# 3 Sampling the Imaginary

Lots of books on Bayesian statistics introduce posterior inference by using a medical testing scenario. To repeat the structure of common examples, suppose there is a blood test that correctly detects vampirism 95% of the time. This implies $\Pr(\text{positive}|\text{vampire}) = 0.95$. It's a very accurate test. It does make mistakes, though, in the form of false positives. One percent of the time, it incorrectly diagnoses normal people as vampires, implying $\Pr(\text{positive}|\text{mortal}) = 0.01$. The final bit of information we are told is that vampires are rather rare, being only 0.1% of the population, implying $\Pr(\text{vampire}) = 0.001$. Suppose now that someone tests positive for vampirism. What's the probability that he or she is a bloodsucking immortal?

The correct approach is just to use Bayes' theorem to invert the probability, to compute $\Pr(\text{vampire}|\text{positive})$. The calculation can be presented as:

$$\Pr(\text{vampire}|\text{positive}) = \frac{\Pr(\text{positive}|\text{vampire})\Pr(\text{vampire})}{\Pr(\text{positive})}$$

where $\Pr(\text{positive})$ is the average probability of a positive test result, that is,

$$\Pr(\text{positive}) = \Pr(\text{positive}|\text{vampire})\Pr(\text{vampire}) + \Pr(\text{positive}|\text{mortal})\big(1 - \Pr(\text{vampire})\big)$$

Performing the calculation in R:

R code 3.1

```
PrPV <- 0.95
PrPM <- 0.01
PrV <- 0.001
PrP <- PrPV*PrV + PrPM*(1-PrV)
( PrVP <- PrPV*PrV / PrP )
```

```
[1] 0.08683729
```

That corresponds to an 8.7% chance that the suspect is actually a vampire.

Most people find this result counterintuitive. And it's a very important problem, because it mimics the structure of many realistic testing contexts, such as HIV and DNA testing, criminal profiling, and even statistical significance testing (see the rethinking box at the end of this section). Whenever the condition of interest is very rare, having a test that finds all the true cases is still no guarantee that a positive result carries much information at all. The reason is that most positive results are false positives, even when all the true positives are detected correctly.

But I don't like these examples, for two reasons. First, there's nothing really "Bayesian" about them. Remember: Bayesian inference is distinguished by a broad view of probability, not by the use of Bayes' theorem. Since all of the probabilities I provided above reference frequencies of events, rather than theoretical parameters, all major statistical philosophies would agree to use Bayes' theorem in this case. Second, and more important to our work in this chapter, these examples make Bayesian inference seem much harder than it has to be. Few people find it easy to remember which number goes where, probably because they never grasp the logic of the procedure. It's just a formula that descends from the sky.

There is a way to present the same problem that does make it more intuitive, however. Suppose that instead of reporting probabilities, as before, I tell you the following:

(1) In a population of 100,000 people, 100 of them are vampires.
(2) Of the 100 who are vampires, 95 of them will test positive for vampirism.
(3) Of the 99,900 mortals, 999 of them will test positive for vampirism.

Now tell me, if we test all 100,000 people, what proportion of those who test positive for vampirism actually are vampires? Many people, although certainly not all people, find this presentation a lot easier.[47] Now we can just count up the number of people who test positive: $95 + 999 = 1094$. Out of these 1094 positive tests, 95 of them are real vampires, so that implies:

$$\Pr(\text{vampire}|\text{positive}) = \frac{95}{1094} \approx 0.087$$

It's exactly the same answer as before, but without a seemingly arbitrary rule.

The second presentation of the problem, using counts rather than probabilities, is often called the *frequency format* or *natural frequencies*. Why a frequency format helps people intuit the correct approach remains contentious. Some people think that human psychology naturally works better when it receives information in the form a person in a natural environment would receive it. In the real world, we encounter counts only. No one has ever seen a probability, the thinking goes. But everyone sees counts ("frequencies") in their daily lives. Maybe so.

**Rethinking: The natural frequency phenomenon is not unique.** Changing the representation of a problem often makes it easier to address or inspires new ideas that were not available in an old representation.[48] In physics, switching between Newtonian and Lagrangian mechanics can make problems much easier. In evolutionary biology, switching between inclusive fitness and multilevel selection sheds new light on old models. And in statistics, switching between Bayesian and non-Bayesian representations often teaches us new things about both approaches.

Regardless of the explanation for this phenomenon, we can exploit it. And in this chapter we exploit it by taking the probability distributions from the previous chapter and sampling from them to produce counts. The posterior distribution is a probability distribution. And like all probability distributions, we can imagine drawing *samples* from it. The sampled events in this case are parameter values. Most parameters have no exact empirical realization. The Bayesian formalism treats parameter distributions as relative plausibility, not as any physical random process. In any event, randomness is always a property of information, never of the real world. But inside the computer, parameters are just as empirical as the outcome of a coin flip or a die toss or an agricultural experiment. The posterior defines the expected frequency that different parameter values will appear, once we start plucking parameters out of it.

This chapter teaches you basic skills for working with samples from the posterior distribution. It will seem a little silly to work with samples at this point, because the posterior distribution for the globe tossing model is very simple. It's so simple that it's no problem to work directly with the grid approximation or even the exact mathematical form.[49] But there are two reasons to adopt the sampling approach early on, before it's really necessary.

First, many scientists are quite shaky about integral calculus, even though they have strong and valid intuitions about how to summarize data. Working with samples transforms a problem in calculus into a problem in data summary, into a frequency format problem. An integral in a typical Bayesian context is just the total probability in some interval. That can be a challenging calculus problem. But once you have samples from the probability distribution, it's just a matter of counting values in the interval. Even seemingly simple calculations, like confidence intervals, are made difficult once a model has many parameters. In those cases, one must average over the uncertainty in all other parameters, when describing the uncertainty in a focal parameter. This requires a complicated integral, but only a very simple data summary. An empirical attack on the posterior allows the scientist to ask and answer more questions about the model, without relying upon a captive mathematician. For this reason, it is often easier and more intuitive to work with samples from the posterior, than to work with probabilities and integrals directly.

Second, some of the most capable methods of computing the posterior produce nothing but samples. Many of these methods are variants of Markov chain Monte Carlo techniques (MCMC). So if you learn early on how to conceptualize and process samples from the posterior, when you inevitably must fit a model to data using MCMC—and chances are you will—you will already know how to make sense of the output. Beginning with Chapter 8 of this book, you will use MCMC to open up the types and complexity of models you can practically fit to data. MCMC is no longer a technique only for experts, but rather part of the standard toolkit of quantitative science. So it's worth planning ahead.

So in this chapter we'll begin to use samples to summarize and simulate model output. The skills you learn here will apply to every problem in the remainder of the book, even though the details of the models, how they are fit to data, and how the samples are produced will vary.

**Rethinking: Why statistics can't save bad science.** The vampirism example at the start of this chapter has the same logical structure as many different *signal detection* problems: (1) There is some binary state that is hidden from us; (2) we observe an imperfect cue of the hidden state; (3) we (should) use Bayes' theorem to logically deduce the impact of the cue on our uncertainty.

Scientific inference is often framed in similar terms: (1) An hypothesis is either true or false; (2) we use a statistical procedure and get an imperfect cue of the hypothesis' falsity; (3) we (should) use Bayes' theorem to logically deduce the impact of the cue on the status of the hypothesis. It's the third step that is hardly ever done. But let's do it, for a toy example, so you can see how little statistical procedures—Bayesian or not—may do for us.

