{1-p}{2nh} \tag{20.20}$$

This matches intuition too: On the one hand, for fixed h, the larger n is, the smaller the variance of our estimator—i.e. larger samples are better, as expected. On the other hand, the smaller we set h, the larger the variance, because with small h there just won't be many R_i falling into our interval (t-h,t+h).

So, you can really see the bias-variance tradeoff here, in terms of what value we choose for h.[3]

---

[3]You might ask about finding the h to minimize (20.15). This would not make sense in our present context, in which we are simply assuming a known density in order to explore the bias and variance issues here. In practice, of
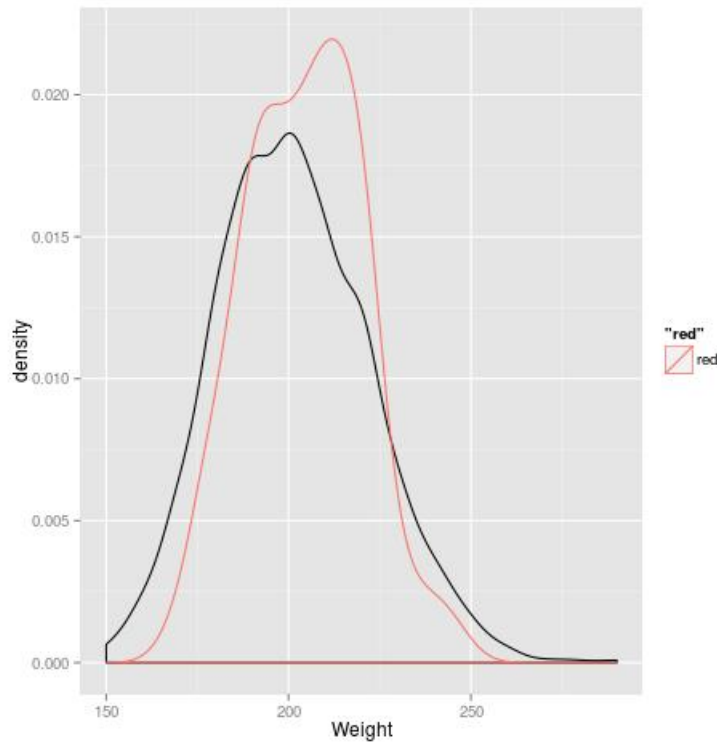
Figure 20.4: Kernel estimate, all players plus catchers (red), default bandwidth

20.7.2 Aliasing

There is another issue here to recognize: The integration in (20.17) tacitly assumed that $t - h > 0$ and $t + h < 1$. But suppose we are interested in $f_R(1)$. Then the upper limit in the integral in (20.17) will be 1, not t+h, which will approximately reduce the value of the integral by a factor of 2.

This results in strong bias near the endpoints of the support.[4] Let's illustrate this with the same density explored in our last section.

Using the general method in Section 7.11 for generating random numbers from a specified distribution, we have that this function will generate n random numbers from the density $4t^3$ on (0,1):

---

course, we don't know the density—that's why we are estimating it! However, some schemes ("plug-in" methods) for selecting h find a rough estimate of the density first, and then find the best h under the assumption that that estimate is correct.

[4]Recall that this term was defined in Section 7.5.
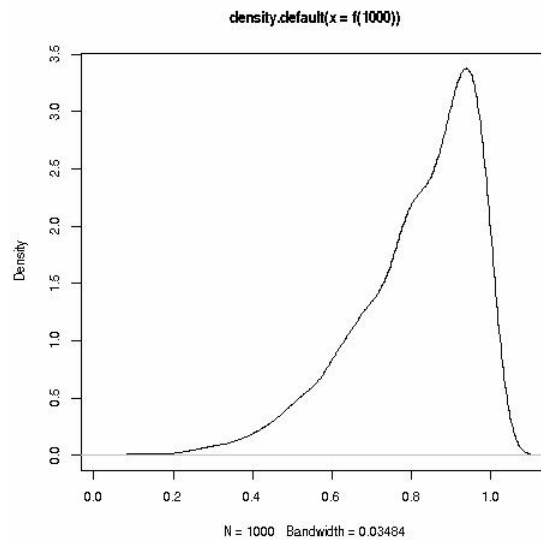
Figure 20.5: Example of Aliasing

```
f <- function(n) runif(n)^0.25
```

So, let's generate some numbers and plot the density estimate:

```
> plot(density(f(1000)))
```

The result is shown in Figure 20.5. Sure enough, the estimated density drops after about t = 0.9, instead of continuing to rise.

20.8   Nearest-Neighbor Methods

Consider (20.11) again. We count data points that are within a <u>fixed distance</u> from t; the <u>number</u> of such points will be random. With the nearest-neighbor approach, it's just the opposite: Now the <u>number</u> will be fixed, while the maximum <u>distance</u> from t will be random.

Specifically, at any point t we find the k nearest R_i to t, where k is chosen by the analyst just like h is selected in the kernel case. (And the method is are usually referred to as the *k-Nearest Neighbor* method, kNN.) The estimate is now
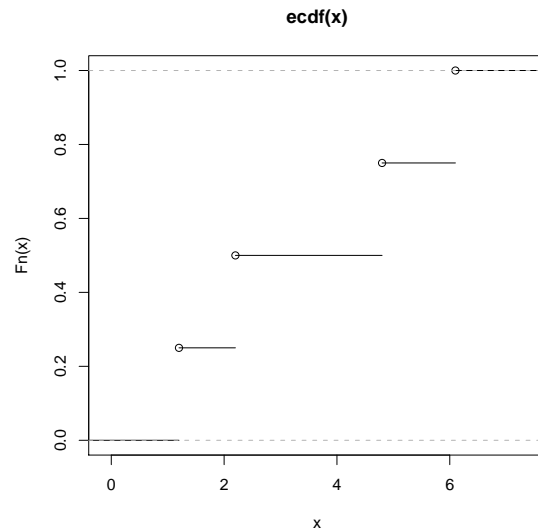
Figure 20.6: Empirical cdf, toy example

$$\widehat{f}_R(t) \quad = \quad \frac{k/n}{2\max_i |R_i - t|} \tag{20.21}$$

$$= \quad \frac{k}{2n\max_i |R_i - t|} \tag{20.22}$$

$$\tag{20.23}$$

20.9   Estimating a cdf

Let's introduce the notion of the **empirical distribution function** (ecdf), based on a sample $X_1, ..., X_n$. It is a sample estimate of a cdf, defined to be the proportion of X_i that are below t in the sample. Graphically, $\widehat{F}_X$ is a step function, with jumps at the values of the X_i.

As a small example, say n = 4 and our data are 4.8, 1.2, 2.2 and 6.1. We can plot the empirical cdf by calling R's **ecdf()** function:

```
> plot(ecdf(x))
```

The graph is in Figure 20.6. (In **ggplot2**, the function **stat_ecdf()** is similar.)

20.10 Hazard Function Estimation

In principle, estimating a hazard function from data should be a direct application of nonparametric density function methods. In (**??**) we would estimate the numerator with a kernel-based method, say, and the cdf in the denominator using the ecdf (Section 20.9).

However, the situation is complicated in that in many applications we have **censored data**, meaning that not all the data is available, due to an event not yet happening.

Say we measure lifetimes of batteries, and that at the time we collect our data, some of the batteries have died but some are still working. The problem is that we don't know the lifetimes of the latter. If one has run, say, for 212 hours so far, we know that it's lifetime will be at least 212, but we don't know the actual value yet.

This is an advanced topic, but a good starting point would be R's **muhaz** library in the CRAN repository. See the references in the documentation.

20.11 For Further Reading

To see an example of nonparametric density estimation applied to biology, see this paper by a UCD professor:

Kernel Methods in Line and Point Transect Sampling. *Biometrics*, Mack, Y. P. and P. X. Quang (1998). 54, 609-619.

Also see *All of Nonparametric Statistics*, Larry Wasserman Springer, 2007.

Chapter 21

Introduction to Model Building

All models are wrong, but some are useful.—George Box[1]

[Mathematical models] should be made as simple as possible, but not simpler.—Albert Einstein[2]

Beware of geeks bearing formulas.—Warrent Buffett, 2009, on the role of "quants" (Wall Street analysts who form probabilistic models for currency, bonds etc.) in the 2008 financial collapse.

The above quote by Box says it all. Consider for example the family of normal distributions. In real life, random variables are bounded—no person's height is negative or greater than 500 inches—and are inherently discrete, due to the finite precision of our measuring instruments. Thus, technically, no random variable in practice can have an exact normal distribution. Yet the assumption of normality pervades statistics, and has been enormously successful, provided one understands its approximate nature.

The situation is similar to that of physics. Paraphrasing Box, we might say "Tthe physical models used when engineers design an airplane wing are all wrong—but they are useful." We know that in many analyses of bodies in motion, we can neglect the effect of air resistance. But we also know that in some situations one must include that factor in our model.

So, the field of probability and statistics is fundamentally about *modeling*. The field is extremely useful, provided the user understands the modeling issues well. For this reason, this book contains this separate chapter on modeling issues.

---

[1]George Box (1919-) is a famous statistician, with several statistical procedures named after him.

[2]The reader is undoubtedly aware of Einstein's (1879-1955) famous theories of relativity, but may not know his connections to probability theory. His work on **Brownian motion**, which describes the path of a molecule as it is bombarded by others, is probabilistic in nature, and later developed into a major branch of probability theory. Einstein was also a pioneer in quantum mechanics, which is probabilistic as well. At one point, he doubted the validity of quantum theory, and made his famous remark, "God does not play dice with the universe."

21.1   "Desperate for Data"

Suppose we have the samples of men's and women's heights, $X_1, ..., X_n$ and $Y_1, ..., Y_n$. Assume for simplicity that the variance of height is the same for each gender, σ^2. The means of the two populations are designated by μ_1 and μ_2.

Say we wish to guess the height of a new person who we know to be a man but for whom we know nothing else. We do not see him, etc.

21.1.1   Known Distribution

Suppose for just a moment that we actually know the distribution of X, i.e. the *population* distribution of male heights. What would be the best constant g to use as our guess for a person about whom we know nothing other than gender?

Well, we might borrow from Section 18.2 and use mean squared error,

$$E[(g - X)^2] \tag{21.1}$$

as our criterion of goodness of guessing. But we already know what the best g is, from Section 3.66: The best g is μ_1. Our best guess for this unseen man's height is the mean height of all men in the population.

21.1.2   Estimated Mean

Of course, we don't know μ_1, but we can do the next-best thing, i.e. use an estimate of it from our sample.

The natural choice for that estimator would be

$$T_1 = \overline{X}, \tag{21.2}$$

the mean height of men in our sample.

But what if n is really small, say n = 5? That's awfully small. We may wish to consider adding the women's heights to our estimate, in order to get a larger sample. Then we would estimate μ_1 by

$$T_2 = \frac{\overline{X} + \overline{Y}}{2}, \tag{21.3}$$

It may at first seem obvious that T_1 is the better estimator. Women tend to be shorter, after all, so pooling the data from the two genders would induce a bias. On the other hand, we found in Section 18.2 that for any estimator,

$$\text{MSE} = \text{variance of the estimator} + \text{bias of the estimator}^2 \qquad (21.4)$$

In other words, *some amount of bias may be tolerable*, if it will buy us a subtantial reduction in variance. After all, women are not that much shorter than men, so the bias might not be too bad. Meanwhile, the pooled estimate should have lower variance, as it is based on 2n observations instead of n; (15.8) indicates that.

Before continuing, note first that T_2 is based on a simpler model than is T_1, as T_2 ignores gender. We thus refer to T_1 as being based on the more complex model.

Which one is better? The answer will need a criterion for goodness of estimation, which we will take to be mean squared error, MSE. So, the question becomes, which has the smaller MSE, T_1 or T_2? In other words:

Which is smaller, $E[(T_1 - \mu_1)^2]$ or $E[(T_2 - \mu_1)^2]$?

21.1.3   The Bias/Variance Tradeoff

We could calculate MSE from scratch, but it would probably be better to make use of the work we already went through, producing (18.80). This is especially true in that we know a lot about variance of sample means, and we will take this route.

So, let's find the biases of the two estimators.

- T_1

  T_1 is unbiased, from (15.8). So,

    bias of $T_1 = 0$

- T_2

$$
\begin{aligned}
E(T_2) &= E(0.5\overline{X} + 0.5\overline{Y}) \quad \text{(definition)} & (21.5) \\
&= 0.5E\overline{X} + 0.5E\overline{Y} \quad \text{(linearity of E())} & (21.6) \\
&= 0.5\mu_1 + 0.5\mu_2 \quad \text{[from (15.8)]} & (21.7)
\end{aligned}
$$

  So,

bias of $T_2 = (0.5\mu_1 + 0.5\mu_2) - \mu_1$

On the other hand, T_2 has a smaller variance than T_1:

- T_1

  Recalling (15.13), we have

  $$Var(T_1) = \frac{\sigma^2}{n} \tag{21.8}$$

- T_2

$$
\begin{aligned}
Var(T_2) &= Var(0.5\overline{X} + 0.5\overline{Y}) && \text{(21.9)} \\
&= 0.5^2 Var(\overline{X}) + 0.5^2 Var(\overline{Y}) \quad \text{(properties of Var())} && \text{(21.10)} \\
&= 2 \cdot 0.25 \cdot \frac{\sigma^2}{n} \quad \text{[from 15.13]} && \text{(21.11)} \\
&= \frac{\sigma^2}{2n} && \text{(21.12)}
\end{aligned}
$$

These findings are highly instructive. You might at first think that "of course" T_1 would be the better predictor than T_2. But for a small sample size, the smaller (actually 0) bias of T_1 is not enough to counteract its larger variance. T_2 is biased, yes, but it is based on double the sample size and thus has half the variance.

In light of (18.80), we see that T_1, the "true" predictor, may not necessarily be the better of the two predictors. Granted, it has no bias whereas T_2 does have a bias, but the latter has a smaller variance.

So, under what circumstances will T_1 be better than T_2? Let's answer this by using (18.79):

$$MSE(T_1) = \frac{\sigma^2}{n} + 0^2 = \frac{\sigma^2}{n} \tag{21.13}$$

$$MSE(T_2) = \frac{\sigma^2}{2n} + \left(\frac{\mu_1 + \mu_2}{2} - \mu_1\right)^2 = \frac{\sigma^2}{2n} + \left(\frac{\mu_2 - \mu_1}{2}\right)^2 \tag{21.14}$$

T_1 is a better predictor than T_2 if (21.13) is smaller than (21.14), which is true if

$$\left(\frac{\mu_2 - \mu_1}{2}\right)^2 > \frac{\sigma^2}{2n} \tag{21.15}$$

Granted, we don't know the values of the μ_1 and σ^2, so in a real situation, we won't really know whether to use T_1 or T_2. But the above analysis makes the point that under some circumstances, it really is better to pool the data in spite of bias.

21.1.4 Implications

So you can see that T_1 is better only if either

- n is large enough, or

- the difference in population mean heights between men and women is large enough, or

- there is not much variation within each population, e.g. most men have very similar heights

Since that third item, small within-population variance, is rarely seen, let's concentrate on the first two items. The big revelation here is that:

> A more complex model is more accurate than a simpler one only if either
>
> - we have enough data to support it, or
> - the complex model is sufficiently different from the simpler one

In height/gender example above, if n is too small, we are "desperate for data," and thus make use of the female data to augment our male data. Though women tend to be shorter than men, the bias that results from that augmentation is offset by the reduction in estimator variance that we get. But if n is large enough, the variance will be small in either model, so when we go to the more complex model, the advantage gained by reducing the bias will more than compensate for the increase in variance.

THIS IS AN ABSOLUTELY FUNDAMENTAL NOTION IN STATISTICS.

This was a very simple example, but you can see that in complex settings, fitting too rich a model can result in very high MSEs for the estimates. In essence, everything becomes noise. (Some people have cleverly coined the term **noise mining**, a play on the term **data mining**.) This is the famous **overfitting** problem.

In our unit on statistical relations, Chapter 22, we will show the results of a scary experiment done at the Wharton School, the University of Pennsylvania's business school. The researchers deliberately added fake data to a prediction equation, and standard statistical software identified it as "significant"! This is partly a problem with the word itself, as we saw in Section 17.11, but also a problem of using far too complex a model, as will be seen in that future unit.

Note that of course (21.15) contains several unknown population quantities. I derived it here merely to establish a <u>principle</u>, namely that a more complex model may perform more poorly under some circumstances.

It would be possible, though, to make (21.15) into a practical decision tool, by estimating the unknown quantities, e.g. replacing μ_1 by $\overline{X}$. This then creates possible problems with confidence intervals, whose derivation did not include this extra decision step. Such estimators, termed **adaptive**, are beyond the scope of this book.

21.2   Assessing "Goodness of Fit" of a Model

Our example in Section 18.1.4 concerned how to estimate the parameters of a gamma distribution, given a sample from the distribution. But that assumed that we had already decided that the gamma model was reasonable in our application. Here we will be concerned with how we might come to such decisions.

Assume we have a random sample $X_1, ..., X_n$ from a distribution having density f_X. To keep things simple, let's suppose we are considering an exponential model.

21.2.1   The Chi-Square Goodness of Fit Test

The classic way to do this would be the **Chi-Square Goodness of Fit Test**. We would set

$$H_0 : f_X \text{ is a member of the exponential parametric family} \qquad (21.16)$$

This would involve partitioning $(0, \infty)$ into k intervals (s_{i-1}, s_i) of our choice, and setting

$$N_i = \text{number of } X_i \text{ in } (s_{i-1}, s_i) \qquad (21.17)$$

We would then find the Maximum Likelihood Estimate (MLE) of λ, on the assumption that the distribution of X really is exponential. The MLE turns out to be the reciprocal of the sample mean, i.e.

$$\widehat{\lambda} = 1/\overline{X} \qquad (21.18)$$

This would be considered the parameter of the "best-fitting" exponential density for our data. We

would then estimate the probabilities

$$p_i = P[X \epsilon (s_{i-1}, s_i)] = e^{-\lambda s_{i-1}} - e^{-\lambda s_i}, \ i = 1, ..., k. \tag{21.19}$$

by

$$\widehat{p_i} = e^{-\widehat{\lambda} s_{i-1}} - e^{-\widehat{\lambda} s_i}, \ i = 1, ..., k. \tag{21.20}$$

Note that N_i has a binomial distribution, with n trials and success probability p_i. Using this, the expected value of EN_i is estimated to be

$$\nu_i = n(e^{-\widehat{\lambda} s_{i-1}} - e^{-\widehat{\lambda} s_i}), \ i = 1, ..., k. \tag{21.21}$$

Our test statistic would then be

$$Q = \sum_{i=1}^{k} \frac{(N_i - v_i)^2}{v_i} \tag{21.22}$$

where v_i is the expected value of N_i under the assumption of "exponentialness." It can be shown that Q is approximately chi-square distributed with k-2 degrees of freedom.[3] Note that only large values of Q should be suspicious, i.e. should lead us to reject H_0; if Q is small, it indicates a good fit. If Q were large enough to be a "rare event," say larger than $\chi_{0.95,k-2}$, we would decide NOT to use the exponential model; otherwise, we would use it.

Hopefully the reader has immediately recognized the problem here. If we have a large sample, this procedure will pounce on tiny deviations from the exponential distribution, and we would decide not to use the exponential model—even if those deviations were quite minor. Again, no model is 100% correct, and thus a goodness of fit test will eventually tell us not to use *any* model at all.

21.2.2 Kolmogorov-Smirnov Confidence Bands

Again consider the problem above, in which we were assessing the fit of a exponential model. In line with our major point that confidence intervals are far superior to hypothesis tests, we now present **Kolmogorov-Smirnov confidence bands**, which work as follows.

---

[3]We have k intervals, but the N_i must sum to n, so there are only k-1 free values. We then subtract one more degree of freedom, having estimated the parameter λ.

Since this method relies on cdfs, recall notion of the **empirical distribution function** (ecdf), Section 20.9. It is a sample estimate of a cdf, defined to be the proportion of X_i that are below t in the sample. Graphically, $\widehat{F}_X$ is a step function, with jumps at the values of the X_i.

What Kolmogorov-Smirnov does is form a **confidence band** around the empirical cdf of a sample. The basis for this is that the distribution of

$$M = \max_{-\infty < t\infty} |\widehat{F}_X(t) - F_X(t)| \tag{21.23}$$

is the same for all distributions having a density. This fact (whose proof is related to the general method for simulating random variables having a given density, in Section 7.11) tells us that, without knowing anything about the distribution of X, we can be sure that M has the same distribution. And it turns out that

$$F_M(1.358n^{-1/2}) \approx 0.95 \tag{21.24}$$

Define "upper" and "lower" functions

$$U(t) = \widehat{F}_X(t) + 1.358n^{-1/2}, \quad L(t) = \widehat{F}_X(t) - 1.358n^{-1/2} \tag{21.25}$$

So, what (21.23) and (21.24) tell us is

$$0.95 \approx P (\text{the curve } F_X \text{ is entirely between U and L}) \tag{21.26}$$

So, the pair of curves, (L(t), U(t)) is called a a **95% confidence band** for F_X.

Now suppose we wish to see how well, say, the gamma distribution family fits our application. If the band is very wide, we know we really don't have enough data to decide much about the distribution of X. But if the band is narrow but some member of the family is in the band or is close to it, we would probably decide that the model is a good one. Once again, we should NOT pounce on tiny deviations from the model.

Warning: The Kolmogorov-Smirnov procedure available in the R language performs only a hypothesis test, rather than forming a confidence band. In other words, it simply checks to see whether a member of the family falls within the band. This is not what we want, because we may be perfectly happy if a member is only *near* the band.

21.2.3 Less Formal Methods

Of course, another way, this one less formal, of assessing data for suitability for some model is to plot the data in a histogram or something of that nature. In this section, let's explore using ecdfs for this. We can plot the ecdf against a fitted model.

Let's try this with the baseball player data (Section 16.9). Here is the code:

```
> library(ggplot2)
> p <- ggplot(bb,aes(Age))
> p + stat_ecdf()  # plot ecdf
# define a function for the cdf of a fitted normal distribution
> ncdf <- function(t) pnorm(t,mean=mean(bb$Age),sd=sd(bb$Age))
> p + stat_ecdf() + stat_function(fun=ncdf)
```

The resulting plot is in Figure 21.1. The fitted curve, in red, is higher than the ecdf on the low end, i.e. age in the early 20s. In other words, modeling age as normally distributed in the player population overestimated the number of 20-somethings. It also overestimates the number in the late 30s. But overall, not a bad approximation.[4]

Of course, we could do the same with densities. But that would meaning choose bin width, bandwidth, number of nearest neighbors or something like that. **BUT THERE IS NO GOOD WAY TO CHOOSE THE BIN WIDTH OR h**. Even though there is a lot of theory to suggest how to choose the bin width or h, no method is foolproof. This is made even worse by the fact that the theory generally has a goal of minimizing *integrated* mean squared error,

$$\int_{-\infty}^{\infty} E\left[\left(\widehat{f}_R(t) - f_R(t)\right)^2\right] \, dt \tag{21.27}$$

rather than, say, the mean squared error at a particular point of interest, v:

$$E\left[\left(\widehat{f}_R(t) - f_R(t)\right)^2\right] \tag{21.28}$$

21.3 Robustness

Traditionally, the term *robust* in statistics has meant resilience to violations in assumptions. For example, in Section 16.7, we presented Student-t, a method for finding exact confidence intervals for

---

[4]Don't confuse this with the Central Limit Theorem. Here we are modeling the population itself has having a normal distribution, not sample means from it.
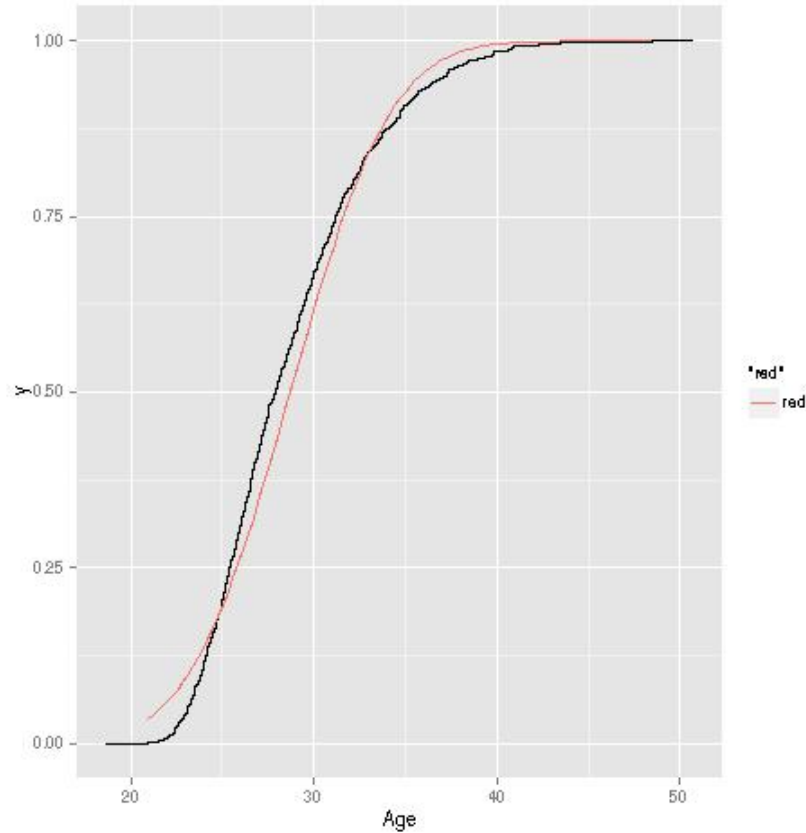
Figure 21.1: Ecdf and fitted cdf

means, assuming normally-distributed populations. But as noted at the outset of this chapter, no population in the real world has an exact normal distribution. The question at hand (which we will address below) is, does the Student-t method still give approximately correct results if the sample population is not normal? If so, we say that Student-t is **robust** to the normality assumption.

Later, there was quite a lot of interest among statisticians in estimation procedures that do well even if there are **outliers** in the data, i.e. erroneous observations that are in the fringes of the sample. Such procedures are said to be robust to outliers.

Our interest here is on robustness to assumptions. Let us first consider the Student-t example. As

discussed in Section 16.7, the main statistic here is

$$T = \frac{\bar{X} - \mu}{s/\sqrt{n}} \tag{21.29}$$

where μ is the population mean and s is the unbiased version of the sample variance:

$$s = \sqrt{\frac{\sum_{i=1}^{n}(X_i - \bar{X})^2}{n-1}} \tag{21.30}$$

The distribution of T, under the assumption of a normal population, has been tabulated, and tables for it appear in virtually every textbook on statistics. But what if the population is not normal, as is inevitably the case?

The answer is that it doesn't matter. For large n, even for samples having, say, n = 20, the distribution of T is close to N(0,1) by the Central Limit Theorem regardless of whether the population is normal.

By contrast, consider the classic procedure for performing hypothesis tests and forming confidence intervals for a population variance σ^2, which relies on the statistic

$$K = \frac{(n-1)s^2}{\sigma^2} \tag{21.31}$$

where again s^2 is the unbiased version of the sample variance. If the sampled population is normal, then K can be shown to have a chi-square distribution with n-1 degrees of freedom. This then sets up the tests or intervals. However, it has been shown that these procedures are not robust to the assumption of a normal population. See *The Analysis of Variance: Fixed, Random, and Mixed Models*, by Hardeo Sahai and Mohammed I. Ageel, Springer, 2000, and the earlier references they cite, especially the pioneering work of Scheffe'.

21.4 Real Populations and Conceptual Populations

In our example in Section 16.3.1, we were sampling from a real population. However, in many, probably most applications of statistics, either the population or the sampling is more conceptual.

Consider an experiment we will discuss in Section 22.2, in which we compare the programmability of three scripting languages. (You need not read ahead.) We divide our programmers into three groups, and assign each group to program in one of the languages. We then compare how long it took the three groups to finish writing and debugging the code, and so on.

We think of our programmers as being a random sample from the population of all programmers, but that is probably an idealization. We probably did NOT choose our programmers randomly; we just used whoever we had available. But we can think of them as a "random sample" from the rather conceptual "population" of all programmers who *might* work at this company.[5]

You can see from this that if one chooses to apply statistics carefully—which you absolutely should do—there sometimes are some knotty problems of interpretation to think about.

---

[5]You're probably wondering why we haven't discussed other factors, such as differing levels of experience among the programmers. This will be dealt with in our unit on regression analysis, Chapter 22.

Chapter 22

Linear Regression

In many senses, this chapter and several of the following ones form the real core of statistics, especially from a computer science point of view.

In this chapter and the next, we are interested in relations between variables, in two main senses:

- In **regression analysis**, we are interested in the relation of one variable with one or more others.

- In other kinds of analyses, such as **principal components analysis**, we are interested in relations among several variables, symmetrically, i.e. not having one variable play a special role.

Note carefully that *many types of methods that go by another name are actually regression methods.* Examples are the **classification problem**, **discriminant analysis**, **pattern recognition**, **machine learning** and so on. We'll return to this point in Chapter 23.

22.1 The Goals: Prediction and Description

Prediction is difficult, especially when it's about the future.—Yogi Berra[1]

Before beginning, it is important to understand the typical goals in regression analysis.

---

[1]Yogi Berra (1925-2015) is a former baseball player and manager, famous for his malapropisms, such as "When you reach a fork in the road, take it"; "That restaurant is so crowded that no one goes there anymore"; and "I never said half the things I really said."

- **Prediction:** Here we are trying to predict one variable from one or more others.

- **Description:** Here we wish to determine which of several variables have a greater effect on (or relation to) a given variable. An important special case is that in which we are interested in determining the effect of one predictor variable, **after the effects of the other predictors are removed.**

Denote the **predictor variables** by, $X^{(1)}, ..., X^{(r)}$, alluding to the Prediction goal. They are also called **independent variables** or **explanatory variables** (the latter term highlighting the Description goal) The variable to be predicted, Y, is often called the **response variable**, or the **dependent variable**. Note that one or more of the variables—whether the predictors or the response variable—may be indicator variables (Section 3.9). Another name for response variables of that type is **dummy variables**.

Methodology for this kind of setting is called **regression analysis**. If the response variable Y is an indicator variable, the values 1 and 0 to indicate class membership, we call this the **classification problem**. (If we have more than two classes, we need several Ys.)

