2 Small Worlds and Large Worlds

When Cristoforo Colombo (Christopher Columbus) infamously sailed west in the year 1492, he believed that the Earth was spherical. In this, he was like most educated people of his day. He was unlike most people, though, in that he also believed the planet was much smaller than it actually is—only 30,000 km around its middle instead of the actual 40,000 km (Figure 2.1).[37] This was one of the most consequential mistakes in European history. If Colombo had believed instead that the Earth was 40,000 km around, he would have correctly reasoned that his fleet could not carry enough food and potable water to complete a journey all the way westward to Asia. But at 30,000 km around, Asia would lie a bit west of the coast of California. It was possible to carry enough supplies to make it that far. Emboldened in part by his unconventional estimate, Colombo set sail, eventually landing in the Bahamas.

Colombo made a prediction based upon his view that the world was small. But since he lived in a large world, aspects of the prediction were wrong. In his case, the error was lucky. His small world model was wrong in an unanticipated way: There was a lot of land in the way. If he had been wrong in the expected way, with nothing but ocean between Europe and Asia, he and his entire expedition would have run out of supplies long before reaching the East Indies.

Colombo's small and large worlds provide a contrast between model and reality. All statistical modeling has these two frames: the *small world* of the model itself and the *large world* we hope to deploy the model in.[38] Navigating between these two worlds remains a central challenge of statistical modeling. The challenge is greater when we forget the distinction.

The **small world** is the self-contained logical world of the model. Within the small world, all possibilities are nominated. There are no pure surprises, like the existence of a huge continent between Europe and Asia. Within the small world of the model, it is important to be able to verify the model's logic, making sure that it performs as expected under favorable assumptions. Bayesian models have some advantages in this regard, as they have reasonable claims to optimality: No alternative model could make better use of the information in the data and support better decisions, assuming the small world is an accurate description of the real world.[39]

The **large world** is the broader context in which one deploys a model. In the large world, there may be events that were not imagined in the small world. Moreover, the model is always an incomplete representation of the large world, and so will make mistakes, even if all kinds of events have been properly nominated. The logical consistency of a model in the small world is no guarantee that it will be optimal in the large world. But it is certainly a warm comfort.

FIGURE 2.1. Illustration of Martin Behaim's 1492 globe, showing the small world that Colombo anticipated. Europe lies on the right-hand side. Asia lies on the left. The big island labeled "Cipangu" is Japan.

In this chapter, you will begin to build Bayesian models. The way that Bayesian models learn from evidence is arguably optimal in the small world. When their assumptions approximate reality, they also perform well in the large world. But large world performance has to be demonstrated rather than logically deduced. Passing back and forth between these two worlds allows both formal methods, like Bayesian inference, and informal methods, like peer review, to play an indispensable role.

This chapter focuses on the small world. It explains probability theory in its essential form: counting the ways things can happen. Bayesian inference arises automatically from this perspective. Then the chapter presents the stylized components of a Bayesian statistical model, a model for learning from data. Then it shows you how to animate the model, to produce estimates.

All this work provides a foundation for the next chapter, in which you'll learn to summarize Bayesian estimates, as well as begin to consider large world obligations.

Rethinking: Fast and frugal in the large world. The natural world is complex, as trying to do science serves to remind us. Yet everything from the humble tick to the industrious squirrel to the idle sloth manages to frequently make adaptive decisions. But it's a good bet that most animals are not Bayesian, if only because being Bayesian is expensive and depends upon having a good model. Instead, animals use various heuristics that are fit to their environments, past or present. These heuristics take adaptive shortcuts and so may outperform a rigorous Bayesian analysis, once costs of information gathering and processing (and overfitting, Chapter 7) are taken into account.[40] Once you already know which information to ignore or attend to, being fully Bayesian is a waste. It's neither necessary nor sufficient for making good decisions, as real animals demonstrate. But for human animals, Bayesian analysis provides a general way to discover relevant information and process it logically. Just don't think that it is the only way.

2.1. The garden of forking data

Our goal in this section will be to build Bayesian inference up from humble beginnings, so there is no superstition about it. Bayesian inference is really just counting and comparing of possibilities. Consider by analogy Jorge Luis Borges' short story "The Garden of Forking Paths." The story is about a man who encounters a book filled with contradictions. In most books, characters arrive at plot points and must decide among alternative paths. A protagonist may arrive at a man's home. She might kill the man, or rather take a cup of tea. Only

one of these paths is taken—murder or tea. But the book within Borges' story explores all paths, with each decision branching outward into an expanding garden of forking paths.

This is the same device that Bayesian inference offers. In order to make good inference about what actually happened, it helps to consider everything that could have happened. A Bayesian analysis is a garden of forking data, in which alternative sequences of events are cultivated. As we learn about what did happen, some of these alternative sequences are pruned. In the end, what remains is only what is logically consistent with our knowledge.

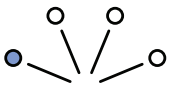
This approach provides a quantitative ranking of hypotheses, a ranking that is maximally conservative, given the assumptions and data that go into it. The approach cannot guarantee a correct answer, on large world terms. But it can guarantee the best possible answer, on small world terms, that could be derived from the information fed into it.

Consider the following toy example.

2.1.1. Counting possibilities. Suppose there's a bag, and it contains four marbles. These marbles come in two colors: blue and white. We know there are four marbles in the bag, but we don't know how many are of each color. We do know that there are five possibilities: (1) [○○○○], (2) [●○○○], (3) [●●○○], (4) [●●●○], (5) [●●●●]. These are the only possibilities consistent with what we know about the contents of the bag. Call these five possibilities the *conjectures*.

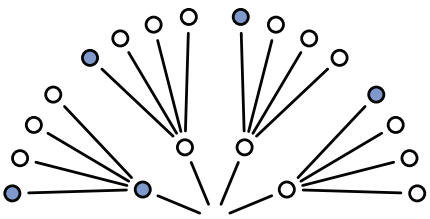
Our goal is to figure out which of these conjectures is most plausible, given some evidence about the contents of the bag. We do have some evidence: A sequence of three marbles is pulled from the bag, one at a time, replacing the marble each time and shaking the bag before drawing another marble. The sequence that emerges is: ●○●, in that order. These are the data.

So now let's plant the garden and see how to use the data to infer what's in the bag. Let's begin by considering just the single conjecture, [●○○○], that the bag contains one blue and three white marbles. On the first draw from the bag, one of four things could happen, corresponding to one of four marbles in the bag. So we can visualize the possibilities branching outward:



Notice that even though the three white marbles look the same from a data perspective—we just record the color of the marbles, after all—they are really different events. This is important, because it means that there are three more ways to see ○ than to see ●.

Now consider the garden as we get another draw from the bag. It expands the garden out one layer:



Now there are 16 possible paths through the garden, one for each pair of draws. On the second draw from the bag, each of the paths above again forks into four possible paths. Why?

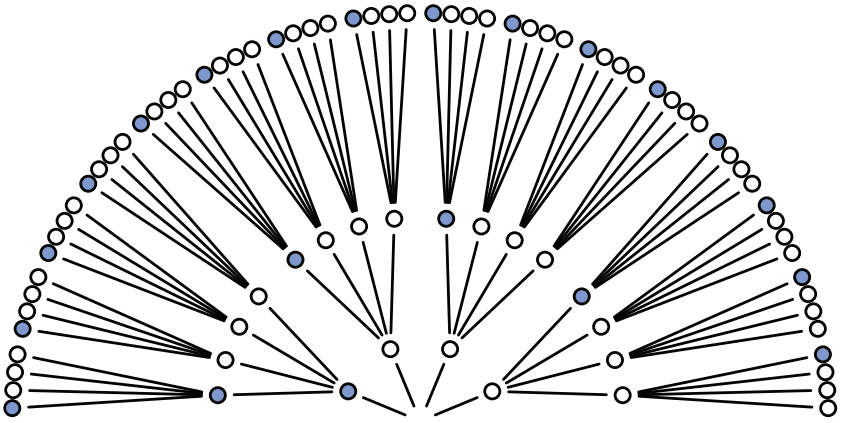


FIGURE 2.2. The 64 possible paths generated by assuming the bag contains one blue and three white marbles.

Because we believe that our shaking of the bag gives each marble a fair chance at being drawn, regardless of which marble was drawn previously. The third layer is built in the same way, and the full garden is shown in FIGURE 2.2. There are $4^3 = 64$ possible paths in total.

As we consider each draw from the bag, some of these paths are logically eliminated. The first draw tuned out to be ●, recall, so the three white paths at the bottom of the garden are eliminated right away. If you imagine the real data tracing out a path through the garden, it must have passed through the one blue path near the origin. The second draw from the bag produces ○, so three of the paths forking out of the first blue marble remain. As the data trace out a path, we know it must have passed through one of those three white paths (after the first blue path), but we don't know which one, because we recorded only the color of each marble. Finally, the third draw is ●. Each of the remaining three paths in the middle layer sustain one blue path, leaving a total of three ways for the sequence ●○● to appear, assuming the bag contains [●○○○]. FIGURE 2.3 shows the garden again, now with logically eliminated paths grayed out. We can't be sure which of those three paths the actual data took. But as long as we're considering only the possibility that the bag contains one blue and three white marbles, we can be sure that the data took one of those three paths. Those are the only paths consistent with both our knowledge of the bag's contents (four marbles, white or blue) and the data (●○●).

This demonstrates that there are three (out of 64) ways for a bag containing [●○○○] to produce the data ●○●. We have no way to decide among these three ways. The inferential power comes from comparing this count to the numbers of ways each of the other conjectures of the bag's contents could produce the same data. For example, consider the conjecture [○○○○]. There are zero ways for this conjecture to produce the observed data, because even one ● is logically incompatible with it. The conjecture [●●●●] is likewise logically incompatible with the data. So we can eliminate these two conjectures, because neither provides even a single path that is consistent with the data.