Suppose the probability of a positive finding, when an hypothesis is true, is $\Pr(\text{sig}|\text{true}) = 0.95$. That's the *power* of the test. Suppose that the probability of a positive finding, when an hypothesis is false, is $\Pr(\text{sig}|\text{false}) = 0.05$. That's the false-positive rate, like the 5% of conventional significance testing. Finally, we have to state the *base rate* at which hypotheses are true. Suppose for example that 1 in every 100 hypotheses turns out to be true. Then $\Pr(\text{true}) = 0.01$. No one knows this value, but the history of science suggests it's small. See Chapter 15 for more discussion. Now use Bayes' to

compute the posterior:

$$\Pr(\text{true}|\text{pos}) = \frac{\Pr(\text{pos}|\text{true})\Pr(\text{true})}{\Pr(\text{pos})} = \frac{\Pr(\text{pos}|\text{true})\Pr(\text{true})}{\Pr(\text{pos}|\text{true})\Pr(\text{true}) + \Pr(\text{pos}|\text{false})\Pr(\text{false})}$$

Plug in the appropriate values, and the answer is approximately $\Pr(\text{true}|\text{pos}) = 0.16$. So a positive finding corresponds to a 16% chance that the hypothesis is true. This is the same low base-rate phenomenon that applies in medical (and vampire) testing. You can shrink the false-positive rate to 1% and get this posterior probability up to around 0.5, only as good as a coin flip. The most important thing to do is to improve the base rate, $\Pr(\text{true})$, and that requires thinking, not testing.[50]

## 3.1. Sampling from a grid-approximate posterior

Before beginning to work with samples, we need to generate them. Here's a reminder for how to compute the posterior for the globe tossing model, using grid approximation:

R code 3.2

```
p_grid <- seq( from=0 , to=1 , length.out=1000 )
prior <- rep( 1 , 1000 )
likelihood <- dbinom( 6 , size=9 , prob=p_grid )
posterior <- likelihood * prior
posterior <- posterior / sum(posterior)
```

Now we wish to draw 10,000 samples from this posterior. Imagine the posterior is a bucket full of parameter values, numbers such as 0.1, 0.7, 0.5, 1, etc. Within the bucket, each value exists in proportion to its posterior probability, such that values near the peak are much more common than those in the tails. We're going to scoop out 10,000 values from the bucket. Provided the bucket is well mixed, the resulting samples will have the same proportions as the exact posterior density. Therefore the individual values of $p$ will appear in our samples in proportion to the posterior plausibility of each value.

Here's how you can do this in R, with one line of code:

R code 3.3

```
samples <- sample( p_grid , prob=posterior , size=1e4 , replace=TRUE )
```

The workhorse here is `sample`, which randomly pulls values from a vector. The vector in this case is `p_grid`, the grid of parameter values. The probability of each value is given by `posterior`, which you computed just above.

The resulting samples are displayed in FIGURE 3.1. On the left, all 10,000 (`1e4`) random samples are shown sequentially.

R code 3.4

```
plot( samples )
```

In this plot, it's as if you are flying over the posterior distribution, looking down on it. There are many more samples from the dense region near 0.6 and very few samples below 0.25. On the right, the plot shows the *density estimate* computed from these samples.

R code 3.5

```
library(rethinking)
dens( samples )
```

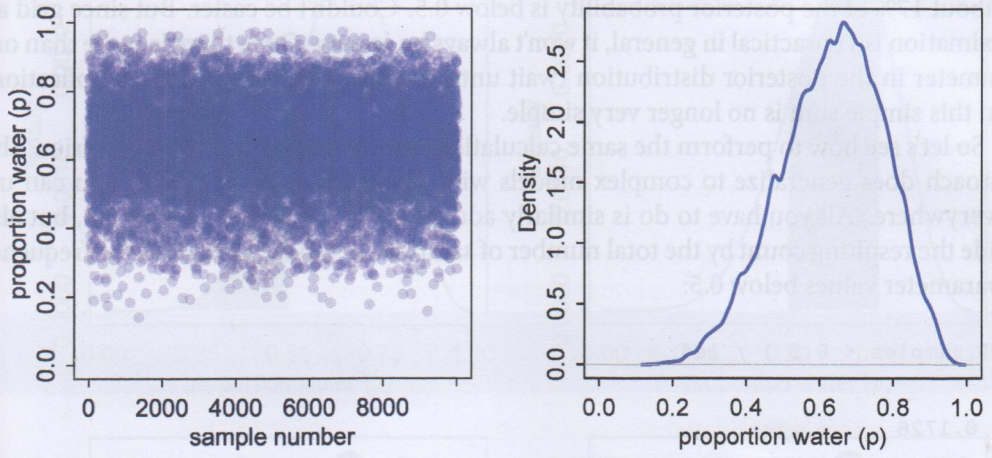


FIGURE 3.1. Sampling parameter values from the posterior distribution. Left: 10,000 samples from the posterior implied by the globe tossing data and model. Right: The density of samples (vertical) at each parameter value (horizontal).

You can see that the estimated density is very similar to ideal posterior you computed via grid approximation. If you draw even more samples, maybe `1e5` or `1e6`, the density estimate will get more and more similar to the ideal.

All you've done so far is crudely replicate the posterior density you had already computed. So next it is time to use these samples to describe and understand the posterior.

## 3.2. Sampling to summarize

Once your model produces a posterior distribution, the model's work is done. But your work has just begun. It is necessary to summarize and interpret the posterior distribution. Exactly how it is summarized depends upon your purpose. But common questions include:

- How much posterior probability lies below some parameter value?
- How much posterior probability lies between two parameter values?
- Which parameter value marks the lower 5% of the posterior probability?
- Which range of parameter values contains 90% of the posterior probability?
- Which parameter value has highest posterior probability?

These simple questions can be usefully divided into questions about (1) intervals of *defined boundaries*, (2) questions about intervals of *defined probability mass*, and (3) questions about *point estimates*. We'll see how to approach these questions using samples from the posterior.

**3.2.1. Intervals of defined boundaries.** Suppose I ask you for the posterior probability that the proportion of water is less than 0.5. Using the grid-approximate posterior, you can just add up all of the probabilities, where the corresponding parameter value is less than 0.5:

R code 3.6

```
# add up posterior probability where p < 0.5
sum( posterior[ p_grid < 0.5 ] )
```

```
[1] 0.1718746
```

So about 17% of the posterior probability is below 0.5. Couldn't be easier. But since grid approximation isn't practical in general, it won't always be so easy. Once there is more than one parameter in the posterior distribution (wait until the next chapter for that complication), even this simple sum is no longer very simple.

So let's see how to perform the same calculation, using samples from the posterior. This approach does generalize to complex models with many parameters, and so you can use it everywhere. All you have to do is similarly add up all of the samples below 0.5, but also divide the resulting count by the total number of samples. In other words, find the frequency of parameter values below 0.5:

R code 3.7

```
sum( samples < 0.5 ) / 1e4
```

```
[1] 0.1726
```

And that's nearly the same answer as the grid approximation provided, although your answer will not be exactly the same, because the exact samples you drew from the posterior will be different. This region is shown in the upper-left plot in FIGURE 3.2. Using the same approach, you can ask how much posterior probability lies between 0.5 and 0.75:

R code 3.8

```
sum( samples > 0.5 & samples < 0.75 ) / 1e4
```

```
[1] 0.6059
```

So about 61% of the posterior probability lies between 0.5 and 0.75. This region is shown in the upper-right plot of FIGURE 3.2.