In the above context, we are interested in the relation of a single variable Y with other variables $X^{(i)}$. But in some applications, we are interested in the more symmetric problem of relations *among* variables $X^{(i)}$ (with there being no Y). A typical tool for the case of continuous random variables is **principal components analysis**, and a popular one for the discrete case is **log-linear model**; both will be discussed later in this chapter.

22.2   Example Applications: Software Engineering, Networks, Text Mining

Example: As an aid in deciding which applicants to admit to a graduate program in computer science, we might try to predict Y, a faculty rating of a student after completion of his/her first year in the program, from $X^{(1)}$ = the student's CS GRE score, $X^{(2)}$ = the student's undergraduate GPA and various other variables. Here our goal would be Prediction, but educational researchers might do the same thing with the goal of Description. For an example of the latter, see Predicting Academic Performance in the School of Computing & Information Technology (SCIT), *35th ASEE/IEEE Frontiers in Education Conference*, by Paul Golding and Sophia McNamarah, 2005.

Example: In a paper, Estimation of Network Distances Using Off-line Measurements, *Computer Communications*, by Prasun Sinha, Danny Raz and Nidhan Choudhuri, 2006, the authors wanted to predict Y, the round-trip time (RTT) for packets in a network, using the predictor variables $X^{(1)}$ = geographical distance between the two nodes, $X^{(2)}$ = number of router-to-router hops, and other offline variables. The goal here was primarily Prediction.

Example: In a paper, Productivity Analysis of Object-Oriented Software Developed in a Commercial Environment, *Software—Practice and Experience*, by Thomas E. Potok, Mladen Vouk and Andy Rindos, 1999, the authors mainly had an Description goal: What impact, positive or negative, does the use of object-oriented programming have on programmer productivity? Here they predicted Y = number of person-months needed to complete the project, from $X^{(1)}$ = size of the project as measured in lines of code, $X^{(2)} = 1$ or 0 depending on whether an object-oriented or procedural approach was used, and other variables.

Example: Most **text mining** applications are classification problems. For example, the paper Untangling Text Data Mining, *Proceedings of ACL'99*, by Marti Hearst, 1999 cites, *inter alia*, an application in which the analysts wished to know what proportion of patents come from publicly funded research. They were using a patent database, which of course is far too huge to feasibly search by hand. That meant that they needed to be able to (reasonably reliably) predict Y = 1 or 0, according to whether the patent was publicly funded from a number of $X^{(i)}$, each of which was an indicator variable for a given key word, such as "NSF." They would then treat the predicted Y values as the real ones, and estimate their proportion from them.

Example: A major health insurance company wanted to have a tool to predict which of its members would be likely to need hospitalization in the next year. Here Y = 1 or 0, according to whether the patient turns out to be hospitalized, and the predictor variables were the members' demographics, previous medical history and so on. (Interestingly, rather hiring its own data scientist to do the analysis, the company put the problem on Kaggle, a site that holds predictive analytics competitions, `www.kaggle.com`.)

22.3 Adjusting for Covariates

The first statistical consulting engagement I ever worked involved something called *adjusting for covariates*. I was retained by the Kaiser hospital chain to investigate how heart attack patients fared at the various hospitals—did patients have a better chance to survive in some hospitals than in others? There were four hospitals of particular interest.

I could have simply computed raw survival rates, say the proportion of patients who survive for a month following a heart attack, and then used the methods of Section 16.4, for instance. This could have been misleading, though, because one of the four hospitals served a largely elderly population. A straight comparison of survival rates might then unfairly paint that particular hospital as giving lower quality of care than the others.

So, we want to somehow adjust for the effects of age. I did this by setting Y to 1 or 0, for survival, $X^{(1)}$ to age, and $X^{(2+i)}$ to be an indicator random variable for whether the patient was at hospital i, i = 1,2,3.[2]

---

[2]Note that there is no i = 4 case, since if the first three hospital variables are all 0, that already tells us that this

22.4   What Does "Relationship" Really Mean?

Consider the Davis city population example again. In addition to the random variable W for weight, let H denote the person's height. Suppose we are interested in exploring the relationship between height and weight.

As usual, we must first ask, **what does that really mean**? What do we mean by "relationship"? Clearly, there is no exact relationship; for instance, a person's weight is not an exact function of his/her height.

Effective use of the methods to be presented here requires an understanding of what exactly is meant by the term *relationship* in this context.

22.4.1   Precise Definition

Intuitively, we would guess that mean weight increases with height. To state this precisely, the key word in the previous sentence is *mean*.

Take Y to be the weight W and $X^{(1)}$ to be the height H, and define

$$m_{W;H}(t) = E(W|H = t) \tag{22.1}$$

This looks abstract, but it is just common-sense stuff. For example, $m_{W;H}(68)$ would be the mean weight of all people in the population of height 68 inches. The value of $m_{W;H}(t)$ varies with t, and we would expect that a graph of it would show an increasing trend with t, reflecting that taller people tend to be heavier.

We call $m_{W;H}$ the **regression function of W on H**. In general, $m_{Y;X}(t)$ means the mean of Y among all units in the population for which $X = t$.[3]

Note the word *population* in that last sentence. The function m() is a <u>population</u> function.

So we have:

> **Major Point 1:** When we talk about the *relationship* of one variable to one or more others, we are referring to the regression function, which expresses the mean of the first variable as a function of the others. The key word here is *mean*!

---

patient was at the fourth hospital.

[3]The word "regression" is an allusion to the famous comment of Sir Francis Galton in the late 1800s regarding "regression toward the mean." This referred to the fact that tall parents tend to have children who are less tall—closer to the mean—with a similar statement for short parents. The predictor variable here might be, say, the father's height F, with the response variable being, say, the son's height S. Galton was saying that E(S | F) < F.

i ↓, j →	0	1	2	3
0	0.0079	0.0952	0.1429	0.0317
1	0.0635	0.2857	0.1905	0.1587
2	0.0476	0.0952	0.0238	0.000

Table 22.1: Bivariate pmf for the Marble Problem

22.4.2 (Rather Artificial) Example: Marble Problem

Recall the marble selection example in Section 13.1: Suppose we have a bag containing two yellow marbles, three blue ones and four green ones. We choose four marbles from the bag at random, without replacement. Let Y and B denote the number of yellow and blue marbles that we get. Let's find $m_{Y;B}(2)$.

For convenience, Table 22.1 shows what we found before for P(Y = i and B = j).

Now keep in mind that since $m_{Y;B}(t)$ is the conditional mean of Y given B, we need to use conditional probabilities to compute it. For our example here of $m_{Y;B}(2)$, we need the probabilities $P(Y = k|B = 2)$. For instance,

$$P(Y = 1|B = 2) = \frac{p_{Y,B}(1,2)}{p_B(2)} \tag{22.2}$$

$$= \frac{0.1905}{0.1492 + 0.1905 + 0.0238} \tag{22.3}$$

$$= 0.5333 \tag{22.4}$$

The other conditional $P(Y = k|B = 2)$ are then found to be $0.1429/0.3572 = 0.4001$ for k = 0 and $0.0238/0.3572 = 0.0667$ for k = 2.

$$m_{Y;B}(2) = 0.4001 \cdot 0 + 0.5333 \cdot 1 + 0.0667 \cdot 2 = 0.667 \tag{22.5}$$

22.5   Estimating That Relationship from Sample Data

The marble example in the last section was rather artificial, in that the exact distribution of the variables was known (Table 22.1). In real applications, we don't know this distribution, and must estimate it from sample data.

As noted, $m_{W;H}(t)$ is a population function, dependent on population distributions. How can we estimate this function from sample data?

Toward that end, let's again suppose we have a random sample of 1000 people from Davis, with

$$(H_1, W_1), ..., (H_{1000}, W_{1000}) \tag{22.6}$$

being their heights and weights. We again wish to use this data to estimate population values, meaning the population regression function of W on H, $m_{W;H}(t)$. But the difference here is that we are estimating a whole function now, the whole curve $m_{W;H}(t)$. That means we are estimating infinitely many values, with one $m_{W;H}(t)$ value for each t.[4] How do we do this?

One approach would be as follows. Say we wish to find $\widehat{m}_{W;H}(t)$ (note the hat, for "estimate of"!) at t = 70.2. In other words, we wish to estimate the mean weight—in the population—among all people of height 70.2. What we could do is look at all the people in our sample who are within, say, 1.0 inch of 70.2, and calculate the average of all their weights. This would then be our $\widehat{m}_{W;H}(t)$.

22.5.1   Parametric Models for the Regression Function m()

There are many methods like the above (Chapter 24), but the traditional method is to choose a parametric model for the regression function. That way we estimate only a finite number of quantities instead of an infinite number. This would be good in light of Section 21.1.

Typically the parametric model chosen is linear, i.e. we assume that $m_{W;H}(t)$ is a linear function of t:

$$m_{W;H}(t) = ct + d \tag{22.7}$$

for some constants c and d. If this assumption is reasonable—meaning that though it may not be exactly true it is reasonably close—then it is a huge gain for us over a nonparametric model. Do you see why? Again, the answer is that instead of having to estimate an infinite number of quantities, we now must estimate only two quantities—the parameters c and d.

---

[4]Of course, the population of Davis is finite, but there is the conceptual population of all people who *could* live in Davis.

Equation (22.7) is thus called a **parametric** model of $m_{W;H}()$. The set of straight lines indexed by c and d is a two-parameter family, analogous to parametric families of distributions, such as the two-parametric gamma family; the difference, of course, is that in the gamma case we were modeling a density function, and here we are modeling a regression function.

Note that c and d are indeed population parameters in the same sense that, for instance, r and λ are parameters in the gamma distribution family. We must estimate c and d from our sample data.

So we have:

> **Major Point 2:** The function $m_{W;H}(t)$ is a population entity, so we must estimate it from our sample data. To do this, we have a choice of either assuming that $m_{W;H}(t)$ takes on some parametric form, or making no such assumption.
>
> If we opt for a parametric approach, the most common model is linear, i.e. (22.7). Again, the quantities c and d in (22.7) are population values, and as such, we must estimate them from the data.

22.5.2 Estimation in Parametric Regression Models

So, how can we estimate these population values c and d? We'll go into details in Section 22.10, but here is a preview:

Using the result on page 57, together with the principle of iterated expectation, (4.68) and (7.78), we can show that the minimum value of the quantity

$$E\left[(W - g(H))^2\right] \qquad (22.8)$$

overall all possible functions $g(H)$, is attained by setting

$$g(H) = m_{W;H}(H) \qquad (22.9)$$

In other words, $m_{W;H}(H)$ is the optimal predictor of W among all possible functions of H, in the sense of minimizing mean squared prediction error.[5]

Since we are assuming the model (22.7), this in turn means that:

---

[5]But if we wish to minimize the mean absolute prediction error, $E\left(|W - g(H)|\right)$, the best function turns out to be is $g(H) = \text{median}(W|H)$.

The quantity

$$E\left[(W - (uH + v))^2\right] \tag{22.10}$$

is minimized by setting u = c and v = d.

This then gives us a clue as to how to estimate c and d from our data, as follows.

If you recall, in earlier chapters we've often chosen estimators by using sample analogs, e.g. s^2 as an estimator of σ^2. Well, the sample analog of (22.10) is

$$\frac{1}{n}\sum_{i=1}^{n}[W_i - (uH_i + v)]^2 \tag{22.11}$$

Here (22.10) is the mean squared prediction error using u and v in the population, and (22.11) is the mean squared prediction error using u and v in our sample. Since u = c and v = d minimize (22.10), it is natural to estimate c and d by the u and v that minimize (22.11).

Using the "hat" notation common for estimators, we'll denote the u and v that minimize (22.11) by $\hat{c}$ and $\hat{d}$, respectively. These numbers are then the classical **least-squares estimators** of the population values c and d.

> **Major Point 3:** In statistical regression analysis, one uses a linear model as in (22.7), estimating the coefficients by minimizing (22.11).

We will elaborate on this in Section 22.10.

22.5.3   More on Parametric vs. Nonparametric Models

Suppose we're interested in the distribution of battery lifetimes, and we have a sample of them, say $B_1, ..., B_{100}$. We wish to estimate the density of lifetimes in the population of all batteries of this kind, $f_B(t)$.

We have two choices:

(a) We can simply plot a histogram of our data, which we found in Chapter 20 is actually a density estimator. We are estimating infinitely many population quantities, namely the heights of the curve $f_B(t)$ at infinitely many values of t.

(b) We could postulate a model for the distribution of battery lifetime, say using the gamma family (Section 7.6.4). Then we would estimate just two parameters, λ and r.

What are the pros and cons of (a) versus (b)? The approach (a) is nice, because we don't have to make any assumptions about the form of the curve $f_B(t)$; we just estimate it directly, with the histogram or other method from Chapter 20. But we are, in essence, using a finite amount of data to estimate an infinite values.

As to (b), it requires us to estimate only two parameters, which is nice. Also, having a nice, compact parametric form for our estimate is appealing. But we have the problem of having to make an assumption about the form of the model. We then have to see how well the model fits the data, say using the methods in Chapter 21. If it turns out not to fit well, we may try other models (e.g. from the Weibull family, not presented in this book).

The above situation is exactly parallel to what we are studying in the present chapter. The analogy here of estimating a density function is estimating a regression function. The analog of the histogram in (a) is the "average the people near a given height" method. The analog here of using a parametric family of densities, such as the gamma, is using a parametric family of straight lines. And the analog of comparing several candidate parametric density models is to compare several regression models, e.g. adding quadratic or cubic terms (t^2, t^3) for height in (22.7). (See Section 22.18.3 for reading on model assessment methods.)

Most statistical analysts prefer parameteric models, but nonparametric approaches are becoming increasingly popular.

22.6 Example: Baseball Data

Let's do a regression analysis of weight against height in the baseball player data introduced in Section 16.9.

22.6.1 R Code

I ran R's **lm()** ("linear model") function to perform the regression analysis:

```
> summary(lm(players$Weight ~ players$Height))

Call:
lm(formula = players$Weight ~ players$Height)

Residuals:
```

```
    Min       1Q  Median       3Q      Max
-51.988  -13.147    1.218   11.694   70.012


Coefficients:
               Estimate Std. Error t value Pr(>|t|)
(Intercept)    -155.092     17.699  -8.763   <2e-16 ***
players$Height    4.841      0.240  20.168   <2e-16 ***
---
Signif. codes:  0    ***    0.001    **    0.01    *    0.05    .    0.1

Residual standard error: 17.78 on 1031 degrees of freedom
  (1 observation deleted due to missingness)
Multiple R-squared: 0.2829,      Adjusted R-squared: 0.2822
F-statistic: 406.7 on 1 and 1031 DF,  p-value: < 2.2e-16
```

This illustrates the **polymorphic** nature of R: The same function can be applied to different classes of objects. Here **summary()** is such a function; another common one is **plot()**. So, we can call **summary()** on an object of any class, at least, any one for which a **summary()** function has been written. In the above R output, we called **summary()** on an object of type **"lm"**; the R interpreter checked the class of our object, and then accordingly called **summary.lm()**. But it's convenient for us, since we ignore all that and simply call **summary()** no matter what our object is.

The call `lm(players$Weight ∼players$Height)` specified that my response and predictor variables were the Weight and Height columns in the **players** data frame.

Note: The variables here are specified in an R data frame. One can also specify via a matrix, which gives more flexibility. For example,

```
lm(y ∼ x[,c(2,3,7)])
```

to predict **y** from columns 2, 3 and 7 of **x**.

22.6.2   A Look through the Output

Next, note that **lm()** returns a lot of information (even more than shown above), all packed into an object of type **"lm"**.[6] By calling **summary()** on that object, I got some of the information. It gave me more than we'll cover for now, but the key is that it told me that the sample estimates of

---

[6]R class names are quoted.

c and d are

$$\hat{d} = -155.092 \tag{22.12}$$

$$\hat{c} = 4.841 \tag{22.13}$$

In other words, our estimate for the function giving mean weight in terms of height is

```
mean weight = -155.092 + 4.841 height
```

Do keep in mind that this is just an estimate, based on the sample data; it is not the population mean-weight-versus-height function. So for example, our *sample estimate* is that an extra inch in height corresponds on average to about 4.8 more pounds in weight.

We can form a confidence interval to make that point clear, and get an idea of how accurate our estimate is. The R output tells us that the standard error of $\hat{d}$ is 0.240. Making use of Section 16.5, we add and subtract 1.96 times this number to $\hat{d}$ to get our interval: (4.351,5.331). So, we are about 95% confident that the true slope, c, is in that interval.

Note the column of output labled "t value." This is again a Student-t test, with the p-value given in the last column, labeled "$Pr(> |t|)$." Let's discuss this. In the row of the summary above regarding the Height variable, for example, we are testing

$$H_0 : c = 0 \tag{22.14}$$

R is using a Student-t distribution for this, while we have been using the the N(0,1) distribution, based on the Central Limit Theorem approximation. For all but the smallest samples, the difference is negligible. Consider:

Using (17.6), we would test (22.14) by forming the quotient

$$\frac{4.841 - 0}{0.240} = 20.17 \tag{22.15}$$

This is essentially the same as the 20.168 we see in the above summary. In other words, don't worry that R uses the Student-t distribution while we use (17.6).

At any rate, 20.17 is way larger than 1.96, thus resulting in rejection of H_0. The p-value is then the area to the left of -20.17 and to the right of 20.17, which we could compute using **pnorm()**. But R has already done this for us, reporting that the p-value is 2×10^{-16}.

What about the **residuals**? Here we go back to the original (H_i, W_i) data with our slope and intercept estimates, and "predict" each W_i from the corresponding H_i. The residuals are the resulting prediction errors. In other words, the ith residual is

$$W_i - (\hat{d} + \hat{c}H_i) \tag{22.16}$$

You might wonder why we would try to predict the data that we already know! But the reason for doing this is to try to assess how well we can predict future cases, in which we know height but not weight. If we can "predict" well in our known data, maybe we'll do well later with unknown data. This will turn out to be somewhat overoptimistic, we'll see, but again, the residuals should be of at least *some* value in assessing the predictive ability of our model. So, the R output reports to us what the smallest and largest residual values were.

The R^2 values will be explained in Section 22.15.4.

Finally, the F-test is a significance test that c = d = 0. Since this book does not regard testing as very useful, this aspect will not be pursued here.

22.7   Multiple Regression: More Than One Predictor Variable

Note that X and t could be vector-valued. For instance, we could have Y be weight and have X be the pair

$$X = \left(X^{(1)}, X^{(2)}\right) = (H, A) = \text{(height, age)} \tag{22.17}$$

so as to study the relationship of weight with height and age. If we used a linear model, we would write for $t = (t_1, t_2)$,

$$m_{W;H,A}(t) = \beta_0 + \beta_1 t_1 + \beta_2 t_2 \tag{22.18}$$

In other words

$$\text{mean weight} = \beta_0 + \beta_1 \text{ height} + \beta_2 \text{ age} \tag{22.19}$$

Once again, keep in mind that (22.18) and (22.19) are models for the population. We assume that (22.18), (22.19) or whichever model we use is an exact representation of the relation in the population. And of course, our derivations below assume our model is correct.

(It is traditional to use the Greek letter β to name the coefficients in a linear regression model.)

So for instance $m_{W;H,A}(68, 37.2)$ would be the mean weight in the population of all people having height 68 and age 37.2.

In analogy with (22.11), we would estimate the β_i by minimizing

$$\frac{1}{n} \sum_{i=1}^{n} [W_i - (u + vH_i + wA_i)]^2 \tag{22.20}$$

with respect to u, v and w. The minimizing values would be denoted $\widehat{\beta}_0$, $\widehat{\beta}_1$ and $\widehat{\beta}_2$.

We might consider adding a third predictor, gender:

$$\text{mean weight} = \beta_0 + \beta_1 \text{ height} + \beta_2 \text{ age} + \beta_3 \text{ gender} \tag{22.21}$$

where **gender** is an indicator variable, 1 for male, 0 for female. Note that we would not have two gender variables, since knowledge of the value of one such variable would tell us for sure what the other one is. (It would also make a certain matrix noninvertible, as we'll discuss later.)

22.8 Example: Baseball Data (cont'd.)

So, let's regress weight against height and age:

```
> summary(lm(players$Weight ~ players$Height + players$Age))

Call:
lm(formula = players$Weight ~ players$Height + players$Age)

Residuals:
    Min      1Q  Median      3Q     Max
-50.794 -12.141  -0.304  10.737  74.206

Coefficients:
                Estimate Std. Error t value Pr(>|t|)
(Intercept)    -192.6564    17.8905 -10.769  < 2e-16 ***
players$Height    4.9746     0.2341  21.247  < 2e-16 ***
players$Age       0.9647     0.1249   7.722  2.7e-14 ***
---
Signif. codes:  0    ***    0.001    **    0.01    *    0.05    .    0.1
```

```
Residual standard error: 17.3 on 1030 degrees of freedom
  (1 observation deleted due to missingness)
Multiple R-squared: 0.3221,     Adjusted R-squared: 0.3208
F-statistic: 244.8 on 2 and 1030 DF,  p-value: < 2.2e-16
```

So, our regression function coefficient estimates are $\hat{\beta}_0 = -192.6564$, $\hat{\beta}_1 = 4.9746$ and $\hat{\beta}_2 = 0.9647$. For instance, we estimate from our sample data that 10 years' extra age results, on average, of a weight gain about about 9.6 pounds—for people of a given height. This last condition is very important.

22.9   Interaction Terms

Equation (22.18) implicitly says that, for instance, the effect of age on weight is the same at all height levels. In other words, the difference in mean weight between 30-year-olds and 40-year-olds is the same regardless of whether we are looking at tall people or short people. To see that, just plug 40 and 30 for age in (22.18), with the same number for height in both, and subtract; you get $10\beta_2$, an expression that has no height term.

That assumption is not a good one, since the weight gain in aging tends to be larger for tall people than for short ones. If we don't like this assumption, we can add an **interaction term** to (22.18), consisting of the product of the two original predictors. Our new predictor variable $X^{(3)}$ is equal to $X^{(1)}X^{(2)}$, and thus our regression function is

$$m_{W;H}(t) = \beta_0 + \beta_1 t_1 + \beta_2 t_2 + \beta_3 t_1 t_2 \tag{22.22}$$

If you perform the same subtraction described above, you'll see that this more complex model does not assume, as the old did, that the difference in mean weight between 30-year-olds and 40-year-olds is the same regardless of we are looking at tall people or short people.

Recall the study of object-oriented programming in Section 22.1. The authors there set $X^{(3)} = X^{(1)}X^{(2)}$. The reader should make sure to understand that without this term, we are basically saying that the effect (whether positive or negative) of using object-oriented programming is the same for any code size.

Though the idea of adding interaction terms to a regression model is tempting, it can easily get out of hand. If we have k basic predictor variables, then there are $\binom{k}{2}$ potential two-way interaction terms, $\binom{k}{3}$ three-way terms and so on. Unless we have a very large amount of data, we run a

big risk of overfitting (Section 22.15.1). And with so many interaction terms, the model would be difficult to interpret.

We can add even more interaction terms by introducing powers of variables, say the square of height in addition to height. Then (22.22) would become

$$m_{W;H}(t) = \beta_0 + \beta_1 t_1 + \beta_2 t_2 + \beta_3 t_1 t_2 + \beta_4 t_1^2 \tag{22.23}$$

This square is essentially the "interaction" of height with itself. If we believe the relation between weight and height is quadratic, this might be worthwhile, but again, this means more and more predictors.

So, we may have a decision to make here, as to whether to introduce interaction terms. For that matter, it may be the case that age is actually not that important, so we even might consider dropping that variable altogether. These questions will be pursued in Section 22.15.

22.10   Parametric Estimation of Linear Regression Functions

So, how did R compute those estimated regression coefficients? Let's take a look.

22.10.1   Meaning of "Linear"

Here we model $m_{Y;X}$ as a linear function of $X^{(1)}, ..., X^{(r)}$:

$$m_{Y;X}(t) = \beta_0 + \beta_1 t^{(1)} + ... + \beta_r t^{(r)} \tag{22.24}$$

Note that the term **linear regression** does NOT necessarily mean that the graph of the regression function is a straight line or a plane. We could, for instance, have one predictor variable set equal to the square of another, as in (22.23).

Instead, the word *linear* refers to the regression function being linear in the parameters. So, for instance, (22.23) is a linear model; if for example we multiple β_0, β_1 and β_2 by 8, then $m_{A;b}(s)$ is multiplied by 8.

A more literal look at the meaning of "linear" comes from the matrix formulation (**??**) below.

22.10.2   Random-X and Fixed-X Regression

Consider our earlier example of estimating the regression function of weight on height. To make things, simple, say we sample only 5 people, so our data is $(H_1, W_1), ..., (H_5, W_5)$. and we measure height to the nearest inch.

In our "notebook" view, each line of our notebook would have 5 heights and 5 weights. Since we would have a different set of 5 people on each line, in the H_1 column will generally have different values from line to line, though occasionally two consecutive lines will have the same value. H_1 is a random variable. We can regression analysis in this setting **random-X** regression.

We could, on the other hand, set up our sampling plan so that we sample one person each of heights 65, 67, 69, 71 and 73. These values would then stay the same from line to line. The H_1 column, for instance, would consist entirely of 65s. This is called **fixed-X regression**.

So, the probabilistic structure of the two settings is different. However, it turns out not to matter much, for the following reason.

Recall that the definition of the regression function, concerns the *conditional* distribution of W given H. So, our analysis below will revolve around that conditional distribution, in which case H becomes nonrandom anyway.

22.10.3   Point Estimates and Matrix Formulation

So, how do we estimate the β_i? Keep in mind that the β_i are population values, which we need to estimate them from our data. How do we do that? For instance, how did R compute the $\widehat{\beta_i}$ in Section 22.6? As previewed in Section 22.5, the usual method is least-squares. Here we will go into the details.

For concreteness, think of the baseball data, and let H_i, A_i and W_i denote the height, age and weight of the ith player in our sample, i = 1,2,...,1033. As in (22.11), the estimation methodology involves finding the values of u_i which minimize the sum of squared differences between the actual W values and their predicted values using the u_i:

$$\sum_{i=1}^{1033} [W_i - (u_0 + u_1 H_i + u_2 A_i)]^2 \qquad (22.25)$$

When we find the minimizing u_i, we will set our estimates for the population regression coefficients β_i in (22.24):

$$\widehat{\beta_0} = u_0 \qquad (22.26)$$

$$\widehat{\beta_1} = u_1 \tag{22.27}$$

$$\widehat{\beta_2} = u_2 \tag{22.28}$$

Obviously, this is a calculus problem. We set the partial derivatives of (22.25) with respect to the u_i to 0, giving use three linear equations in three unknowns, and then solve.

In linear algebra terms, define

$$V = \begin{pmatrix} W_1 \\ W_2 \\ \dots \\ W_{1033} \end{pmatrix}, \tag{22.29}$$

$$u = \begin{pmatrix} u_0 \\ u_1 \\ u_2 \end{pmatrix} \tag{22.30}$$

and

$$Q = \begin{pmatrix} 1 & H_1 & A_1 \\ 1 & H_2 & A_2 \\ \dots & & \\ 1 & H_{1033} & A_{1033} \end{pmatrix} \tag{22.31}$$

Then

$$E(V \mid Q) = Q\beta \tag{22.32}$$

To see this, look at the first player, of height 74 and age 22.99 (Section 16.9). We are modeling the mean weight in the population for all players of that height and weight as

$$\text{mean weight} = \beta_0 + \beta_1 \ 74 + \beta_2 \ 22.99 \tag{22.33}$$

The top row of Q will be (1,74,22.99), so the top row of $Q\beta$ will be $\beta_0 + \beta_1 \ 74 + \beta_2 \ 22.99$ — which exactly matches (22.33). Note the need for the 1s column in Q.

we can write (22.25) as

$$(V - Qu)'(V - Qu) \tag{22.34}$$

Whatever vector u minimizes (22.34), we set our estimated β vector $\widehat{\beta} = (\widehat{\beta_0}, \widehat{\beta_1}, \widehat{\beta_2})'$ to that u.

Then it can be shown that, after all the partial derivatives are taken and set to 0, the solution is

$$\hat{\beta} = (Q'Q)^{-1}Q'V \tag{22.35}$$

For the general case (22.24) with n observations (n = 1033 in the baseball data), the matrix Q has n rows and r+1 columns. Column i+1 has the sample data on predictor variable i.

Note that we are conditioning on Q in (22.32). This is the standard approach, especially since there is the case of nonrandom X. Thus we will later get conditional confidence intervals, which is fine. To avoid clutter, I will sometimes not show the conditioning explicitly, and thus for instance will write, for example, Cov(V) instead of Cov(V|Q).

It turns out that $\hat{\beta}$ is an unbiased estimate of β:[7]

$$
\begin{aligned}
E\hat{\beta} &= E[(Q'Q)^{-1}Q'V] \quad (22.35) & (22.36)\\
&= (Q'Q)^{-1}Q'EV \quad \text{(linearity of E())} & (22.37)\\
&= (Q'Q)^{-1}Q' \cdot Q\beta \quad \text{(??)} & (22.38)\\
&= \beta & (22.39)
\end{aligned}
$$

In some applications, we assume there is no constant term β_0 in (22.24). This means that our Q matrix no longer has the column of 1s on the left end, but everything else above is valid.

22.10.4   Approximate Confidence Intervals

As noted, R gives you standard errors for the estimated coefficients. Where do they come from?

As usual, we should not be satisfied with just point estimates, in this case the $\widehat{\beta_i}$. We need an indication of how accurate they are, so we need confidence intervals. In other words, we need to use the $\widehat{\beta_i}$ to form confidence intervals for the β_i.

---

[7]Note that here we are taking the expected value of a vector, as in Chapter 12.

For instance, recall the study on object-oriented programming in Section 22.1. The goal there was primarily Description, specifically assessing the impact of OOP. That impact is measured by β_2. Thus, we want to find a confidence interval for β_2.

Equation (22.35) shows that the $\widehat{\beta}_i$ are sums of the components of V, i.e. the W_j. So, the Central Limit Theorem implies that the $\widehat{\beta}_i$ are approximately normally distributed. That in turn means that, in order to form confidence intervals, we need standard errors for the β_i. How will we get them?