FIGURE 2.4 displays the full garden now, for the remaining three conjectures: [●○○○], [●●○○], and [●●●○]. The upper-left wedge displays the same garden as FIGURE 2.3. The upper-right shows the analogous garden for the conjecture that the bag contains three blue marbles and one white marble. And the bottom wedge shows the garden for two blue

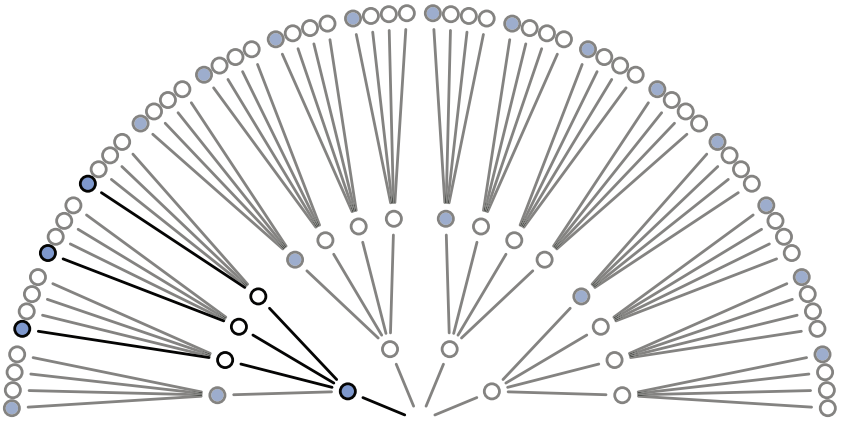


FIGURE 2.3. After eliminating paths inconsistent with the observed sequence, only 3 of the 64 paths remain.

and two white marbles. Now we count up all of the ways each conjecture could produce the observed data. For one blue and three white, there are three ways, as we counted already. For two blue and two white, there are eight paths forking through the garden that are logically consistent with the observed sequence. For three blue and one white, there are nine paths that survive.

To summarize, we've considered five different conjectures about the contents of the bag, ranging from zero blue marbles to four blue marbles. For each of these conjectures, we've counted up how many sequences, paths through the garden of forking data, could potentially produce the observed data, ●○●:

Conjecture	Ways to produce ●○●
[○○○○]	$0 \times 4 \times 0 = 0$
[●○○○]	$1 \times 3 \times 1 = 3$
[●●○○]	$2 \times 2 \times 2 = 8$
[●●●○]	$3 \times 1 \times 3 = 9$
[●●●●]	$4 \times 0 \times 4 = 0$

Notice that the number of ways to produce the data, for each conjecture, can be computed by first counting the number of paths in each "ring" of the garden and then by multiplying these counts together. This is just a computational device. It tells us the same thing as FIGURE 2.4, but without having to draw the garden. The fact that numbers are multiplied during calculation doesn't change the fact that this is still just counting of logically possible paths. This point will come up again, when you meet a formal representation of Bayesian inference.

So what good are these counts? By comparing these counts, we have part of a solution for a way to rate the relative plausibility of each conjectured bag composition. But it's only a part of a solution, because in order to compare these counts we first have to decide how many ways each conjecture could itself be realized. We might argue that when we have no reason to assume otherwise, we can just consider each conjecture equally plausible and compare the counts directly. But often we do have reason to assume otherwise.

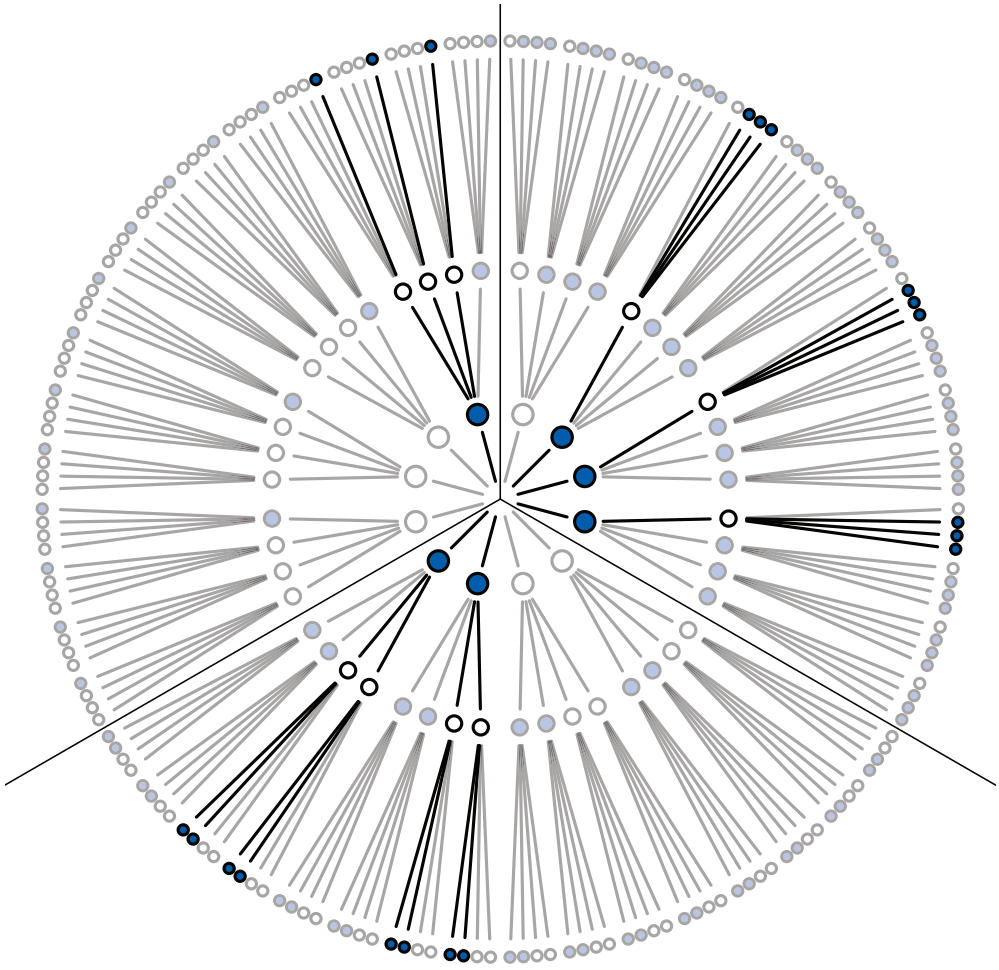


FIGURE 2.4. The garden of forking data, showing for each possible composition of the bag the forking paths that are logically compatible with the data.

Rethinking: Justification. My justification for using paths through the garden as measures of relative plausibility is humble: If we wish to reason about plausibility and remain consistent with ordinary logic—statements about *true* and *false*—then we should obey this procedure.[41] There are other justifications that lead to the same mathematical procedure. Regardless of how you choose to philosophically justify it, notice that it actually works. Justifications and philosophy motivate procedures, but it is the results that matter. The many successful real world applications of Bayesian inference may be all the justification you need. Twentieth century opponents of Bayesian data analysis argued that Bayesian inference was easy to justify, but hard to apply.[42] That is luckily no longer true. Indeed, the opposite is often true—scientists are switching to Bayesian approaches because it lets them use the models they want. Just be careful not to assume that because Bayesian inference is justified that no other approach can also be justified. Golems come in many types, and some of all types are useful.

2.1.2. Combining other information. We may have additional information about the relative plausibility of each conjecture. This information could arise from knowledge of how the contents of the bag were generated. It could also arise from previous data. Whatever the source, it would help to have a way to combine different sources of information to update the plausibilities. Luckily there is a natural solution: Just multiply the counts.

To grasp this solution, suppose we're willing to say each conjecture is equally plausible at the start. So we just compare the counts of ways in which each conjecture is compatible with the observed data. This comparison suggests that [●●●○] is slightly more plausible than [●●○○], and both are about three times more plausible than [●○○○]. Since these are our initial counts, and we are going to update them next, let's label them *prior*.

Now suppose we draw another marble from the bag to get another observation: ●. Now you have two choices. You could start all over again, making a garden with four layers to trace out the paths compatible with the data sequence ●○●●. Or you could take the previous counts—the prior counts—over conjectures (0, 3, 8, 9, 0) and just update them in light of the new observation. It turns out that these two methods are mathematically identical, as long as the new observation is logically independent of the previous observations.

Here's how to do it. First we count the numbers of ways each conjecture could produce the new observation, ●. Then we multiply each of these new counts by the prior numbers of ways for each conjecture. In table form:

Conjecture	Ways to produce ●	Prior counts	New count
[○○○○]	0	0	$0 \times 0 = 0$
[●○○○]	1	3	$3 \times 1 = 3$
[●●○○]	2	8	$8 \times 2 = 16$
[●●●○]	3	9	$9 \times 3 = 27$
[●●●●]	4	0	$0 \times 4 = 0$

The new counts in the right-hand column above summarize all the evidence for each conjecture. As new data arrive, and provided those data are independent of previous observations, then the number of logically possible ways for a conjecture to produce all the data up to that point can be computed just by multiplying the new count by the old count.

This updating approach amounts to nothing more than asserting that (1) when we have previous information suggesting there are W_{prior} ways for a conjecture to produce a previous observation D_{prior} and (2) we acquire new observations D_{new} that the same conjecture can produce in W_{new} ways, then (3) the number of ways the conjecture can account for both D_{prior} as well as D_{new} is just the product $W_{\text{prior}} \times W_{\text{new}}$. For example, in the table above the conjecture [●●○○] has $W_{\text{prior}} = 8$ ways to produce $D_{\text{prior}} =$ ●○●. It also has $W_{\text{new}} = 2$ ways to produce the new observation $D_{\text{new}} =$ ●. So there are $8 \times 2 = 16$ ways for the conjecture to produce both D_{prior} and D_{new}. Why multiply? Multiplication is just a shortcut to enumerating and counting up all of the paths through the garden that could produce all the observations.

In this example, the prior data and new data are of the same type: marbles drawn from the bag. But in general, the prior data and new data can be of different types. Suppose for example that someone from the marble factory tells you that blue marbles are rare. So for every bag containing [●●●○], they made two bags containing [●●○○] and three bags containing [●○○○]. They also ensured that every bag contained at least one blue and one white marble. We can update our counts again:

Conjecture	Prior count	Factory count	New count
[○○○○]	0	0	$0 \times 0 = 0$
[●○○○]	3	3	$3 \times 3 = 9$
[●●○○]	16	2	$16 \times 2 = 32$
[●●●○]	27	1	$27 \times 1 = 27$
[●●●●]	0	0	$0 \times 0 = 0$

Now the conjecture [●●○○] is most plausible, but barely better than [●●●○]. Is there a threshold difference in these counts at which we can safely decide that one of the conjectures is the correct one? You'll spend the next chapter exploring that question.