---

**Overthinking: Counting with `sum`.** In the R code examples just above, I used the function `sum` to effectively count up how many samples fulfill a logical criterion. Why does this work? It works because R internally converts a logical expression, like `samples < 0.5`, to a vector of `TRUE` and `FALSE` results, one for each element of `samples`, saying whether or not each element matches the criterion. Go ahead and enter `samples < 0.5` on the R prompt, to see this for yourself. Then when you `sum` this vector of `TRUE` and `FALSE`, R counts each `TRUE` as 1 and each `FALSE` as 0. So it ends up counting how many `TRUE` values are in the vector, which is the same as the number of elements in `samples` that match the logical criterion.

---

**3.2.2. Intervals of defined mass.** It is more common to see scientific journals reporting an interval of defined mass, usually known as a **CONFIDENCE INTERVAL**. An interval of posterior probability, such as the ones we are working with, may instead be called a **CREDIBLE INTERVAL**, although the terms may also be used interchangeably, in the usual polysemy that arises when commonplace words are used in technical definitions. It's easy to keep track of what's being summarized, however, as long as you pay attention to how the model is defined.

These posterior intervals report two parameter values that contain between them a specified amount of posterior probability, a probability mass. For this type of interval, it is easier to find the answer by using samples from the posterior than by using a grid approximation. Suppose for example you want to know the boundaries of the lower 80% posterior probability. You know this interval starts at $p = 0$. To find out where it stops, think of the samples as data and ask where the 80th percentile lies:

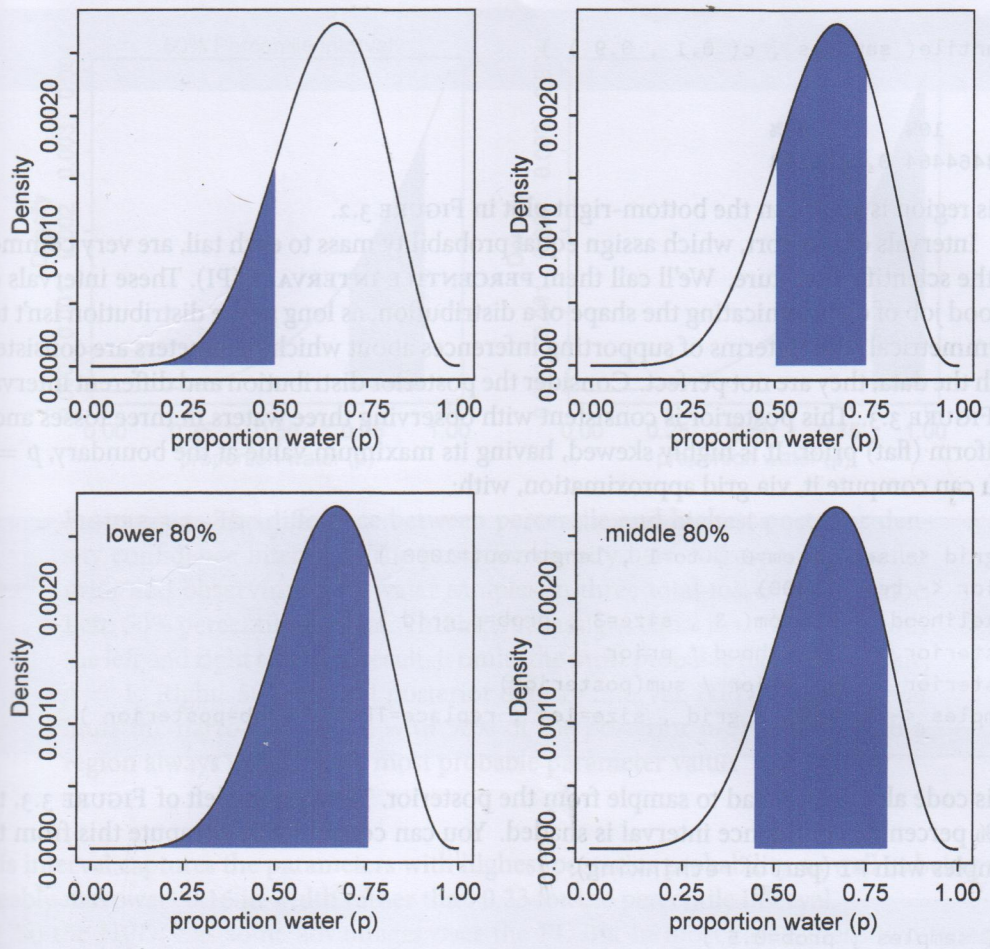


FIGURE 3.2. Two kinds of posterior interval. Top row: Intervals of defined boundaries. Top-left: The blue area is the posterior probability below a parameter value of 0.5. Top-right: The posterior probability between 0.5 and 0.75. Bottom row: Intervals of defined mass. Bottom-left: Lower 80% posterior probability exists below a parameter value of about 0.75. Bottom-right: Middle 80% posterior probability lies between the 10% and 90% quantiles.

```
quantile( samples , 0.8 )
```

R code 3.9

```
      80%
0.7607608
```

This region is shown in the bottom-left plot in FIGURE 3.2. Similarly, the middle 80% interval lies between the 10th percentile and the 90th percentile. These boundaries are found using the same approach:

R code 3.10
```
quantile( samples , c( 0.1 , 0.9 ) )
```

```
      10%       90%
0.4464464 0.8118118
```

This region is shown in the bottom-right plot in FIGURE 3.2.

quantile interval

Intervals of this sort, which assign equal probability mass to each tail, are very common in the scientific literature. We'll call them **PERCENTILE INTERVALS** (PI). These intervals do a good job of communicating the shape of a distribution, as long as the distribution isn't too asymmetrical. But in terms of supporting inferences about which parameters are consistent with the data, they are not perfect. Consider the posterior distribution and different intervals in FIGURE 3.3. This posterior is consistent with observing three waters in three tosses and a uniform (flat) prior. It is highly skewed, having its maximum value at the boundary, $p = 1$. You can compute it, via grid approximation, with:

R code 3.11
```
p_grid <- seq( from=0 , to=1 , length.out=1000 )
prior <- rep(1,1000)
likelihood <- dbinom( 3 , size=3 , prob=p_grid )
posterior <- likelihood * prior
posterior <- posterior / sum(posterior)
samples <- sample( p_grid , size=1e4 , replace=TRUE , prob=posterior )
```

This code also goes ahead to sample from the posterior. Now, on the left of FIGURE 3.3, the 50% percentile confidence interval is shaded. You can conveniently compute this from the samples with `PI` (part of `rethinking`):

R code 3.12
```
PI( samples , prob=0.5 )
```

```
      25%       75%
0.7037037 0.9329329
```

This interval assigns 25% of the probability mass above and below the interval. So it provides the central 50% probability. But in this example, it ends up excluding the most probable parameter values, near $p = 1$. So in terms of describing the shape of the posterior distribution—which is really all these intervals are asked to do—the percentile interval can be misleading.