Note carefully that so far we have made NO assumptions other than (22.24). Now, though, we need to add an assumption:[8]

$$Var(Y|X = t) = \sigma^2 \tag{22.40}$$

for all t. Note that this and the independence of the sample observations (e.g. the various people sampled in the Davis height/weight example are independent of each other) implies that

$$Cov(V|Q) = \sigma^2 I \tag{22.41}$$

where I is the usual identiy matrix (1s on the diagonal, 0s off diagonal).

Be sure you understand what this means. In the Davis weights example, for instance, it means that the variance of weight among 72-inch tall people is the same as that for 65-inch-tall people. That is not quite true—the taller group has larger variance—but research into this has found that as long as the discrepancy is not too bad, violations of this assumption won't affect things much.

We can derive the covariance matrix of $\hat{\beta}$ as follows. Again to avoid clutter, let $B = (Q'Q)^{-1}$. A theorem from linear algebra says that Q'Q is symmetric and thus B is too. Another theorem says that for any conformable matrices U and V, then (UV)' = V'U'. Armed with that knowledge, here we go:

$$
\begin{aligned}
Cov(\hat{\beta}) &= Cov(BQ'V) \;((22.35)) & (22.42)\\
&= BQ'Cov(V)(BQ')' \;(12.54) & (22.43)\\
&= BQ'\sigma^2 I(BQ')' \;(22.41) & (22.44)\\
&= \sigma^2 BQ'QB \;\text{(lin. alg.)} & (22.45)\\
&= \sigma^2 (Q'Q)^{-1} \;\text{(def. of B)} & (22.46)
\end{aligned}
$$

---

[8]Actually, we could derive some usable, though messy,standard errors without this assumption.

Whew! That's a lot of work for you, if your linear algebra is rusty. But it's worth it, because (22.46) now gives us what we need for confidence intervals. Here's how:

First, we need to estimate σ^2. Recall first that for any random variable U, $Var(U) = E[(U-EU)^2]$, we have

$$
\begin{aligned}
\sigma^2 &= Var(Y|X = t) & (22.47)\\
&= Var(Y|X^{(1)} = t_1, ..., X^{(r)} = t_r) & (22.48)\\
&= E\left[\{Y - m_{Y;X}(t)\}^2\right] & (22.49)\\
&= E\left[(Y - \beta_0 - \beta_1 t_1 - ... - \beta_r t_r)^2\right] & (22.50)
\end{aligned}
$$

Thus, a natural estimate for σ^2 would be the sample analog, where we replace E() by averaging over our sample, and replace population quantities by sample estimates:

$$
s^2 = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{\beta}_0 - \hat{\beta}_1 X_i^{(1)} - ... - \hat{\beta}_r X_i^{(r)})^2 \qquad (22.51)
$$

As in Chapter 18, this estimate of σ^2 is biased, and classicly one divides by n-(r+1) instead of n. But again, it's not an issue unless r+1 is a substantial fraction of n, in which case you are overfitting and shouldn't be using a model with so large a value of r.

So, the estimated covariance matrix for $\hat{\beta}$ is

$$
\widehat{Cov}(\hat{\beta}) = s^2(Q'Q)^{-1} \qquad (22.52)
$$

The diagonal elements here are the squared standard errors (recall that the standard error of an estimator is its estimated standard deviation) of the β_i. (And the off-diagonal elements are the estimated covariances between the β_i.) Since the first standard errors you ever saw, in Section 16.5, included factors like $1/\sqrt{n}$, you might wonder why you don't see such a factor in (22.52).

The answer is that such a factor is essentially there, in the following sense. Q'Q consists of various sums of products of the X values, and the larger n is, then the larger the elements of Q'Q are. So, $(Q'Q)^{-1}$ already has something like a "1/n" factor in it.

R's **vcov()** function, applied to the output of **lm()** will give you (22.52) (subject to a bias correction factor that we'll discuss in Section 22.18.1, but that we'll dismiss as unimportant.

22.11 Example: Baseball Data (cont'd.)

Let us use **vcov()** to obtain the estimated covariance matrix of the vector $\widehat{\beta}$ for our baseball data.

```
> lmout <- lm(players$Weight ~ players$Height + players$Age)
> vcov(lmout)
                (Intercept) players$Height  players$Age
(Intercept)      320.0706223    -4.102047105 -0.607718793
players$Height    -4.1020471     0.054817211  0.002160128
players$Age       -0.6077188     0.002160128  0.015607390
```

The first command saved the output of **lm()** in a variable that we chose to name **lmout**; we then called **vcov()** on that object.

For instance, the estimated variance of $\widehat{\beta}_1$ is 0.054817211. Actually, we already knew this, because the standard error of $\widehat{\beta}_1$ was reported earlier to be 0.2341, and $0.2341^2 = 0.054817211$.

But now we can find more. Say we wish to compute a confidence interval for the population mean weight of players who are 72 inches tall and age 30. That quantity is equal to

$$\beta_0 + 72\beta_1 + 30\beta_2 = (1, 72, 30)\beta \tag{22.53}$$

which we will estimate by

$$(1, 72, 30)\widehat{\beta} \tag{22.54}$$

Thus, using (12.56), we have

$$\widehat{Var}(\widehat{\beta}_0 + 72\widehat{\beta}_1 + 30\widehat{\beta}_2) = (1, 72, 30)A \begin{pmatrix} 1 \\ 72 \\ 30 \end{pmatrix} \tag{22.55}$$

where A is the matrix in the R output above.

The square root of this quantity is the standard error of $\widehat{\beta}_0 + 72\widehat{\beta}_1 + 30\widehat{\beta}_2$. We add and subtract 1.96 times that square root to $\widehat{\beta}_0 + 72\widehat{\beta}_1 + 30\widehat{\beta}_2$, and then have an approximate 95% confidence interval for the population mean weight of players who are 72 inches tall and age 30.

22.12   Dummy Variables

Recall our example in Section 22.2 concerning a study of software engineer productivity. To review, the authors of the study predicted Y = number of person-months needed to complete the project, from $X^{(1)}$ = size of the project as measured in lines of code, $X^{(2)} = 1$ or 0 depending on whether an object-oriented or procedural approach was used, and other variables.

As mentioned at the time, $X^{(2)}$ is an indicator variable, often called a "dummy" variable in the regression context.

Let's generalize that a bit. Suppose we are comparing two different object-oriented languages, C++ and Java, as well as the procedural language C. Then we could change the definition of $X^{(2)}$ to have the value 1 for C++ and 0 for non-C++, and we could add another variable, $X^{(3)}$, which has the value 1 for Java and 0 for non-Java. Use of the C language would be implied by the situation $X^{(2)} = X^{(3)} = 0$.

Note that we do NOT want to represent Language by a single value having the values 0, 1 and 2, which would imply that C has, for instance, double the impact of Java.

22.13   Example: Baseball Data (cont'd.)

Let's now bring the Position variable into play. First, what is recorded for that variable?

```
> levels(players$Position)
[1] "Catcher"           "Designated_Hitter" "First_Baseman"
[4] "Outfielder"        "Relief_Pitcher"    "Second_Baseman"
[7] "Shortstop"         "Starting_Pitcher"  "Third_Baseman"
```

So, all the outfield positions have been simply labeled "Outfielder," though pitchers have been separated into starters and relievers.

Technically, this variable, **players$Position**, is an R **factor**. This is a fancy name for an integer vector with labels, such that the labels are normally displayed rather than the codes. So actually catchers are coded 1, designated hitters 2, first basemen 3 and so on, but in displaying the data frame, the labels are shown rather than the codes.

The designated hitters are rather problematic, as they only exist in the American League, not the National League. Let's restrict our analysis to the other players:

```
> nondh <- players[players$Position != "Designated_Hitter",]
> nrow(players)
[1] 1034
```

```
> nrow(nondh)
[1] 1016
```

This requires some deconstruction. The expression `players$Position !="Designated_Hitter"`
gives us a vector of True and False values. Then `players[players$Position !="Designated_Hitter",]`
consists of all rows of **players** corresponding to a True value. Result: We've deleted the designated
hitters, assigning the result to **nondh**. A comparison of numbers of rows show that there were only
18 designated hitters in the data set anyway.

Let's consolidate into four kinds of positions: infielders, outfielders, catchers and pitchers. First,
switch to numeric codes, in a vector we'll name **poscodes**:

```
> poscodes <- as.integer(nondh$Position)
> head(poscodes)
[1] 1 1 1 3 3 6
> head(nondh$Position)
[1] Catcher        Catcher        Catcher        First_Baseman  First_Baseman
[6] Second_Baseman
9 Levels: Catcher Designated_Hitter First_Baseman ... Third_Baseman
```

Now consolidate into three dummy variables:

```
> infld <- as.integer(poscodes==3 | poscodes==6 | poscodes==7 | poscodes==9)
> outfld <- as.integer(poscodes==4)
> pitcher <- as.integer(poscodes==5 | poscodes==8)
```

Again, remember that catchers are designated via the other three dummies being 0.

So, let's run the regression:

```
> summary(lm(nondh$Weight ~ nondh$Height + nondh$Age + infld + outfld + pitcher

Call:
lm(formula = nondh$Weight ~ nondh$Height + nondh$Age + infld +
    outfld + pitcher)

Residuals:
    Min      1Q  Median      3Q     Max
-49.669 -12.083  -0.386  10.410  75.081

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  -193.2557    19.0127 -10.165  < 2e-16 ***
nondh$Height    5.1075     0.2520  20.270  < 2e-16 ***
```

```
nondh$Age          0.8844       0.1251    7.068 2.93e-12 ***
infld             -7.7727       2.2917   -3.392 0.000722 ***
outfld            -6.1398       2.3169   -2.650 0.008175 **
pitcher           -8.3017       2.1481   -3.865 0.000118 ***
---
Signif. codes:  0     ***     0.001     **     0.01     *     0.05     .     0.1

Residual standard error: 17.1 on 1009 degrees of freedom
  (1 observation deleted due to missingness)
Multiple R-squared: 0.3286,      Adjusted R-squared: 0.3253
F-statistic: 98.76 on 5 and 1009 DF,  p-value: < 2.2e-16
```

The estimated coefficients for the position variables are all negative. For example, for a given height and age, pitchers are on average about 8.3 pounds lighter than catchers, while outfielders are about 6.1 pounds lighter than catchers.

What if we want to compare infielders and outfielders, say form a confidence interval for $\beta_3 - \beta_4$? Then we'd do a computation like (22.55), with a vector (0,0,0,1,-1,0) instead of (1,72,30).

22.14   What Does It All Mean?—Effects of Adding Predictors

Keep in mind the twin goals of regression analysis, Prediction and Description. In applications in which Description is the goal, we are keenly interested in the signs and magnitudes of the β_i,[9] especially their signs. We do need to be careful, just as we saw in Section 17.11; the sign of a coefficient usually won't be of much interest if the magnitude is near 0. Subject to that caution, discussion of regression results often centers on the sign of a coefficient: Is there a positive relationship between the response variable and a predictor, holding the other predictors constant?

That latter phrase, *holding the other predictors constant*, is key. Recall for example our example at the start of this chapter on a study of the effects of using the object-oriented programming paradigm. Does OOP help or hurt productivity? Since longer programs often take longer to write, the researchers wanted to correct for program length, so they used that as a predictor, in addition to a dummy variable for OOP. In other words, they wanted to know the impact of OOP on productivity, holding program length constant.

So, in studying a predictor variable, it may matter greatly which other predictors one is using. Let's examine the baseball data in this regard.

In Section 22.8, we added the age variable as our second predictor, height being the first. This resulted in the coefficient of height increasing from 4.84 to 4.97. This is not a large change, but

---

[9]As estimated from the $\widehat{\beta}_i$.

what does it tell us? It suggests that older players tend to be shorter. No, this doesn't mean the players shrink with age—shrinkage does occur among the elderly, but likely not here—but rather that other phenomena are at work. It could be, for instance, that shorter players tend to have longer careers. This in turn might be due to a situation in which certain positions whose players tend to be tall have shorter careers. All of this could be explored, say starting with calculating the correlation between height and age.[10]

To develop some intuition on this, consider the following artificial population of eight people:

gender	height	weight
male	66	150
male	70	165
male	70	175
male	70	185
female	66	120
female	66	130
female	66	140
female	70	155

Here is the weight-height relationship for men, i.e. the mean weight for each height group:

men:

height	mean weight
66	150
70	175

$$\beta_{height} = (175 - 150)/4 = 6.25 \tag{22.56}$$

women:

height	mean weight
66	130
70	155

$$\beta_{height} = (155 - 130)/4 = 6.25 \tag{22.57}$$

The coefficient of height is the same for both gender subpopulations.

---

[10]The R function **cor()** computes the correlation between its first two arguments if they are vectors.

But look what happens when we remove gender from the analysis:

all:

height	mean weight
66	135
70	170

$$\beta_{height} = (170 - 135)/4 = 8.75 \tag{22.58}$$

In other words, the beta coefficient for height is 8.75 if gender is not in the equation, but is only 6.25 if we add in gender. For a given height, men in this population tend to be heavier, and since the men tend to be taller, that inflated the height coefficient in the genderless analysis.

Returning to the baseball example, recall that in in Section 22.13, we added the position variables to height and age as predictors. The coefficient for height, which had increased when we added in the age variable, now increased further, while the coefficient for age decreased, compared to the results in Section 22.8. Those heavy catchers weren't separated out from the other players in our previous analysis, and now that we are separating them from the rest, the relationship of weight versus height and age is now clarified.

Such thinking was central to another baseball example, in *Mere Mortals: Retract This Article*, Gregory Matthews blog,

`http://statsinthewild.wordpress.com/2012/08/23/mere-mortals-retract-this-article/`.

There the author took exception to someone else's analysis that purported to show that professional baseball players have a higher mortality rate than do pro football players. This was counterintuitive, since football is more of a contact sport. It turned out that the original analysis had been misleading, as it did not use age as a predictor.

Clearly, the above considerations are absolutely crucial to effective use of regression analysis for the Description goal. This insight is key—don't do regression without it! And for the same reasons, whenever you read someone else's study, do so with a skeptical eye.

22.15   Model Selection

The issues raised in Chapter 21 become crucial in regression and classification problems. In the context of this chapter, we typically deal with models having large numbers of parameters. In regression analysis, we often have many predictor variables, and of course the number of parameters can become even larger if we add in interaction and polynomial terms.

The **model selection** problem concerns simplifying a given model to one with fewer parameters. There are two motivations for this:

- A central principle will be that simpler models are preferable, provided of course they fit the data well. Hence the Einstein quote that opens Chapter 21! Simpler models are often called **parsimonious**.

- A simpler model may actually predict new cases better than a complex one, due to the **overfitting** problem discussed below.

So, in this section we discuss methods of selecting which predictor variables (including powers and interactions) we will use.

22.15.1 The Overfitting Problem in Regression

Recall that in Section 22.9 we mentioned that we could add polynomial terms to a regression model. But you can see that if we carry this notion to its extreme, we get absurd results. If we fit a polynomial of degree 99 to our 100 points, we can make our fitted curve exactly pass through every point! This clearly would give us a meaningless, useless curve. We are simply fitting the noise.

Recall that we analyzed this problem in Section 21.1.4 in our chapter on modeling. There we noted an absolutely fundamental principle in statistics:

> In choosing between a simpler model and a more complex one, the latter is more accurate only if either
>
> - we have enough data to support it, or
> - the complex model is sufficiently different from the simpler one

This is extremely important in regression analysis, because we often have so many variables we can use, thus often can make highly complex models.

In the regression context, the phrase "we have enough data to support the model" means (in the parametric model case) we have enough data so that the confidence intervals for the β_i will be reasonably narrow. For fixed n, the more complex the model, the wider the resulting confidence intervals will tend to be.

If we use too many predictor variables,[11] our data is "diluted," by being "shared" by so many β_i. As a result, $Var(\widehat{\beta_i})$ will tend to be large, with big implications: Whether our goal is Prediction or Description, our estimates will be so poor that neither goal is achieved.

---

[11]In the ALOHA example above, b, b^2, b^3 and b^4 are separate predictors, even though they are of course correlated.

On the other hand, if some predictor variable is really important (i.e. its β_i is far from 0), then it may pay to include it, even though the confidence intervals might get somewhat wider.

The questions raised in turn by the above considerations, i.e. **How much** data is enough data?, and **How different** from 0 is "quite different"?, are addressed below in Section 22.15.4.

A detailed mathematical example of overfitting in regression is presented in my paper A Careful Look at the Use of Statistical Methodology in Data Mining (book chapter), by N. Matloff, in *Foundations of Data Mining and Granular Computing*, edited by T.Y. Lin, Wesley Chu and L. Matzlack, Springer-Verlag Lecture Notes in Computer Science, 2005.

22.15.2   Relation to the Bias-vs.-Variance Tradeoff

Above we mentioned that the overfitting issue can be viewed as due to the bias-vs.-variance tradeoff. The variance portion of this was explained above: As the number of predictors increases, $Var(\widehat{\beta}_i)$ will also tend to increase.

But the bias will decrease. To see this, suppose we have data on two predictors. Let Model I be the result of using just $X^{(1)}$, and Model II be the corresponding result using both $X^{(1)}$ and $X^{(2)}$. Then from the point of view of Model II, our Model I will be biased.

Specifically, omitting a predictor makes the conditional expected response, given all the other predictors AND this additional one, is not modeled correctly

22.15.3   Multicollinearity

In typical applications, the $X^{(i)}$ are correlated with each other, to various degrees. If the correlation is high—a condition termed **multicollinearity**—problems may occur.

Consider (22.35). Suppose one predictor variable were to be fully correlated with another. That would mean that the first is exactly equal to a linear function of the other, which would mean that in Q one column is an exact linear combination of the first column and another column. Then $(Q'Q)^{-1}$ would not exist.

Well, if one predictor is strongly (but not fully) correlated with another, $(Q'Q)^{-1}$ will exist, but it will be numerically unstable. Moreover, even without numeric roundoff errors, $(Q'Q)^{-1}$ would be very large, and thus (22.46) would be large, giving us large standard errors—not good!

Thus we have yet another reason to limit our set of predictor variables.

22.15.4 Methods for Predictor Variable Selection

So, we typically must discard some, maybe many, of our predictor variables. In the weight/height/age example, we may need to discard the age variable. In the ALOHA example, we might need to discard b^4 and even b^3. How do we make these decisions?

Note carefully that **this is an unsolved problem.** If anyone claims they have a foolproof way to do this, then they do not understand the problem in the first place. Entire books have been written on this subject, e.g. *Subset Selection in Regression*, by Alan Miller, pub. by Chapman and Hall, second ediiton 2002. In his preface to the second edition of the book, Miller laments that almost no progress had been made in the field since the first edition had been published, a dozen years earlier! The same statement could be made today.

Myriad different methods have been developed. but again, none of them is foolproof.

22.15.4.1 Hypothesis Testing

The most commonly used methods for variable selection use hypothesis testing in one form or another. Typically this takes the form

$$H_0 : \beta_i = 0 \tag{22.59}$$

In the context of (22.19), for instance, a decision as to whether to include age as one of our predictor variables would mean testing

$$H_0 : \beta_2 = 0 \tag{22.60}$$

If we reject H_0, then we use the age variable; otherwise we discard it.

This approach is extended in a method called *stepwise regression* (which actually should be called "stepwise variable selection." It comes in *forward* and *backward* varieties. In the former, one keeps adding more and more predictors to the model, until there are no remaining "significant" ones. At each step, one enters the variable that is most "significant," meaning the one with the smallest p-value. In the backward variation, one starts with all predictors, and removes one at each step.

I hope I've convinced the reader, in Sections 17.11 and 21.2.1, that using significance testing for variable selection is not a good idea. As usual, the hypothesis test is asking the wrong question. For instance, in the weight/height/age example, the test is asking whether β_2 is zero or not—yet we know it is not zero, before even looking at our data. *What we want to know is whether β_2 is far enough from 0 for age to give us better predictions of weight.* Those are two very, very different questions.

A very interesting example of overfitting using real data may be found in the paper, Honest Confidence Intervals for the Error Variance in Stepwise Regression, by Foster and Stine, `www-stat.wharton.upenn.edu/~stine/research/honests2.pdf`. The authors, of the University of Pennsylvania Wharton School, took real financial data and deliberately added a number of extra "predictors" that were in fact random noise, independent of the real data. They then tested the hypothesis (22.59). They found that each of the fake predictors was "significantly" related to Y! This illustrates both the dangers of hypothesis testing and the possible need for multiple inference procedures.[12] This problem has always been known by thinking statisticians, but the Wharton study certainly dramatized it.

22.15.4.2 Confidence Intervals

Well, then, what can be done instead? First, there is the same alternative to hypothesis testing that we discussed before—confidence intervals. If the interval is very wide, telling us that it would be nice to have more data. But if the lower bound of that interval is far from zero, say, it would look like the corresponding variable is worth using as a predictor.

On the other hand, suppose in the weight/height/age example our confidence interval for β_2 is (0.04,0.06). In other words, we estimate β_2 to be 0.05, with a margin of error of 0.01. The 0.01 is telling us that our sample size is good enough for an accurate assessment of the situation, but the interval's location—centered at 0.05–says, for instance, a 10-year difference in age only makes about half a pound difference in mean weight. In that situation age would be of almost no value in predicting weight.

An example of this using real data is given in Section 23.3.

22.15.4.3 Predictive Ability Indicators

Suppose you have several competing models, some using more predictors, some using fewer. If we had some measure of predictive power, we could decide to use whichever model has the maximum value of that measure. Here are some of the more commonly used methods of this type:

- One such measure is called *adjusted R-squared*. To explain it, we must discuss ordinary R^2 first.

   Let ρ denote the population correlation between actual Y and predicted Y, i.e. the correlation between Y and $m_{Y;X}(X)$, where X is the vector of predictor variables in our model. Then $|\rho|$ is a measure of the power of X to predict Y, but it is traditional to use ρ^2 instead.[13]

---

[12] They added so many predictors that r became greater than n. However, the problems they found would have been there to a large degree even if r were less than n but r/n was substantial.

[13] That quantity can be shown to be the proportion of variance of Y attributable to X.

R is then the *sample* correlation between the Y_i and the vectors X_i. The sample R^2 is then an estimate of ρ^2. However, the former is a **biased** estimate—over infinitely many samples, the long-run average value of R^2 is higher than ρ^2. And the worse the overfitting, the greater the bias. Indeed, if we have n-1 predictors and n observations, we get a perfect fit, with $R^2 = 1$, yet obviously that "perfection" is meaningless.

Adjusted R^2 is a tweaked version of R^2 with less bias. So, in deciding which of several models to use, we might choose the one with maximal adjusted R^2. Both measures are reported when one calls **summary()** on the output of **lm()**.

- The most popular alternative to hypothesis testing for variable selection today is probably **cross validation**. Here we split our data into a **training set**, which we use to estimate the β_i, and a **validation set**, in which we see how well our fitted model predicts new data, say in terms of average squared prediction error. We do this for several models, i.e. several sets of predictors, and choose the one which does best in the validation set. I like this method very much, though I often simply stick with confidence intervals.

- A method that enjoys some popularity in certain circles is the **Akaike Information Criterion** (AIC). It uses a formula, backed by some theoretical analysis, which creates a tradeoff between richness of the model and size of the standard errors of the $\hat{\beta}_i$. Here we choose the model with minimal AIC.

  The R statistical package includes a function **AIC()** for this, which is used by **step()** in the regression case.

22.15.4.4 The LASSO

Consider again Equation (22.25). Ordinarily, we finding the u_i that minimize this quantity. But the LASSO (Least Absolute Shrinkage and Selection Operator) minimizes

$$\sum_{i=1}^{1033} [W_i - (u_0 + u_1 H_i + u_2 A_i)]^2 + s \sum_{i=1}^{3} |u_i| \qquad (22.61)$$

for some value of s (generally chosen by cross-validation). The importance of this in our context here is that this method turns out to force some of the $\hat{u}_i$ to 0—in effect, becoming a variable selection procedure. The LASSO has many fans, though again see the Miller book why you ought to be more careful with it.

22.15.5   Rough Rules of Thumb

A rough rule of thumb is that one should have $r < \sqrt{n}$, where r is the number of predictors and n is the sample size.[14] This result is general, not just restricted to regression models.

Also, if the adjusted R^2 is close to the unadjusted value, this is some indication that you are not overfitting.

22.16   Prediction

As noted, regression analysis is motivated by prediction. This is true even if ones goal is Description. We pursue this point further here.

22.16.1   Height/Weight Age Example

Let's return to our weight/height/age example. We are informed of a certain person, of height 70.4 and age 24.8, but weight unknown. What should we predict his weight to be?

The intuitive answer (justified formally on page 389) is that we predict his weight to be the mean weight for his height/age group,

$$m_{W;H,A}(70.4, 24.8) \tag{22.62}$$

But that is a population value. Say we estimate the function $m_{W;H}$ using that data, yielding $\widehat{m}_{W;H}$. Then we could take as our prediction for the new person's weight

$$\widehat{m}_{W;H,A}(70.4, 24.8) \tag{22.63}$$

If our model is (22.18), then (22.63) is

$$\widehat{m}_{W;H}(t) = \widehat{\beta}_0 + \widehat{\beta}_1 70.4 + \widehat{\beta}_2 24.8 \tag{22.64}$$

where the $\widehat{\beta}_i$ are estimated from our data by least-squares.

---

[14] Asymptotic Behavior of Likelihood Methods for Exponential Families When the Number of Parameters Tends to Infinity, Stephen Portnoy, *Annals of Statistics*, 1988.

22.16.2   R's predict() Function

We can automate the prediction process in (22.64), which is handy when we are doing a lot of predictions. An important example of such a situation was seen in Section 22.15.4.3, with the idea of breaking our data into training and validation sets.

R's **predict()** function makes this much easier. It is actually a collection of functions, with the one corresponding to **lm()** being **predict.lm()**. We just call **predict()**, and R will sense which version to call.[15]

With the arguments used here, the call form is

```
predict(lmobj,newxmatrix)
```

where **lmobj** is an object returned from a call to **lm()**, and **newmatrix** is the matrix of predictor values from which we wish to predict Y. The return value will be the vector of predicted Y values.

22.17   Example: Turkish Teaching Evaluation Data

This data, again from the UCI Machine Learning Repository, consists of 5820 student evaluations of professors in Turkey.

22.17.1   The Data

There are 28 questions of the type agree/disagree, scale of 1 to 5. Here are the first few:

Q1: The semester course content, teaching method and evaluation system were provided at the start.

Q2: The course aims and objectives were clearly stated at the beginning of the period.

Q3: The course was worth the amount of credit assigned to it.

Q4: The course was taught according to the syllabus announced on the first day of class.

Q5: The class discussions, homework assignments, applications and studies were satisfactory.

Q6: The textbook and other courses resources were sufficient and up to date.

Q7: The course allowed field work, applications, laboratory, discussion and other studies.

---

[15]R's object orientation includes the notion of **generic functions**, where a single function, say **plot()**, actually transfers control to the proper class-specific version.

There are also several "miscellaneous" questions, e.g. concerning the difficulty of the class. I chose
to use just this one.

22.17.2   Data Prep

I used a text editor to remove quotation marks in the original file, making it easier to read in. I
then did

```
> turk <-
read.csv("~/Montreal/Data/TurkEvals/turkiye-student-evaluation.csv",header=T)
> names(turk)
 [1] "instr"       "class"        "nb.repeat"   "attendance" "difficulty"
 [6] "Q1"          "Q2"           "Q3"          "Q4"         "Q5"
[11] "Q6"          "Q7"           "Q8"          "Q9"         "Q10"
[16] "Q11"         "Q12"          "Q13"         "Q14"        "Q15"
[21] "Q16"         "Q17"          "Q18"         "Q19"        "Q20"
[26] "Q21"         "Q22"          "Q23"         "Q24"        "Q25"
[31] "Q26"         "Q27"          "Q28"
> turk <- turk[,c(6:33,5)]
```

In that last operation, I reordered the data, so that the new column numbers would reflect the
question numbers:

```
> head(turk)
  Q1 Q2 Q3 Q4 Q5 Q6 Q7 Q8 Q9 Q10 Q11 Q12 Q13 Q14 Q15 Q16 Q17 Q18 Q19 Q20
Q21
1  3  3  3  3  3  3  3  3  3   3   3   3   3   3   3   3   3   3
3
3
2  3  3  3  3  3  3  3  3  3   3   3   3   3   3   3   3   3   3
3
3
3  5  5  5  5  5  5  5  5  5   5   5   5   5   5   5   5   5   5
5
5
4  3  3  3  3  3  3  3  3  3   3   3   3   3   3   3   3   3   3
3
3
5  1  1  1  1  1  1  1  1  1   1   1   1   1   1   1   1   1   1
1
1
```

```
6    4    4    4    4    4    4    4    4    4    4    4    4    4    4    4    4    4    4    4
4
4
   Q22 Q23 Q24 Q25 Q26 Q27 Q28 difficulty
1    3    3    3    3    3    3    3            4
2    3    3    3    3    3    3    3            3
3    5    5    5    5    5    5    5            4
4    3    3    3    3    3    3    3            3
5    1    1    1    1    1    1    1            1
6    4    4    4    4    4    4    4            3
```

Let's also split the rows of the data into training and validation sets, as in Section 22.15.4.3, and fit the model to the training set:

```
nr <- nrow(turk)
train <- sample(1:nr,floor(0.8*nr),replace=F)
val <- setdiff(1:nr,train)
lmout <- lm(turk[train,9] ~ .,data=turk[train,c(1:8,10:29)])
```

22.17.3 Analysis

So, let's run a regression on the training set. Question 9, "Q9: I greatly enjoyed the class and was eager to actively participate during the lectures," is the closest one to an overall evaluation of an instructor. Let's predict the outcome of Q9 from the other variables, in order to understand what makes a popular teacher in Turkey. Here is part of the output:

```
> lmout <- lm(turk[train,9] ~ .,data=turk[train,c(1:8,10:29)])
> summary(lmout)
...
Coefficients:
               Estimate  Std. Error  t value  Pr(>|t|)
(Intercept)   1.639e-01   2.852e-02    5.748  9.61e-09 ***
Q1           -9.078e-05   1.373e-02   -0.007  0.994726
Q2            5.148e-02   1.762e-02    2.923  0.003489 **
Q3            6.134e-02   1.541e-02    3.980  7.00e-05 ***
Q4            7.941e-03   1.626e-02    0.488  0.625343
Q5           -3.858e-02   1.851e-02   -2.084  0.037179 *
Q6           -2.991e-02   1.690e-02   -1.770  0.076874 .
Q7            8.543e-02   1.898e-02    4.501  6.92e-06 ***
Q8            1.172e-01   1.767e-02    6.631  3.73e-11 ***
Q10           3.386e-01   1.973e-02   17.162   < 2e-16 ***
```

```
Q11              1.744e-01  1.528e-02  11.414   < 2e-16 ***
Q12              4.206e-02  1.524e-02   2.760 0.005795 **
Q13             -2.283e-02  2.090e-02  -1.092 0.274879
Q14              2.871e-02  2.329e-02   1.233 0.217664
Q15             -6.692e-02  2.164e-02  -3.093 0.001993 **
Q16              7.670e-02  2.007e-02   3.821 0.000135 ***
Q17              1.005e-01  1.716e-02   5.857 5.04e-09 ***
Q18             -3.766e-03  1.940e-02  -0.194 0.846072
Q19              2.268e-02  1.983e-02   1.143 0.252990
Q20             -4.538e-02  2.074e-02  -2.189 0.028676 *
Q21              1.022e-01  2.280e-02   4.484 7.52e-06 ***
Q22              5.248e-02  2.288e-02   2.294 0.021860 *
Q23             -8.160e-03  2.160e-02  -0.378 0.705668
Q24             -1.228e-01  1.924e-02  -6.380 1.95e-10 ***
Q25              7.248e-02  2.057e-02   3.523 0.000431 ***
Q26             -6.819e-03  1.775e-02  -0.384 0.700820
Q27             -6.771e-03  1.584e-02  -0.428 0.668958
Q28             -2.506e-02  1.782e-02  -1.407 0.159615
difficulty      -5.367e-03  6.151e-03  -0.873 0.382925
---
Signif. codes:  0    ***     0.001    **     0.01    *    0.05    .    0.1

Residual standard error: 0.556 on 4627 degrees of freedom
Multiple R-squared:  0.8061,    Adjusted R-squared:  0.8049
F-statistic: 686.8 on 28 and 4627 DF,  p-value: < 2.2e-16
```

Questions 10 ("Q10: My initial expectations about the course were met at the end of the period or year") had the largest coefficient by far, 0.3386.