Rethinking: Original ignorance. Which assumption should we use, when there is no previous information about the conjectures? The most common solution is to assign an equal number of ways that each conjecture could be correct, before seeing any data. This is sometimes known as the **PRINCIPLE OF INDIFFERENCE**: When there is no reason to say that one conjecture is more plausible than another, weigh all of the conjectures equally. This book does not use nor endorse "ignorance" priors. As we'll see in later chapters, the structure of the model and the scientific context always provide information that allows us to do better than ignorance.

For the sort of problems we examine in this book, the principle of indifference results in inferences very comparable to mainstream non-Bayesian approaches, most of which contain implicit equal weighting of possibilities. For example a typical non-Bayesian confidence interval weighs equally all of the possible values a parameter could take, regardless of how implausible some of them are. In addition, many non-Bayesian procedures have moved away from equal weighting, through the use of penalized likelihood and other methods. We'll discuss this in Chapter 7.

2.1.3. From counts to probability. It is helpful to think of this strategy as adhering to a principle of honest ignorance: *When we don't know what caused the data, potential causes that may produce the data in more ways are more plausible.* This leads us to count paths through the garden of forking data. We're counting the implications of assumptions.

It's hard to use these counts though, so we almost always standardize them in a way that transforms them into probabilities. Why is it hard to work with the counts? First, since relative value is all that matters, the size of the counts 3, 8, and 9 contain no information of value. They could just as easily be 30, 80, and 90. The meaning would be the same. It's just the relative values that matter. Second, as the amount of data grows, the counts will very quickly grow very large and become difficult to manipulate. By the time we have 10 data points, there are already more than one million possible sequences. We'll want to analyze data sets with thousands of observations, so explicitly counting these things isn't practical.

Luckily, there's a mathematical way to compress all of this. Specifically, we define the updated plausibility of each possible composition of the bag, after seeing the data, as:

$$\begin{gathered}\text{plausibility of } [●○○○] \text{ after seeing } ●○● \\ \propto \\ \text{ways } [●○○○] \text{ can produce } ●○● \\ \times \\ \text{prior plausibility } [●○○○]\end{gathered}$$

That little $\propto$ means *proportional to*. We want to compare the plausibility of each possible bag composition. So it'll be helpful to define p as the proportion of marbles that are blue. For

[●○○○], $p = 1/4 = 0.25$. Also let D_{new} = ●○●. And now we can write:

$$\text{plausibility of } p \text{ after } D_{new} \propto \text{ways } p \text{ can produce } D_{new} \times \text{prior plausibility of } p$$

The above just means that for any value p can take, we judge the plausibility of that value p as proportional to the number of ways it can get through the garden of forking data. This expression just summarizes the calculations you did in the tables of the previous section.

Finally, we construct probabilities by standardizing the plausibility so that the sum of the plausibilities for all possible conjectures will be one. All you need to do in order to standardize is to add up all of the products, one for each value p can take, and then divide each product by the sum of products:

$$\text{plausibility of } p \text{ after } D_{new} = \frac{\text{ways } p \text{ can produce } D_{new} \times \text{prior plausibility } p}{\text{sum of products}}$$

A worked example is needed for this to really make sense. So consider again the table from before, now updated using our definitions of p and "plausibility":

Possible composition	p	Ways to produce data	Plausibility
[○○○○]	0	0	0
[●○○○]	0.25	3	0.15
[●●○○]	0.5	8	0.40
[●●●○]	0.75	9	0.45
[●●●●]	1	0	0

You can quickly compute these plausibilities in R:

R code 2.1

```
ways <- c( 0 , 3 , 8 , 9 , 0 )
ways/sum(ways)
```

```
[1] 0.00 0.15 0.40 0.45 0.00
```

The values in `ways` are the products mentioned before. And `sum(ways)` is the denominator "sum of products" in the expression near the top of the page.

These plausibilities are also *probabilities*—they are non-negative (zero or positive) real numbers that sum to one. And all of the mathematical things you can do with probabilities you can also do with these values. Specifically, each piece of the calculation has a direct partner in applied probability theory. These partners have stereotyped names, so it's worth learning them, as you'll see them again and again.

- A conjectured proportion of blue marbles, p, is usually called a **PARAMETER** value. It's just a way of indexing possible explanations of the data.
- The relative number of ways that a value p can produce the data is usually called a **LIKELIHOOD**. It is derived by enumerating all the possible data sequences that could have happened and then eliminating those sequences inconsistent with the data.
- The prior plausibility of any specific p is usually called the **PRIOR PROBABILITY**.
- The new, updated plausibility of any specific p is usually called the **POSTERIOR PROBABILITY**.

In the next major section, you'll meet the more formal notation for these objects and see how they compose a simple statistical model.

Rethinking: Randomization. When you shuffle a deck of cards or assign subjects to treatments by flipping a coin, it is common to say that the resulting deck and treatment assignments are *randomized*. What does it mean to randomize something? It just means that we have processed the thing so that we know almost nothing about its arrangement. Shuffling a deck of cards changes our state of knowledge, so that we no longer have any specific information about the ordering of cards. However, the bonus that arises from this is that, if we really have shuffled enough to erase any prior knowledge of the ordering, then the order the cards end up in is very likely to be one of the many orderings with high INFORMATION ENTROPY. The concept of information entropy will be increasingly important as we progress, and will be unpacked in Chapters 7 and 10.

2.2. Building a model

By working with probabilities instead of raw counts, Bayesian inference is made much easier, but it looks much harder. So in this section, we follow up on the garden of forking data by presenting the conventional form of a Bayesian statistical model. The toy example we'll use here has the anatomy of a typical statistical analysis, so it's the style that you'll grow accustomed to. But every piece of it can be mapped onto the garden of forking data. The logic is the same.

Suppose you have a globe representing our planet, the Earth. This version of the world is small enough to hold in your hands. You are curious how much of the surface is covered in water. You adopt the following strategy: You will toss the globe up in the air. When you catch it, you will record whether or not the surface under your right index finger is water or land. Then you toss the globe up in the air again and repeat the procedure.[43] This strategy generates a sequence of samples from the globe. The first nine samples might look like:

W L W W W L W L W

where W indicates water and L indicates land. So in this example you observe six W (water) observations and three L (land) observations. Call this sequence of observations the *data*.

To get the logic moving, we need to make assumptions, and these assumptions constitute the model. Designing a simple Bayesian model benefits from a design loop with three steps.

(1) Data story: Motivate the model by narrating how the data might arise.
(2) Update: Educate your model by feeding it the data.
(3) Evaluate: All statistical models require supervision, leading to model revision.

The next sections walk through these steps, in the context of the globe tossing evidence.

2.2.1. A data story. Bayesian data analysis usually means producing a story for how the data came to be. This story may be *descriptive*, specifying associations that can be used to predict outcomes, given observations. Or it may be *causal*, a theory of how some events produce other events. Typically, any story you intend to be causal may also be descriptive. But many descriptive stories are hard to interpret causally. But all data stories are complete, in the sense that they are sufficient for specifying an algorithm for simulating new data. In the next chapter, you'll see examples of doing just that, as simulating new data is useful not only for model criticism but also for model construction.

You can motivate your data story by trying to explain how each piece of data is born. This usually means describing aspects of the underlying reality as well as the sampling process. The data story in this case is simply a restatement of the sampling process:

(1) The true proportion of water covering the globe is p.

(2) A single toss of the globe has a probability p of producing a water (W) observation. It has a probability $1 - p$ of producing a land (L) observation.
(3) Each toss of the globe is independent of the others.

The data story is then translated into a formal probability model. This probability model is easy to build, because the construction process can be usefully broken down into a series of component decisions. Before meeting these components, however, it'll be useful to visualize how a Bayesian model behaves. After you've become acquainted with how such a model learns from data, we'll pop the machine open and investigate its engineering.

Rethinking: The value of storytelling. The data story has value, even if you quickly abandon it and never use it to build a model or simulate new observations. Indeed, it is important to eventually discard the story, because many different stories correspond to the same model. As a result, showing that a model does a good job does not in turn uniquely support our data story. Still, the story has value because in trying to outline the story, often one realizes that additional questions must be answered. Most data stories are much more specific than are the verbal hypotheses that inspire data collection. Hypotheses can be vague, such as "it's more likely to rain on warm days." When you are forced to consider sampling and measurement and make a precise statement of how temperature predicts rain, many stories and resulting models will be consistent with the same vague hypothesis. Resolving that ambiguity often leads to important realizations and model revisions, before any model is fit to data.

2.2.2. Bayesian updating. Our problem is one of using the evidence—the sequence of globe tosses—to decide among different possible proportions of water on the globe. These proportions are like the conjectured marbles inside the bag, from earlier in the chapter. Each possible proportion may be more or less plausible, given the evidence. A Bayesian model begins with one set of plausibilities assigned to each of these possibilities. These are the prior plausibilities. Then it updates them in light of the data, to produce the posterior plausibilities. This updating process is a kind of learning, called **Bayesian updating**. The details of this updating—how it is mechanically achieved—can wait until later in the chapter. For now, let's look only at how such a machine behaves.

For the sake of the example only, let's program our Bayesian machine to initially assign the same plausibility to every proportion of water, every value of p. We'll do better than this later. Now look at the top-left plot in Figure 2.5. The dashed horizontal line represents this initial plausibility of each possible value of p. After seeing the first toss, which is a "W," the model updates the plausibilities to the solid line. The plausibility of $p = 0$ has now fallen to exactly zero—the equivalent of "impossible." Why? Because we observed at least one speck of water on the globe, so now we know there is *some* water. The model executes this logic automatically. You don't have it instruct it to account for this consequence. Probability theory takes care of it for you, because it is essentially counting paths through the garden of forking data, as in the previous section.

Likewise, the plausibility of $p > 0.5$ has increased. This is because there is not yet any evidence that there is land on the globe, so the initial plausibilities are modified to be consistent with this. Note however that the relative plausibilities are what matter, and there isn't yet much evidence. So the differences in plausibility are not yet very large. In this way, the amount of evidence seen so far is embodied in the plausibilities of each value of p.