In contrast, the right-hand plot in FIGURE 3.3 displays the 50% **HIGHEST POSTERIOR DENSITY INTERVAL** (HPDI).[51] The HPDI is the narrowest interval containing the specified probability mass. If you think about it, there must be an infinite number of posterior intervals with the same mass. But if you want an interval that best represents the parameter values most consistent with the data, then you want the densest of these intervals. That's what the HPDI is. Compute it from the samples with `HPDI` (also part of `rethinking`):

R code 3.13
```
HPDI( samples , prob=0.5 )
```

```
    |0.5       0.5|
0.8408408 1.0000000
```

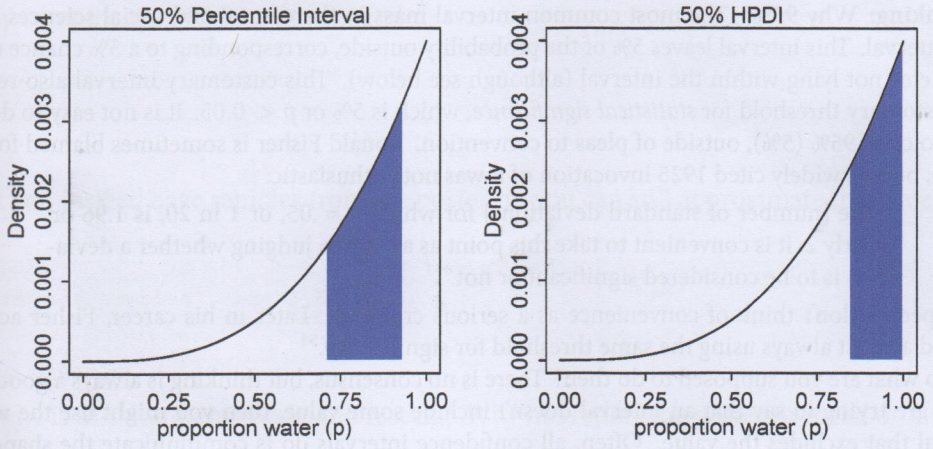


FIGURE 3.3. The difference between percentile and highest posterior density confidence intervals. The posterior density here corresponds to a flat prior and observing three water samples in three total tosses of the globe. Left: 50% percentile interval. This interval assigns equal mass (25%) to both the left and right tail. As a result, it omits the most probable parameter value, $p = 1$. Right: 50% highest posterior density interval, HPDI. This interval finds the narrowest region with 50% of the posterior probability. Such a region always includes the most probable parameter value.

This interval captures the parameters with highest posterior probability, as well as being noticeably narrower: 0.16 in width rather than 0.23 for the percentile interval.

So the HPDI has some advantages over the PI. But in most cases, these two types of interval are very similar.[52] They only look so different in this case because the posterior distribution is highly skewed. If we instead used samples from the posterior distribution for six waters in nine tosses, these intervals would be nearly identical. Try it for yourself, using different probability masses, such as `prob=0.8` and `prob=0.95`. When the posterior is bell shaped, it hardly matters which type of interval you use. Remember, we're not launching rockets or calibrating atom smashers, so fetishizing precision to the 5th decimal place will not improve your science.

The HPDI also has some disadvantages. HPDI is more computationally intensive than PI and suffers from greater *simulation variance*, which is a fancy way of saying that it is sensitive to how many samples you draw from the posterior. It is also harder to understand and many scientific audiences will not appreciate its features, while they will immediately understand a percentile interval, as ordinary non-Bayesian intervals are nearly always percentile intervals (although of sampling distributions, not posterior distributions).

Overall, if the choice of interval type makes a big difference, then you shouldn't be using intervals to summarize the posterior. Remember, the entire posterior distribution is the Bayesian estimate. It summarizes the relative plausibilities of each possible value of the parameter. Intervals of the distribution are just helpful for summarizing it. If choice of interval leads to different inferences, then you'd be better off just plotting the entire posterior distribution.

**Rethinking: Why 95%?** The most common interval mass in the natural and social sciences is the 95% interval. This interval leaves 5% of the probability outside, corresponding to a 5% chance of the parameter not lying within the interval (although see below). This customary interval also reflects the customary threshold for *statistical significance*, which is 5% or $p < 0.05$. It is not easy to defend the choice of 95% (5%), outside of pleas to convention. Ronald Fisher is sometimes blamed for this choice, but his widely cited 1925 invocation of it was not enthusiastic:

> "The [number of standard deviations] for which $P = .05$, or 1 in 20, is 1.96 or nearly 2; it is convenient to take this point as a limit in judging whether a deviation is to be considered significant or not."[53]

Most people don't think of convenience as a serious criterion. Later in his career, Fisher actively advised against always using the same threshold for significance.[54]

So what are you supposed to do then? There is no consensus, but thinking is always a good idea. If you are trying to say that an interval doesn't include some value, then you might use the widest interval that excludes the value. Often, all confidence intervals do is communicate the shape of a distribution. In that case, a series of nested intervals may be more useful than any one interval. For example, why not present 67%, 89%, and 97% intervals, along with the median? Why these values? No reason. They are prime numbers, which makes them easy to remember. But all that matters is they be spaced enough to illustrate the shape of the posterior. And these values avoid 95%, since conventional 95% intervals encourage many readers to conduct unconscious hypothesis tests.

**Rethinking: What do confidence intervals mean?** It is common to hear that a 95% confidence interval means that there is a probability 0.95 that the true parameter value lies within the interval. In strict non-Bayesian statistical inference, such a statement is never correct, because strict non-Bayesian inference forbids using probability to measure uncertainty about parameters. Instead, one should say that if we repeated the study and analysis a very large number of times, then 95% of the computed intervals would contain the true parameter value. If the distinction is not entirely clear to you, then you are in good company. Most scientists find the definition of a confidence interval to be bewildering, and many of them slip unconsciously into a Bayesian interpretation.

But whether you use a Bayesian interpretation or not, a 95% interval does not contain the true value 95% of the time. The history of science teaches us that confidence intervals exhibit chronic overconfidence.[55] The word *true* should set off alarms that something is wrong with a statement like "contains the true value." The 95% is a *small world* number (see the introduction to Chapter 2), only true in the model's logical world. So it will never apply exactly to the real or *large world*. It is what the golem believes, but you are free to believe something else. Regardless, the width of the interval, and the values it covers, can provide valuable advice.

**3.2.3. Point estimates.** The third and final common summary task for the posterior is to produce point estimates of some kind. Given the entire posterior distribution, what value should you report? This seems like an innocent question, but it is difficult to answer. The Bayesian parameter estimate is precisely the entire posterior distribution, which is not a single number, but instead a function that maps each unique parameter value onto a plausibility value. So really the most important thing to note is that you don't have to choose a point estimate. It's hardly ever necessary.

But if you must produce a point estimate from the posterior, you'll have to ask and answer more questions. Consider the following example. Suppose again the globe tossing experiment in which we observe 3 waters out of 3 tosses, as in FIGURE 3.3. Let's consider three alternative point estimates. First, it is very common for scientists to report the parameter

value with highest posterior probability, a *maximum a posteriori* (MAP) estimate. You can easily compute the MAP in this example:

R code 3.14

```
p_grid[ which.max(posterior) ]
```

```
[1] 1
```

Or if you instead have samples from the posterior, you can still approximate the same point:

R code 3.15

```
chainmode( samples , adj=0.01 )
```

```
[1] 0.9985486
```

But why is this point, the mode, interesting? Why not report the posterior mean or median?

R code 3.16

```
mean( samples )
median( samples )
```

```
[1] 0.8005558
[1] 0.8408408
```

These are also point estimates, and they also summarize the posterior. But all three—the mode (MAP), mean, and median—are different in this case. How can we choose among them? FIGURE 3.4 shows this posterior distribution and the locations of these point summaries.