In terms of significance testing, it was very highly significant, but since we have a large sample size, 5802, we should be careful not to automatically conclude that this is an important predictor. Let's take a closer look.

The intercept term, $\widehat{\beta}_0$, is 0.1639. This is considerably smaller than the coefficient for Q10; increments in Q10 are of size 1, and we see that this increment will make a sizeable impact on our overall estimated regression function.

So, let's try **predict()** on the validation set:

```
newy <- predict(lmout,newdata=turk[val,c(1:8,10:29)])
```

Just to make sure that **predict()** works as advertised, let's do a check. Here's is the predicted Y for the first observation in our validation set, calculated "by hand":

```
> newx1 <- turk[val[1],c(1:8,10:29)]
> newx1 <- as.numeric(newx1)  # was a data frame
> newx1 <- c(1,newx1)  # need to handle the intercept term
> betas <- lmout$coef
> betas %*% newx1
          [,1]
[1,]  3.976574
```

Here is what **predict()** tells us:

```
> newy[1]
       6
3.976574  # it checks out!
```

Now, let's see how accurate our predictions are on the new data:

```
> truey <- turk[val,9]
> mean(abs(newy - truey))
[1] 0.2820834
```

Not bad at all—on average, our preduction is off by about 0.3 point, on a scale of 5.

The basic point is to fit a model to one data set and then try it out on new, "fresh" data, which we've done above. But just for fun, let's go back and "predict" the original data set:

```
 predold <- predict(lmout,newdata=turk[train,c(1:8,10:29)])
> mean(abs(predold - turk[train,9]))
[1] 0.290844
```

Given the concern about overfitting brought up in Section 22.15.1, one might expect the mean absolute error to be smaller on the original data, but it turns out to actually be a bit larger. The is presumably due to sampling error, but the real issue here is that the mean absolute error did not decrease a lot. This is because our sampling size, 5802, is large enough to support the 28 predictor variables we are using. This is seen above, where the adjusted R^2, 0.8049, was almost the same as the unadjusted version, 0.8061.

22.18 What About the Assumptions?

We have made two assumptions in this chapter:

- Linearity of our model: (22.24). (But recall that this doesn't prevent us from including powers of variables etc.)

- Homogeneity of variance (termed **homoscedasticity**) of Y given the $X^{(i)}$: (22.40).[16]

The classical analysis makes one more assumption:

- The conditional distribution of Y given the $X^{(i)}$ is normal.

We discuss these points further in this section.

22.18.1   Exact Confidence Intervals and Tests

Note carefully that we have not assumed that Y, given X, is normally distributed. In the height/weight context, for example, such an assumption would mean that weights in a specific height subpopulation, say all people of height 70 inches, have a normal distribution. We have not needed this assumption, as wse have relied on the Central Limit Theorem to give us approximate normal distributions for the $\widehat{\beta}_i$, enabling confidence intervals and significance tests. This issue is similar to that of Section 16.7.

If we do make such a normality assumption, then we can get exact confidence intervals (which of course, only hold if we really do have an exact normal distribution in the population). This again uses Student-t distributions. In that analysis, s^2 has n-(r+1) in its denominator instead of our n, just as there was n-1 in the denominator for s^2 when we estimated a single population variance. The number of degrees of freedom in the Student-t distribution is likewise n-(r+1).

But as before, for even moderately large n, it doesn't matter. And for small n, the normal population assumption almost never holds, or literally never. Thus exact methods are overrated, in this author's opinion.

22.18.2   Is the Homoscedasticity Assumption Important?

What about the assumption (22.40), which we made and which the "exact" methods assume too? This assumption is seldom if ever exactly true in practice, but studies have shown that the analysis is **robust** to that assumption. This means that even with fairly substantial violation of the assumption, the confidence intervals work fairly well.

22.18.3   Regression Diagnostics

Researchers in regression analysis have devised some **diagnostic** methods, meaning methods to check the fit of a model, the validity of assumptions [e.g. (22.40)], search for data points that may

---

[16]We also assume that the observations are independent.

have an undue influence (and may actually be in error), and so on. The residuals tend to play a central role here.

For instance, to check a model such as (22.19), we could plot our residuals against our age values. Suppose the pattern is that the residuals tend to be negative for the very young or very old people in our sample (i.e. overpredicting), and positive for the ones in between (underpredicting). This may suggest trying a model quadratic in age.

The R package has tons of diagnostic methods. See for example *Linear Models with R*, Julian Faraway, Chapman and Hall, 2005, and *An R and S-Plus Companion to Applied Regression*, John Fox, Sage, 2002.

22.19 Case Studies

22.19.1 Example: Prediction of Network RTT

Recall the paper by Raz *et al*, introduced in Section 22.2. They wished to predict network round-trip travel time (RTT) from offline variables. Now that we know how regression analysis works, let's look at some details of that paper.

First, they checked for multicollinearity. one measure of that is the ratio of largest to smallest eigenvalue of the matrix of correlations among the predictors. A rule of thumb is that there are problems if this value is greater than 15, but they found it was only 2.44, so they did not worry about multicollinearity.

They took a *backwards stepwise* approach to predictor variable selection, meaning that they started with all the variables, and removed them one-by-one while monitoring a goodness-of-fit criterion. They chose AIC for the latter.

Their initial predictors were DIST, the geographic distance between source and destination node, HOPS, the number of network hops (router processing) and an online variable, AS, the number of **autonomous systems**—large network routing regions—a message goes through. They measured the latter using the network tool **traceroute**.

But AS was the first variable they ended up eliminating. They found that removing it increased AIC only slightly, from about 12.6 million to 12.9 million, and reduced R^2 only a bit, from 0.785 to 0.778. They decided that AS was expendable, especially since they were hoping to use only offline variables.

Based on a scatter plot of RTT versus DIST, they then decided to try adding a quadratic term in that variable. This increased R^2 substantially, to 0.877. So, the final prediction equation they settled on predicts RTT from a quadratic function of DIST and a linear term for HOPS.

22.19.2   Transformations

It is common in some fields, especially economics, to apply logarithm transformations to regression variables.[17]

One of the motivations for this is to deal with the homoscedasticity assumption: Say we have just one predictor variable, for simplicity. If $Var(Y|X = t)$ is increasing in t, it is hoped that $Var[ln(Y)|X = t]$ is more stable.

22.19.3   Example: OOP Study

Consider again the OOP study cited in Section 22.2. It was actually a bit different from our description above. Among other things, they took natural logarithms of the variables. The model was

$$\text{mean } Y = \beta_0 + \beta_1 X^{(1)} + \beta_2 X^{(2)} + \beta_3 X^{(1)} X^{(2)} \tag{22.65}$$

where now: Y is the log of Person Months (PM); $X^{(1)}$ is the log of KLOC, the number of thousands of lines of code; and $X^{(2)}$ is a dummy variable for OOP. The results were:

coef.	betahat	std.err.
β_0	4.37	0.23
β_1	0.49	0.07
β_2	0.56	1.57
β_3	-0.13	-1.34

Let's find the estimated difference in mean log completion time under OOP and using procedural language (former minus the latter), for 1000-line programs:

$$(4.37 + 0.49 \cdot 1 + 0.56 \cdot 1 - 0.13 \cdot 1 \cdot 1) - (4.37 + 0.49 \cdot 1 + 0.5 \cdot 0 - 0.13 \cdot 0 \cdot 0) = 0.92 \tag{22.66}$$

While it is not the case that the mean of the log is the log of the mean, those who use log transformations treat this as an approximation. The above computation would then be viewed as the difference between two logs, thus the log of a quotient. That quotient would then be exp(0.92) = 2.51. In other words, OOP takes much longer to write. However, the authors note that neither of the beta coefficients for OOP and KLOC × OOP was significantly different from 0 at the 0.05 level, and thus consider the whole thing a wash.

---

[17]I personally do not take this approach.

Chapter 23

Classification

In prediction problems, in the special case in which Y is an indicator variable, with the value 1 if the object is in a class and 0 if not, the regression problem is called the **classification problem**.[1]

We'll formalize this idea in Section 23.1, but first, here are some examples:

- A forest fire is now in progress. Will the fire reach a certain populated neighborhood? Here Y would be 1 if the fire reaches the neighborhood, 0 otherwise. The predictors might be wind direction, distance of the fire from the neighborhood, air temperature and humidity, and so on.

- Is a patient likely to develop diabetes? This problem has been studied by many researchers, e.g. Using Neural Networks To Predict the Onset of Diabetes Mellitus, Murali S. Shanker *J. Chem. Inf. Comput. Sci.*, 1996, 36 (1), pp 3541. A famous data set involves Pima Indian women, with Y being 1 or 0, depending on whether the patient does ultimately develop diabetes, and the predictors being the number of times pregnant, plasma glucose concentration, diastolic blood pressure, triceps skin fold thickness, serum insulin level, body mass index, diabetes pedigree function and age.

- Is a disk drive likely to fail soon? This has been studied for example in Machine Learning Methods for Predicting Failures in Hard Drives: A Multiple-Instance Application, by Joseph F. Murray, Gordon F. Hughes, and Kenneth Kreutz-Delgado, *Journal of Machine Learning Research* 6 (2005) 783-816. Y was 1 or 0, depending on whether the drive failed, and the predictors were temperature, number of read errors, and so on.

- An online service has many customers come and go. It would like to predict who is about to leave, so as to offer them a special deal for staying with this firm.

---

[1] The case of $c > 2$ classes will be treated in Section 23.4.

- Of course, a big application is character recognition, based on pixel data. This is different from the above examples, as there are more than two classes, many more. We'll return to this point soon.

In electrical engineering the classification is called **pattern recognition**, and the predictors are called **features**. In computer science the term **machine learning** usually refers to classification problems. Different terms, same concept.

23.1   Classification = Regression

All of the many machine learning algorithms, despite their complexity, really boil down to regression at their core. Here's why:

23.1.1   What Happens with Regression in the Case Y = 0,1?

As we have frequently noted the mean of any indicator random variable is the probability that the variable is equal to 1 (Section 3.9). Thus in the case in which our response variable Y takes on only the values 0 and 1, i.e. classification problems, the regression function reduces to

$$m_{Y;X}(t) = P(Y = 1|X = t) \tag{23.1}$$

(Remember that X and t are vector-valued.)

As a simple but handy example, suppose Y is gender (1 for male, 0 for female), $X^{(1)}$ is height and $X^{(2)}$ is weight, i.e. we are predicting a person's gender from the person's height and weight. Then for example, $m_{Y;X}(70, 150)$ is the probability that a person of height 70 inches and weight 150 pounds is a man. Note again that this probability is a population fraction, the fraction of men among all people of height 70 and weight 150 in our population.

Make a mental note of the optimal prediction rule, if we know the population regression function:

> Given X = t, the optimal prediction rule is to predict that Y = 1 if and only if $m_{Y;X}(t) >$ 0.5.

So, if we known a certain person is of height 70 and weight 150, our best guess for the person's gender is to predict the person is male if and only if $m_{Y;X}(70, 150) > 0.5$.[2]

The optimality makes intuitive sense, and is shown in Section 23.6.

---

[2]Things change in the multiclass case, though, as will be seen in Section 23.4.

23.2 Logistic Regression: a Common Parametric Model for the Regression Function in Classification Problems

Remember, we often try a parametric model for our regression function first, as it means we are estimating a finite number of quantities, instead of an infinite number. Probably the most commonly-used model is that of the **logistic function** (often called "logit"). Its r-predictor form is

$$m_{Y;X}(t) = P(Y = 1|X = t) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 t_1 + \ldots + \beta_r t_r)}} \qquad (23.2)$$

For instance, consider the patent example in Section 22.2. Under the logistic model, the population proportion of all patents that are publicly funded, among those that contain the word "NSF," do not contain "NIH," and make five claims would have the value

$$\frac{1}{1 + e^{-(\beta_0 + \beta_1 + 5\beta_3)}} \qquad (23.3)$$

23.2.1 The Logistic Model: Motivations

The logistic function itself,

$$\frac{1}{1 + e^{-u}} \qquad (23.4)$$

has values between 0 and 1, and is thus a good candidate for modeling a probability. Also, it is monotonic in u, making it further attractive, as in many classification problems we believe that $m_{Y;X}(t)$ should be monotonic in the predictor variables.

But there are additional reasons to use the logit model, as it includes many common parametric models for X. To see this, note that we can write, for vector-valued discrete X and t,

$$P(Y = 1 | X = t) \quad = \quad \frac{P(Y = 1 \text{ and } X = t)}{P(X = t)} \tag{23.5}$$

$$= \quad \frac{P(Y = 1)P(X = t | Y = 1)}{P(X = t)} \tag{23.6}$$

$$= \quad \frac{P(Y = 1)P(X = t | Y = 1)}{P(Y = 1)P(X = t | Y = 1) + P(Y = 0)P(X = t | Y = 0)} \tag{23.7}$$

$$= \quad \frac{1}{1 + \frac{(1-q)P(X=t|Y=0)}{qP(X=t|Y=1)}} \tag{23.8}$$

where $q = P(Y = 1)$ is the proportion of members of the population which have $Y = 1$. (Keep in mind that this probability is unconditional!!!! In the patent example, for instance, if say $q = 0.12$, then 12% of all patents in the patent population—without regard to words used, numbers of claims, etc.—are publicly funded.)

If X is a continuous random vector, then the analog of (23.8) is

$$P(Y = 1 | X = t) = \frac{1}{1 + \frac{(1-q)f_{X|Y=0}(t)}{qf_{X|Y=1}(t)}} \tag{23.9}$$

Now for simplicity, suppose X is scalar, i.e. r = 1. And suppose that, given Y, X has a normal distribution. In other words, within each class, Y is normally distributed. Suppose also that the two within-class variances of X are equal, with common value σ^2, but with means μ_0 and μ_1. Then

$$f_{X|Y=i}(t) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left[-0.5\left(\frac{t - \mu_i}{\sigma}\right)^2\right] \tag{23.10}$$

After doing some elementary but rather tedious algebra, (23.9) reduces to the logistic form

$$\frac{1}{1 + e^{-(\beta_0 + \beta_1 t)}} \tag{23.11}$$

where

$$\beta_0 = -\ln\left(\frac{1 - q}{q}\right) + \frac{\mu_0^2 - \mu_1^2}{2\sigma^2}, \tag{23.12}$$

and

$$\beta_1 = \frac{\mu_1 - \mu_0}{\sigma^2}, \tag{23.13}$$

In other words, if X is normally distributed in both classes, with the same variance but different means, then $m_{Y;X}()$ has the logistic form! And the same is true if X is multivariate normal in each class, with different mean vectors but equal covariance matrices. (The algebra is even more tedious here, but it does work out.) Given the central importance of the multivariate normal family—the word *central* here is a pun, alluding to the (multivariate) Central Limit Theorem—this makes the logit model even more useful.

If you retrace the derivation above, you will see that the logit model will hold for any within-class distributions for which

$$\ln \left(\frac{f_{X|Y=0}(t)}{f_{X|Y=1}(t)} \right) \tag{23.14}$$

(or its discrete analog) is linear in t. Well guess what—this condition is true for exponential distributions too! Work it out for yourself.

In fact, a number of famous distributions imply the logit model. So, logit is not only a good intuitive model, as discussed above, but in addition there are some good theoretical recommendations for it.

23.2.2 Esimation and Inference for Logit Coefficients

We fit a logit model in R using the **glm()** function, with the argument **family=binomial**. The function finds Maximum Likelihood Estimates (Section 18.1.3) of the β_i.[3]

The output gives standard errors for the $\widehat{\beta}_i$ as in the linear model case. This enables the formation of confidence intervals and significance tests on individual $\widehat{\beta}_i$. For inference on linear combinations of the $\widehat{\beta}_i$, use the **vcov()** function as in the linear model case.

---

[3]As in the case of linear regression, estimation and inference are done conditionally on the values of the predictor variables X_i.

23.3   Example: Forest Cover Data

Let's look again at the forest cover data we saw in Section 16.6.4.[4] Recall that this application has the Prediction goal, rather than the Description goal;[5] we wish to predict the type of forest cover. There were seven classes of forest cover.

23.3.0.1   R Code

For simplicity, I restricted my analysis to classes 1 and 2.[6] In my R analysis I had the class 1 and 2 data in objects **cov1** and **cov2**, respectively. I combined them,

```
> cov1and2 <- rbind(cov1,cov2)
```

and created a new variable to serve as Y, recoding the 1,2 class names to 1,0:

```
cov1and2[,56] <- ifelse(cov1and2[,55] == 1,1,0)
```

Let's see how well we can predict a site's class from the variable HS12 (hillside shade at noon) that we investigated in Chapter 17, using a logistic model.

As noted earlier, in R we fit logistic models via the **glm()** function, for generalized linear models. The word *generalized* here refers to models in which some function of $m_{Y;X}(t)$ is linear in parameters β_i. For the classification model,

$$\ln\left(m_{Y;X}(t)/[1 - m_{Y;X}(t)]\right) = \beta_0 + \beta_1 t^{(1)} + ... + \beta_r t^{(r)} \qquad (23.15)$$

(Recall the discussion surrounding (23.14).)

This kind of generalized linear model is specified in R by setting the named argument **family** to **binomial**. Here is the call:

```
> g <- glm(cov1and2[,56] ~ cov1and2[,8],family=binomial)
```

The result was:

---

[4]There is a problem here, to be discussed in Section 23.4.2, but which will not affect the contents of this section.
[5]Recall these concepts from Section 22.1.
[6]This will be generalized in Section 23.4.

```
> summary(g)

Call:
glm(formula = cov1and2[, 56] ~ cov1and2[, 8], family = binomial)

Deviance Residuals:
   Min      1Q  Median      3Q     Max
-1.165  -0.820  -0.775   1.504   1.741

Coefficients:
               Estimate Std. Error z value Pr(>|z|)
(Intercept)    1.515820   1.148665   1.320   0.1870
cov1and2[, 8] -0.010960   0.005103  -2.148   0.0317 *
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 959.72  on 810  degrees of freedom
Residual deviance: 955.14  on 809  degrees of freedom
AIC: 959.14

Number of Fisher Scoring iterations: 4
```

23.3.1 Analysis of the Results

You'll immediately notice the similarity to the output of **lm()**. In particular, note the Coefficients section. There we have the estimates of the population coefficients β_i, their standard errors, and p-values for the tests of $H_0 : \beta_i = 0$.

One difference from the linear case is that in that case, the tests of

$$H_0 : \beta_i = 0 \tag{23.16}$$

were "exact," based on the Student-t distribution, rather than being approximate tests based on the Central Limit Theorem. The assumption is that the conditional distribution of the response given the predictors is exactly normal. As noted before, those tests can't possibly be exact, since the assumption cannot exactly hold.

But in the logit case, no "exact" test is available anyway, so R does indeed do approximate tests based on the Central Limit Theorem. Accordingly, the test column in the output is labeled "z value," rather than "t value" as before.

At any rate, we see that for example $\widehat{\beta}_1 = -0.01$. This is tiny, reflecting our analysis of this data in Chapter 17. There we found that the estimated mean values of HS12 for cover types 1 and 2 were 223.8 and 226.3, a difference of only 2.5, minuscule in comparison to the estimated means themselves. That difference in essence now gets multiplied by 0.01. Let's see the effect on the

regression function, i.e. the probability of cover type 1 given HS12.

Note first, though, that things are a little more complicated than they were in the linear case. Recall our first baseball data analysis, in Section 22.6. Our model for $m_{Y;X}()$ was

$$\text{mean weight} = \beta_0 + \beta_1 \text{ height} \tag{23.17}$$

After estimating the β_i from the data, we had the estimated regression equation,

```
mean weight = -155.092 + 4.841 height
```

In presenting these results to others, we can illustrate the above equation by noting, for instance, that a 3-inch difference in height corresponds to an estimated $3 \times 4.841 = 14.523$ difference in mean weight. The key point, though, is that this difference is the same whether we are comparing 70-inch-tall players with 73-inch-tall ones, or comparing 72-inch-tall players with 75-inch-tall ones, etc.

In our nonlinear case, the logit, the different in regression value will NOT depend only on the difference between the predictor values, in this case HS12. We cannot simply say something like "If two forest sites differ in HS12 by 15, then the difference in probability of cover type 1 differs by such-and-such an amount."

Instead we'll have to choose two specific HS11 values for our illustration. Let's use the estimated mean HS12 values for the two cover types, 223.8 and then 226.3, found earlier.

So, in (23.2), let's plug in our estimates 1.52 and -0.01 from our R output above, twice, first with HS12 = 223.8 and then with HS12 = 226.3

In other words, let's imagine two forest sites, with unknown cover type, but known HS12 values 223.8 and 226.8 that are right in the center of the HS12 distribution for the two cover types. What would we predict for the cover types to be for those two sites?

Plugging in to (23.2), the results are 0.328 and 0.322, respectively. Remember, these numbers are the estimated probabilities that we have cover type 1, given HS12. So, our guess—predicting whether we have cover type 1 or 2—isn't being helped much by knowing HS12.

In other words, HS12 isn't having much effect on the probability of cover type 1, and so it cannot be a good predictor of cover type.

And yet... the R output says that β_1 is "significantly" different from 0, with a p-value of 0.03. Thus, we see once again that significance testing does not achieve our goal.

23.4 The Multiclass Case

In classification problems, we often have more than two classes. In the forest cover example above, we simplified to having just two types of forest cover, but there were actually seven. In an optical character recognition application, there may be dozens of classes, maybe even hundreds or thousands for some written languages.

We will use m to denote the number of classes, and code the classes as $0, 1, ..., m-1$. Y will the class number, and let

$$q_i = P(Y = i) = P(\text{class i}), i = 0, 1, ..., m-1 \tag{23.18}$$

Note that the q_i are *un*conditional probabilities.

23.4.1 One vs. All Approach

The methods remain basically the same. Consider the forest cover example, for instance. There are seven cover types, so we could run seven logistic models, predicting a dummy Y for each type. In the first run, Y would be 1 for cover type 1, 0 for everything else. In predicting a new case, we plug the predictor values into each of the seven models, and guess Y to be the one with maximal probability among the seven.

The above approach is called One versus All (OVA). To see where the name comes from, again consider the forest cover example, with seven logistic regression models. In the ith model, we are predicting class i, against the collective alternative of all the other classes.

There is also the All versus All method (AVA). Here we run a regression analysis for each possible pair of classes ($7 \times 6/2 = 21$ of them for the forest cover data). For exach pair, we restrict our analysis only to data for the two classes. In predicting a new case, we use a "voting" procedure: For each pair of classes, we decide which of the two classes is more likely for this new case, and then take our guess for the class to be whichever class "won" the most times.

AVA has the disadvantage of using much less data for each pair. But if for example we are using a logistic model, which implies a linear boundary between each class and all the others,[7], an assumption that may not be very accurate. AVA essentially makes the weaker assumption that the boundary is linear between each *pair* of classes.

Note that under either OVA or AVA, we are always predicting one class at a time. Let's call this the Two Classes Suffice Principle, which will have implications below.

---

[7]The boundary is implied by setting (23.11) to 0.5.

There are multivariate versions of the logit model, but they make certain restrictive assumptions.

23.4.2   Issues of Data Balance

Here will discuss a topic that is usually glossed over in treatments of the classification problem, and indeed is often misunderstood in the machine learning (ML) research literature, where one sees much questionable handwringing over "the problem of unbalanced data." On the contrary, the real problem is often that the data are *balanced*.

23.4.2.1   Statement of the Problem

Consider the UCI Letters Recognition data set (`https://archive.ics.uci.edu/ml/datasets/Letter+Recognition`). There are between 700 and 800 cases for each English capital letter, which does not reflect that wide variation in letter frequencies. The letter 'E', for example, is more than 100 times as frequent as the letter 'Z', according to published data (see below). This would be considered "balanced" data, but it would be artificially so.

Now think of a different version of that data, in which the letter frequencies reflect their true nature in general English. There would be rather few instances of 'Z' here, but lots of cases of 'A'. This would be "unbalanced" but natural. And our predictions for data as a whole would be optimal in the sense of minimizing overall misclassification error (Section 23.6).

From here on, we will use the term *unbalanced* to mean that our data form a sample from our target population, in this case the population of all English text. Then we can estimate the q_i from the data:

$$\widehat{q_i} = \frac{N_i}{n} \tag{23.19}$$

where N_i is the number of our data points in class i, and n is the total number of observations in the data set. For balanced data, our data arose from sampling independently from the various classes, without attempting to have our data's class frequencies be representative of the population.

In the balanced case, due to a paucity of data for 'Z', our misclassification rate for that particular letter (guessing another letter when the letter is actually 'Z') might not be good. The latter issue is the one of concern in the machine learning community.

In other words, for general prediction accuracy, unbalanced data is the best. If we are concerned with misclassification rates for specific classes, things are more complex.

23.4.2.2 Solutions

There are really two problems to be solved:

(a) Our data is balanced, when it shouldn't be, as with the Letter Recognition data.

(b) Our data is unbalanced, and we are concerned about misclassification with the rarer classes.

We will assume just two classes here, per the Two Classes Suffice Principle we set up earlier. Let q and $1 - q$ denote the class probabilities for $Y = 1$ and $Y = 0$.

Let's first treat the context of Problem (a), and consider the quantity q in (23.9). If our data were balanced, we actually has an easy solution, if we know the correct value of q from some external source. For instance, in the English letters example above, there is much published data, such as at the Web page Practical Cryptography (`http://practicalcryptography.com/cryptanalysis/letter-frequencies-various-languages/english-letter-frequencies/`). They find that $\pi_A = 0.08$, $\pi_B = 0.0160$, $\pi_C = 0.0316$ and so on. So, if we use logistic regression (26 times), then when we run the one for 'C', we take $q = 0.0316$.

Now recall from Section 23.2.1 that one of the motivations for the logistic model is that it holds if the density of our predictor vector X is multivariate normal within each class (with equal covariance matrices in each class). It turns out that Equation (23.12) holds for a general number of predictors, not just 1 as in that equation.

If we then run **glm()** for our logistic regression, that function will act as if our data is unbalanced (the situation it was designed for). That means that our $\widehat{\beta}_0$ term will be incorrect, essentially using (23.19) . This suggests the following adjustment procedure.

(i) Add $\ln(Ni0/n)$.

(ii) Subtract $\ln[(1 - \pi)/\pi)]$, where π is the true class probability.

If we use a nonparametric approach (Chapter 24), Equation (23.9) shows us how to make the necessary adjustment. When we have a new case to predict, we do the following:

(a) Our software has given us an estimate of the left-hand side of that equation, for t equal to our new data point.

(b) We know the value that our software has used for its estimate of $(1 - q)/q$, which is N_0/N_1.

(c) Using (a) and (b), we can solve for the estimate of $f_o(t)/f_1(t)$.

(d) Now plug the correct estimate of $(1-q)/q$, and the result of (c), back into (23.9) to get the proper estimate of the desired conditional probability.

Now what about Problem (b) above, the unbalanced situation? The easiest solution is to deliberately use the *wrong* value of q in the above adjustment procedure.

23.5 Model Selection in Classification

Since, as has been emphasized here, classification is merely a special case of regression, the same issues arise for model selection as in Section 22.15. The only new issue is what to do with nonparametric models. For instance, with k-Nearest Neighbor estimation, how do we choose k?

Here the standard technique is again, as in Section 22.15, to split the data into training and validation sets. We could fit models with various values of k to the training set, and then see how well each does on the validation set.

The same points apply to deciding which predictors to use, and which to discard, as well as (in the parametric model case) whether to add polynomial terms and so on.

23.6 Optimality of the Regression Function for 0-1-Valued Y (optional section)

Remember, our context is that we want to guess Y, knowing X. Since Y is 0-1 valued, our guess for Y based on X, g(X), should be 0-1 valued too. What is the best function g()?