In the remaining plots in Figure 2.5, the additional samples from the globe are introduced to the model, one at a time. Each dashed curve is just the solid curve from the previous

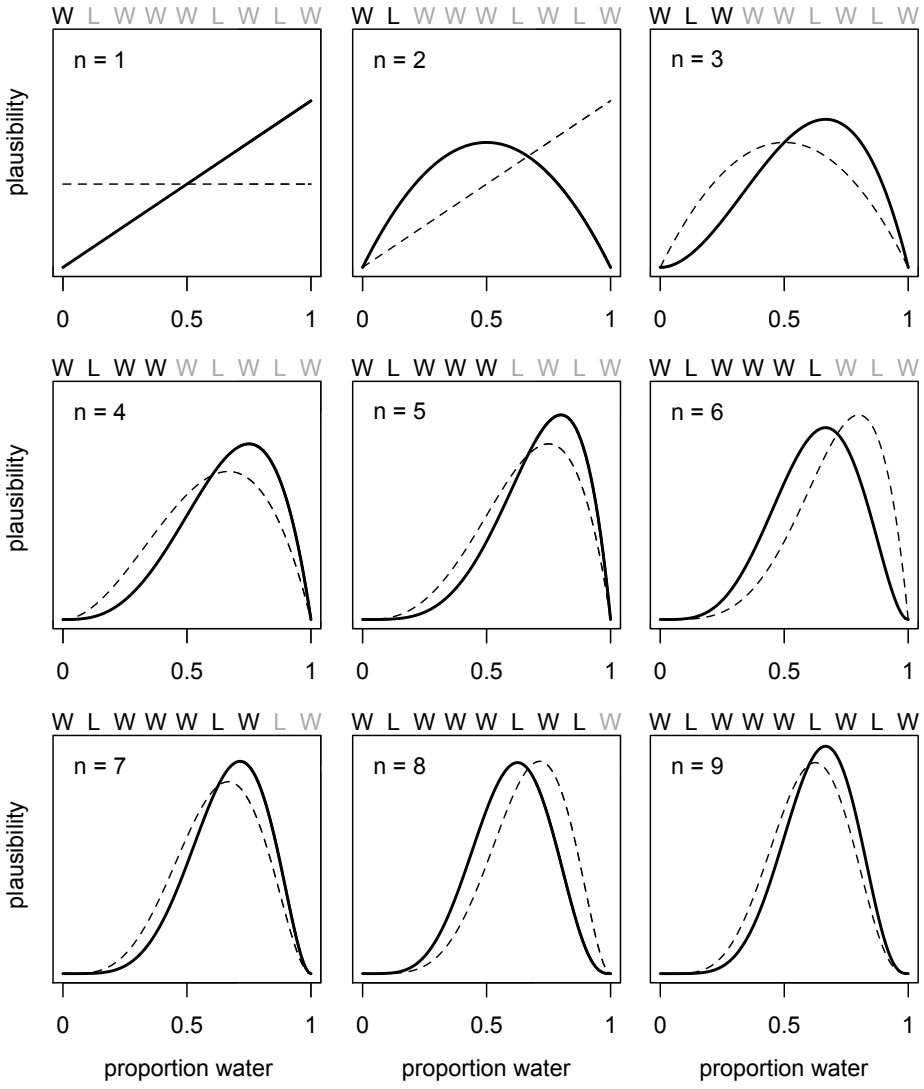


FIGURE 2.5. How a Bayesian model learns. Each toss of the globe produces an observation of water (W) or land (L). The model's estimate of the proportion of water on the globe is a plausibility for every possible value. The lines and curves in this figure are these collections of plausibilities. In each plot, previous plausibilities (dashed curve) are updated in light of the latest observation to produce a new set of plausibilities (solid curve).

plot, moving left to right and top to bottom. Every time a "W" is seen, the peak of the plausibility curve moves to the right, towards larger values of p. Every time an "L" is seen, it moves the other direction. The maximum height of the curve increases with each sample, meaning that fewer values of p amass more plausibility as the amount of evidence increases. As each new observation is added, the curve is updated consistent with all previous observations.

Notice that every updated set of plausibilities becomes the initial plausibilities for the next observation. Every conclusion is the starting point for future inference. However, this updating process works backwards, as well as forwards. Given the final set of plausibilities in the bottom-right plot of Figure 2.5, and knowing the final observation (W), it is possible to mathematically divide out the observation, to infer the previous plausibility curve. So the data could be presented to your model in any order, or all at once even. In most cases, you will present the data all at once, for the sake of convenience. But it's important to realize that this merely represents abbreviation of an iterated learning process.

Rethinking: Sample size and reliable inference. It is common to hear that there is a minimum number of observations for a useful statistical estimate. For example, there is a widespread superstition that 30 observations are needed before one can use a Gaussian distribution. Why? In non-Bayesian statistical inference, procedures are often justified by the method's behavior at very large sample sizes, so-called *asymptotic* behavior. As a result, performance at small samples sizes is questionable.

In contrast, Bayesian estimates are valid for any sample size. This does not mean that more data isn't helpful—it certainly is. Rather, the estimates have a clear and valid interpretation, no matter the sample size. But the price for this power is dependency upon the initial plausibilities, the prior. If the prior is a bad one, then the resulting inference will be misleading. There's no free lunch,[44] when it comes to learning about the world. A Bayesian golem must choose an initial plausibility, and a non-Bayesian golem must choose an estimator. Both golems pay for lunch with their assumptions.

2.2.3. Evaluate. The Bayesian model learns in a way that is demonstrably optimal, provided that it accurately describes the real, large world. This is to say that your Bayesian machine guarantees perfect inference within the small world. No other way of using the available information, beginning with the same state of information, could do better.

Don't get too excited about this logical virtue, however. The calculations may malfunction, so results always have to be checked. And if there are important differences between the model and reality, then there is no logical guarantee of large world performance. And even if the two worlds did match, any particular sample of data could still be misleading. So it's worth keeping in mind at least two cautious principles.

First, the model's certainty is no guarantee that the model is a good one. As the amount of data increases, the globe tossing model will grow increasingly sure of the proportion of water. This means that the curves in Figure 2.5 will become increasingly narrow and tall, restricting plausible values within a very narrow range. But models of all sorts—Bayesian or not—can be very confident about an inference, even when the model is seriously misleading. This is because the inferences are conditional on the model. What your model is telling you is that, given a commitment to this particular model, it can be very sure that the plausible values are in a narrow range. Under a different model, things might look differently. There will be examples in later chapters.

Second, it is important to supervise and critique your model's work. Consider again the fact that the updating in the previous section works in any order of data arrival. We could shuffle the order of the observations, as long as six W's and three L's remain, and still end up with the same final plausibility curve. That is only true, however, because the model assumes that order is irrelevant to inference. When something is irrelevant to the machine, it won't affect the inference directly. But it may affect it indirectly, because the data will depend upon order. So it is important to check the model's inferences in light of aspects of the data it does

not know about. Such checks are an inherently creative enterprise, left to the analyst and the scientific community. Golems are very bad at it.

In Chapter 3, you'll see some examples of such checks. For now, note that the goal is not to test the truth value of the model's assumptions. We know the model's assumptions are never exactly right, in the sense of matching the true data generating process. Therefore there's no point in checking if the model is true. Failure to conclude that a model is false must be a failure of our imagination, not a success of the model. Moreover, models do not need to be exactly true in order to produce highly precise and useful inferences. All manner of small world assumptions about error distributions and the like can be violated in the large world, but a model may still produce a perfectly useful estimate. This is because models are essentially information processing machines, and there are some surprising aspects of information that cannot be easily captured by framing the problem in terms of the truth of assumptions.[45]

Instead, the objective is to check the model's adequacy for some purpose. This usually means asking and answering additional questions, beyond those that originally constructed the model. Both the questions and answers will depend upon the scientific context. So it's hard to provide general advice. There will be many examples, throughout the book, and of course the scientific literature is replete with evaluations of the suitability of models for different jobs—prediction, comprehension, measurement, and persuasion.

Rethinking: Deflationary statistics. It may be that Bayesian inference is the best general purpose method of inference known. However, Bayesian inference is much less powerful than we'd like it to be. There is no approach to inference that provides universal guarantees. No branch of applied mathematics has unfettered access to reality, because math is not discovered, like the proton. Instead it is invented, like the shovel.[46]

2.3. Components of the model

Now that you've seen how the Bayesian model behaves, it's time to open up the machine and learn how it works. Consider three different things that we counted in the previous sections.

(1) The number of ways each conjecture could produce an observation
(2) The accumulated number of ways each conjecture could produce the entire data
(3) The initial plausibility of each conjectured cause of the data

Each of these things has a direct analog in conventional probability theory. And so the usual way we build a statistical model involves choosing distributions and devices for each that represent the relative numbers of ways things can happen.

In this section, you'll meet these components in some detail and see how each relates to the counting you did earlier in the chapter. The job in front of us is really nothing more than naming all of the variables and defining each. We'll take these tasks in turn.

2.3.1. Variables. Variables are just symbols that can take on different values. In a scientific context, variables include things we wish to infer, such as proportions and rates, as well as things we might observe, the data. In the globe tossing model, there are three variables.

The first variable is our target of inference, p, the proportion of water on the globe. This variable cannot be observed. Unobserved variables are usually called **PARAMETERS**. But while p itself is unobserved, we can infer it from the other variables.

The other variables are the observed variables, the counts of water and land. Call the count of water W and the count of land L. The sum of these two variables is the number of globe tosses: $N = W + L$.

2.3.2. Definitions. Once we have the variables listed, we then have to define each of them. In defining each, we build a model that relates the variables to one another. Remember, the goal is to count all the ways the data could arise, given the assumptions. This means, as in the globe tossing model, that for each possible value of the unobserved variables, such as p, we need to define the relative number of ways—the probability—that the values of each observed variable could arise. And then for each unobserved variable, we need to define the prior plausibility of each value it could take. I appreciate that this is all a bit abstract. So here are the specifics, for the globe.

2.3.2.1. *Observed variables.* For the count of water W and land L, we define how plausible any combination of W and L would be, for a specific value of p. This is very much like the marble counting we did earlier in the chapter. Each specific value of p corresponds to a specific plausibility of the data, as in Figure 2.5.

So that we don't have to literally count, we can use a mathematical function that tells us the right plausibility. In conventional statistics, a distribution function assigned to an observed variable is usually called a **likelihood**. That term has special meaning in non-Bayesian statistics, however.[47] We will be able to do things with our distributions that non-Bayesian models forbid. So I will sometimes avoid the term *likelihood* and just talk about distributions of variables. But when someone says, "likelihood," they will usually mean a distribution function assigned to an observed variable.