One principled way to go beyond using the entire posterior as the estimate is to choose a **LOSS FUNCTION**. A loss function is a rule that tells you the cost associated with using any particular point estimate. While statisticians and game theorists have long been interested in loss functions, and how Bayesian inference supports them, scientists hardly ever use them explicitly. The key insight is that *different loss functions imply different point estimates*.

Here's an example to help us work through the procedure. Suppose I offer you a bet. Tell me which value of $p$, the proportion of water on the Earth, you think is correct. I will pay you \$100, if you get it exactly right. But I will subtract money from your gain, proportional to the distance of your decision from the correct value. Precisely, your loss is proportional to the absolute value of $d - p$, where $d$ is your decision and $p$ is the correct answer. We could change the precise dollar values involved, without changing the important aspects of this problem. What matters is that the loss is proportional to the distance of your decision from the true value.

Now once you have the posterior distribution in hand, how should you use it to maximize your expected winnings? It turns out that the parameter value that maximizes expected winnings (minimizes expected loss) is the median of the posterior distribution. Let's calculate that fact, without using a mathematical proof. Those interested in the proof should follow the endnote.[56]

Calculating expected loss for any given decision means using the posterior to average over our uncertainty in the true value. Of course we don't know the true value, in most cases. But if we are going to use our model's information about the parameter, that means using the entire posterior distribution. So suppose we decide $p = 0.5$ will be our decision. Then the expected loss will be:

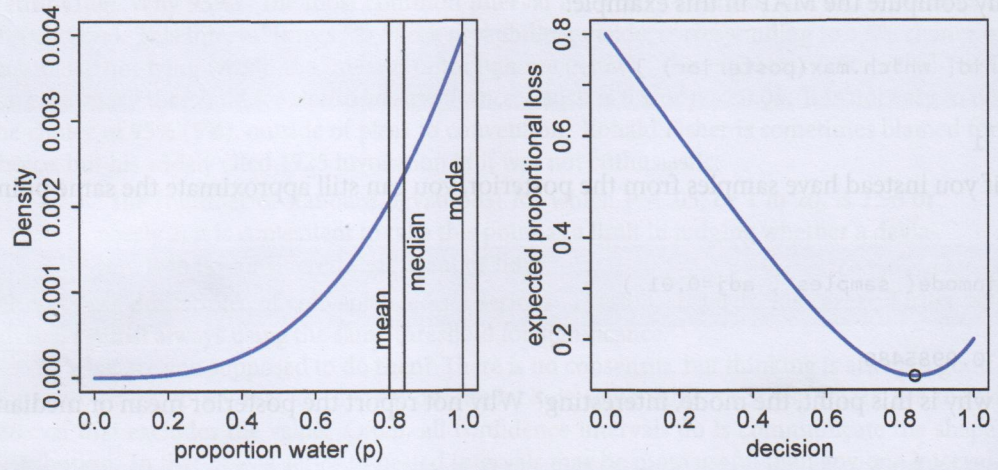


FIGURE 3.4. Point estimates and loss functions. Left: Posterior distribution (blue) after observing 3 water in 3 tosses of the globe. Vertical lines show the locations of the mode, median, and mean. Each point implies a different loss function. Right: Expected loss under the rule that loss is proportional to absolute distance of decision (horizontal axis) from the true value. The point marks the value of $p$ that minimizes the expected loss, the posterior median.

R code 3.17

```
sum( posterior*abs( 0.5 - p_grid ) )
```

```
[1] 0.3128752
```

The symbols `posterior` and `p_grid` are the same ones we've been using throughout this chapter, containing the posterior probabilities and the parameter values, respectively. All the code above does is compute the weighted average loss, where each loss is weighted by its corresponding posterior probability. There's a trick for repeating this calculation for every possible decision, using the function `sapply`.

R code 3.18

```
loss <- sapply( p_grid , function(d) sum( posterior*abs( d - p_grid ) ) )
```

Now the symbol `loss` contains a list of loss values, one for each possible decision, corresponding the values in `p_grid`. From here, it's easy to find the parameter value that minimizes the loss:

R code 3.19

```
p_grid[ which.min(loss) ]
```

```
[1] 0.8408408
```

And this is actually the posterior median, the parameter value that splits the posterior density such that half of the mass is above it and half below it. Try `median(samples)` for comparison. It may not be exactly the same value, due to sampling variation, but it will be close.

So what are we to learn from all of this? In order to decide upon a *point estimate*, a single-value summary of the posterior distribution, we need to pick a loss function. Different loss functions nominate different point estimates. The two most common examples are the absolute loss as above, which leads to the median as the point estimate, and the quadratic loss $(d - p)^2$, which leads to the posterior mean (`mean(samples)`) as the point estimate. When the posterior distribution is symmetrical and normal-looking, then the median and mean converge to the same point, which relaxes some anxiety we might have about choosing a loss function. For the original globe tossing data (6 waters in 9 tosses), for example, the mean and median are barely different.

In principle, though, the details of the applied context may demand a rather unique loss function. Consider a practical example like deciding whether or not to order an evacuation, based upon an estimate of hurricane wind speed. Damage to life and property increases very rapidly as wind speed increases. There are also costs to ordering an evacuation when none is needed, but these are much smaller. Therefore the implied loss function is highly asymmetric, rising sharply as true wind speed exceeds our guess, but rising only slowly as true wind speed falls below our guess. In this context, the optimal point estimate would tend to be larger than posterior mean or median. Moreover, the real issue is whether or not to order an evacuation, and so producing a point estimate of wind speed may not be necessary at all.

Usually, research scientists don't think about loss functions. And so any point estimate like the mean or MAP that they may report isn't intended to support any particular decision, but rather to describe the shape of the posterior. You might argue that the decision to make is whether or not to accept an hypothesis. But the challenge then is to say what the relevant costs and benefits would be, in terms of the knowledge gained or lost.[57] Usually it's better to communicate as much as you can about the posterior distribution, as well as the data and the model itself, so that others can build upon your work. Premature decisions to accept or reject hypotheses can cost lives.[58]

It's healthy to keep these issues in mind, if only because they remind us that many of the routine questions in statistical inference can only be answered under consideration of a particular empirical context and applied purpose. Statisticians can provide general outlines and standard answers, but a motivated and attentive scientist will always be able to improve upon such general advice.

## 3.3. Sampling to simulate prediction

Another common job for samples from the posterior is to ease SIMULATION of the model's implied observations. Generating implied observations from a model is useful for at least four distinct reasons.

(1) *Model checking.* After a model is fit to real data, it is worth simulating implied observations, to check both whether the fit worked correctly and to investigate model behavior.

(2) *Software validation.* In order to be sure that our model fitting software is working, it helps to simulate observations under a known model and then attempt to recover the values of the parameters the data were simulated under.

(3) *Research design.* If you can simulate observations from your hypothesis, then you can evaluate whether the research design can be effective. In a narrow sense, this means doing *power analysis*, but the possibilities are much broader.

(4) *Forecasting.* Estimates can be used to simulate new predictions, for new cases and future observations. These forecasts can be useful as applied prediction, but also for model criticism and revision.

In this final section of the chapter, we'll look at how to produce simulated observations and how to perform some simple model checks.

**3.3.1. Dummy data.** Let's summarize the globe tossing model that you've been working with for two chapters now. A fixed true proportion of water $p$ exists, and that is the target of our inference. Tossing the globe in the air and catching it produces observations of "water" and "land" that appear in proportion to $p$ and $1 - p$, respectively.