Again, since Y and g are 0-1 valued, our criterion should be what will I call Probability of Correct Classification (PCC):[8]

$$\text{PCC} = P[Y = g(X)] \tag{23.20}$$

We'll show intuitively that the best rule, i.e. the g() that maximizes (23.20), is given by the function

$$g(t) = \begin{cases} 0, & \text{if } g(t) \le 0.5 \\ 1, & \text{if } g(t) > 0.5 \end{cases} \tag{23.21}$$

---

[8]This assumes that our goal is to minimize the overall misclassification error rates, which in terms assumes equal costs for the two kinds of classification errors, i.e. that guessing $Y = 1$ when $Y = 0$ is no more or no less serious than the opposite error.

Think of this simple situation: There is a biased coin, with <u>known</u> probability of heads p. The coin will be tossed once, and you are supposed to guess the outcome.

Let's name your guess q, and let C denote the as-yet-unknown outcome of the toss (1 for heads, 0 for tails). Then the probability that you guess correctly is

$$
\begin{aligned}
P(C = q) &= P(C = 1)q + P(C = 0)(1 - q) & (23.22) \\
&= P(C = 1)q + [1 - P(C = 1)](1 - q) & (23.23) \\
&= [2P(C = 1) - 1]q + 1 - P(C = 1) & (23.24) \\
&= [2p - 1]q + 1 - p & (23.25)
\end{aligned}
$$

(That first equation merely accounts for the two cases, q = 1 and q = 0. For example, if you choose q = 0, then the right-hand side reduces to $P(C = 0)$, as it should.)

Inspecting the last of the three equations above, we see that if we set q = 1, then $P(C = q)$, i.e. the probability that we correctly predict the coin toss, is p. If we set q to 0, then $P(C = q)$ is 1-p. That in turn says that if $p > 0.5$ (remember, p is known), we should set q to 1; otherwise we should set q to 0.

The above reasoning gives us very intuitive—actually trivial—result:

If the coin is biased toward heads, we should guess heads. If the coin is biased toward tails, we should guess tails.

Now returning to (23.21), would take $P(C = q)$ above as the conditional probability $P(Y = g(X) \mid X)$. The above coin example says we should predict Y to be 1 or 0, depending on whether g(X) is larger or smaller than 0.5. Then use (4.68) to complete the proof.

Chapter 24

Nonparametric Estimation of Regression and Classification Functions

In some applications, there may be no good parametric model, say linear or logistic, for $m_{Y;X}$. Or, we may have a parametric model that we are considering, but we would like to have some kind of nonparametric estimation method available as a means of checking the validity of our parametric model. So, how do we estimate a regression function nonparametrically?

Many, many methods have been developed. We introduce a few here.

24.1   Methods Based on Estimating $m_{Y;X}(t)$

To guide our intuition on this, let's turn again to the example of estimating the relationship between height and weight. Consider estimation of the quantity $m_{W;H}(68.2)$, the *population* mean weight of all people of height 68.2.

We could take our estimate of $m_{W;H}(68.2)$, $\widehat{m}_{W;H}(68.2)$, to be the average weight of all the people in our sample who have that height. But we may have very few people of that height (or even none), so that our estimate may have a high variance, i.e. may not be very accurate.

What we could do instead is to take the mean weight of all the people in our sample whose heights are *near* 68.2, say between 67.7 and 68.7. That would bias things a bit, but we'd get a lower variance. This is again an illustration of the variance/bias tradeoff introduced in Section 21.1.3.

All nonparametric regression/classification (or "machine learning") methods work like this. There

are many variations, but at their core they all have this same theme. (Again, note the Hillel quote at the beginning of Section 23.1.)

As noted earlier, the classification problem is a special case of regression, so in the following material we will usually not distinguish between the two.

24.1.1 Nearest-Neighbor Methods

In Chapter**??**, we presented both kernel and nearest-neighbors for density estimation. The same two approaches can be used to estimate regression functions.

In the **nearest-neighbor** approach, we for instance estimating $m_{Y;X}(68.2)$ to be the mean weight of the k people in our sample with heights nearest 68.2. Here k controls bias/variance tradeoff.

Note that if we have more than one predictor variable, the distance used to determine "nearest" is multivariate, e.g. the distance in the plane in the case of two predictors.

In spite of the apparently simple notion here, nearest-neighbor regression and classification methods are quite effective and popular. Several contributed packages on the CRAN site for R implement this idea.

Here is simple (nonoptimized) code to do all this:

```
1  # the function knn() does k-nearest neighbor regression; the user has a
2  # choice of either just fitting to the x,y dataset or using that data to
3  # predict new observations newobs for which only the predictors are
4  # known
5
6  # arguments:
7
8  # x:   matrix or data frame of the predictor variable data, one row per
9  #      observation
10 #
11 # y:   vector of the response variables corresponding to x; in the
12 #      classification case, these are assumed to be 1s and 0s
13 #
14 # k:   the number of nearest neighbors to use for estimating the regression
15 #      or predicting the new data
16 #
17 # newobs:  a matrix of values of the predictors, one row per observation,
18 #          on which to predict the responses; default value is NULL
19 #
20 # regtype:  "reg" for prediction of continuous variables, "cls" for
```

```
21  #                 classification problems; default value "reg"
22  #
23
24  # return value: an R list with the following components
25  #
26  #     regvals:   estimated values of the regression function at x
27  #
28  #     predvals:   if newobs is not NULL, predicted values for y from newobs
29  #                 otherwise NULL
30  #
31  #     predsuccess:   if newobs is NULL, then R^2 in the "reg" case, proportion
32  #                    of correctly classified observations in the "cls" case;
33  #                    otherwise NULL
34
35  library(RANN)   # fast nearest-neighbor finder on CRAN
36
37  knn <- function(x,y,k,newobs=NULL,regtype="reg") {
38     # make sure x is a matrix or data frame for use with RANN
39     if (is.vector(x)) x <- matrix(x,ncol=1)
40     retval <- list()
41     # just trying out on current data set?
42     if (is.null(newobs)) {
43        nearones <- nn2(data=x,k=k,query=x)$nn.idx
44     } else {
45        nearones <- nn2(data=x,k=k,query=newobs)$nn.idx
46     }
47     # row i of nearones now consists of the indices in x of the k closest
48     # observations in x to row i of x or row i of newobs
49     #
50     # now find the estimated regression function at each row
51     regvals <- apply(nearones,1,pred1y,y)
52     if (is.null(newobs)) {
53        if (regtype=="reg") {
54           tmp <- cor(regvals,y)
55           predsuccess <- tmp^2
56        } else {
57           predvals <- as.integer(regvals > 0.5)
58           predsuccess <- mean(predvals == y)
59        }
60        predvals <- NULL
```

```
61     } else {
62        predsuccess <- NULL
63        newregvals <- apply(nearones,1,pred1y,y)
64        if (regtype == "reg") predvals <- newregvals else {
65           predvals <- as.integer(regvals > 0.5)
66        }
67     }
68     retval$regvals <- regvals
69     retval$predvals <- predvals
70     retval$predsuccess <- predsuccess
71     retval
72  }
73
74  # for a single observation, calculate the value of the regression
75  # function there, knowing the indices xidxs of the values in the
76  # original data x that are closest to the given observation
77  pred1y <- function(xidxs,y) predval <- mean(y[xidxs])
```

24.1.2 Kernel-Based Methods

As our definition of "near," we could take all people in our sample whose heights are within h amount of 68.2. This should remind you of our density estimators in Chapter 20. A generalization would be to use a **kernel** method. For instance, for univariate X and t:

$$\widehat{m}_{Y;X}(t) = \frac{\sum_{i=1}^{n} Y_i k\left(\frac{t-X_i}{h}\right)}{\sum_{i=1}^{n} k\left(\frac{t-X_i}{h}\right)} \tag{24.1}$$

Again note that if we have more than one predictor variable, the function k() has a multivariate argument.

Here k() is a density, i.e. a nonnegative function that integrates to 1. Also, it is almost always chosen so that k() is symmetric around 0, with a peak at 0 and then tapering off as one moves away from 0 in either direction.

This looks imposing! But it is simply a weighted average of the Y values in our sample, with the larger weights being placed on observations for which X is close to t.

Note the word *chosen*. The analyst makes this choice (or takes a default value, say in an R library), simply from considerations of weighting: Choosing k() to be a "tall, narrow" function will make the weights drop off more rapidly to 0.

In fact, the choice of kernel is not very important (often it is taken to be the N(0,1) density.) What does matter is the parameter h. The smaller h is, the more our weighting will concentrate on nearby observations.

In other words, setting a smaller value of h is quite analogous to choosing a smaller value of k (the number of nearest neighbors, not our kernel function here) in nearest-neighbor regression.

As before, the choice of h here involves a bias/variance tradeoff. We might try choosing h via cross validation, as discussed in Section 22.15.4.

There is an R package that includes a function **nkreg()** for kernel regression. The R base has a similar method, called **LOESS**. Note: That is the class name, but the R function is called **lowess()**.

24.1.3   The Naive Bayes Method

This method is for the classification problem only.

The Naive Bayes method is not "Bayesian" in the sense of Section 18.4. Instead, its name comes simply from its usage of Bayes' Rule for conditional probability. It basically makes the same computations as in Section 23.2.1, for the case in which the predictors are indicator variables and are independent of each other, given the class.

The term *naive* is an allusion to analysts who naively assume independent predictors, without realizing that they are making a serious restriction.

Under that assumption, the numerator in (23.8) becomes

$$P(Y = 1) \ \ P[X^{(1)} = t_1 | Y = 1] \ \ ... \ \ P[X^{(r)} = t_r | Y = 1] \tag{24.2}$$

All of those quantities (and similarly, those in the denominator of (23.8) can be estimated directly as sample proportions. For example, $\widehat{P}[X^{(1)} = t_1 | Y = 1]$ would be the fraction of $X_j^{(1)}$ that are equal to t_1, among those observations for which $Y_j = 1$.

A common example of the use of Naive Bayes is text mining, as in Section 13.5.1.4. Our independence assumption in this case means that the probability that, for instance, a document of a certain class contains both of the words *baseball* and *strike* is the product of the individual probabilities of those words.

Clearly the independence assumption is not justified in this application. But if our vocabulary is large, that assumption limits the complexity of our model, which may be necessary from a bias/variance tradeoff point of view (Section 21.1.3).

24.2 Methods Based on Estimating Classification Boundaries

In the methods presented above, we are estimating the function $m_{Y;X}(t)$. But with support vector machines and CART below, we are in a way working backwards. In the classification case (which is what we will focus on), for instance, our goal is to estimate the values of t for which the regression function equals 0.5:

$$B = \{t : m_{Y;X}(t) = 0.5\} \tag{24.3}$$

Recall that r is the number of predictor variables we have. Then note the geometric form that the set B in (24.3) will take on: discrete points if r = 1; a curve if r = 2; a surface if r = 3; and a hypersurface if r > 3.

The motivation for using (24.3) stems from the fact, noted in Section 23.1, that if we know $m_{Y;X}(t)$, we will predict Y to be 1 if and only if $m_{Y;X}(t) > 0.5$. Since (24.3) represents the boundary between the portions of the X space for which $m_{Y;X}(t)$ is either larger or smaller than 0.5, it is the boundary for our prediction rule, i.e. the boundary separating the regions in X space in which we predict Y to be 1 or 0.

Lest this become too abstract, again consider the simple example of predicting gender from height and weight. Consider the (u,v) plane, with u and v representing height and weight, respectively. Then (24.3) is some curve in that plane. If a person's (height,weight) pair is on one side of the curve, we guess that the person is male, and otherwise guess female.

If the logistic model (23.2) holds, then that curve is actually a straight line. To see this, note that in (23.2), the equation (24.3) boils down to

$$\beta_0 + \beta_1 u + \beta_2 v = 0 \tag{24.4}$$

whose geometric form is a straight line.

24.2.1 Support Vector Machines (SVMs)

This method has been getting a lot of publicity in computer science circles (maybe too much; see below). It is better explained for the classification case.

In the form of dot product (or inner product) from linear algebra, (24.4) is

$$(\beta_1, \beta_2)'(u, v) = -\beta_0 \tag{24.5}$$

What SVM does is to generalize this, for instance changing the criterion to, say

$$\beta_0 u^2 + \beta_1 uv + \beta_2 v^2 + \beta_3 u + \beta_4 v = 1 \tag{24.6}$$

Now our (u, v) plane is divided by a curve instead of by a straight line (though it includes straight lines as special cases), thus providing more flexibility and thus potentially better accuracy.

In SVM terminology, (24.6) uses a different **kernel** than regular dot product. (This of course should not be confused our the term *kernel* in kernel-based regression above.) The actual method is more complicated than this, involving transforming the original predictor variables and then using an ordinary inner product in the transformed space. In the above example, the transformation consists of squaring and multiplying our variables. That takes us from two-dimensional space (just u and v) to five dimensions (u, v, u^2, v^2 and uv).

There are various other details that we've omitted here, but the essence of the method is as shown above.

Of course, a good choice of the kernel is crucial to the successful usage of this method. It is the analog of h and k in the nearness-based methods above.

A former UCD professor, Nello Cristiani, is one of the world leaders in SVM research. See *An Introduction to Support Vector Machines*, N. Cristianini and J. Shawe-Taylor, Cambridge University Press, 2000.

24.2.2   CART

Another nonparametric method is that of **Classification and Regression Trees** (CART). It's again easiest explained in the classification context, say the diabetes example above.

In the diabetes example, we might try to use glucose variable as our first predictor. The data may show that a high glucose value implies a high likelihood of developing diabetes, while a low value does the opposite. We would then find a **split** on this variable, meaning a a cutoff value that defines "high" and "low." Pictorially, we draw this as the root of a tree, with the left branch indicating a tentative guess of no diabetes and the right branch corresponding to a guess of diabetes.

Actually, we could do this for all our predictor variables, and find which one produces the best split at the root stage. But let's assume that we find that glucose is that variable.

Now we repeat the process. For the left branch—all the subset of our data corresponding to "low" glucose—we find the variable that best splits that branch, say body mass index. We do the same for the right branch, say finding that age gives the best split. We keep going until the resulting cells are too small for a reasonable split.

An example with real data is given in a tutorial on the use of **rpart**, an R package that does analysis of the CART type, *An Introduction to Recursive Partitioning Using the RPART Routines*, by Terry Therneau and Elizabeth Atkinson. The data was on treatment of cardiac arrest patients by emergency medical technicians.

The response variable here is whether the technicians were able to revive the patient, with predictors $X^{(1)} =$ initial heart rhythm, $X^{(2)} =$ initial response to defrillabration, and $X^{(3)} =$ initial response to drugs. The resulting tree was
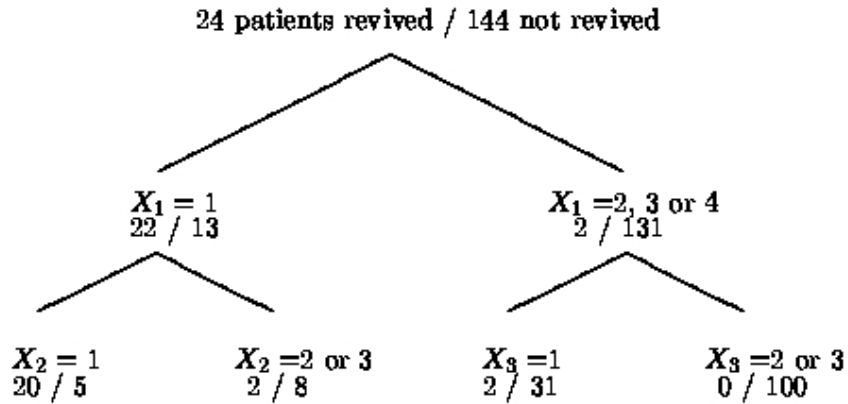
Figure 1: Revival data

So, if for example a patient has $X^{(1)} = 1$ and $X^{(2)} = 3$, we would guess him to be revivable.

CART is a boundary method, as SVM is. Say for instance we have two variables, represented graphically by s and t, and our root node rule is $s > 0.62$. In the left branch, the rule is $t > 0.8$ and in the right branch it's $t > 0.58$. This boils down to a boundary line as follows:
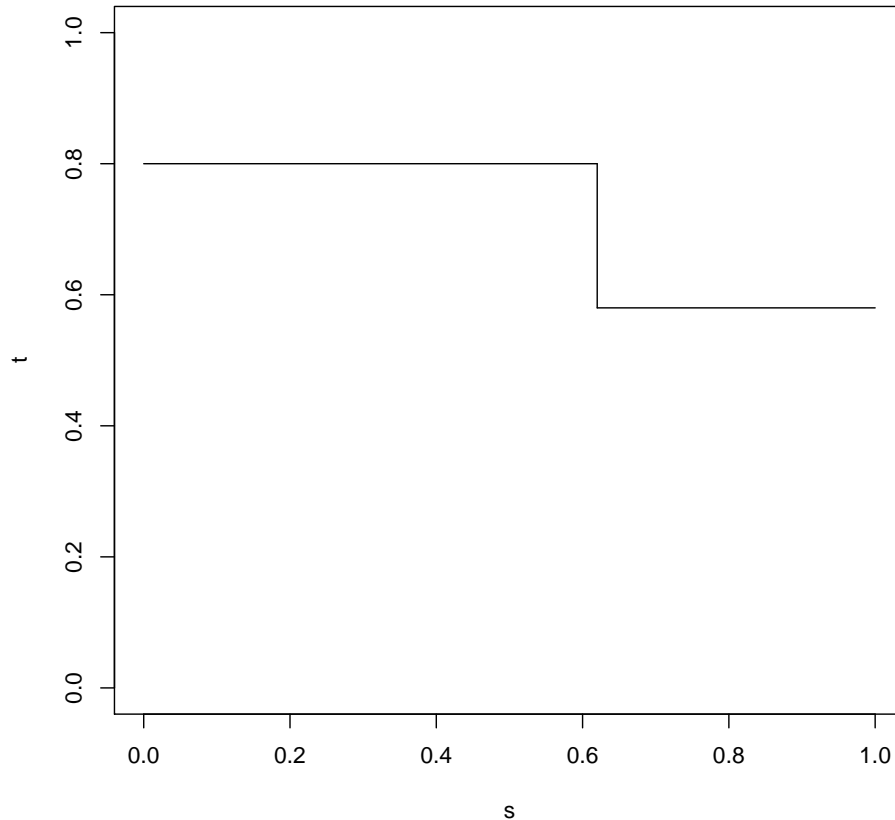
CART obviously has an intuitive appeal, easily explained to nonstatisticians, and easy quite easy to implement. It also has the virtue of working equally well with discrete or continuous predictor variables.

The analogs here of the h in the kernel method and k in nearest-neighbor regression are the choice of where to define the splits, and when to stop splitting. Cross validation is often used for making such decisions.

24.3   Comparison of Methods

Beware!   There are no "magic" solutions to statistical problems. The statements one sees by some computer science researchers to the effect that SVMs are generally superior to other prediction methods are, unfortunately, unfounded; there just is no generally superior method.

First, note that every one of the above methods involves some choice of tuning parameter, such as

h in the kernel method, k in the nearest-neighbor method, the split points in CART, and in the case of SVM, the form of kernel to use. For SVM the choice of kernel is crucial, yet difficult.

Second, the comparisons are often unfair, notably comparisons of the logit model to SVM. Such comparisons usually limit the logit experiments to first-degree terms without interactions. But the kernel in SVM is essentially analogous to throwing in second-degree and interaction terms, and so oni, (23.2) for the logit case, thus producing a curved partitioning line just like SVM does.

I highly recommend the site `www.dtreg.com/benchmarks.htm`, which compares six different types of classification function estimators—including logistic regression and SVM—on several dozen real data sets. The overall percent misclassification rates, averaged over all the data sets, was fairly close, ranging from a high of 25.3% to a low of 19.2%. The much-vaunted SVM came in at an overall score across all data sets of 20.3%. That's nice, but it was only a tad better than logit's 20.9%—and remember, that's with logit running under the handicap of having only first-degree terms.

Or consider the annual KDDCup competition, in which teams from around the world compete to solve a given classification problem with the lowest misclassification rate. In KDDCup2009, for instance, none of the top teams used SVM. See *SIGKDD Explorations*, December 2009 issue.

Considering that logit has a big advantage in that one gets an actual equation for the classification function, complete with parameters which we can estimate and make confidence intervals for, it is not clear just what role SVM and the other nonparametric estimators should play, in general, though in specific applications they may be appropriate.

Chapter 25

Relations <u>Among</u> Variables

It is a very sad thing that nowadays there is so little useless information—Oscar Wilde, famous 19th century writer

Unlike the case of regression analysis, where the response/dependent variable plays a central role, we are now interested in symmetric relations among several variables. Often our goal is **dimension reduction**, meaning compressing our data into just a few important variables.

Dimension reduction ties in to the Oscar Wilde quote above, which is a complaint that there is too *much* information of the use*ful* variety. We are concerned here with reducing the complexity of that information to a more manageable, simple set of variables.

Here we cover two of the most widely-used methods, **principal components analysis** for continuous variables, and the **log-linear model** for the discrete case. We also introduce **clustering**.

25.1 Principal Components Analysis (PCA)

Consider a random vector $X = (X_1, X_2)'$. Suppose the two components of X are highly correlated with each other. Then for some constants c and d,

$$X_2 \approx c + dX_1 \tag{25.1}$$

Then in a sense there is really just one random variable here, as the second is nearly equal to some linear combination of the first. The second provides us with almost no new information, once we have the first.

In other words, even though the vector X roams in two-dimensional space, it usually sticks close to

a one-dimensional object, namely the line (25.1). We saw a graph illustrating this in our chapter on multivariate distributions, page 238.

In general, consider a k-component random vector

$$X = (X_1, ..., X_k)' \tag{25.2}$$

We again wish to investigate whether just a few, say w, of the X_i tell almost the whole story, i.e. whether most X_j can be expressed approximately as linear combinations of these few X_i. In other words, even though X is k-dimensional, it tends to stick close to some w-dimensional subspace.

Note that although (25.1) is phrased in prediction terms, we are not (or more accurately, not necessarily) interested in prediction here. We have not designated one of the $X^{(i)}$ to be a response variable and the rest to be predictors.

Once again, the Principle of Parsimony is key. If we have, say, 20 or 30 variables, it would be nice if we could reduce that to, for example, three or four. This may be easier to understand and work with, albeit with the complication that our new variables would be linear combinations of the old ones.

25.1.1   How to Calculate Them

Here's how it works. Let Σ denote the covariance matrix of X. The theory of linear algebra says that since Σ is a symmetric matrix, it is diagonalizable, i.e. there is a real matrix Q for which

$$Q'\Sigma Q = D \tag{25.3}$$

where D is a diagonal matrix. (A related approach is **singular value decomposition**.) The columns C_i of Q are the eigenvectors of Σ, and it turns out that they are orthogonal to each other, i.e. their dot product is 0.

Let

$$W_i = C_i'X, \ \ i = 1, ..., k \tag{25.4}$$

so that the W_i are scalar random variables, and set

$$W = (W_1, ..., W_k)' \tag{25.5}$$

Then

$$W = Q'X \tag{25.6}$$

Now, use the material on covariance matrices from our chapter on random vectors, page 208,

$$Cov(W) = Cov(Q'X) = Q'Cov(X)Q = D \quad \text{(from (25.3))} \tag{25.7}$$

Note too that if X has a multivariate normal distribution (which we are not assuming), then W does too.

Let's recap:

- We have created new random variables W_i as linear combinations of our original X_j.

- The W_i are uncorrelated. Thus if in addition X has a multivariate normal distribution, so that W does too, then the W_i will be independent.

- The variance of W_i is given by the ith diagonal element of D.

The W_i are called the **principal components** of the distribution of X.

It is customary to relabel the W_i so that W_1 has the largest variance, W_2 has the second-largest, and so on. We then choose those W_i that have the larger variances, and discard the others, because the latter, having small variances, are close to constant and thus carry no information.

Note that an alternate definition of the first principal component is a value of u that maximizes $u'X$ subject to u having length 1. The second principal component maximizes $u'X$ subject to u having length 1 and subject to the second component being orthogonal to the first, and so on.

All this will become clearer in the example below.

25.1.2   Example: Forest Cover Data

Let's try using principal component analysis on the forest cover data set we've looked at before. There are 10 continuous variables.[1]

In my R run, the data set (not restricted to just two forest cover types, but consisting only of the first 1000 observations) was in the object **f**. Here are the call and the results:

_____

[1]There are also many discrete ones.

```
> prc <- prcomp(f[,1:10])
> summary(prc)
Importance of components:
                      PC1      PC2      PC3      PC4      PC5 PC6
Standard deviation  1812.394 1613.287 1.89e+02 1.10e+02 96.93455 30.16789
Proportion of Variance 0.552    0.438 6.01e-03 2.04e-03  0.00158 0.00015
Cumulative Proportion  0.552    0.990 9.96e-01 9.98e-01  0.99968 0.99984
                      PC7      PC8 PC9  PC10
Standard deviation  25.95478 16.78595 4.2 0.783
Proportion of Variance 0.00011 0.00005 0.0 0.000
Cumulative Proportion  0.99995 1.00000 1.0 1.000
```

You can see from the variance values here that R has scaled the W_i so that their variances sum to 1.0. (It has not done so for the standard deviations, which are for the nonscaled variables.) This is fine, as we are only interested in the variances relative to each other, i.e. saving the principal components with the larger variances.

What we see here is that eight of the 10 principal components have very small variances, i.e. are close to constant. In other words, though we have 10 variables $X_1, ..., X_{10}$, there is really only two variables' worth of information carried in them.

So for example if we wish to predict forest cover type from these 10 variables, we should only use two of them. We could use W_1 and W_2, but for the sake of interpretability we stick to the original X vector. We can use any two of the X_i, though typically it would be two that have large coefficients in the top two principal components..

The coefficients of the linear combinations which produce W from X, i.e. the Q matrix, are available via **prc$rotation**.

25.1.3   Scaling

If your original variables range quite a bit in variance, you should have **prcomp()** scale them first, so they all have standard deviation 1.[2] The argument name is **scale**, of course.

Without scaling, the proportion-of-total-variance type of analysis discussed above may be misleading, as large-variance variables may dominate.

25.1.4   Scope of Application

PCA makes no assumptions about the data. It is strictly an exploratory/descriptive tool.

However, it should be noted that the motivation we presented for PCA at the beginning of this chapter involved correlations among our original variables. This is further highlighted by the fact

---

[2]And mean 0, though this is irrelevant, as Σ is all that matter.

that the PCs are calculated based on the covariance matrix of the data, which except for scale is the same as the correlation matrix.

This in turn implies that each variable is at least *ordinal* in nature, i.e. that it makes sense to speak of the impact of larger or smaller values of a variable.

Note, though, that an indicator random variable is inherently ordinal! So, if you have a *categorical* variable, i.e. one that simply codes what category an individual falls into (such as Democratic, Republican, independent), then you can convert it to a set of indicator variables, and potentially get some insight into the relation between this variable and others.

This can be especially valuable if, as is often the case, your data consists of a mixture of ordinal and categorical variables.

25.1.5 Example: Turkish Teaching Evaluation Data

This data, again from the UCI Machine Learning Repository, was introduced in Section 22.17. Let's try PCA on it:

```
> tpca <- prcomp(turk,scale=T)
> summary(tpca)
Importance of components:
                          PC1     PC2     PC3     PC4     PC5     PC6
PC7
Standard deviation      4.8008  1.1296  0.98827  0.62725  0.59837  0.53828  0.50587
Proportion of Variance  0.7947  0.0440  0.03368  0.01357  0.01235  0.00999  0.00882
Cumulative Proportion   0.7947  0.8387  0.87242  0.88598  0.89833  0.90832  0.91714
                          PC8     PC9    PC10    PC11    PC12    PC13
PC14
Standard deviation      0.45182  0.42784  0.41517  0.37736  0.37161  0.36957  0.3450
Proportion of Variance  0.00704  0.00631  0.00594  0.00491  0.00476  0.00471  0.0041
Cumulative Proportion   0.92418  0.93050  0.93644  0.94135  0.94611  0.95082  0.9549
                          PC15    PC16    PC17    PC18    PC19    PC20
PC21
Standard deviation      0.34114  0.33792  0.33110  0.32507  0.31687  0.30867  0.3046
Proportion of Variance  0.00401  0.00394  0.00378  0.00364  0.00346  0.00329  0.0032
Cumulative Proportion   0.95894  0.96288  0.96666  0.97030  0.97376  0.97705  0.9802
                          PC22    PC23    PC24    PC25    PC26    PC27
PC28
Standard deviation      0.29083  0.29035  0.28363  0.27815  0.26602  0.26023  0.23621
Proportion of Variance  0.00292  0.00291  0.00277  0.00267  0.00244  0.00234  0.00192
Cumulative Proportion   0.98316  0.98607  0.98884  0.99151  0.99395  0.99629  0.99821
```

```
                        PC29
Standard deviation     0.22773
Proportion of Variance 0.00179
Cumulative Proportion  1.00000
```

This is remarkable—the first PC accounts for 79% of the variance of the set of 29 variables. In other words, in spite of the survey asking supposedly 29 different aspects of the course, they can be summarized largely in just one variable. Let's see what that variable is:

```
> tpca$rotation[,1]
          Q1              Q2              Q3              Q4              Q5
Q6
-0.16974120  -0.18551431  -0.18553930  -0.18283025  -0.18973563  -0.18635256
          Q7              Q8              Q9             Q10             Q11             Q12
-0.18730028  -0.18559928  -0.18344211  -0.19241585  -0.18388873  -0.18184118
         Q13             Q14             Q15             Q16             Q17             Q18
-0.19430111  -0.19462822  -0.19401115  -0.19457451  -0.18249389  -0.19320936
         Q19             Q20             Q21             Q22             Q23             Q24
-0.19412781  -0.19335127  -0.19232101  -0.19232914  -0.19554282  -0.19328500
         Q25             Q26             Q27             Q28      difficulty
-0.19203359  -0.19186433  -0.18751777  -0.18855570  -0.01712709
```

This is even more remarkable. Except for the "difficulty" variable, all the Qi have about the same coefficients (the same **loadings**). In other words, just one question would have been enough, and it wouldn't matter mcuh which one were used.