In the case of the globe tossing model, the function we need can be derived directly from the data story. Begin by nominating all of the possible events. There are two: *water* (W) and *land* (L). There are no other events. The globe never gets stuck to the ceiling, for example. When we observe a sample of W's and L's of length N (nine in the actual sample), we need to say how likely that exact sample is, out of the universe of potential samples of the same length. That might sound challenging, but it's the kind of thing you get good at very quickly, once you start practicing.

In this case, once we add our assumptions that (1) every toss is independent of the other tosses and (2) the probability of W is the same on every toss, probability theory provides a unique answer, known as the *binomial distribution*. This is the common "coin tossing" distribution. And so the probability of observing W waters and L lands, with a probability p of water on each toss, is:

$$\Pr(W, L|p) = \frac{(W+L)!}{W!L!} p^W (1-p)^L$$

Read the above as:

> *The counts of "water" W and "land' L are distributed binomially, with probability p of "water" on each toss.*

And the binomial distribution formula is built into R, so you can easily compute the likelihood of the data—six W's in nine tosses—under any value of p with:

R code 2.2

```
dbinom( 6 , size=9 , prob=0.5 )
```

```
[1] 0.1640625
```

That number is the relative number of ways to get six water, holding p at 0.5 and $N = W + L$ at nine. So it does the job of counting relative number of paths through the garden. Change the `0.5` to any other value, to see how the value changes.

Much later in the book, in Chapter 10, we'll see that the binomial distribution is rather special, because it represents the **MAXIMUM ENTROPY** way to count binary events. "Maximum entropy" might sound like a bad thing. Isn't entropy disorder? Doesn't "maximum entropy" mean the death of the universe? Actually it means that the distribution contains no additional information other than: There are two events, and the probabilities of each in each trial are p and $1 - p$. Chapter 10 explains this in more detail, and the details can certainly wait.

Overthinking: Names and probability distributions. The "`d`" in `dbinom` stands for *density*. Functions named in this way almost always have corresponding partners that begin with "`r`" for random samples and that begin with "`p`" for cumulative probabilities. See for example the help `?dbinom`.

Rethinking: A central role for likelihood. A great deal of ink has been spilled focusing on how Bayesian and non-Bayesian data analyses differ. Focusing on differences is useful, but sometimes it distracts us from fundamental similarities. Notably, the most influential assumptions in both Bayesian and many non-Bayesian models are the distributions assigned to data, the likelihood functions. The likelihoods influence inference for every piece of data, and as sample size increases, the likelihood matters more and more. This helps to explain why Bayesian and non-Bayesian inferences are often so similar. If we had to explain Bayesian inference using only one aspect of it, we should describe likelihood, not priors.

2.3.2.2. *Unobserved variables.* The distributions we assign to the observed variables typically have their own variables. In the binomial above, there is p, the probability of sampling water. Since p is not observed, we usually call it a **PARAMETER**. Even though we cannot observe p, we still have to define it.

In future chapters, there will be more parameters in your models. In statistical modeling, many of the most common questions we ask about data are answered directly by parameters:

- What is the average difference between treatment groups?
- How strong is the association between a treatment and an outcome?
- Does the effect of the treatment depend upon a covariate?
- How much variation is there among groups?

You'll see how these questions become extra parameters inside the distribution function we assign to the data.

For every parameter you intend your Bayesian machine to consider, you must provide a distribution of prior plausibility, its **PRIOR**. A Bayesian machine must have an initial plausibility assignment for each possible value of the parameter, and these initial assignments do useful work. When you have a previous estimate to provide to the machine, that can become the prior, as in the steps in FIGURE 2.5. Back in FIGURE 2.5, the machine did its learning one piece of data at a time. As a result, each estimate becomes the prior for the next step. But this doesn't resolve the problem of providing a prior, because at the dawn of time, when $N = 0$, the machine still had an initial state of information for the parameter p: a flat line specifying equal plausibility for every possible value.

So where do priors come from? They are both engineering assumptions, chosen to help the machine learn, and scientific assumptions, chosen to reflect what we know about a phenomenon. The flat prior in FIGURE 2.5 is very common, but it is hardly ever the best prior. Later chapters will focus on prior choice a lot more.

There is a school of Bayesian inference that emphasizes choosing priors based upon the personal beliefs of the analyst.[48] While this **SUBJECTIVE BAYESIAN** approach thrives in some statistics and philosophy and economics programs, it is rare in the sciences. Within Bayesian data analysis in the natural and social sciences, the prior is considered to be just part of the model. As such it should be chosen, evaluated, and revised just like all of the other components of the model. In practice, the subjectivist and the non-subjectivist will often analyze data in nearly the same way.

None of this should be understood to mean that any statistical analysis is not inherently subjective, because of course it is—lots of little subjective decisions are involved in all parts of science. It's just that priors and Bayesian data analysis are no more inherently subjective than are likelihoods and the repeat sampling assumptions required for significance testing.[49] Anyone who has visited a statistics help desk at a university has probably experienced this subjectivity—statisticians do not in general exactly agree on how to analyze anything but the simplest of problems. The fact that statistical inference uses mathematics does not imply that there is only one reasonable or useful way to conduct an analysis. Engineering uses math as well, but there are many ways to build a bridge.

Beyond all of the above, there's no law mandating we use only one prior. If you don't have a strong argument for any particular prior, then try different ones. Because the prior is an assumption, it should be interrogated like other assumptions: by altering it and checking how sensitive inference is to the assumption. No one is required to swear an oath to the assumptions of a model, and no set of assumptions deserves our obedience.

Overthinking: Prior as probability distribution. You could write the prior in the example here as:

$$\Pr(p) = \frac{1}{1-0} = 1.$$

The prior is a probability distribution for the parameter. In general, for a uniform prior from a to b, the probability of any point in the interval is $1/(b-a)$. If you're bothered by the fact that the probability of every value of p is 1, remember that every probability distribution must sum (integrate) to 1. The expression $1/(b-a)$ ensures that the area under the flat line from a to b is equal to 1. There will be more to say about this in Chapter 4.

Rethinking: Datum or parameter? It is typical to conceive of data and parameters as completely different kinds of entities. Data are measured and known; parameters are unknown and must be estimated from data. Usefully, in the Bayesian framework the distinction between a datum and a parameter is not so fundamental. Sometimes we observe a variable, but sometimes we do not. In that case, the same distribution function applies, even though we didn't observe the variable. As a result, the same assumption can look like a "likelihood" or a "prior," depending upon context, without any change to the model. Much later in the book (Chapter 15), you'll see how to exploit this deep identity between certainty (data) and uncertainty (parameters) to incorporate measurement error and missing data into your modeling.

Rethinking: Prior, prior pants on fire. Historically, some opponents of Bayesian inference objected to the arbitrariness of priors. It's true that priors are very flexible, being able to encode many different states of information. If the prior can be anything, isn't it possible to get any answer you want? Indeed it is. Regardless, after a couple hundred years of Bayesian calculation, it hasn't turned out that people use priors to lie. If your goal is to lie with statistics, you'd be a fool to do it with priors, because such a lie would be easily uncovered. Better to use the more opaque machinery of the likelihood. Or better yet—don't actually take this advice!—massage the data, drop some "outliers," and otherwise engage in motivated data transformation.

It is true though that choice of the likelihood is much more conventionalized than choice of prior. But conventional choices are often poor ones, smuggling in influences that can be hard to discover. In this regard, both Bayesian and non-Bayesian models are equally harried, because both traditions depend heavily upon likelihood functions and conventionalized model forms. And the fact that the non-Bayesian procedure doesn't have to make an assumption about the prior is of little comfort. This is because non-Bayesian procedures need to make choices that Bayesian ones do not, such as choice of estimator or likelihood penalty. Often, such choices can be shown to be equivalent to some Bayesian choice of prior or rather choice of loss function. (You'll meet loss functions later in Chapter 3.)

2.3.3. A model is born. With all the above work, we can now summarize our model. The observed variables W and L are given relative counts through the binomial distribution. So we can write, as a shortcut:

$$W \sim \text{Binomial}(N, p)$$

where $N = W + L$. The above is just a convention for communicating the assumption that the relative counts of ways to realize W in N trials with probability p on each trial comes from the binomial distribution. And the unobserved parameter p similarly gets:

$$p \sim \text{Uniform}(0, 1)$$

This means that p has a uniform—flat—prior over its entire possible range, from zero to one. As I mentioned earlier, this is obviously not the best we could do, since we know the Earth has more water than land, even if we do not know the exact proportion yet.

Next, let's see how to use these assumptions to generate inference.

2.4. Making the model go

Once you have named all the variables and chosen definitions for each, a Bayesian model can update all of the prior distributions to their purely logical consequences: the **posterior distribution**. For every unique combination of data, likelihood, parameters, and prior, there is a unique posterior distribution. This distribution contains the relative plausibility of different parameter values, conditional on the data and model. The posterior distribution takes the form of the probability of the parameters, conditional on the data. In this case, it would be $\Pr(p|W, L)$, the probability of each possible value of p, conditional on the specific W and L that we observed.

2.4.1. Bayes' theorem. The mathematical definition of the posterior distribution arises from **Bayes' theorem**. This is the theorem that gives Bayesian data analysis its name. But the theorem itself is a trivial implication of probability theory. Here's a quick derivation of it, in the context of the globe tossing example. Really this will just be a re-expression of the garden of forking data derivation from earlier in the chapter. What makes it look different

is that it will use the rules of probability theory to coax out the updating rule. But it is still just counting.

The joint probability of the data W and L and any particular value of p is:

$$\Pr(W, L, p) = \Pr(W, L|p) \Pr(p)$$

This just says that the probability of W, L and p is the product of $\Pr(W, L|p)$ and the prior probability $\Pr(p)$. This is like saying that the probability of rain and cold on the same day is equal to the probability of rain, when it's cold, times the probability that it's cold. This much is just definition. But it's just as true that:

$$\Pr(W, L, p) = \Pr(p|W, L) \Pr(W, L)$$

All I've done is reverse which probability is conditional, on the right-hand side. It is still a true definition. It's like saying that the probability of rain and cold on the same day is equal to the probability that it's cold, when it's raining, times the probability of rain. Compare this statement to the one in the previous paragraph.