Now note that these assumptions not only allow us to infer the plausibility of each possible value of $p$, after observation. That's what you did in the previous chapter. These assumptions also allow us to simulate the observations that the model implies. They allow this, because likelihood functions work in both directions. Given a realized observation, the likelihood function says how plausible the observation is. And given only the parameters, the likelihood defines a distribution of possible observations that we can sample from, to simulate observation. In this way, Bayesian models are always *generative*, capable of simulating predictions. Many non-Bayesian models are also generative, but many are not.

We will call such simulated data **DUMMY DATA**, to indicate that it is a stand-in for actual data. With the globe tossing model, the dummy data arises from a binomial likelihood:

$$\Pr(w|n,p) = \frac{n!}{w!(n-w)!}p^w(1-p)^{n-w}$$

where $w$ is an observed count of "water" and $n$ is the number of tosses. Suppose $n = 2$, two tosses of the globe. Then there are only three possible observations: 0 water, 1 water, 2 water. You can quickly compute the likelihood of each, for any given value of $p$. Let's use $p = 0.7$, which is just about the true proportion of water on the Earth:

R code 3.20

```
dbinom( 0:2 , size=2 , prob=0.7 )
```

```
[1] 0.09 0.42 0.49
```

This means that there's a 9% chance of observing $w = 0$, a 42% chance of $w = 1$, and a 49% chance of $w = 2$. If you change the value of $p$, you'll get a different distribution of implied observations.

Now we're going to simulate observations, using these likelihoods. This is done by sampling from the distribution just described above. You could use `sample` to do this, but R provides convenient sampling functions for all the ordinary probability distributions, like the binomial. So a single dummy data observation of $w$ can be sampled with:

R code 3.21

```
rbinom( 1 , size=2 , prob=0.7 )
```

```
[1] 1
```

The "r" in `rbinom` stands for "random." It can also generate more than one simulation at a time. A set of 10 simulations can be made by:

R code 3.22

```
rbinom( 10 , size=2 , prob=0.7 )
```

```
[1] 2 2 2 1 2 1 1 1 0 2
```

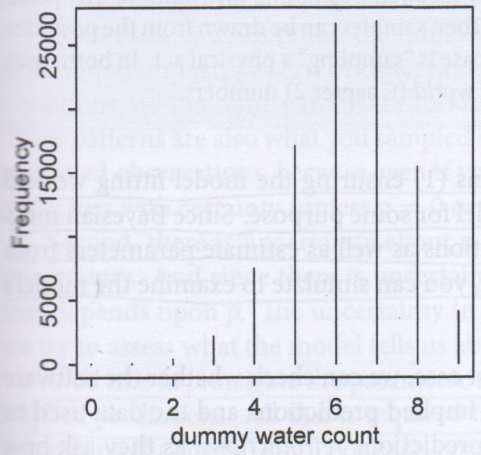


FIGURE 3.5. Distribution of simulated sample observations from 9 tosses of the globe. These samples assume the proportion of water is 0.7.

Let's generate 100,000 dummy observations, just to verify that each value (0, 1, or 2) appears in proportion to its likelihood:

R code 3.23

```
dummy_w <- rbinom( 1e5 , size=2 , prob=0.7 )
table(dummy_w)/1e5
```

```
dummy_w
      0       1       2
0.08904 0.41948 0.49148
```

And those values are very close to the analytically calculated likelihoods further up. You will see slightly different values, due to simulation variance. Execute the code above multiple times, to see how the exact realized frequencies fluctuate from simulation to simulation.

Only two tosses of the globe isn't much of a sample, though. So now let's simulate the same sample size as before, 9 tosses.

R code 3.24

```
dummy_w <- rbinom( 1e5 , size=9 , prob=0.7 )
simplehist( dummy_w , xlab="dummy water count" )
```

The resulting plot is shown in FIGURE 3.5. Notice that most of the time the expected observation does not contain water in its true proportion, 0.7. That's the nature of observation: There is a one-to-many relationship between data and data-generating processes. You should experiment with sample size, the `size` input in the code above, as well as the `prob`, to see how the distribution of simulated samples changes shape and location.

So that's how to perform a basic simulation of observations. What good is this? There are many useful jobs for these samples. In this chapter, we'll put them to use in examining the implied predictions of a model. But to do that, we'll have to combine them with samples from the posterior distribution. That's next.

**Rethinking: Sampling distributions.** Many readers will already have seen simulated observations. SAMPLING DISTRIBUTIONS are the foundation of common non-Bayesian statistical traditions. In those approaches, inference about parameters is made through the sampling distribution. In this

book, inference about parameters is never done directly through a sampling distribution. The posterior distribution is not sampled, but deduced logically. Then samples can be drawn from the posterior, as earlier in this chapter, to aid in inference. In neither case is "sampling" a physical act. In both cases, it's just a mathematical device and produces only *small world* (Chapter 2) numbers.

**3.3.2. Model checking.** MODEL CHECKING means (1) ensuring the model fitting worked correctly and (2) evaluating the adequacy of a model for some purpose. Since Bayesian models are always *generative*, able to simulate observations as well as estimate parameters from observations, once you condition a model on data, you can simulate to examine the model's empirical expectations.

3.3.2.1. *Did the software work?* In the simplest case, we can check whether the software worked by checking for correspondence between implied predictions and the data used to fit the model. You might also call these implied predictions *retrodictions*, as they ask how well the model reproduces the data used to educate it. An exact match is neither expected nor desired. But when there is no correspondence at all, it probably means the software did something wrong.

There is no way to really be sure that software works correctly. Even when the retrodictions correspond to the observed data, there may be subtle mistakes. And when you start working with multilevel models, you'll have to expect a certain pattern of lack of correspondence between retrodictions and observations. Despite there being no perfect way to ensure software has worked, the simple check I'm encouraging here often catches silly mistakes, mistakes of the kind everyone makes from time to time.

In the case of the globe tossing analysis, the software implementation is simple enough that it can be checked against analytical results. So instead let's move directly to considering the model's adequacy.

3.3.2.2. *Is the model adequate?* After assessing whether the posterior distribution is the correct one, because the software worked correctly, it's useful to also look for aspects of the data that are not well described by the model's expectations. The goal is not to test whether the model's assumptions are "true," because all models are false. Rather, the goal is to assess exactly how the model fails to describe the data, as a path towards model comprehension, revision, and improvement.

All models fail in some respect, so you have to use your judgment—as well as the judgments of your colleagues—to decide whether any particular failure is or is not important. Few scientists want to produce models that do nothing more than re-describe existing samples. So imperfect prediction (retrodiction) is not a bad thing. Typically we hope to either predict future observations or understand enough that we might usefully tinker with the world. We'll consider these problems again, in Chapter 6.

For now, we need to learn how to combine sampling of simulated observations, as in the previous section, with sampling parameters from the posterior distribution. We expect to do better when we use the entire posterior distribution, not just some point estimate derived from it. Why? Because there is a lot of information about uncertainty in the entire posterior distribution. We lose this information when we pluck out a single parameter value and then perform calculations with it. This loss of information leads to overconfidence.

Let's do some basic model checks, using simulated observations for the globe tossing model. The observations in our example case are counts of water, over tosses of the globe.

The implied predictions of the model are uncertain in two ways, and it's important to be aware of both.