The second PC, though only accounting for 4% of the total variation, is still worth a look:

```
> tpca$rotation[,2]
          Q1              Q2              Q3              Q4              Q5
Q6
 0.32009850   0.22046468   0.11432028   0.23340347   0.20236372   0.19890471
          Q7              Q8              Q9             Q10             Q11             Q12
 0.24025046   0.24477543   0.13198060   0.19239207   0.11064523   0.20881773
         Q13             Q14             Q15             Q16             Q17             Q18
-0.09943140  -0.15193169  -0.15089563  -0.03494282  -0.26163096  -0.11646066
         Q19             Q20             Q21             Q22             Q23             Q24
-0.14424468  -0.18729978  -0.21208705  -0.21650494  -0.09349599  -0.05372049
         Q25             Q26             Q27             Q28      difficulty
-0.20342350  -0.10790888  -0.05928032  -0.20370705  -0.27672177
```

Here the "difficulty" variable now shows up, and some of the Qi become unimportant.

25.2 Log-Linear Models

Here we discuss a procedure which is something of an analog of principal components for discrete variables, especially *nominal* ones, i.e. variables we would convert to dummies in a regression setting.

25.2.1 The Setting

Let's consider a variation on the software engineering example in Section 22.2. Assume we have the factors, IDE, Language and Education. One change—**of extreme importance**—is that we will now assume that these factors are **random**. What does this mean?

In the original example described in Section 22.2, programmers were *assigned* to languages, and in our extensions of that example, we continued to assume this. Thus for example the number of programmers who use an IDE and program in Java was fixed; if we repeated the experiment, that number would stay the same. If we were sampling from some programmer population, our new sample would have new programmers, but the number using and IDE and Java would be the same as before, as our study procedure specifies this.

By contrast, let's now assume that we simply sample programmers at random, and ask them whether they prefer to use an IDE or not, and which language they prefer.[3] Then for example the number of programmers who prefer to use an IDE and program in Java will be random, not fixed; if we repeat the experiment, we will get a different count.

Suppose we now wish to investigate relations between the factors. Are choice of platform and language related to education, for instance?

25.2.2 The Data

Denote our three factors by $X^{(s)}$, s = 1,2,3.[4] Here $X^{(1)}$, IDE, will take on the values 1 and 2 instead of 1 and 0 as before, 1 meaning that the programmer prefers to use an IDE, and 2 meaning not so. $X^{(3)}$, Education, changes this way too, and $X^{(2)}$ will take on the values 1 for C++, 2 for Java and 3 for C.

Note that we no longer use indicator variables. Indeed, the log-linear model is in contrast to PCA, which as noted above tacitly assumes ordinal variables. Here we are working with strictly categorical variables, whose values are merely labels. We could, for example, relabel $X^{(2)}$ to have the value 1 for Java, 2 for C++ and 3 for C, and yet we would still have the same results (though

---

[3]Other sampling schemes are possible too.

[4]All the examples here will have three factors, but there can be any number.

the results would be relabeled too).

Let $X_r^{(s)}$ denote the value of $X^{(s)}$ for the rth programmer in our sample, r = 1,2,...,n. Our data are the counts

$$N_{ijk} = \text{number of r such that } X_r^{(1)} = i, X_r^{(2)} = j \text{ and } X_r^{(3)} = k \qquad (25.8)$$

For instance, if we sample 100 programmers, our data might look like this:

```
prefers to use IDE:

            Bachelor's or less      Master's or more
    C++                     18                     15
    Java                    22                     10
      C                      6                      4


prefers not to use IDE:

            Bachelor's or less      Master's or more
    C++                      7                      4
    Java                     6                      2
      C                      3                      3
```

So for example $N_{122} = 10$ and $N_{212} = 4$.

Here we have a three-dimensional **contingency table**. Each N_{ijk} value is a **cell** in the table.

25.2.3   The Models

Let p_{ijk} be the population probability of a randomly-chosen programmer falling into cell ijk, i.e.

$$p_{ijk} = P\left(X^{(1)} = i \text{ and } X^{(2)} = j \text{ and } X^{(3)} = k\right) = E(N_{ijk})/n \qquad (25.9)$$

As mentioned, we are interested in relations among the factors, in the form of independence, both full and partial. Indeed, it is common for an analyst to fit successively more refined models to the data, each assuming a more complex dependence structure than the last. This will unfold in detail below.

Consider first the model that assumes full independence:

$$
\begin{aligned}
p_{ijk} &= P\left(X^{(1)} = i \text{ and } X^{(2)} = j \text{ and } X^{(3)} = k\right) & (25.10) \\
&= P\left(X^{(1)} = i\right) \cdot P\left(X^{(2)} = j\right) \cdot P\left(X^{(3)} = k\right) & (25.11)
\end{aligned}
$$

Taking logs of both sides in (25.11), we see that independence of the three factors is equivalent to saying

$$\log(p_{ijk}) = a_i + b_j + c_k \qquad (25.12)$$

for some numbers a_i, b_j and c_j; e.g.

$$b_2 = \log[P(X^{(2)} = 2)] \qquad (25.13)$$

The point is that (25.12) looks like our no-interaction linear regression models, whose analog of noninteraction here is independence of our variables. On the other hand, if we assume instead that Education is independent of IDE and Language but that IDE and Language are not independent of each other, our model would include an i-j interaction term, as follows.[5]

We would have

$$p_{ijk} \;\; = \;\; P\left(X^{(1)} = i \text{ and } X^{(2)} = j\right) \cdot P\left(X^{(3)} = k\right) \qquad (25.14)$$

so we would set

$$\log(p_{ijk}) = a_{ij} + b_k \qquad (25.15)$$

Most formal models rewrite this as

$$a_{ij} = u + v_i + w_j + r_{ij} \qquad (25.16)$$

Here we have written $P\left(X^{(1)} = i \text{ and } X^{(2)} = j\right)$ as a sum of an "overall effect" u, "main effects" v_i and w_j, and "interaction effects," r_{ij}, again analogous to linear regression.

Note, though, that this actually gives us too many parameters. For the i-j part of the model, we have $2 \times 3 = 6$ actual probabilities, but $1 + 2 + 3 + 2 \times 3 = 12$ parameters (1 for u, 2 for the v_i and so on). So the model needs constraints, e.g.

$$\sum v_i = 0 \qquad (25.17)$$

---

[5]In order to simplify the discussion below, we will often write i as a shorthand for $X^{(i)}$ and so on.

This is similar to classical Analysis of Variance (ANOVA), not covered in this book. We will not state the constraints below, but they are used in most if not all software packages for the log-linear model..

Another possible model would have IDE and Language conditionally independent, given Education, meaning that at any level of education, a programmer's preference to use IDE or not, and his choice of programming language, are not related. We'd write the model this way:

$$
\begin{aligned}
p_{ijk} &= P\left(X^{(1)} = i \text{ and } X^{(2)} = j \text{ and } X^{(3)} = k\right) & (25.18) \\
&= P\left(X^{(1)} = i \text{ and } X^{(2)} = j \mid X^{(3)} = k\right) \cdot P\left(X^{(3)} = k\right) & (25.19) \\
&= P\left(X^{(1)} = i \mid X^{(3)} = k\right) \cdot P\left(X^{(2)} = j \mid X^{(3)} = k\right) \cdot P\left(X^{(3)} = k\right) & (25.20)
\end{aligned}
$$

and thus set

$$
\log(p_{ijk}) == u + a_i + f_{ik} + b_j + h_{jk} + c_k \tag{25.21}
$$

25.2.4   Interpretation of Parameters

Note carefully that the type of independence in (25.21) has a quite different interpretation than that in (25.15). Actually, our old trick coin example, Section 19.1, was like (25.20); given the choice of coin, the B_i were independent. On the other hand, among people, height and weight are correlated but they are presumably independent of preference for flavor of ice cream, a situation like (25.15).

This is an excellent example of what can go wrong with mindless use of packaged software. Log-linear analysis is all about distinguishing between various kinds of dependence. Without an understanding of how the models reflect this, the software can produce highly misleading results.

So, pay close attention to which interactions are in the model, and which are not. In (25.15) we see a two-factor interaction between the i and j factors, but no interaction with k. So i and j are being modeled as completely independent of k, though not with each other. On the other hand, in (25.21), the i and j factors have interactions with k, but not with each other.

Now consider the model

$$
\log(p_{ijk}) = u + a_i + f_{ik} + b_j + h_{jk} + l_{ij} + c_k \tag{25.22}
$$

Here things are a little harder to interpret. Given k, i and j are no longer modeled as independent.

However, the degree of "correlation" between i and j, given k, is modeled as being independent of k. In other words, the strength of the relation between i and j is the same, for all levels of k.

If we had included an m_{ijk} term—which would now make the model **full** or **saturated**—then it may be possible that i and j are highly related for some values of k, and less related for others.

Clearly, the more variables we have, and the higher the order of interactions we include, the harder it is to interpret the model

25.2.5 Parameter Estimation

Remember, whenever we have parametric models, the statistician's "Swiss army knife" is Maximum Likelihood Estimation (MLE, Section 18.1.3). That is what is most often used in the case of log-linear models.

How, then, do we compute the likelihood of our data, the N_{ijk}? It's actually quite straightforward, because the N_{ijk} have a multinomial distribution (Section 13.5.1.1). Then the likelihood function is

$$L = \frac{n!}{\Pi_{i,j,k} N_{ijk}!} p_{ijk}^{N_{ijk}} \tag{25.23}$$

We then write the p_{ijk} in terms of our model parameters. Take for example (25.21), where we write

$$p_{ijk} = e^{u+v_i+w_j+r_{ij}+c_k} \tag{25.24}$$

We then substitute (25.24) in (25.23), and maximize the latter with respect to the a_i, b_j, d_{ij} and c_k, subject to constraints as mentioned earlier.

The maximization may be messy. But certain cases have been worked out in closed form, and in any case today one would typically do the computation by computer. In R, for example, there is the **loglin()** function for this purpose, illustrated below.

Unfortunately, most books and software packages for the log-linear model put almost all of their focus on significance testing, rather than point estimation and confidence intervals. In the popular **loglin()** package, for instance, the parameter estimates, e.g. $\hat{v}_i$ above, are not even reported unless the user requests them. Even then, no standard errors are reported. This is counter to the universally recognized—though unfortunately widely ignored—point that significance testing can be quite misleading, especially in large samples (Section 17.11).

The tests themselves assess fit. For instance, say we fit the model (25.21). The program estimates parameters using maximum likelihood, and then tests whether the model is "correct." The actual

test will either be chi-squared (Section 21.2.1) or the Likelihood Ratio Test (related to MLEs); both test statistics have chi-square distributions under H_0. We can call **pchisq()** to find the p-value. Again, note that H_0 is that the specified model is correct.

25.2.6   Example: Hair, Eye Color

Here we will look at R's built-in data set **HairEyeColor**. There are variables Hair, Eye and Sex, with 4 levels for hair color and 4 for eye color, over 592 people. Type **?HairEyeColor** to see the details.

This is actually an example from the online help for the function **loglin()**, though we'll look at a different model in terms of interactions.

25.2.6.1   The loglin() Function

We'll use the built-in R function **loglin()**, whose input data must be of class **"table"**. Let's see first how the latter works.

Say we have two variables, the first having levels 1 and 2, and the second having levels 1, 2 and 3. Suppose our data frame **d** is

```
> d
  V1 V2
1  1  3
2  2  3
3  2  2
4  1  1
5  1  2
```

The first person (or other entity) in our dataset has $X^{(1)} = 1$, $X^{(2)} = 3$ and so on. The function **table()** does what its name implies: It tabulates the counts in each cell:

```
> table(d)
    V2
V1   1 2 3
   1 1 1 1
   2 0 1 1
```

This says there was one instance in which $X^{(1)} = 1, X^{(2)} = 3$ etc., but no instances of $X^{(1)} = 2, X^{(2)} = 1$.

So, our data is input as an object of type **"table"**, specified in the argument **table**.

Our model is input via the argument **margin**, which is an R list of vectors. For instance **c(1,3)** specifies an interaction between variables 1 and 3, and **c(1,2,3)** means a three-way interaction. Once a higher-order interaction is specified, we need not specify its lower-order "subset." If, say, we specify **c(2,5,6)**, we need not specify **c(2,6)**. On the other hand, if a variable m is involved in no interactions, we need to specify it as **c(m)**, in order to keep its individual (i.e. non-interaction) effect in the model.

The model used is actually for the cell counts, not the cell probabilities. Thus the constant term, e.g. u in (25.22) is smaller by an amount equal to the log of the total number of observations in the table.

25.2.7 Hair/Eye Color Analysis

Let's get an overview of the data first:

```
> HairEyeColor
, , Sex = Male

       Eye
Hair     Brown Blue Hazel Green
  Black     32   11    10     3
  Brown     53   50    25    15
  Red       10   10     7     7
  Blond      3   30     5     8

, , Sex = Female

       Eye
Hair     Brown Blue Hazel Green
  Black     36    9     5     2
  Brown     66   34    29    14
  Red       16    7     7     7
  Blond      4   64     5     8
```

Note that this is a 3-dimensional array, with Hair being rows, Eye being columns, and Sex being layers. The data above show, for instance, that there are 25 men with brown hair and hazel eyes. Let's check this:

```
> HairEyeColor[2,3,1]
[1] 25
```

Let's fit a model (as noted, for the N_{ijk} rather than the p_{ijk}) in which hair and eye color are independent of gender, but not with each other, i.e. the model (25.16):

```
> fm <- loglin(HairEyeColor, list(c(1, 2),3),param=T,fit=T)
2 iterations: deviation 5.684342e-14
> fm
$lrt
[1] 19.85656

$pearson
[1] 19.56712

$df
[1] 15

$margin
$margin[[1]]
[1] "Hair" "Eye"

$margin[[2]]
[1] "Sex"

$fit
, , Sex = Female

        Eye
Hair          Brown       Blue      Hazel      Green
  Black 35.952703 10.574324   7.930743   2.643581
  Brown 62.917230 44.412162 28.550676 15.332770
  Red   13.746622  8.988176   7.402027   7.402027
  Blond  3.701014 49.699324   5.287162   8.459459


$param
$param$`(Intercept)`
[1] 2.494748

$param$Hair
      Black        Brown          Red        Blond
-0.3063649   0.9520081  -0.3471908  -0.2984523
```

```
$param$Eye
      Brown         Blue        Hazel        Green
 0.3611125   0.5112173  -0.2798778  -0.5924520


$param$Sex
       Male       Female
-0.0574957   0.0574957


$param$Hair.Eye
        Eye
Hair            Brown          Blue        Hazel        Green
   Black   0.9752132  -0.3986671   0.1047460  -0.6812921
   Brown   0.2764560  -0.2219556   0.1273068  -0.1818072
   Red     0.0546279  -0.5203601   0.0765790   0.3891532
   Blond  -1.3062970   1.1409828  -0.3086318   0.4739461
```

The Likelihood Ratio Test, which has a chi-square distribution with 16 degrees of freedom under H_0 here, is not rejected:

```
> 1 - pchisq(19.85656,16)
[1] 0.2267507
```

The p-value here is about 0.23. So, if you allow your analyses to be dictated by tests, you would accept the above model (which is plausible anyway). But even if you accept the hypothesis, you should still be interested in the magnitudes of the effects, shown above. Of course, for them to make sense, you need to exponentiate them back into probabilities, a fair amount of work.

At any rate, one generally entertains several different models for a data set, in much the same way as one considers several different sets of predictor variables in a regression setting. If one does not use significance testing for this, one goes through a less structured, but hopefully more fruitful, process of comparing numerical model results.

Let's again consider the case of brown-haired, hazel-eyed men. The model fit is

$$\exp\left(\hat{u} + \hat{v}_2 + \hat{w}_3 + \hat{r}_{23}\right) = \exp(2.494748 + 0.9520081 - 0.279877 - 0.0574957 + 0.1273068) = 25.44934 \tag{25.25}$$

Actually, we could have gotten this directly, from

```
> fm$fit[2,3,1]
[1] 25.44932
```

(there is a bit of roundoff error above), but you should look at (25.25) to make sure you understand what this number 25.44932 means. It is the estimate of $E(N_{231})$ under our hypothesized model, and (25.25) shows how this breaks down with respect to main effects and interactions.

In particular, the interaction term, 0.1273068, is rather small compared to most of the other numbers in the sum. This suggests that although there is some relation between hair and eye color, the relation is not strong. Of course, we'd have to browse through the other interaction values to make a stronger statement.

25.2.8   Obtaining Standard Errors

It was mentioned above that the output of **loglin()**, being significance test-oriented, does not report standard errors for the coefficients. However, there is a "trick" one can use to get them, as follows.[6]

As noted earlier, our cell counts follow a multinomial distribution. But there is another model, in which the cell counts are independent, each one following a Poisson distribution. (Note carefully that this is not saying that the factors are independent.) The "trick" is that it turns out that both models, multinomial and Poisson, yield the same MLEs. Thus even if we assume the multinomial model, we can run our analyses as if it follows the Poisson model.

The remaining point is to note that R's **glm()** function, used before for logistic regression by setting **family=binomial**, can be used here with **family=poisson**. Since **glm()** output reports standard errors, this workaround enables us to obtain standard errors for log-linear analysis.

25.3   Clustering

The **clustering** problem is somewhat like the reverse of the classification problem. In the latter, we our original data to enable ourselves to later input new data points and output class identities. With clustering, we use our original data to hunt for classes themselves.

Clustering is treated as merely a technique for exploration/description, for browsing through the data. It is not generally regarded as "estimating" anything (though we will return to this issue later).

25.3.1   K-Means Clustering

This is probably the oldest clustering method, and it is still in wide use.

---

[6]At the expense of computation, one can generally use the **bootstrap** for obtaining standard errors. See Section **??**.

25.3.1.1 The Algorithm

The method itself is quite simple, using an iterative algorithm. The user specifies **k**, the number of clusters he hopes to find, and at any step during the iteration process, the current **k** clusters are summarized by their centroids. (If we have m variables, then the centroid of a group is the m-element vector of means of those variables within this group.) We iterate the following:

1. For each data point, i.e. each row of our data matrix, determine which centroid this point is closest to.

2. Add this data point to the group corresponding to that centroid.

3. After all data points are processed in this manner, update the centroids to reflect the current group memberships.

4. Next iteration.

-]5.] Iterate until the centroids don't change much from one iteration to the next.

25.3.1.2 Example: the Baseball Player Data

We use the R function **kmeans()** here. Its simplest call form is

```
kmeans(x,k)
```

where **x** is our data matrix and **k** is as above.

Let's try it on our baseball player data Height, Weight and Age, which are in columns 4-6 of our data frame:

```
> head(baseball)
              Name Team        Position Height Weight   Age PosCategory
1   Adam_Donachie  BAL          Catcher     74    180 22.99     Catcher
2       Paul_Bako  BAL          Catcher     74    215 34.69     Catcher
3 Ramon_Hernandez  BAL          Catcher     72    210 30.78     Catcher
4   Kevin_Millar   BAL    First_Baseman     72    210 35.43   Infielder
5     Chris_Gomez  BAL    First_Baseman     73    188 35.71   Infielder
6   Brian_Roberts  BAL   Second_Baseman     69    176 29.39   Infielder
kmb <- kmeans(baseball[,4:6],4)
```

The return value, which we've stored here in **kmb**, is of class (of course) **"kmeans"**. One of its components is the final centroids:

```
> kmb$centers
    Height    Weight      Age
1 73.81267 202.6584 29.00882
2 72.45013 180.3181 27.90776
3 74.79147 221.7725 29.48867
4 76.30000 244.4571 29.04129
```

Cluster 2 seems interesting, consisting of shorter, lighter and younger players, while Cluster 4 seems the opposite. Clusters 1 and 3 are similar except in weight.

Let's see if these clusters correlate to player position. Note that we've consolidated the original Position variable into PosCategory, which consists of catchers, pitchers, infielders and outfielders. Let's take a look at the relation to the clusters. The **cluster** component of the object returned by the function shows which cluster each observations fell into. Here's are table of the results:

```
> kmb$size   # number of points in each cluster
[1] 363 371 211   70
> baseball$cls <- kmb$cluster
> table(baseball$cls,baseball$PosCategory)

   Catcher Infielder Outfielder Pitcher
  1      31        69         74     201
  2      32        42         51     143
  3       9        84         57     107
  4       4        15         12      84
```

Clusters 1 and 2 were the largest overall, but there is an interesting pattern here. Catchers were almost entirely in Clusters 1 and 2, while the other position categories, especially infielders, were more evenly dispersed.

25.3.2   Mixture Models

Even though k-means and other clustering methods don't claim to be estimating anything—again, they are just intended as exploratory tools—the basic motivation is that each cluster represents some multivariate distribution, e.g. trivariate normal in the baseball example above. Let q_i be the probability that an individual is in cluster i. Then the overall distribution has the form (19.2)—a mixture!

So, we really are modeling the population by a mixture of subpopulations, and the EM algorithm for mixtures (Section 19.5) can be used to estimate the centroids and the mixture weights q_i. in particular, the **mixtools** library includes a function **mvnormalmixEM()** which does exactly this.

25.3.3 Spectral Models

The word *spectral* here refers in turn to *spectrum*, which in linear algebra terms is the set of eigenvalues of a matrix. It turns out that this is useful for clustering.

The details are complex, and many variant algorithms exist, but here is an overview:

Start with a **similarity matrix** S, calculated from your original data matrix X. S_{ij} is defined in terms of some distance between rows i and j in X. One then computes

$$L = I - D^{-1/2}SD^{-1/2} \tag{25.26}$$

where D is a diagonal matrix whose (i,i) element is $\sum_j S_{ij}$.

One then finds an eigenvector v corresponding to the second-smallest eigenvalue of L. One then partitions the rows of X according to some criterion involving v.

Then one partitions those partitions! The same method is used, and in this way, the data points are chopped up into groups, our clusters. This is an example of what is called **hierarchical** clustering.

25.3.4 Other R Functions

The CRAN code repository includes a number of clustering packages. See the CRAN Task View: Cluster Analysis & Finite Mixture Models,[7] `http://cran.r-project.org/web/views/Cluster.html`.

25.3.5 Further Reading

UCD professor Catherine Yang is a specialist in clustering. See for instance Segmenting Customer Transactions Using a Pattern-Based Clustering Approach, *Proceedings of the Thrid IEEE International Conference on Data Mining.*

25.4 Simpson's (Non-)Paradox

Every serious user of statistics must keep in mind **Simpson's Paradox** at all times. It's an example of "what can go wrong" in multivariate studies. Its central importance is reflected, for instance, in an extensive entry in Wikipedia.

---

[7]CRAN has a number of these "task views" on various topics.

And yet...a very simple precaution will enable you to avoid the problem.

So, what is this paradox? In short, the relation between variables X and Y can be positive, conditional on every level of a third variable Z, and yet be negative overall. (Or vice versa, i.e. change from negative to positive.)

As you will see below, it arguably is not really a paradox at all. Instead, the real problem is a failure to examine the more important variables before the less important ones.

25.4.1   Example: UC Berkeley Graduate Admission Data

This is a very famous data set, much analyzed over the years, well known partly because a lawsuit was involved.

25.4.1.1   Overview

The suit claimed that UC Berkeley graduate admissions practices discriminated against women. The plaintiffs pointed out that male applicants had a 44% acceptance rate, compared to only 35% for women.

On the surface, the claims of discrimination seemed quite plausible. Here X, called Admit, was being admitted to graduate school, and Y, called Gender, was an indicator variable for being male. X and Y appeared to be positively related.

However, things changed a lot when a third variable Z, called Dept, was brought in. This coded which department the applicant applied to. Upon closer inspection by UCB statistics professors, it was found that conditional on Z, X and Y were actually <u>negatively</u> related for (almost) every value of Z. In other words, it turned out that in (almost) every department, the women applicants were actually being slightly fared better than the men.[8]

25.4.1.2   Log-Linear Analysis

Let's analyze this with a log-linear model. As it happens, the data is one of R's built-in data sets, **UCBAdmissions** (which we will copy to a new variable **ucb**, for brevity):

```
> ucb <- UCBAdmissions
```

As noted, our "Z," is Dept; six departments were represented in the data, i.e. six different graduate programs.

---

[8]One department was an exception, but the difference was small and possibly due to sampling variation. Thus most accounts treat this as an instance of Simpson's Paradox, rather than "almost" an example.

We can easily determine how many applicants there were:

```
> sum(ucb)
[1] 4526
```

Since the Admit and Gender variables had two levels each, there were $2 \times 2 \times 6 = 24$ different cells in the table. In other words, our log-linear analysis is modeling 24 different cell probabilities (actually 23, since they must sum to 1.0).

So, even a saturated model would easily satisfy our rule of thumb (Section 22.15.5) regarding overfitting, which recommended keeping the number of parameters below the square root of the sample size.

Let's fit a model that includes all two-way interactions.[9]

```
> llout <- loglin(ucb,list(1:2,c(1,3),2:3),param=T)
2 iterations: deviation 3.552714e-15
> llout$param
$'(Intercept)'
[1] 4.802457

$Admit
  Admitted    Rejected
-0.3212111   0.3212111

$Gender
      Male      Female
 0.3287569  -0.3287569

$Dept
          A            B            C            D            E
F
 0.15376258  -0.76516841   0.53972054   0.43021534  -0.02881353  -0.32971651

$Admit.Gender
         Gender
Admit            Male        Female
  Admitted  -0.02493703   0.02493703
  Rejected   0.02493703  -0.02493703

$Admit.Dept
```

---

[9]A full model was fit, with similar patterns.

```
          Dept
Admit
             A           B           C           D           E
F
 Admitted   0.6371804   0.6154772   0.005914624  -0.01010004  -0.2324371  -1.016035
 Rejected  -0.6371804  -0.6154772  -0.005914624   0.01010004   0.2324371
1.016035


$Gender.Dept
       Dept
Gender          A           B           C           D           E
F
  Male    0.6954949   1.232906  -0.6370521  -0.2836477  -0.7020726  -0.3056282
  Female -0.6954949  -1.232906   0.6370521   0.2836477   0.7020726   0.3056282
```

Again, there is a lot here, but it's not hard to spot the salient numbesr. Let's look at the main effects first, labeled **$Admit**. **$Gender** and **$Dept**, near the top of the output, reproduced here for convenience:

```
$Admit
  Admitted    Rejected
-0.3212111   0.3212111


$Gender
      Male      Female
 0.3287569  -0.3287569


$Dept
          A           B           C           D           E
F
 0.15376258  -0.76516841   0.53972054   0.43021534  -0.02881353  -0.32971651
```

One sees that the main effect for Gender, ±0.3287569, is small compared to the main effects for some of the departments. So some departments attract a lot of applicants, notably C, while others such as F attract fewer than average. If this varies by gender, then it's important to take department into account in our analysis.

In view of that latter fact, in viewing the two-way interactions, consider the interaction between Admit and Dept. The Admit values were strongly positive for Departments A and B, small for C and D, and moderately to strongly negative for E and F. In other words, A and B were especially easy to get into, while E and F were particularly difficult.

But now turn to the Gender.Dept interaction, showing the pattern of which departments were applied to by which genders. The men were extra likely to apply to Departments A and B, while for the women it was C, D, E and F. In other words, men were more likely to apply to the easier departments, while the women were more likely to apply to the difficult ones! No wonder the women seemed to be faring poorly overall in the data presented in the lawsuit, which did not break things down by department.

The Admit.Gender interaction says it all: Everything else equal, the women were slightly more likely to be admitted than men.

25.4.2   Toward Making It Simpson's NON-Paradox

The above example shows that Simpson's Paradox really isn't a paradox at all. The so-called "anomaly" occurred simply because an omitted variable (department) was a stronger factor than was one of the included variables (gender).

Indeed, it can be argued that the "paradox," i.e. the reversal of signs, wouldn't occur if one used the correct ordering when bringing factors into consideration, as follows. Let's use the UCB data above to make this concrete.

Recall the method of *forward stepwise regression* (Section 22.15.4.1). The analog here might be start by considering just two of the three variables at a time, before looking at three-variable models. For the UCB data, we would first look at Admit and Gender, then look at Admit and Dept. For example, we would fit the latter model by running

```
> loglin(ucb,list(c(1,3)),param=TRUE)
```

The results are not shown here, but they reveal that indeed Dept has a much stronger association with Admit than does Gender. We would thus take Admit-Dept as our "first-step" model, and then add Gender to get our second-step model (again, "step" as in the spirit of stepwise regression). That would produce the model we analyzed above, WITH NO PARADOXICAL RESULT.

Appendix A

R Quick Start

Here we present a quick introduction to the R data/statistical programming language. Further learning resources are listed at `http://heather.cs.ucdavis.edu//r.html`.

R syntax is similar to that of C. It is object-oriented (in the sense of *encapsulation*, *polymorphism* and everything being an object) and is a functional language (i.e. almost no side effects, every action is a function call, etc.).

A.1   Correspondences

aspect	C/C++	R
assignment	=	`<-` (or =)
array terminology	array	vector, matrix, array
subscripts	start at 0	start at 1
array notation	m[2][3]	m[2,3]
2-D array storage	row-major order	column-major order
mixed container	struct, members accessed by .	list, members acessed by $ or [[ ]]
return mechanism	return	return() or last value computed
primitive types	int, float, double, char, bool	integer, float, double, character, logical
logical values	true, false	TRUE, FALSE (abbreviated T, F)
mechanism for combining modules	include, link	library()
run method	batch	interactive, batch
comment symbol	//	#

A.2   Starting R

To invoke R, just type "R" into a terminal window, e.g. **xterm** in Linux or Macs, **CMD** in Windows.

If you prefer to run from an IDE, you may wish to consider ESS for Emacs, StatET for Eclipse or RStudio, all open source. ESS is the favorite among the "hard core coder" types, while the colorful, easy-to-use, RStudio is a big general crowd pleaser. If you are already an Eclipse user, StatET will be just what you need.[1]

R is normally run in interactive mode, with > as the prompt. Among other things, that makes it easy to try little experiments to learn from; remember my slogan, "When in doubt, try it out!" For batch work, use **Rscript**, which is in the R package.

A.3   First Sample Programming Session

Below is a commented R session, to introduce the concepts. I had a text editor open in another window, constantly changing my code, then loading it via R's **source()** command. The original contents of the file **odd.R** were:

```
1  oddcount <- function(x)  {
2     k <- 0  # assign 0 to k
3     for (n in x)  {
4        if (n %% 2 == 1) k <- k+1   # %% is the modulo operator
5     }
6     return(k)
7  }
```

By the way, we could have written that last statement as simply

```
    k
```

because the last computed value of an R function is returned automatically. This is actually preferred style in the R community.