Now since both right-hand sides above are equal to the same thing, $\Pr(W, L, p)$, they are also equal to one another:

$$\Pr(W, L|p) \Pr(p) = \Pr(p|W, L) \Pr(W, L)$$

So we can now solve for the thing that we want, $\Pr(p|W, L)$:

$$\Pr(p|W, L) = \frac{\Pr(W, L|p) \Pr(p)}{\Pr(W, L)}$$

And this is Bayes' theorem. It says that the probability of any particular value of p, considering the data, is equal to the product of the relative plausibility of the data, conditional on p, and the prior plausibility of p, divided by this thing $\Pr(W, L)$, which I'll call the *average probability of the data*. In word form:

$$\text{Posterior} = \frac{\text{Probability of the data} \times \text{Prior}}{\text{Average probability of the data}}$$

The average probability of the data, $\Pr(W, L)$, can be confusing. It is commonly called the "evidence" or the "average likelihood," neither of which is a transparent name. The probability $\Pr(W, L)$ is literally the *average* probability of the data. Averaged over what? Averaged over the prior. It's job is just to standardize the posterior, to ensure it sums (integrates) to one. In mathematical form:

$$\Pr(W, L) = \mathrm{E}\big(\Pr(W, L|p)\big) = \int \Pr(W, L|p) \Pr(p) dp$$

The operator E means to take an *expectation*. Such averages are commonly called *marginals* in mathematical statistics, and so you may also see this same probability called a *marginal likelihood*. And the integral above just defines the proper way to compute the average over a continuous distribution of values, like the infinite possible values of p.

The key lesson is that the posterior is proportional to the product of the prior and the probability of the data. Why? Because for each specific value of p, the number of paths through the garden of forking data is the product of the prior number of paths and the new number of paths. Multiplication is just compressed counting. The average probability on the bottom just standardizes the counts so they sum to one. So while Bayes' theorem looks complicated, because the relationship with counting paths is obscured, it just expresses the counting that logic demands.

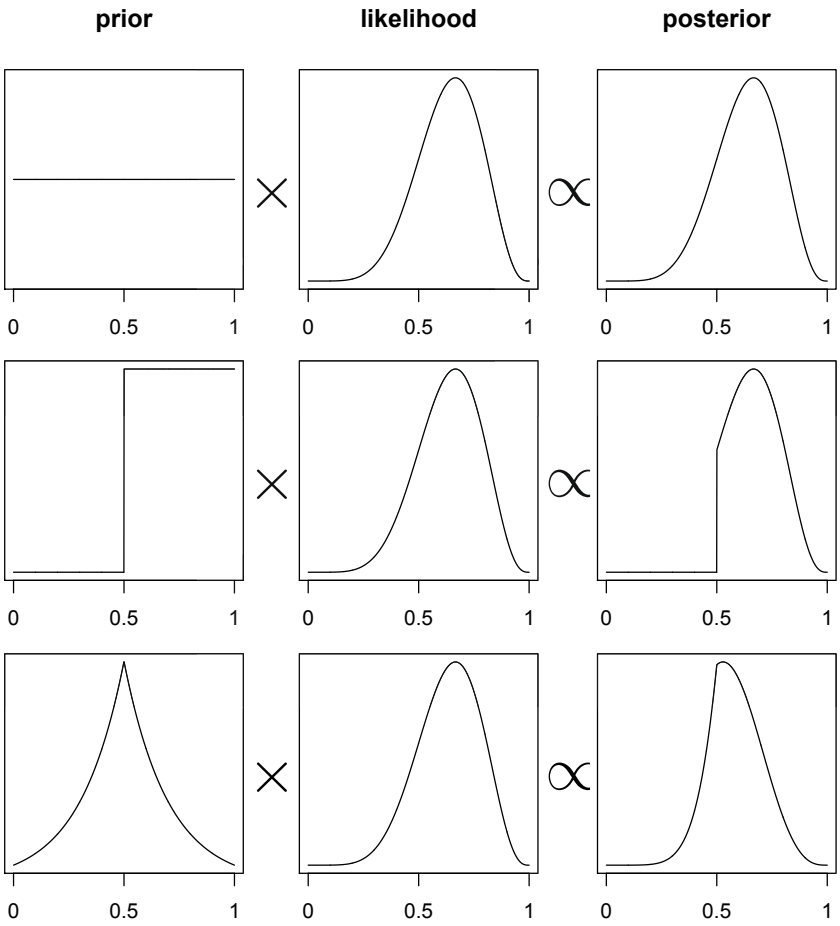


FIGURE 2.6. The posterior distribution as a product of the prior distribution and likelihood. Top: A flat prior constructs a posterior that is simply proportional to the likelihood. Middle: A step prior, assigning zero probability to all values less than 0.5, results in a truncated posterior. Bottom: A peaked prior that shifts and skews the posterior, relative to the likelihood.

FIGURE 2.6 illustrates the multiplicative interaction of a prior and a probability of data. On each row, a prior on the left is multiplied by the probability of data in the middle to produce a posterior on the right. The probability of data in each case is the same. The priors however vary. As a result, the posterior distributions vary.

Rethinking: Bayesian data analysis isn't about Bayes' theorem. A common notion about Bayesian data analysis, and Bayesian inference more generally, is that it is distinguished by the use of Bayes' theorem. This is a mistake. Inference under any probability concept will eventually make use of Bayes' theorem. Common introductory examples of "Bayesian" analysis using HIV and DNA testing are not

uniquely Bayesian. Since all of the elements of the calculation are frequencies of observations, a non-Bayesian analysis would do exactly the same thing. Instead, Bayesian approaches get to use Bayes' theorem more generally, to quantify uncertainty about theoretical entities that cannot be observed, like parameters and models. Powerful inferences can be produced under both Bayesian and non-Bayesian probability concepts, but different justifications and sacrifices are necessary.

2.4.2. Motors. Recall that your Bayesian model is a machine, a figurative golem. It has built-in definitions for the likelihood, the parameters, and the prior. And then at its heart lies a motor that processes data, producing a posterior distribution. The action of this motor can be thought of as *conditioning* the prior on the data. As explained in the previous section, this conditioning is governed by the rules of probability theory, which defines a uniquely logical posterior for set of assumptions and observations.

However, knowing the mathematical rule is often of little help, because many of the interesting models in contemporary science cannot be conditioned formally, no matter your skill in mathematics. And while some broadly useful models like linear regression can be conditioned formally, this is only possible if you constrain your choice of prior to special forms that are easy to do mathematics with. We'd like to avoid forced modeling choices of this kind, instead favoring conditioning engines that can accommodate whichever prior is most useful for inference.

What this means is that various numerical techniques are needed to approximate the mathematics that follows from the definition of Bayes' theorem. In this book, you'll meet three different conditioning engines, numerical techniques for computing posterior distributions:

(1) Grid approximation
(2) Quadratic approximation
(3) Markov chain Monte Carlo (MCMC)

There are many other engines, and new ones are being invented all the time. But the three you'll get to know here are common and widely useful. In addition, as you learn them, you'll also learn principles that will help you understand other techniques.

Rethinking: How you fit the model is part of the model. Earlier in this chapter, I implicitly defined the model as a composite of a prior and a likelihood. That definition is typical. But in practical terms, we should also consider how the model is fit to data as part of the model. In very simple problems, like the globe tossing example that consumes this chapter, calculation of the posterior density is trivial and foolproof. In even moderately complex problems, however, the details of fitting the model to data force us to recognize that our numerical technique influences our inferences. This is because different mistakes and compromises arise under different techniques. The same model fit to the same data using different techniques may produce different answers. When something goes wrong, every piece of the machine may be suspect. And so our golems carry with them their updating engines, as much slaves to their engineering as they are to the priors and likelihoods we program into them.

2.4.3. Grid approximation. One of the simplest conditioning techniques is grid approximation. While most parameters are *continuous*, capable of taking on an infinite number of values, it turns out that we can achieve an excellent approximation of the continuous posterior distribution by considering only a finite grid of parameter values. At any particular

value of a parameter, p', it's a simple matter to compute the posterior probability: just multiply the prior probability of p' by the likelihood at p'. Repeating this procedure for each value in the grid generates an approximate picture of the exact posterior distribution. This procedure is called **GRID APPROXIMATION**. In this section, you'll see how to perform a grid approximation, using simple bits of R code.

Grid approximation will mainly be useful as a pedagogical tool, as learning it forces the user to really understand the nature of Bayesian updating. But in most of your real modeling, grid approximation isn't practical. The reason is that it scales very poorly, as the number of parameters increases. So in later chapters, grid approximation will fade away, to be replaced by other, more efficient techniques. Still, the conceptual value of this exercise will carry forward, as you graduate to other techniques.

In the context of the globe tossing problem, grid approximation works extremely well. So let's build a grid approximation for the model we've constructed so far. Here is the recipe:

(1) Define the grid. This means you decide how many points to use in estimating the posterior, and then you make a list of the parameter values on the grid.
(2) Compute the value of the prior at each parameter value on the grid.
(3) Compute the likelihood at each parameter value.
(4) Compute the unstandardized posterior at each parameter value, by multiplying the prior by the likelihood.
(5) Finally, standardize the posterior, by dividing each value by the sum of all values.

In the globe tossing context, here's the code to complete all five of these steps:

R code 2.3

```
# define grid
p_grid <- seq( from=0 , to=1 , length.out=20 )

# define prior
prior <- rep( 1 , 20 )

# compute likelihood at each value in grid
likelihood <- dbinom( 6 , size=9 , prob=p_grid )

# compute product of likelihood and prior
unstd.posterior <- likelihood * prior

# standardize the posterior, so it sums to 1
posterior <- unstd.posterior / sum(unstd.posterior)
```

The above code makes a grid of only 20 points. To display the posterior distribution now:

R code 2.4

```
plot( p_grid , posterior , type="b" ,
    xlab="probability of water" , ylab="posterior probability" )
mtext( "20 points" )
```

You'll get the right-hand plot in FIGURE 2.7. Try sparser grids (5 points) and denser grids (100 or 1000 points). The correct density for your grid is determined by how accurate you want your approximation to be. More points means more precision. In this simple example, you can go crazy and use 100,000 points, but there won't be much change in inference after the first 100.