First, there is observation uncertainty. For any unique value of the parameter $p$, there is a unique implied pattern of observations that the model expects. These patterns of observations are the same gardens of forking data that you explored in the previous chapter. These patterns are also what you sampled in the previous section. There is uncertainty in the predicted observations, because even if you know $p$ with certainty, you won't know the next globe toss with certainty (unless $p = 0$ or $p = 1$).

Second, there is uncertainty about $p$. The posterior distribution over $p$ embodies this uncertainty. And since there is uncertainty about $p$, there is uncertainty about everything that depends upon $p$. The uncertainty in $p$ will interact with the sampling variation, when we try to assess what the model tells us about outcomes.

We'd like to *propagate* the parameter uncertainty—carry it forward—as we evaluate the implied predictions. All that is required is averaging over the posterior density for $p$, while computing the predictions. For each possible value of the parameter $p$, there is an implied distribution of outcomes. So if you were to compute the sampling distribution of outcomes at each value of $p$, then you could average all of these prediction distributions together, using the posterior probabilities of each value of $p$, to get a **POSTERIOR PREDICTIVE DISTRIBUTION**.

FIGURE 3.6 illustrates this averaging. At the top, the posterior distribution is shown, with 10 unique parameter values highlighted by the vertical lines. The implied distribution of observations specific to each of these parameter values is shown in the middle row of plots. Observations are never certain for any value of $p$, but they do shift around in response to it. Finally, at the bottom, the sampling distributions for all values of $p$ are combined, using the posterior probabilities to compute the weighted average frequency of each possible observation, zero to nine water samples.

The resulting distribution is for predictions, but it incorporates all of the uncertainty embodied in the posterior distribution for the parameter $p$. As a result, it is honest. While the model does a good job of predicting the data—the most likely observation is indeed the observed data—predictions are still quite spread out. If instead you were to use only a single parameter value to compute implied predictions, say the most probable value at the peak of posterior distribution, you'd produce an overconfident distribution of predictions, narrower than the posterior predictive distribution in FIGURE 3.6 and more like the sampling distribution shown for $p = 0.6$ in the middle row. The usual effect of this overconfidence will be to lead you to believe that the model is more consistent with the data than it really is—the predictions will cluster around the observations more tightly. This illusion arises from tossing away uncertainty about the parameters.

So how do you actually do the calculations? To simulate predicted observations for a single value of $p$, say $p = 0.6$, you can use `rbinom` to generate random binomial samples:

R code 3.25

```
w <- rbinom( 1e4 , size=9 , prob=0.6 )
```

This generates 10,000 (`1e4`) simulated predictions of 9 globe tosses (`size=9`), assuming $p = 0.6$. The predictions are stored as counts of water, so the theoretical minimum is zero and the theoretical maximum is nine. You can use `simplehist(w)` (in the `rethinking` package) to get a clean histogram of your simulated outcomes.

All you need to propagate parameter uncertainty into these predictions is replace the value `0.6` with samples from the posterior:

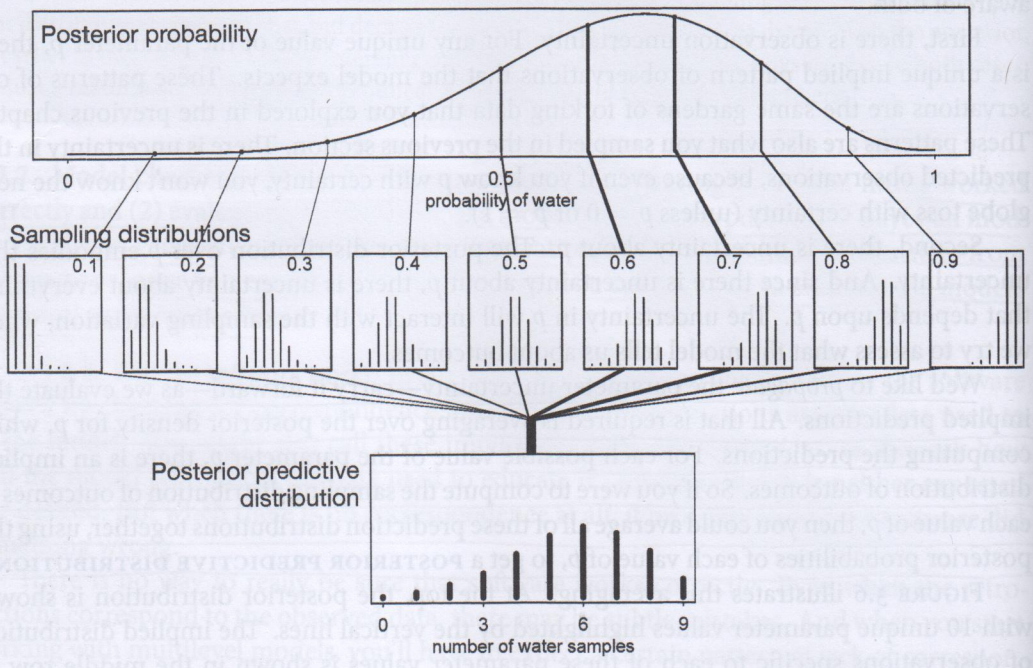


FIGURE 3.6. Simulating predictions from the total posterior. Top: The familiar posterior distribution for the globe tossing data. Ten example parameter values are marked by the vertical lines. Values with greater posterior probability indicated by thicker lines. Middle row: Each of the ten parameter values implies a unique sampling distribution of predictions. Bottom: Combining simulated observation distributions for all parameter values (not just the ten shown), each weighted by its posterior probability, produces the posterior predictive distribution. This distribution propagates uncertainty about parameter to uncertainty about prediction.

R code 3.26

```
w <- rbinom( 1e4 , size=9 , prob=samples )
```

The symbol `samples` above is the same list of random samples from the posterior distribution that you've used in previous sections. For each sampled value, a random binomial observation is generated. Since the sampled values appear in proportion to their posterior probabilities, the resulting simulated observations are averaged over the posterior. You can manipulate these simulated observations just like you manipulate samples from the posterior—you can compute intervals and point statistics using the same procedures. If you plot these samples, you'll see the distribution shown in the right-hand plot in FIGURE 3.6.

The simulated model predictions are quite consistent with the observed data in this case—the actual count of 6 lies right in the middle of the simulated distribution. There is quite a lot of spread to the predictions, but a lot of this spread arises from the binomial process itself, not uncertainty about $p$. Still, it'd be premature to conclude that the model is perfect. So far, we've only viewed the data just as the model views it: Each toss of the globe is completely independent of the others. This assumption is questionable. Unless the person