The R session is shown below. You may wish to type it yourself as you go along, trying little experiments of your own along the way.[2]

---

[1]I personally use **vim**, as I want to have the same text editor no matter what kind of work I am doing. But I have my own macros to help with R work.

[2]The source code for this file is at `https://raw.githubusercontent.com/matloff/probstatbook/master/R5MinIntro.tex` You can download the file, and copy/paste the text from there.

```
 1 > source("odd.R")  # load code from the given file
 2 > ls()  # what objects do we have?
 3 [1] "oddcount"
 4 > # what kind of object is oddcount (well, we already know)?
 5 > class(oddcount)
 6 [1] "function"
 7 > # while in interactive mode, and not inside a function, can print
 8 > # any object by typing its name; otherwise use print(), e.g. print(x+y)
 9 > oddcount  # a function is an object, so can print it
10 function(x)  {
11    k <- 0  # assign 0 to k
12    for (n in x)  {
13       if (n %% 2 == 1) k <- k+1  # %% is the modulo operator
14    }
15    return(k)
16 }
17
18 > # let's test oddcount(), but look at some properties of vectors first
19 > y <- c(5,12,13,8,88)  # c() is the concatenate function
20 > y
21 [1]   5 12 13   8 88
22 > y[2]  # R subscripts begin at 1, not 0
23 [1] 12
24 > y[2:4]  # extract elements 2, 3 and 4 of y
25 [1] 12 13   8
26 > y[c(1,3:5)]  # elements 1, 3, 4 and 5
27 [1]   5 13   8 88
28 > oddcount(y)  # should report 2 odd numbers
29 [1] 2
30
31 > # change code (in the other window) to vectorize the count operation,
32 > # for much faster execution
33 > source("odd.R")
34 > oddcount
35 function(x)  {
36    x1 <- (x %% 2 == 1)  # x1 now a vector of TRUEs and FALSEs
37    x2 <- x[x1]  # x2 now has the elements of x that were TRUE in x1
38    return(length(x2))
39 }
40
```

```
41  > # try it on subset of y, elements 2 through 3
42  > oddcount(y[2:3])
43  [1] 1
44  > # try it on subset of y, elements 2, 4 and 5
45  > oddcount(y[c(2,4,5)])
46  [1] 0
47
48  > # further compactify the code
49  > source("odd.R")
50  > oddcount
51  function(x)  {
52     length(x[x %% 2 == 1])  # last value computed is auto returned
53  }
54  > oddcount(y)  # test it
55  [1] 2
56
57  # and even more compactification, making use of the fact that TRUE and
58  # FALSE are treated as 1 and 0
59  > oddcount <- function(x) sum(x %% 2 == 1)
60  # make sure you understand the steps that that involves:  x is a vector,
61  # and thus x %% 2 is a new vector, the result of applying the mod 2
62  # operation to every element of x; then x %% 2 == 1 applies the == 1
63  # operation to each element of that result, yielding a new vector of TRUE
64  # and FALSE values; sum() then adds them (as 1s and 0s)
65
66  # we can also determine which elements are odd
67  > which(y %% 2 == 1)
68  [1] 1 3
```

Note that, as I like to say, "the function of the R function **function()** is to produce functions!" Thus assignment is used. For example, here is what **odd.R** looked like at the end of the above session:

```
1  oddcount <- function(x)  {
2     x1 <- x[x %% 2 == 1]
3     return(list(odds=x1, numodds=length(x1)))
4  }
```

We created some code, and then used **function()** to create a function object, which we assigned to **oddcount**.

A.4 Vectorization

Note that we eventually **vectorized** our function **oddcount()**. This means taking advantage of
the vector-based, functional language nature of R, exploiting R's built-in functions instead of loops.
This changes the venue from interpreted R to C level, with a potentially large increase in speed.
For example:

```
1  > x <- runif(1000000)  # 1000000 random numbers from the interval (0,1)
2  > system.time(sum(x))
3     user  system elapsed
4    0.008   0.000   0.006
5  > system.time({s <- 0; for (i in 1:1000000) s <- s + x[i]})
6     user  system elapsed
7    2.776   0.004   2.859
```

A.5 Second Sample Programming Session

A matrix is a special case of a vector, with added class attributes, the numbers of rows and columns.

```
1  > # "rowbind() function combines rows of matrices; there's a cbind() too
2  > m1 <- rbind(1:2,c(5,8))
3  > m1
4        [,1] [,2]
5  [1,]     1    2
6  [2,]     5    8
7  > rbind(m1,c(6,-1))
8        [,1] [,2]
9  [1,]     1    2
10 [2,]     5    8
11 [3,]     6   -1
12
13 > # form matrix from 1,2,3,4,5,6, in 2 rows; R uses column-major storage
14 > m2 <- matrix(1:6,nrow=2)
15 > m2
16        [,1] [,2] [,3]
17 [1,]     1    3    5
18 [2,]     2    4    6
19 > ncol(m2)
20 [1] 3
21 > nrow(m2)
```

```
22  [1] 2
23  > m2[2,3]   # extract element in row 2, col 3
24  [1] 6
25  # get submatrix of m2, cols 2 and 3, any row
26  > m3 <- m2[,2:3]
27  > m3
28       [,1] [,2]
29  [1,]    3    5
30  [2,]    4    6
31
32  # or write to that submatrix
33  > m2[,2:3] <- cbind(c(5,12),c(8,0))
34  > m2
35       [,1] [,2] [,3]
36  [1,]    1    5    8
37  [2,]    2   12    0
38
39  > m1 * m3   # elementwise multiplication
40       [,1] [,2]
41  [1,]    3   10
42  [2,]   20   48
43  > 2.5 * m3   # scalar multiplication (but see below)
44       [,1] [,2]
45  [1,]  7.5 12.5
46  [2,] 10.0 15.0
47  > m1 %*% m3   # linear algebra matrix multiplication
48       [,1] [,2]
49  [1,]   11   17
50  [2,]   47   73
51
52  > # matrices are special cases of vectors, so can treat them as vectors
53  > sum(m1)
54  [1] 16
55  > ifelse(m2 %%3 == 1,0,m2) # (see below)
56       [,1] [,2] [,3]
57  [1,]    0    3    5
58  [2,]    2    0    6
```

A.6 Recycling

The "scalar multiplication" above is not quite what you may think, even though the result may be. Here's why:

In R, scalars don't really exist; they are just one-element vectors. However, R usually uses **recycling**, i.e. replication, to make vector sizes match. In the example above in which we evaluated the express `2.5 * m3`, the number 2.5 was recycled to the matrix

$$\begin{pmatrix} 2.5 & 2.5 \\ 2.5 & 2.5 \end{pmatrix} \tag{A.1}$$

in order to conform with **m3** for (elementwise) multiplication.

A.7 More on Vectorization

The **ifelse()** function is another example of vectorization. Its call has the form

```
ifelse(boolean vectorexpression1, vectorexpression2, vectorexpression3)
```

All three vector expressions must be the same length, though R will lengthen some via recycling. The action will be to return a vector of the same length (and if matrices are involved, then the result also has the same shape). Each element of the result will be set to its corresponding element in **vectorexpression2** or **vectorexpression3**, depending on whether the corresponding element in **vectorexpression1** is TRUE or FALSE.

In our example above,

```
> ifelse(m2 %%3 == 1,0,m2) # (see below)
```

the expression `m2 %%3 == 1` evaluated to the boolean matrix

$$\begin{pmatrix} T & F & F \\ F & T & F \end{pmatrix} \tag{A.2}$$

(TRUE and FALSE may be abbreviated to T and F.)

The 0 was recycled to the matrix

$$\begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \tag{A.3}$$

while **vectorexpression3**, **m2**, evaluated to itself.

A.8   Third Sample Programming Session

This time, we focus on vectors and matrices.

```
> m <- rbind(1:3,c(5,12,13))
> m
     [,1] [,2] [,3]
[1,]    1    2    3
[2,]    5   12   13
> t(m)  # transpose
     [,1] [,2]
[1,]    1    5
[2,]    2   12
[3,]    3   13
> ma <- m[,1:2]
> ma
     [,1] [,2]
[1,]    1    2
[2,]    5   12
> rep(1,2)  # "repeat," make multiple copies
[1] 1 1
> ma %*% rep(1,2)   # matrix multiply
     [,1]
[1,]    3
[2,]   17
> solve(ma,c(3,17))  # solve linear system
[1] 1 1
> solve(ma)  # matrix inverse
     [,1] [,2]
[1,]  6.0 -1.0
[2,] -2.5  0.5
```

A.9   Default Argument Values

Consider the **sort()** function, which is built-in to R, though the following points hold for any function, including ones you write yourself.

The online help for this function, invoked by

```
> ?sort
```

shows that the call form (the simplest version) is

```
sort(x, decreasing = FALSE, ...)
```

Here is an example:

```
> x <- c(12,5,13)
> sort(x)
[1]   5 12 13
> sort(x,decreasing=FALSE)
[1] 13 12   5
```

So, the default is to sort in ascending order, i.e. the argument **decreasing** has TRUE as its default value. If we want the default, we need not specify this argument. If we want a descending-order sort, we must say so.

A.10 The R List Type

The R **list** type is, after vectors, the most important R construct. A list is like a vector, except that the components are generally of mixed types.

A.10.1 The Basics

Here is example usage:

```
> g <- list(x = 4:6, s = "abc")
> g
$x
[1] 4 5 6

$s
[1] "abc"

> g$x  # can reference by component name
[1] 4 5 6
> g$s
[1] "abc"
```

```
> g[[1]]   # can reference by index, but note double brackets
[1] 4 5 6
> g[[2]]
[1] "abc"
> for (i in 1:length(g)) print(g[[i]])
[1] 4 5 6
[1] "abc"

# now have ftn oddcount() return odd count AND the odd numbers themselves,
# using the R list type
> source("odd.R")
> oddcount
function(x)  {
   x1 <- x[x %% 2 == 1]
   list(odds=x1, numodds=length(x1))  # last computed value auto-returned
}
> # R's list type can contain any type; components delineated by $
> oddcount(y)
$odds
[1]   5 13

$numodds
[1] 2

> ocy <- oddcount(y)  # save the output in ocy, which will be a list
> ocy
$odds
[1]   5 13

$numodds
[1] 2

> ocy$odds
[1]   5 13
> ocy[[1]]   # can get list elements using [[ ]] instead of $
[1]   5 13
> ocy[[2]]
[1] 2
```

A.10.2   The Reduce() Function

One often needs to combine elements of a list in some way. One approach to this is to use **Reduce()**:

```
> x <- list(4:6,c(1,6,8))
> x
[[1]]
[1] 4 5 6

[[2]]
[1] 1 6 8

> sum(x)
Error in sum(x) : invalid 'type' (list) of argument
> Reduce(sum,x)
[1] 30
```

Here **Reduce()** cumulatively applied R's **sum()** to **x**. Of course, you can use it with functions you write yourself too.

Continuing the above example:

```
> Reduce(c,x)
[1] 4 5 6 1 6 8
```

A.10.3   S3 Classes

R is an object-oriented (and functional) language. It features two types of classes, S3 and S4. I'll introduce S3 here.

An S3 object is simply a list, with a class name added as an *attribute*:

```
> j <- list(name="Joe", salary=55000, union=T)
> class(j) <- "employee"
> m <- list(name="Joe", salary=55000, union=F)
> class(m) <- "employee"
```

So now we have two objects of a class we've chosen to name **"employee"**. Note the quotation marks.

We can write class *generic functions*:

```
> print.employee <- function(wrkr) {
```

```
+       cat(wrkr$name,"\n")
+       cat("salary",wrkr$salary,"\n")
+       cat("union␣member",wrkr$union,"\n")
+ }
> print(j)
Joe
salary 55000
union member TRUE
> j
Joe
salary 55000
union member TRUE
```

What just happened? Well, **print()** in R is a *generic* function, meaning that it is just a placeholder for a function specific to a given class. When we printed **j** above, the R interpreter searched for a function **print.employee()**, which we had indeed created, and that is what was executed. Lacking this, R would have used the print function for R lists, as before:

```
> rm(print.employee)   # remove the function, to see what happens with print
> j
$name
[1] "Joe"

$salary
[1] 55000

$union
[1] TRUE

attr(,"class")
[1] "employee"
```

A.11   Some Workhorse Functions

```
> m <- matrix(sample(1:5,12,replace=TRUE),ncol=2)
> m
[,1] [,2]
[1,]     2     1
[2,]     2     5
[3,]     5     4
```

```
[4,]     5     1
[5,]     2     1
[6,]     1     3
# call sum() on each row
> apply(m,1,sum)
[1] 3 7 9 6 3 4
# call sum() on each column
> apply(m,2,sum)
[1] 17 15
> f <- function(x) sum(x[x >= 4])
# call f() on each row
> apply(m,1,f)
[1] 0 5 9 5 0 0
> l <- list(x = 5, y = 12, z = 13)
# apply the given funciton to each element of l, producing a new list
> lapply(l,function(a) a+1)
$x
[1] 6

$y
[1] 13

$z
[1] 14
# group the first column of m by the second
> sout <- split(m[,1],m[,2])
> sout
$`1`
[1] 2 5 2
$`3`
[1] 1
$`4`
[1] 5

[1] 2
# find the size of each group, by applying the length() function
> lapply(sout,length)
$`1`
[1] 3
```

```
[1] 1
$'4'
[1] 1

$'5'
[1] 1
# like lapply(), but sapply() attempts to make vector output
> sapply(sout,length)
1 3 4 5
3 1 1 1
```

A.12   Handy Utilities

R functions written by others, e.g. in base R or in the CRAN repository for user-contributed code, often return values which are class objects. It is common, for instance, to have lists within lists. In many cases these objects are quite intricate, and not thoroughly documented. In order to explore the contents of an object—even one you write yourself—here are some handy utilities:

- **names()**: Returns the names of a list.

- **str()**: Shows the first few elements of each component.

- **summary()**: General function. The author of a class **x** can write a version specific to **x**, i.e. **summary.x()**, to print out the important parts; otherwise the default will print some bare-bones information.

For example:

```
> z <- list(a = runif(50), b = list(u=sample(1:100,25), v="blue sky"))
> z
$a
 [1] 0.301676229 0.679918518 0.208713522 0.510032893 0.405027042
0.412388038
 [7] 0.900498062 0.119936222 0.154996457 0.251126218 0.928304164
0.979945937
[13] 0.902377363 0.941813898 0.027964137 0.992137908 0.207571134
0.049504986
[19] 0.092011899 0.564024424 0.247162004 0.730086786 0.530251779
0.562163986
[25] 0.360718988 0.392522242 0.830468427 0.883086752 0.009853107
0.148819125
```

```
[31] 0.381143870 0.027740959 0.173798926 0.338813042 0.371025885
0.417984331
[37] 0.777219084 0.588650413 0.916212011 0.181104510 0.377617399
0.856198893
[43] 0.629269146 0.921698394 0.878412398 0.771662408 0.595483477
0.940457376
[49] 0.228829858 0.700500359

$b
$b$u
 [1] 33 67 32 76 29  3 42 54 97 41 57 87 36 92 81 31 78 12 85 73 26 44
86 40 43

$b$v
[1] "blue␣sky"
> names(z)
[1] "a" "b"
> str(z)
List of 2
 $ a: num [1:50] 0.302 0.68 0.209 0.51 0.405 ...
 $ b:List of 2
  ..$ u: int [1:25] 33 67 32 76 29 3 42 54 97 41 ...
  ..$ v: chr "blue␣sky"
> names(z$b)
[1] "u" "v"
> summary(z)
  Length Class  Mode
a 50     -none- numeric
b  2     -none- list
```

A.13 Data Frames

Another workhorse in R is the *data frame*. A data frame works in many ways like a matrix, but differs from a matrix in that it can mix data of different modes. One column may consist of integers, while another can consist of character strings and so on. Within a column, though, all elements must be of the same mode, and all columns must have the same length.

We might have a 4-column data frame on people, for instance, with columns for height, weight, age and name—3 numeric columns and 1 character string column.

Technically, a data frame is an R list, with one list element per column; each column is a vector. Thus columns can be referred to by name, using the **$** symbol as with all lists, or by column number, as with matrices. The matrix **a[i,j]** notation for the element of **a** in row **i**, column **j**, applies to data frames. So do the **rbind()** and **cbind()** functions, and various other matrix operations, such as filtering.

Here is an example using the dataset **airquality**, built in to R for illustration purposes. You can learn about the data through R's online help, i.e.

```
> ?airquality
```

Let's try a few operations:

```
> names(airquality)
[1] "Ozone"   "Solar.R" "Wind"    "Temp"    "Month"   "Day"
> head(airquality)  # look at the first few rows
  Ozone Solar.R Wind Temp Month Day
1    41     190  7.4   67     5   1
2    36     118  8.0   72     5   2
3    12     149 12.6   74     5   3
4    18     313 11.5   62     5   4
5    NA      NA 14.3   56     5   5
6    28      NA 14.9   66     5   6
> airquality[5,3]  # wind on the 5th day
[1] 14.3
> airquality$Wind[3]  # same
[1] 12.6
> nrow(airquality)  # number of days observed
[1] 153
> ncol(airquality)  # number of variables
[1] 6
> airquality$Celsius <- (5/9) * (airquality[,4] - 32)  # new variable
> names(airquality)
[1] "Ozone"   "Solar.R" "Wind"    "Temp"    "Month"   "Day"     "Celsius"
> ncol(airquality)
[1] 7
> airquality[1:3,]
  Ozone Solar.R Wind Temp Month Day  Celsius
1    41     190  7.4   67     5   1 19.44444
2    36     118  8.0   72     5   2 22.22222
3    12     149 12.6   74     5   3 23.33333
> aqjune <- airquality[airquality$Month == 6,]  # filter op
```

```
> nrow(aqjune)
[1] 30
> mean(aqjune$Temp)
[1] 79.1
> write.table(aqjune,"AQJune")  # write data frame to file
> aqj <- read.table("AQJune",header=T)  # read it in
```

A.14 Graphics

R excels at graphics, offering a rich set of capabilities, from beginning to advanced. In addition to the functions in base R, extensive graphics packages are available, such as **lattice** and **ggplot2**.

One point of confusion for beginners involves saving an R graph that is currently displayed on the screen to a file. Here is a function for this, which I include in my R startup file, **.Rprofile**, in my home directory:

```
pr2file
function (filename)
{
    origdev <- dev.cur()
    parts <- strsplit(filename, ".", fixed = TRUE)
    nparts <- length(parts[[1]])
    suff <- parts[[1]][nparts]
    if (suff == "pdf") {
        pdf(filename)
    }
    else if (suff == "png") {
        png(filename)
    }
    else jpeg(filename)
    devnum <- dev.cur()
    dev.set(origdev)
    dev.copy(which = devnum)
    dev.set(devnum)
    dev.off()
    dev.set(origdev)
}
```

The code, which I won't go into here, mostly involves manipulation of various R graphics devices. I've set it up so that you can save to a file of type either PDF, PNG or JPEG, implied by the file

name you give.

A.15   Packages

The analog of a library in C/C++ in R is called a **package** (and often loosely referred to as a **library**). Some are already included in base R, while others can be downloaded, or written by yourself.

```
> library(parallel)  # load the package named 'parallel'
> ls(package:parallel)  # let's see what functions it gave us
 [1] "clusterApply"         "clusterApplyLB"      "clusterCall"
 [4] "clusterEvalQ"         "clusterExport"       "clusterMap"
 [7] "clusterSetRNGStream"  "clusterSplit"        "detectCores"
[10] "makeCluster"          "makeForkCluster"     "makePSOCKcluster"
[13] "mc.reset.stream"      "mcaffinity"          "mccollect"
[16] "mclapply"             "mcMap"               "mcmapply"
[19] "mcparallel"           "nextRNGStream"       "nextRNGSubStream"
[22] "parApply"             "parCapply"           "parLapply"
[25] "parLapplyLB"          "parRapply"           "parSapply"
[28] "parSapplyLB"          "pvec"                "setDefaultCluster"
[31] "splitIndices"         "stopCluster"
> ?pvec  # let's see how one of them works
```

The CRAN repository of contributed R code has thousands of R packages available. It also includes a number of "tables of contents" for specific areas, say time series, in the form of CRAN Task Views. See the R home page, or simply Google "CRAN Task View."

```
> install.packages("cts","~/myr")  # download into desired directory
--- Please select a CRAN mirror for use in this session ---
...
downloaded 533 Kb


The downloaded binary packages are in
        /var/folders/jk/dh9zkds97sj23kjcfkr5v6q00000gn/T//RtmplkKzOU/downloaded
> ?library
> library(cts,lib.loc="~/myr")

Attaching package:   cts
...
```

A.16 Other Sources for Learning R

There are tons of resources for R on the Web. You may wish to start with the links at `http://heather.cs.ucdavis.edu/~matloff/r.html`.

A.17 Online Help

R's **help()** function, which can be invoked also with a question mark, gives short descriptions of the R functions. For example, typing

```
> ?rep
```

will give you a description of R's **rep()** function.

An especially nice feature of R is its **example()** function, which gives nice examples of whatever function you wish to query. For instance, typing

```
> example ( wireframe ())
```

will show examples—R code and resulting pictures—of **wireframe()**, one of R's 3-dimensional graphics functions.

A.18 Debugging in R

The internal debugging tool in R, **debug()**, is usable but rather primitive. Here are some alternatives:

- The RStudio IDE has a built-in debugging tool.

- For Emacs users, there is **ess-tracebug**.

- The StatET IDE for R on Eclipse has a nice debugging tool. Works on all major platforms, but can be tricky to install.

- My own debugging tool, **debugR**, is extensive and easy to install, but for the time being is limited to Linux, Mac and other Unix-family systems. See *http://heather.cs.ucdavis.edu/debugR.html*.

A.19   Complex Numbers

If you have need for complex numbers, R does handle them. Here is a sample of use of the main functions of interest:

```
> za <- complex(real=2,imaginary=3.5)
> za
[1] 2+3.5i
> zb <- complex(real=1,imaginary=-5)
> zb
[1] 1-5i
> za * zb
[1] 19.5-6.5i
> Re(za)
[1] 2
> Im(za)
[1] 3.5
> za^2
[1] -8.25+14i
> abs(za)
[1] 4.031129
> exp(complex(real=0,imaginary=pi/4))
[1] 0.7071068+0.7071068i
> cos(pi/4)
[1] 0.7071068
> sin(pi/4)
[1] 0.7071068
```

Note that operations with complex-valued vectors and matrices work as usual; there are no special complex functions.

A.20   Further Reading

For further information about R as a programming language, there is my book, *The Art of R Programming: a Tour of Statistical Software Design*, NSP, 2011, as well as Hadley Wickham's *Advanced R*, Chapman and Hall, 2014.

For R's statistical functions, a plethora of excellent books is available. such as *The R Book* (2nd Ed.), Michael Crowley, Wiley, 2012. I also very much like *R in a Nutshell* (2nd Ed.), Joseph Adler, O'Reilly, 2012, and even Andrie de Vries' *R for Dummies*, 2012.

Appendix B

Review of Matrix Algebra

This book assumes the reader has had a course in linear algebra (or has self-studied it, always the better approach). This appendix is intended as a review of basic matrix algebra, or a quick treatment for those lacking this background.

B.1 Terminology and Notation

A **matrix** is a rectangular array of numbers. A **vector** is a matrix with only one row (a **row vector** or only one column (a **column vector**).

The expression, "the (i,j) element of a matrix," will mean its element in row i, column j.

Please note the following conventions:

- Capital letters, e.g. A and X, will be used to denote matrices and vectors.

- Lower-case letters with subscripts, e.g. $a_{2,15}$ and x_8, will be used to denote their elements.

- Capital letters with subscripts, e.g. A_{13}, will be used to denote submatrices and subvectors.

If A is a **square** matrix, i.e., one with equal numbers n of rows and columns, then its **diagonal** elements are a_{ii}, i = 1,...,n.

A square matrix is called **upper-triangular** if $a_{ij} = 0$ whenever $i > j$, with a corresponding definition for **lower-triangular** matrices.

The **norm** (or **length**) of an n-element vector X is

$$\| X \| = \sqrt{\sum_{i=1}^{n} x_i^2} \tag{B.1}$$

B.1.1   Matrix Addition and Multiplication

- For two matrices have the same numbers of rows and same numbers of columns, addition is defined elementwise, e.g.

$$\begin{pmatrix} 1 & 5 \\ 0 & 3 \\ 4 & 8 \end{pmatrix} + \begin{pmatrix} 6 & 2 \\ 0 & 1 \\ 4 & 0 \end{pmatrix} = \begin{pmatrix} 7 & 7 \\ 0 & 4 \\ 8 & 8 \end{pmatrix} \tag{B.2}$$

- Multiplication of a matrix by a **scalar**, i.e., a number, is also defined elementwise, e.g.

$$0.4 \begin{pmatrix} 7 & 7 \\ 0 & 4 \\ 8 & 8 \end{pmatrix} = \begin{pmatrix} 2.8 & 2.8 \\ 0 & 1.6 \\ 3.2 & 3.2 \end{pmatrix} \tag{B.3}$$

- The **inner product** or **dot product** of equal-length vectors X and Y is defined to be

$$\sum_{k=1}^{n} x_k y_k \tag{B.4}$$

- The product of matrices A and B is defined if the number of rows of B equals the number of columns of A (A and B are said to be **conformable**). In that case, the (i,j) element of the product C is defined to be

$$c_{ij} = \sum_{k=1}^{n} a_{ik} b_{kj} \tag{B.5}$$

For instance,

$$\begin{pmatrix} 7 & 6 \\ 0 & 4 \\ 8 & 8 \end{pmatrix} \begin{pmatrix} 1 & 6 \\ 2 & 4 \end{pmatrix} = \begin{pmatrix} 19 & 66 \\ 8 & 16 \\ 24 & 80 \end{pmatrix} \tag{B.6}$$

It is helpful to visualize c_{ij} as the inner product of row i of A and column j of B, e.g. as shown in bold face here:

$$\begin{pmatrix} \mathbf{7} & \mathbf{6} \\ 0 & 4 \\ 8 & 8 \end{pmatrix} \begin{pmatrix} \mathbf{1} & 6 \\ \mathbf{2} & 4 \end{pmatrix} = \begin{pmatrix} \mathbf{19} & 66 \\ 8 & 16 \\ 24 & 80 \end{pmatrix} \tag{B.7}$$

• Matrix multiplication is associative and distributive, but in general not commutative:

$$A(BC) = (AB)C \tag{B.8}$$

$$A(B + C) = AB + AC \tag{B.9}$$

$$AB \neq BA \tag{B.10}$$

B.2 Matrix Transpose

• The transpose of a matrix A, denoted A' or A^T, is obtained by exchanging the rows and columns of A, e.g.

$$\begin{pmatrix} 7 & 70 \\ 8 & 16 \\ 8 & 80 \end{pmatrix}' = \begin{pmatrix} 7 & 8 & 8 \\ 70 & 16 & 80 \end{pmatrix} \tag{B.11}$$

• If $A + B$ is defined, then

$$(A + B)' = A' + B' \tag{B.12}$$

• If A and B are conformable, then

$$(AB)' = B'A' \tag{B.13}$$

B.3   Linear Independence

Equal-length vectors $X_1,...,X_k$ are said to be **linearly independent** if it is impossible for

$$a_1 X_1 + ... + a_k X_k = 0 \tag{B.14}$$

unless all the a_i are 0.

B.4   Determinants

Let A be an $n \times n$ matrix. The definition of the determinant of A, det(A), involves an abstract formula featuring permutations. It will be omitted here, in favor of the following computational method.

Let $A_{-(i,j)}$ denote the submatrix of A obtained by deleting its ith row and j^{th} column. Then the determinant can be computed recursively across the kth row of A as

$$det(A) = \sum_{m=1}^{n} (-1)^{k+m} det(A_{-(k,m)}) \tag{B.15}$$

where

$$det \begin{pmatrix} s & t \\ u & v \end{pmatrix} = sv - tu \tag{B.16}$$

Generally, determinants are mainly of theoretical importance, but they often can clarify one's understanding of concepts.

B.5   Matrix Inverse

- The **identity** matrix I of size n has 1s in all of its diagonal elements but 0s in all off-diagonal elements. It has the property that $AI = A$ and $IA = A$ whenever those products are defined.

- The A is a square matrix and $AB = I$, then B is said to be the **inverse** of A, denoted A^{-1}. Then $BA = I$ will hold as well.

- A^{-1} exists if and only if its rows (or columns) are linearly independent.

- A^{-1} exists if and only if $det(A) \neq 0$.

- If A and B are square, conformable and invertible, then AB is also invertible, and

$$(AB)^{-1} = B^{-1}A^{-1} \tag{B.17}$$

A matrix U is said to be **orthogonal** if its rows each have norm 1 and are orthogonal to each other, i.e., their inner product is 0. U thus has the property that $UU' = I$ i.e., $U^{-1} = U$.

The inverse of a triangular matrix is easily obtained by something called **back substitution**.

Typically one does not compute matrix inverses directly. A common alternative is the **QR decomposition**: For a matrix A, matrices Q and R are calculated so that $A = QR$, where Q is an orthogonal matrix and R is upper-triangular.

If A is square and invertible, A^{-1} is easily found:

$$A^{-1} = (QR)^{-1} = R^{-1}Q' \tag{B.18}$$

Again, though, in some cases A is part of a more complex system, and the inverse is not explicitly computed.

B.6 Eigenvalues and Eigenvectors

Let A be a square matrix.[1]

- A scalar λ and a nonzero vector X that satisfy

$$AX = \lambda X \tag{B.19}$$

  are called an **eigenvalue** and **eigenvector** of A, respectively.

- If A is symmetric and real, then it is **diagonalizable**, i.e., there exists an orthogonal matrix U such that

$$U'AU = D \tag{B.20}$$

  for a diagonal matrix D. The elements of D are the eigenvalues of A, and the columns of U are the eigenvectors of A.

---

[1]For nonsquare matrices, the discussion here would generalize to the topic of **singular value decomposition**.

A different sufficient condition for B.20 is that the eigenvalues of A are distinct. In this case, U will not necessarily be orthogonal.

By the way, this latter sufficient condition shows that "most" square matrices are diagonalizable, if we treat their entries as continous random variables. Under such a circumstance, the probability of having repeated eigenvalues would be 0.