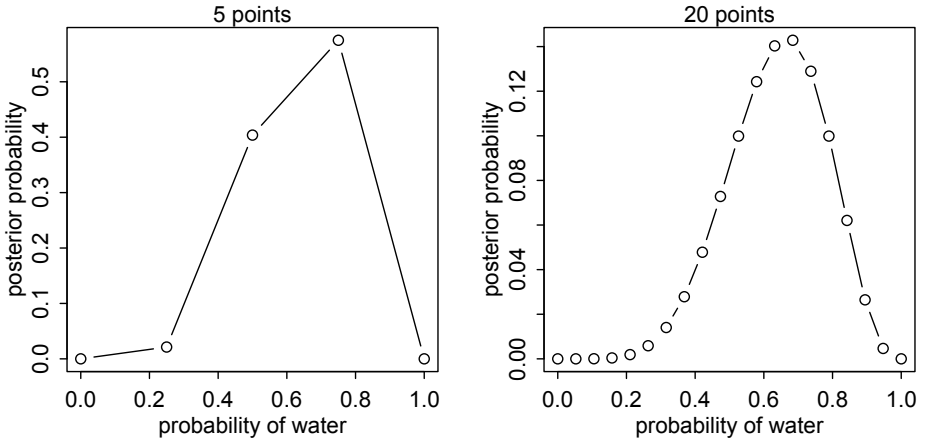


FIGURE 2.7. Computing posterior distribution by grid approximation. In each plot, the posterior distribution for the globe toss data and model is approximated with a finite number of evenly spaced points. With only 5 points (left), the approximation is terrible. But with 20 points (right), the approximation is already quite good. Compare to the analytically solved, exact posterior distribution in FIGURE 2.5 (page 30).

Now to replicate the different priors in FIGURE 2.5, try these lines of code—one at a time—for the prior grid:

R code 2.5

```
prior <- ifelse( p_grid < 0.5 , 0 , 1 )
prior <- exp( -5*abs( p_grid - 0.5 ) )
```

The rest of the code remains the same.

Overthinking: Vectorization. One of R's useful features is that it makes working with lists of numbers almost as easy as working with single values. So even though both lines of code above say nothing about how dense your grid is, whatever length you chose for the vector `p_grid` will determine the length of the vector `prior`. In R jargon, the calculations above are *vectorized*, because they work on lists of values, *vectors*. In a vectorized calculation, the calculation is performed on each element of the input vector—`p_grid` in this case—and the resulting output therefore has the same length. In other computing environments, the same calculation would require a *loop*. R can also use loops, but vectorized calculations are typically faster. They can however be much harder to read, when you are starting out with R. Be patient, and you'll soon grow accustomed to vectorized calculations.

2.4.4. Quadratic approximation. We'll stick with the grid approximation to the globe tossing posterior, for the rest of this chapter and the next. But before long you'll have to resort to another approximation, one that makes stronger assumptions. The reason is that the number of unique values to consider in the grid grows rapidly as the number of parameters in your model increases. For the single-parameter globe tossing model, it's no problem to compute a grid of 100 or 1000 values. But for two parameters approximated by 100 values each, that's already $100^2 = 10{,}000$ values to compute. For 10 parameters, the grid becomes many

billions of values. These days, it's routine to have models with hundreds or thousands of parameters. The grid approximation strategy scales very poorly with model complexity, so it won't get us very far.

A useful approach is **QUADRATIC APPROXIMATION**. Under quite general conditions, the region near the peak of the posterior distribution will be nearly Gaussian—or "normal"—in shape. This means the posterior distribution can be usefully approximated by a Gaussian distribution. A Gaussian distribution is convenient, because it can be completely described by only two numbers: the location of its center (mean) and its spread (variance).

A Gaussian approximation is called "quadratic approximation" because the logarithm of a Gaussian distribution forms a parabola. And a parabola is a quadratic function. So this approximation essentially represents any log-posterior with a parabola.

We'll use quadratic approximation for much of the first half of this book. For many of the most common procedures in applied statistics—linear regression, for example—the approximation works very well. Often, it is even exactly correct, not actually an approximation at all. Computationally, quadratic approximation is very inexpensive, at least compared to grid approximation and MCMC (discussed next). The procedure, which R will happily conduct at your command, contains two steps.

(1) Find the posterior mode. This is usually accomplished by some optimization algorithm, a procedure that virtually "climbs" the posterior distribution, as if it were a mountain. The golem doesn't know where the peak is, but it does know the slope under its feet. There are many well-developed optimization procedures, most of them more clever than simple hill climbing. But all of them try to find peaks.

(2) Once you find the peak of the posterior, you must estimate the curvature near the peak. This curvature is sufficient to compute a quadratic approximation of the entire posterior distribution. In some cases, these calculations can be done analytically, but usually your computer uses some numerical technique instead.

To compute the quadratic approximation for the globe tossing data, we'll use a tool in the `rethinking` package: `quap`. We're going to be using `quap` a lot in the first half of this book. It's a flexible model fitting tool that will allow us to specify a large number of different "regression" models. So it'll be worth trying it out right now. You'll get a more thorough understanding of it later.

To compute the quadratic approximation to the globe tossing data:

R code 2.6

```
library(rethinking)
globe.qa <- quap(
    alist(
        W ~ dbinom( W+L ,p) ,  # binomial likelihood
        p ~ dunif(0,1)     # uniform prior
    ) ,
    data=list(W=6,L=3) )

# display summary of quadratic approximation
precis( globe.qa )
```

To use `quap`, you provide a *formula*, a list of *data*. The formula defines the probability of the data and the prior. I'll say much more about these formulas in Chapter 4. Now let's see the output:

```
  Mean StdDev 5.5% 94.5%
```

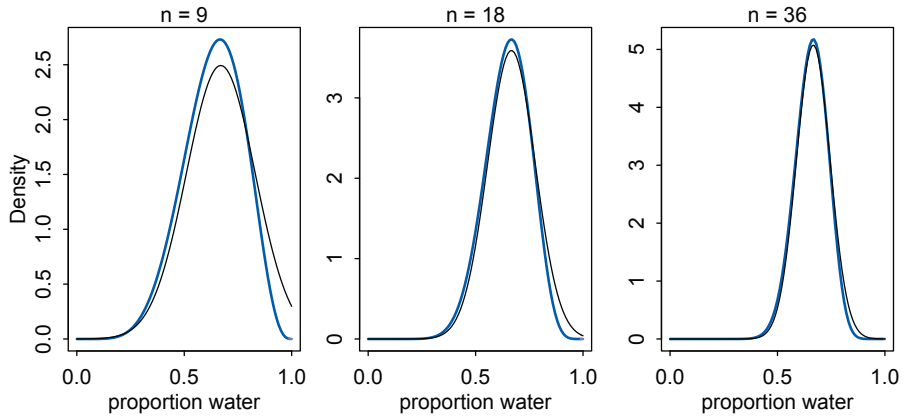


FIGURE 2.8. Accuracy of the quadratic approximation. In each plot, the exact posterior distribution is plotted in blue, and the quadratic approximation is plotted as the black curve. Left: The globe tossing data with $n = 9$ tosses and $w = 6$ waters. Middle: Double the amount of data, with the same fraction of water, $n = 18$ and $w = 12$. Right: Four times as much data, $n = 36$ and $w = 24$.

```
p 0.67   0.16 0.42  0.92
```

The function `precis` presents a brief summary of the quadratic approximation. In this case, it shows the posterior mean value of $p = 0.67$, which it calls the "Mean." The curvature is labeled "StdDev" This stands for *standard deviation*. This value is the standard deviation of the posterior distribution, while the mean value is its peak. Finally, the last two values in the `precis` output show the 89% percentile interval, which you'll learn more about in the next chapter. You can read this kind of approximation like: *Assuming the posterior is Gaussian, it is maximized at 0.67, and its standard deviation is 0.16.*

Since we already know the posterior, let's compare to see how good the approximation is. I'll use the analytical approach here, which uses `dbeta`. I won't explain this calculation, but it ensures that we have exactly the right answer. You can find an explanation and derivation of it in just about any mathematical textbook on Bayesian inference.

R code 2.7

```
# analytical calculation
W <- 6
L <- 3
curve( dbeta( x , W+1 , L+1 ) , from=0 , to=1 )
# quadratic approximation
curve( dnorm( x , 0.67 , 0.16 ) , lty=2 , add=TRUE )
```

You can see this plot (with a little extra formatting) on the left in FIGURE 2.8. The blue curve is the analytical posterior and the black curve is the quadratic approximation. The black curve does alright on its left side, but looks pretty bad on its right side. It even assigns positive probability to $p = 1$, which we know is impossible, since we saw at least one land sample.

As the amount of data increases, however, the quadratic approximation gets better. In the middle of FIGURE 2.8, the sample size is doubled to $n = 18$ tosses, but with the same fraction

of water, so that the mode of the posterior is in the same place. The quadratic approximation looks better now, although still not great. At quadruple the data, on the right side of the figure, the two curves are nearly the same now.

This phenomenon, where the quadratic approximation improves with the amount of data, is very common. It's one of the reasons that so many classical statistical procedures are nervous about small samples: Those procedures use quadratic (or other) approximations that are only known to be safe with infinite data. Often, these approximations are useful with less than infinite data, obviously. But the rate of improvement as sample size increases varies greatly depending upon the details. In some models, the quadratic approximation can remain terrible even with thousands of samples.

Using the quadratic approximation in a Bayesian context brings with it all the same concerns. But you can always lean on some algorithm other than quadratic approximation, if you have doubts. Indeed, grid approximation works very well with small samples, because in such cases the model must be simple and the computations will be quite fast. You can also use MCMC, which is introduced next.

Rethinking: Maximum likelihood estimation. The quadratic approximation, either with a uniform prior or with a lot of data, is often equivalent to a **MAXIMUM LIKELIHOOD ESTIMATE** (MLE) and its **STANDARD ERROR**. The MLE is a very common non-Bayesian parameter estimate. This correspondence between a Bayesian approximation and a common non-Bayesian estimator is both a blessing and a curse. It is a blessing, because it allows us to re-interpret a wide range of published non-Bayesian model fits in Bayesian terms. It is a curse, because maximum likelihood estimates have some curious drawbacks, and the quadratic approximation can share them. We'll explore these drawbacks in later chapters, and they are one of the reasons we'll turn to Markov chain Monte Carlo for the second half of the book.

Overthinking: The Hessians are coming. Sometimes it helps to know more about how the quadratic approximation is computed. In particular, the approximation sometimes fails. When it does, chances are you'll get a confusing error message that says something about the "Hessian." Students of world history may know that the Hessians were German mercenaries hired by the British in the eighteenth century to do various things, including fight against the American revolutionary George Washington. These mercenaries are named after a region of what is now central Germany, Hesse.