B.7   Rank of a Matrix

Definition: The rank of a matrix A is the maximal number of linearly independent columns in A.

Let's denote the rank of A by rk(A). Rank has the following properties:

- rk(A') = rk(A)

- Thus the rank of A is also the maximal number of linearly independent rows in A.

- Let A be $r \times s$. Then

$$rk(A) \leq \min(r, s) \tag{B.21}$$

- rk($A'A$) = rk(A)

B.8   Matrix Algebra in R

The R programming language has extensive facilities for matrix algebra, introduced here. Note by the way that R uses column-major order.

A linear algebra vector can be formed as an R vector, or as a one-row or one-column matrix.

```
> # constructing matrices
> a <- rbind(1:3,10:12)
> a
     [,1] [,2] [,3]
[1,]    1    2    3
[2,]   10   11   12
> b <- matrix(1:9,ncol=3)
> b
     [,1] [,2] [,3]
[1,]    1    4    7
```

```
[2,]     2     5     8
[3,]     3     6     9
# multiplication, etc.
> c <- a %*% b; c + matrix(c(1,-1,0,0,3,8),nrow=2)
      [,1] [,2] [,3]
[1,]    15    32    53
[2,]    67   167   274
> c %*% c(1,5,6)   # note 2 different c's
      [,1]
[1,]   474
# be careful! -- if you extract a submatrix that ends up
# consisting of a single row, the result will be a vector
# rather than a matrix, unless one specifies drop = FALSE:
> x <- rbind(3:5,c(6,2,9),c(5,12,13))
> class(x[1,])
[1] "numeric"
> x[1,]
[1] 3 4 5
> class(x[1,,drop=FALSE])
[1] "matrix"
> x[1,,drop=FALSE]
      [,1] [,2] [,3]
[1,]     3     4     5
> # transpose, inverse
> t(a)   # transpose
      [,1] [,2]
[1,]     1    10
[2,]     2    11
[3,]     3    12
> u <- matrix(runif(9),nrow=3)
> u
           [,1]        [,2]       [,3]
[1,] 0.08446154 0.86335270 0.6962092
[2,] 0.31174324 0.35352138 0.7310355
[3,] 0.56182226 0.02375487 0.2950227
> uinv <- solve(u)
> uinv
          [,1]      [,2]      [,3]
[1,]  0.5818482 -1.594123  2.576995
[2,]  2.1333965 -2.451237  1.039415
```

```
[3,] -1.2798127   3.233115 -1.601586
> u %*% uinv   # note roundoff error
               [,1]            [,2]            [,3]
[1,] 1.000000e+00 -1.680513e-16 -2.283330e-16
[2,] 6.651580e-17  1.000000e+00  4.412703e-17
[3,] 2.287667e-17 -3.539920e-17  1.000000e+00
> # eigenvalues and eigenvectors
> eigen(u)
$values
[1]   1.2456220+0.0000000i  -0.2563082+0.2329172i
-0.2563082-0.2329172i


$vectors
                  [,1]                    [,2]                    [,3]
[1,] -0.6901599+0i  -0.6537478+0.0000000i
-0.6537478+0.0000000i
[2,] -0.5874584+0i  -0.1989163-0.3827132i
-0.1989163+0.3827132i
[3,] -0.4225778+0i   0.5666579+0.2558820i
0.5666579-0.2558820i
> # diagonal matrices (off-diagonals 0)
> diag(3)
     [,1] [,2] [,3]
[1,]    1    0    0
[2,]    0    1    0
[3,]    0    0    1
> diag((c(5,12,13)))
     [,1] [,2] [,3]
[1,]    5    0    0
[2,]    0   12    0
[3,]    0    0   13
```

We can obtain matrix inverse using **solve()**, e.g.

```
> m <- rbind(1:2,3:4)
> m
     [,1] [,2]
[1,]    1    2
[2,]    3    4
> minv <- solve(m)
> minv
```

```
      [,1] [,2]
[1,] -2.0   1.0
[2,]   1.5 -0.5
> m %*% minv   # should get I back
      [,1]          [,2]
[1,]     1 1.110223e-16
[2,]     0 1.000000e+00
```

Note the roundoff error, even with this small matrix. We can try the QR method, provided to us in R via **qr()**. In fact, if we just want the inverse, **qr.solve()** will compute (B.18) for us.

We can in principle obtain rank from, for example, the **rank** component from the output of **qr()**. Note however that although rank is clearly defined in theory, the presence of roundoff error in computation make may rank difficult to determine reliably.

Appendix C

Introduction to the ggplot2 Graphics Package

C.1 Introduction

Hadley Wickham's **ggplot2** package is a hugely popular alternative to R's base graphics package. (Others include **lattice**, **ggobi** and so on.)

The **ggplot2** pacakge is an implementation of the ideas in the book, *The Grammar of Graphics*, by Leland Wilkison, whose goal was to set out a set of general unifying principles for the visualization of data. For this reason, **ggplot2** offers a more elegant and arguably more natural approach than does the base R graphics package.

The package has a relatively small number of primitive functions, making it relatively easy to master. But through combining these functions in various ways, a very large number of types of graphs may be produced. It is considered especially good in setting reasonable default values of parameters, and much is done without the user's asking. Legends are automatically added to graphs, for instance.

The package is quite extensive (only a few functions, but lots of options), and thus this document is merely a brief introduction.

C.2 Installation and Use

Download and install **ggplot2** with the usual **install.packages()** function, and then at each usage, load via **library()**. Here's what I did on my netbook:

```
# did once:
> install.packages("ggplot2","/home/nm/R")
# do each time I use the package (or set in .Rprofile)
> .libPaths("/home/nm/R")
> library(ggplot2)
```

C.3   Basic Structures

One operates in the following pattern:

- One begins with a call to **ggplot()**:

  ```
  > p <- ggplot(yourdataframe)
  ```

  or

  ```
  > p <- ggplot(yourdataframe,aes(yourargs))
  ```

  Here **yourdataframe** could have been read from a file, say using **read.table()**, or generated within the program. If your data is in the form of an R matrix, use **as.data.frame()** to convert it.

  The result **p** is an R S3 object of class **"ggplot"**, consisting of a component named **data**, and other components containing information about the plot.

  Note that at this point, though, there is nothing to plot (if we didn't call **aes()**).

- One adds features to—or even changes—the plot via the **+** operator, which of course is an overloaded version of R's built-in **+**, the function **"+.ggplot"()**.

  Each invocation of **+** adds a new *layer* to the graph, adding to the contents of the previous layer. Typically, each new layer adds new features to the graph, or changes old features. One might, for instance, superimpose several curves on the same graph, by adding one new layer per curve.[1]

  The idea of layering is partly motivated by reusability. One can save a lower layer in a variable (or on disk, using the R **save()** function), so that we can make a different graph, with different features, starting with the same layer.

  To actually display a plot, we print it, i.e. print **p**. Recall that in R, **print()** is a *generic* function, i.e. a stub for a class-specific one. In this case the latter does a plot. At this stage, we don't have anything to display yet, if we didn't call **aes()** above.

---

[1]There are ways to do this in a single layer, but let's not get too complex in this introductory document.

- The function **aes()** ("aesthetics") is used to specify graph attributes. For instance, in a scatter plot, which variable will be on the horizontal axis, and which on the vertical? What colors do we want for the points? Etc.

  We can call **aes()** at various layers, depending on how general (reusable, as noted above) we want each layer to be.

  So for instance we could use **aes()** to specify our data variables either when we call **ggplot()**, so these variables will be used in all operations, or when we later add a layer calling, say, **geom_point()**, to indicate data variables for this specific operation.

There are various types of objects that can be used as the second operand for the + operator. Examples are:

- **geoms** ("geometrics"): Geometric objects to be drawn, such as points, lines, bars, polygons and text.

- **position adjustments**: For instance, in a bar graph, this controls whether bars should be side by side, or stacked on top of each other.

- **facets**: Specifications to draw many graphs together, as panels in a large graph. You can have rows of panels, columns of panels, and rows and columns of panels.

- **themes**: Don't like the gray background in a graph? Want nicer labeling, etc.? You can set each of these individually, but one of the built-in themes, or a user-contributed one, can save you the trouble, or you can write one that you anticipate using a lot.

C.4   Example: Simple Line Graphs

```
> df1
  u v
1 0 3
2 2 4
3 5 5
> ggplot(df1) + geom_line(aes(x=u,y=v))
```
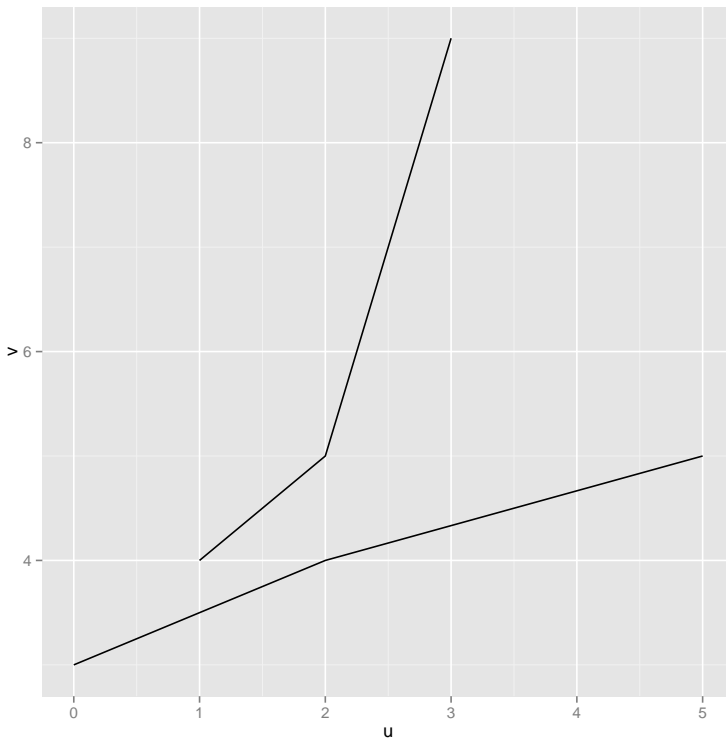
Here **aes()** was called from **geom_line()** rather than from **ggplot()**, so as to apply just to this line. The result is
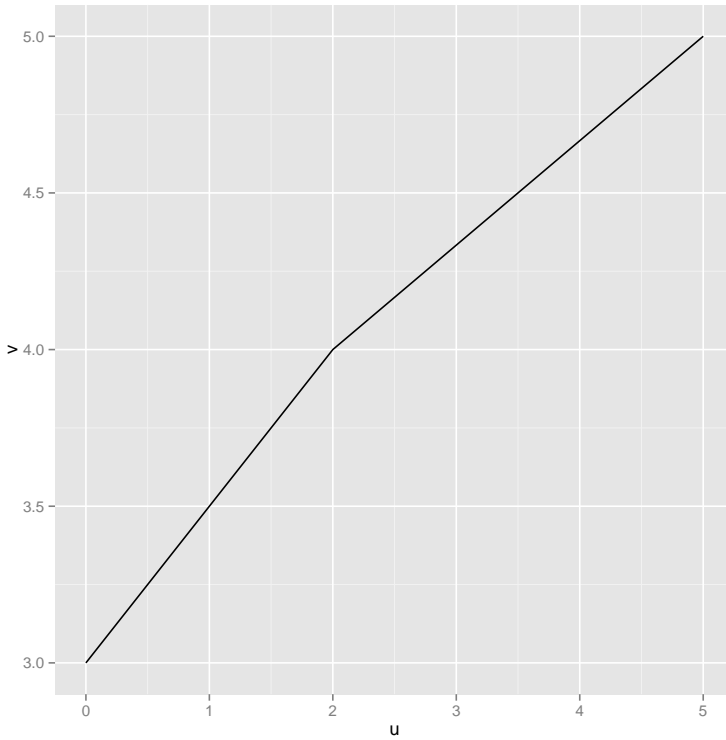
Now let's add a second line, from a *different* data frame:

```
> df2
  w z
1 1 4
2 2 5
3 3 9
ggplot(df1) + geom_line(aes(x=u,y=v)) + geom_line(data=df2,aes(x=w,y=z))
```

Here is the result:

It worked as long as we specified **data** for the second line.

Note that **ggplot2** automatically adjusted that second graph, to make room for the "taller" second line.

C.5 Example: Census Data

The data set here consists of programmers (software engineers, etc.) and electrical engineers in Silicon Valley, in the 2000 Census. I've removed those with less than a Bachelor's degree. The R object was a data frame named **pm**.
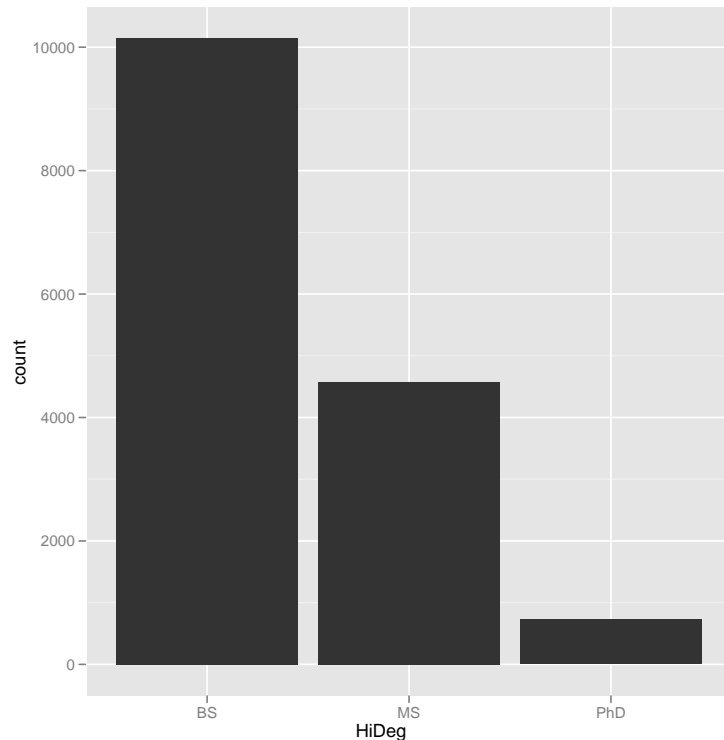
I first ran

```
p <- ggplot(pm)
```

to set up the **ggplot** object. Next, I typed

```
p + geom_histogram(aes(HiDeg))
```

which produced a histogram of a particular variable in the data (i.e. a particular column in the data frame), which was the highest-degree values of the workers:

Note that the + operation yields a new object of class **"ggplot"**. Since the generic print function for that class actually plots the graph, the graph did appear on the screen. I could have saved the new object in a variable if needed.

I then decided to do a scatter plot of salary versus age:

```
> p + geom_point(aes(x=Age,y=WgInc))
```

So, here is an example of the reusability mentioned earlier. For this small data set, it wasn't an issue, but some larger data sets can take a while to render, so you definitely want to save intermediate results for reuse.

Note the roles of **aes()** both here and in the previous example. I used it to specify for the geom what I wanted to do in that layer. Each geom has its own set of aesthetics one can specify. In the case of **geom_point()**, I need to specify which variable to use for the X- and Y-axes. There are other aesthetics for this geom that could be specified, as you'll see below.

This gave me this graph:
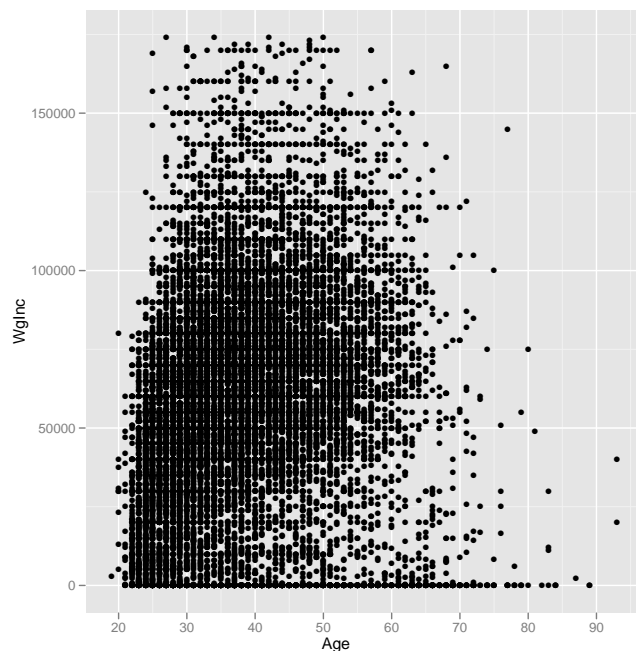
(As is often the case with large data sets, the points tend to "fill in" entire regions. One solution is to graph a random subset of the data, not done here. Data smoothing techniques can also be used. Similar comments apply to some of the graphs below.)

However, I wanted to separate the points according to highest degree level:

```
> p + geom_point(aes(x=Age,y=WgInc,color=HiDeg))
```

Here I have three data variables informing **aes()**: Age, wage income and highest degree. The argument **color** here means that I want the degree to be used for color coding the points:

So the orange points are for Bachelor's people, green for Master's and blue for PhDs. Note the color legend that was automatically included on the right.

Since some people might be viewing a black-and-white version of this document, I ran the command again, specifying coding the highest degree by point shape instead of point color:

```
p + geom_point(aes(x=Age,y=WgInc,shape=HiDeg))
```

Here **ggplot2** decided to use a circle, a triangle and a square to represent Bachelor's, Master's and PhD workers:
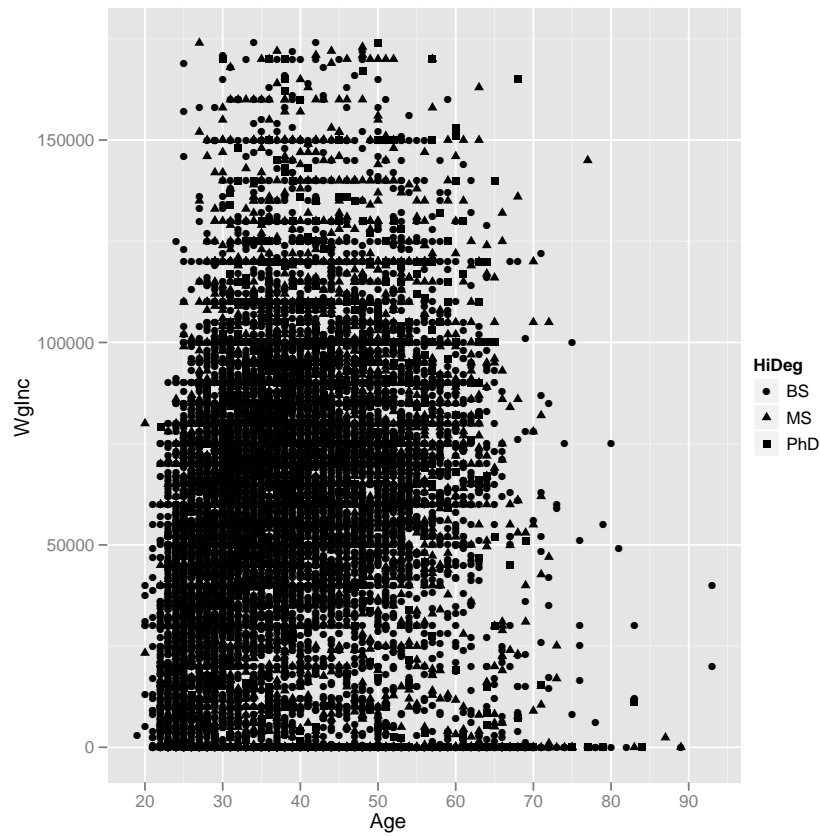
Since I'm interested in age discrimination in the industry, I decided to restrict my graph to those over age 40. The **ggplot2** package cleverly exploits the R **subset()** function, allowing me to write

```
p %+% subset(pm,Age > 40) + geom_point(aes(x=Age,y=WgInc,color=HiDeg))
```

The new operator **%+%** is again mapped to **"+.ggplot"()**. The result was

Look at all those 0-income PhDs! (There was also a business income variable, which I did not pursue here, so maybe some had high incomes after all.)

Even with color, it's a little hard to compare the three degree levels, so let's try faceting:

```
> pt <- p + geom_point(aes(x=Age, y=WgInc))
> pt + facet_grid(HiDeg ~ .)
```

Here I saved an overall graph of wage income versus age to **pt**, and then added faceting by degree. This instructed **ggplot2** to draw three graphs, one for each degree level, as three panels in the same large graph. The `HiDeg` $\sim$`.` argument (which you may recognize as similar notation to R's **lm()** function for linear models) states that I want HiDeg to define the rows of panels, without any variable defining columns. The result was:
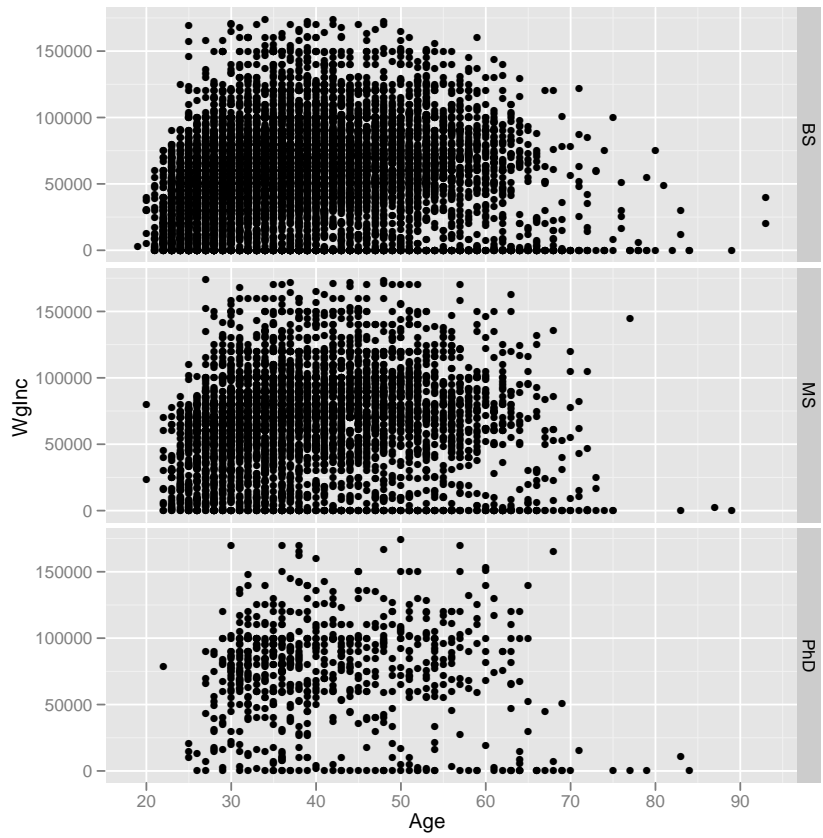
I tried a horizontal presentation too:
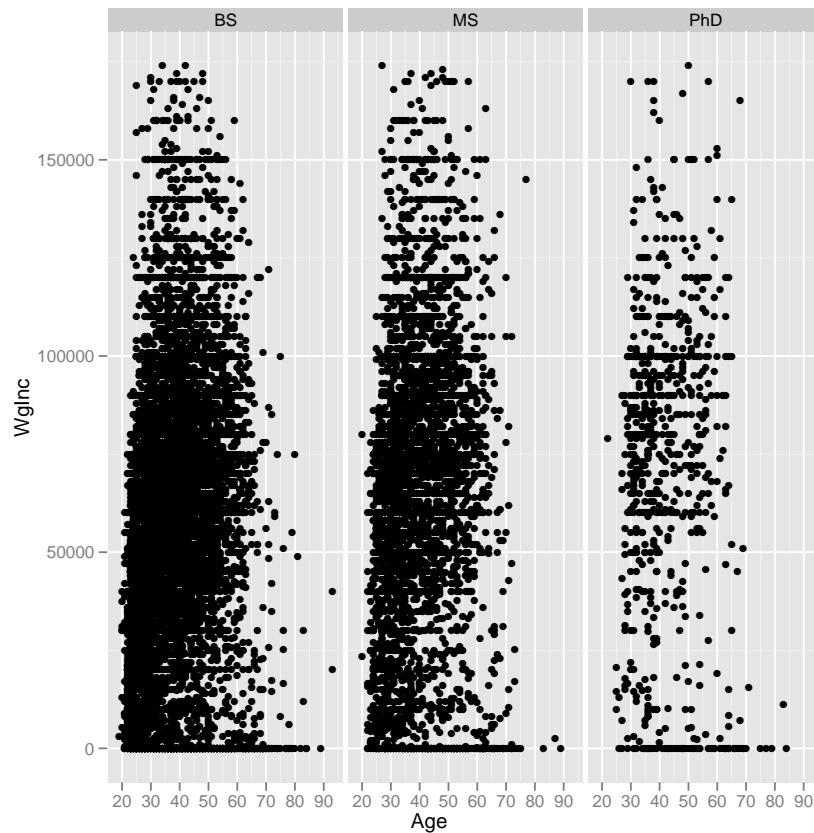
Note that there doesn't seem to be much relation between degree in salary, after age 35 or so.

Note that I could have combined operations, e.g.

```
> p + geom_point(aes(x=Age, y=WgInc)) + facet_grid(HiDeg ~ .)
```

to get the vertical set of panels, without directly going through the intermediate step of drawing the nonfaceted graph.

If there had been a Gender variable in the data, I could have defined rows and columns of panels:

```
> p + geom_point(aes(x=Age, y=WgInc)) + facet_grid(Gender ~ HiDeg)
```

C.6   Function Plots, Density Estimates and Smoothing

The **ggplot2** package has lots of ways to plot known functions and to do smoothing. Here are some examples:

```
# plot a function
```

```
sqfun <- function(t) t^2
# plot over the interval (0,2)
p <- ggplot(data.frame(x=c(0,2)),aes(x))
p + stat_function(fun=sqfun)

# compute and plot a density estimate
w <- rnorm(2500)
p <- ggplot(data.frame(w))
p + geom_density(aes(x=w))   # many other choices
y <- rnorm(2500,sd=2)
p + geom_density(aes(x=w)) + geom_density(aes(x=y))

# generate data from a quadratic function with added noise;
# pretend we don't know it's a quadratic function, and
# smooth the data to get an idea of what the function is like
t <- seq(0.01,2,0.01)
u <- t^2 + rnorm(n=100,sd=0.2)
d <- data.frame(t,u)
p <- ggplot(d)
p + geom_smooth(aes(x=t,y=u),method="loess")
```

C.7   What's Going on Inside

In order to use **ggplot2** most effectively, it helps to understand what happens one level down. Let's do so via the first example in this document:

```
> library(ggplot2)
> df1 <- data.frame(u = c(0,2,5), v = c(3:5))
> df1
  u v
1 0 3
2 2 4
3 5 5
> p <- ggplot(df1)
> p
Error: No layers in plot
```

By just typing **p**, we meant to print it, but there is nothing to print for now. Yet, even at this early stage, **p** has a lot in it:

```
> str(p)
List of 9
 $ data       :'data.frame':    3 obs. of  2 variables:
  ..$ u: num [1:3] 0 2 5
  ..$ v: int [1:3] 3 4 5
 $ layers     : list()
 $ scales     :Reference class 'Scales' [package "ggplot2"] with 1 fields
  ..$ scales: NULL
  ..and 21 methods, of which 9 are possibly relevant:
  ..   add, clone, find, get_scales, has_scale, initialize, input, n,
  ..   non_position_scales
 $ mapping    : list()
 $ theme      : list()
 $ coordinates:List of 1
  ..$ limits:List of 2
  .. ..$ x: NULL
  .. ..$ y: NULL
  ..- attr(*, "class")= chr [1:2] "cartesian" "coord"
 $ facet      :List of 1
  ..$ shrink: logi TRUE
  ..- attr(*, "class")= chr [1:2] "null" "facet"
 $ plot_env   :<environment: R_GlobalEnv>
 $ labels     : list()
 - attr(*, "class")= chr [1:2] "gg" "ggplot"
```

You can see that **p** is indeed a class object, consisting of a list of various components, some of which themselves are lists. Note the **data** component, which sure enough does consist of the data frame we had specified.

Note by the way that the **layer** component, i.e. **p$layers**, is empty, resulting in our failure to plot when we tried to do so above.

Now, let's add a geom:

```
> p1 <- p + geom_line(aes(x=u,y=v))
> str(p1)
List of 9
 $ data       :'data.frame':    3 obs. of  2 variables:
  ..$ u: num [1:3] 0 2 5
  ..$ v: int [1:3] 3 4 5
 $ layers     :List of 1
  ..$ :Classes 'proto', 'environment' <environment: 0x96d8264>
```

```
$ scales     :Reference class 'Scales' [package "ggplot2"] with 1 fields
  ..$ scales: list()
  ..and 21 methods, of which 9 are possibly relevant:
  ..  add, clone, find, get_scales, has_scale, initialize, input, n,
  ..  non_position_scales
$ mapping    : list()
$ theme      : list()
$ coordinates:List of 1
  ..$ limits:List of 2
  .. ..$ x: NULL
  .. ..$ y: NULL
  ..- attr(*, "class")= chr [1:2] "cartesian" "coord"
$ facet      :List of 1
  ..$ shrink: logi TRUE
  ..- attr(*, "class")= chr [1:2] "null" "facet"
$ plot_env   :<environment: R_GlobalEnv>
$ labels     :List of 2
  ..$ x: chr "u"
  ..$ y: chr "v"
 - attr(*, "class")= chr [1:2] "gg" "ggplot"
```

Now the **layers** component is nonempty, as is the **labels** component.

Obviously the rest is complicated, but at least now you have some understanding of what happens to these class objects when we do "+".

Among other things, this insight can help you in debugging, if your **ggplot2** code doesn't produce what you had expected.

C.8 For Further Information

Just plugging "ggplot2 tutorial," "ggplot2 introduction," "ggplot2 examples" and so on into your favorite search engine will give you tons of information.

Hadley's book, *ggplot2: Elegant Graphics for Data Analysis*, is of course the definitive source, but also try his pictorial reference manual, at `http://had.co.nz/ggplot2/`. Winston Chang's O'Reilly series book, the *R Graphics Cookbook*, is chock full of examples, almost all of them using **ggplot2**. Paul Murrell's book, *R Graphics*, gives a more general treatment of graphics in R.

The **ggobi** packated, whose lead author is UCD professor Duncan Temple Lang, takes an interactive approach to graphics.