The Hessian that concerns us here has little to do with mercenaries. It is named after mathematician Ludwig Otto Hesse (1811–1874). A *Hessian* is a square matrix of second derivatives. It is used for many purposes in mathematics, but in the quadratic approximation it is second derivatives of the log of posterior probability with respect to the parameters. It turns out that these derivatives are sufficient to describe a Gaussian distribution, because the logarithm of a Gaussian distribution is just a parabola. Parabolas have no derivatives beyond the second, so once we know the center of the parabola (the posterior mode) and its second derivative, we know everything about it. And indeed the second derivative (with respect to the outcome) of the logarithm of a Gaussian distribution is proportional to its inverse squared standard deviation (its "precision": page 76). So knowing the standard deviation tells us everything about its shape.

The standard deviation is typically computed from the Hessian, so computing the Hessian is nearly always a necessary step. But sometimes the computation goes wrong, and your golem will choke while trying to compute the Hessian. In those cases, you have several options. Not all hope is lost. But for now it's enough to recognize the term and associate it with an attempt to find the standard deviation for a quadratic approximation.

2.4.5. Markov chain Monte Carlo. There are lots of important model types, like multilevel (mixed-effects) models, for which neither grid approximation nor quadratic approximation is always satisfactory. Such models may have hundreds or thousands or tens-of-thousands of parameters. Grid approximation routinely fails here, because it just takes too long—the Sun will go dark before your computer finishes the grid. Special forms of quadratic approximation might work, if everything is just right. But commonly, something is not just right. Furthermore, multilevel models do not always allow us to write down a single, unified function for the posterior distribution. This means that the function to maximize (when finding the MAP) is not known, but must be computed in pieces.

As a result, various counterintuitive model fitting techniques have arisen. The most popular of these is Markov chain Monte Carlo (MCMC), which is a family of conditioning engines capable of handling highly complex models. It is fair to say that MCMC is largely responsible for the insurgence of Bayesian data analysis that began in the 1990s. While MCMC is older than the 1990s, affordable computer power is not, so we must also thank the engineers. Much later in the book (Chapter 9), you'll meet simple and precise examples of MCMC model fitting, aimed at helping you understand the technique.

The conceptual challenge with MCMC lies in its highly non-obvious strategy. Instead of attempting to compute or approximate the posterior distribution directly, MCMC techniques merely draw samples from the posterior. You end up with a collection of parameter values, and the frequencies of these values correspond to the posterior plausibilities. You can then build a picture of the posterior from the histogram of these samples.

We nearly always work directly with these samples, rather than first constructing some mathematical estimate from them. And the samples are in many ways more convenient than having the posterior, because they are easier to think with. And so that's where we turn in the next chapter, to thinking with samples.

Overthinking: Monte Carlo globe tossing. If you are eager to see MCMC in action, a working Markov chain for the globe tossing model does not require much code. The following R code is sufficient for a MCMC estimate of the posterior:

R code 2.8

```
n_samples <- 1000
p <- rep( NA , n_samples )
p[1] <- 0.5
W <- 6
L <- 3
for ( i in 2:n_samples ) {
    p_new <- rnorm( 1 , p[i-1] , 0.1 )
    if ( p_new < 0 ) p_new <- abs( p_new )
    if ( p_new > 1 ) p_new <- 2 - p_new
    q0 <- dbinom( W , W+L , p[i-1] )
    q1 <- dbinom( W , W+L , p_new )
    p[i] <- ifelse( runif(1) < q1/q0 , p_new , p[i-1] )
}
```

The values in p are samples from the posterior distribution. To compare to the analytical posterior:

R code 2.9

```
dens( p , xlim=c(0,1) )
curve( dbeta( x , W+1 , L+1 ) , lty=2 , add=TRUE )
```

It's weird. But it works. I'll explain this algorithm, the Metropolis algorithm, in Chapter 9.

2.5. Summary

This chapter introduced the conceptual mechanics of Bayesian data analysis. The target of inference in Bayesian inference is a posterior probability distribution. Posterior probabilities state the relative numbers of ways each conjectured cause of the data could have produced the data. These relative numbers indicate plausibilities of the different conjectures. These plausibilities are updated in light of observations through Bayesian updating.

More mechanically, a Bayesian model is a composite of variables and distributional definitions for these variables. The probability of the data, often called the likelihood, provides the plausibility of an observation (data), given a fixed value for the parameters. The prior provides the plausibility of each possible value of the parameters, before accounting for the data. The rules of probability tell us that the logical way to compute the plausibilities, after accounting for the data, is to use Bayes' theorem. This results in the posterior distribution.

In practice, Bayesian models are fit to data using numerical techniques, like grid approximation, quadratic approximation, and Markov chain Monte Carlo. Each method imposes different trade-offs.

2.6. Practice

Problems are labeled Easy (**E**), Medium (**M**), and Hard (**H**).

2E1. Which of the expressions below correspond to the statement: *the probability of rain on Monday*?

(1) $\Pr(\text{rain})$
(2) $\Pr(\text{rain}|\text{Monday})$
(3) $\Pr(\text{Monday}|\text{rain})$
(4) $\Pr(\text{rain}, \text{Monday}) / \Pr(\text{Monday})$

2E2. Which of the following statements corresponds to the expression: $\Pr(\text{Monday}|\text{rain})$?

(1) The probability of rain on Monday.
(2) The probability of rain, given that it is Monday.
(3) The probability that it is Monday, given that it is raining.
(4) The probability that it is Monday and that it is raining.

2E3. Which of the expressions below correspond to the statement: *the probability that it is Monday, given that it is raining*?

(1) $\Pr(\text{Monday}|\text{rain})$
(2) $\Pr(\text{rain}|\text{Monday})$
(3) $\Pr(\text{rain}|\text{Monday}) \Pr(\text{Monday})$
(4) $\Pr(\text{rain}|\text{Monday}) \Pr(\text{Monday}) / \Pr(\text{rain})$
(5) $\Pr(\text{Monday}|\text{rain}) \Pr(\text{rain}) / \Pr(\text{Monday})$

2E4. The Bayesian statistician Bruno de Finetti (1906–1985) began his 1973 book on probability theory with the declaration: "PROBABILITY DOES NOT EXIST." The capitals appeared in the original, so I imagine de Finetti wanted us to shout this statement. What he meant is that probability is a device for describing uncertainty from the perspective of an observer with limited knowledge; it has no objective reality. Discuss the globe tossing example from the chapter, in light of this statement. What does it mean to say "the probability of water is 0.7"?

2M1. Recall the globe tossing model from the chapter. Compute and plot the grid approximate posterior distribution for each of the following sets of observations. In each case, assume a uniform prior for p.

(1) W, W, W
(2) W, W, W, L
(3) L, W, W, L, W, W, W

2M2. Now assume a prior for p that is equal to zero when $p < 0.5$ and is a positive constant when $p \geq 0.5$. Again compute and plot the grid approximate posterior distribution for each of the sets of observations in the problem just above.

2M3. Suppose there are two globes, one for Earth and one for Mars. The Earth globe is 70% covered in water. The Mars globe is 100% land. Further suppose that one of these globes—you don't know which—was tossed in the air and produced a "land" observation. Assume that each globe was equally likely to be tossed. Show that the posterior probability that the globe was the Earth, conditional on seeing "land" (Pr(Earth|land)), is 0.23.

2M4. Suppose you have a deck with only three cards. Each card has two sides, and each side is either black or white. One card has two black sides. The second card has one black and one white side. The third card has two white sides. Now suppose all three cards are placed in a bag and shuffled. Someone reaches into the bag and pulls out a card and places it flat on a table. A black side is shown facing up, but you don't know the color of the side facing down. Show that the probability that the other side is also black is 2/3. Use the counting method (Section 2 of the chapter) to approach this problem. This means counting up the ways that each card could produce the observed data (a black side facing up on the table).

2M5. Now suppose there are four cards: B/B, B/W, W/W, and another B/B. Again suppose a card is drawn from the bag and a black side appears face up. Again calculate the probability that the other side is black.

2M6. Imagine that black ink is heavy, and so cards with black sides are heavier than cards with white sides. As a result, it's less likely that a card with black sides is pulled from the bag. So again assume there are three cards: B/B, B/W, and W/W. After experimenting a number of times, you conclude that for every way to pull the B/B card from the bag, there are 2 ways to pull the B/W card and 3 ways to pull the W/W card. Again suppose that a card is pulled and a black side appears face up. Show that the probability the other side is black is now 0.5. Use the counting method, as before.

2M7. Assume again the original card problem, with a single card showing a black side face up. Before looking at the other side, we draw another card from the bag and lay it face up on the table. The face that is shown on the new card is white. Show that the probability that the first card, the one showing a black side, has black on its other side is now 0.75. Use the counting method, if you can. Hint: Treat this like the sequence of globe tosses, counting all the ways to see each observation, for each possible first card.

2H1. Suppose there are two species of panda bear. Both are equally common in the wild and live in the same places. They look exactly alike and eat the same food, and there is yet no genetic assay capable of telling them apart. They differ however in their family sizes. Species A gives birth to twins 10% of the time, otherwise birthing a single infant. Species B births twins 20% of the time, otherwise birthing singleton infants. Assume these numbers are known with certainty, from many years of field research.

Now suppose you are managing a captive panda breeding program. You have a new female panda of unknown species, and she has just given birth to twins. What is the probability that her next birth will also be twins?

2H2. Recall all the facts from the problem above. Now compute the probability that the panda we have is from species A, assuming we have observed only the first birth and that it was twins.

2H3. Continuing on from the previous problem, suppose the same panda mother has a second birth and that it is not twins, but a singleton infant. Compute the posterior probability that this panda is species A.

2H4. A common boast of Bayesian statisticians is that Bayesian inference makes it easy to use all of the data, even if the data are of different types.

So suppose now that a veterinarian comes along who has a new genetic test that she claims can identify the species of our mother panda. But the test, like all tests, is imperfect. This is the information you have about the test:

- The probability it correctly identifies a species A panda is 0.8.
- The probability it correctly identifies a species B panda is 0.65.

The vet administers the test to your panda and tells you that the test is positive for species A. First ignore your previous information from the births and compute the posterior probability that your panda is species A. Then redo your calculation, now using the birth data as well.