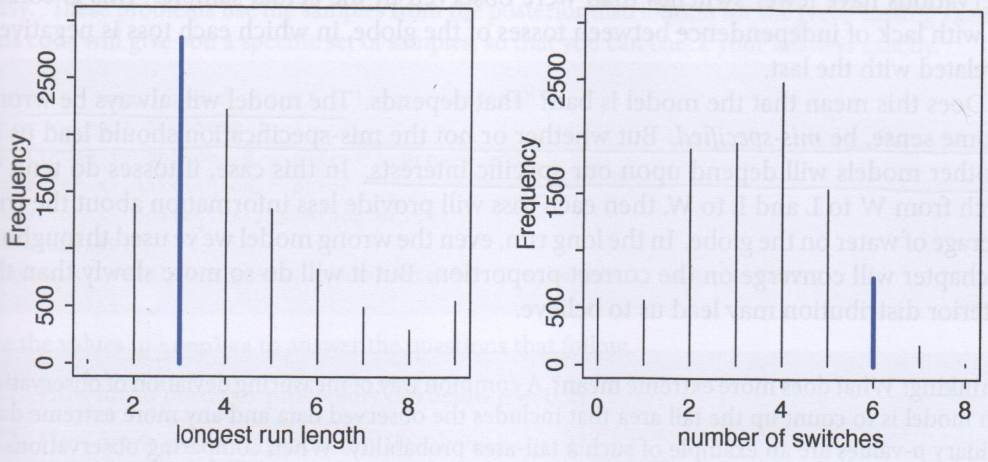


FIGURE 3.7. Alternative views of the same posterior predictive distribution (see FIGURE 3.6). Instead of considering the data as the model saw it, as a sum of water samples, now we view the data as both the length of the maximum run of water or land (left) and the number of switches between water and land samples (right). Observed values highlighted in blue. While the simulated predictions are consistent with the run length (3 water in a row), they are much less consistent with the frequent switches (6 switches in 9 tosses).

tossing the globe is careful, it is easy to induce correlations and therefore patterns among the sequential tosses. Consider for example that about half of the globe (and planet) is covered by the Pacific Ocean. As a result, water and land are not uniformly distributed on the globe, and therefore unless the globe spins and rotates enough while in the air, the position when tossed could easily influence the sample once it lands. The same problem arises in coin tosses, and indeed skilled individuals can influence the outcome of a coin toss, by exploiting the physics of it.[59]

So with the goal of seeking out aspects of prediction in which the model fails, let's look at the data in two different ways. Recall that the sequence of nine tosses was W L W W W L W L W. First, consider the length of the longest run of either water or land. This will provide a crude measure of correlation between tosses. So in the observed data, the longest run is 3 W's. Second, consider the number of times in the data that the sample switches from water to land or from land to water. This is another measure of correlation between samples. In the observed data, the number of switches is 6. There is nothing special about these two new ways of describing the data. They just serve to inspect the data in new ways. In your own modeling, you'll have to imagine aspects of the data that are relevant in your context, for your purposes.

FIGURE 3.7 shows the simulated predictions, viewed in these two new ways. On the left, the length of the longest run of water or land is plotted, with the observed value of 3 highlighted by the bold line. Again, the true observation is the most common simulated observation, but with a lot of spread around it. On the right, the number of switches from water to land and land to water is shown, with the observed value of 6 highlighted in bold. Now

the simulated predictions appear less consistent with the data, as the majority of simulated observations have fewer switches than were observed in the actual sample. This is consistent with lack of independence between tosses of the globe, in which each toss is negatively correlated with the last.

Does this mean that the model is bad? That depends. The model will always be wrong in some sense, be *mis-specified*. But whether or not the mis-specification should lead us to try other models will depend upon our specific interests. In this case, if tosses do tend to switch from W to L and L to W, then each toss will provide less information about the true coverage of water on the globe. In the long run, even the wrong model we've used throughout the chapter will converge on the correct proportion. But it will do so more slowly than the posterior distribution may lead us to believe.

**Rethinking: What does more extreme mean?** A common way of measuring deviation of observation from model is to count up the tail area that includes the observed data and any more extreme data. Ordinary $p$-values are an example of such a tail-area probability. When comparing observations to distributions of simulated predictions, as in FIGURE 3.6 and FIGURE 3.7, we might wonder how far out in the tail the observed data must be before we conclude that the model is a poor one. Because statistical contexts vary so much, it's impossible to give a universally useful answer.

But more importantly, there are usually very many ways to view data and define "extreme." Ordinary $p$-values view the data in just the way the model expects it, and so provide a very weak form of model checking. For example, the far-right plot in FIGURE 3.6 evaluates model fit in the best way for the model. Alternative ways of defining "extreme" may provide a more serious challenge to a model. The different definitions of extreme in FIGURE 3.7 can more easily embarrass it.

Model fitting remains an objective procedure—everyone and every golem conducts Bayesian updating in a way that doesn't depend upon personal preferences. But model checking is inherently subjective, and this actually allows it to be quite powerful, since subjective knowledge of an empirical domain provides expertise. Expertise in turn allows for imaginative checks of model performance. Since golems have terrible imaginations, we need the freedom to engage our own imaginations. In this way, the objective and subjective work together.[60]

## 3.4. Summary

This chapter introduced the basic procedures for manipulating posterior distributions. Our fundamental tool is samples of parameter values drawn from the posterior distribution. Working with samples transforms a problem of integral calculus into a problem of data summary. These samples can be used to produce intervals, point estimates, posterior predictive checks, as well as other kinds of simulations.

Posterior predictive checks combine uncertainty about parameters, as described by the posterior distribution, with uncertainty about outcomes, as described by the assumed likelihood function. These checks are useful for verifying that your software worked correctly. They are also useful for prospecting for ways in which your models are inadequate.

Once models become more complex, posterior predictive simulations will be used for a broader range of applications. Even understanding a model often requires simulating implied observations. We'll keep working with samples from the posterior, to make these tasks as easy and customizable as possible.

**Easy.** These problem
This code will give y

```
p_grid <- seq( fro
prior <- rep( 1 ,
likelihood <- dbin
posterior <- likel
posterior <- poste
set.seed(100)
samples <- sample(
```

Use the values in sa

**3E1.** How much po

**3E2.** How much po

**3E3.** How much po

**3E4.** 20% of the po

**3E5.** 20% of the po

**3E6.** Which values

**3E7.** Which values
ity both below and a

**Medium.**

**3M1.** Suppose the
rior distribution, us

**3M2.** Draw 10,000
culate the 90% HPI

**3M3.** Construct a
bution of samples, a
8 water in 15 tosses

**3M4.** Using the po
ability of observing

**3M5.** Start over at
This corresponds t
problem above and
compare inference

## 3.5. Practice

**Easy.** These problems use the samples from the posterior distribution for the globe tossing example. This code will give you a specific set of samples, so that you can check your answers exactly.

R code 3.27

```
p_grid <- seq( from=0 , to=1 , length.out=1000 )
prior <- rep( 1 , 1000 )
likelihood <- dbinom( 6 , size=9 , prob=p_grid )
posterior <- likelihood * prior
posterior <- posterior / sum(posterior)
set.seed(100)
samples <- sample( p_grid , prob=posterior , size=1e4 , replace=TRUE )
```

Use the values in `samples` to answer the questions that follow.

**3E1.** How much posterior probability lies below $p = 0.2$?

**3E2.** How much posterior probability lies above $p = 0.8$?

**3E3.** How much posterior probability lies between $p = 0.2$ and $p = 0.8$?

**3E4.** 20% of the posterior probability lies below which value of $p$?

**3E5.** 20% of the posterior probability lies above which value of $p$?

**3E6.** Which values of $p$ contain the narrowest interval equal to 66% of the posterior probability?

**3E7.** Which values of $p$ contain 66% of the posterior probability, assuming equal posterior probability both below and above the interval?

**Medium.**

**3M1.** Suppose the globe tossing data had turned out to be 8 water in 15 tosses. Construct the posterior distribution, using grid approximation. Use the same flat prior as before.

**3M2.** Draw 10,000 samples from the grid approximation from above. Then use the samples to calculate the 90% HPDI for $p$.

**3M3.** Construct a posterior predictive check for this model and data. This means simulate the distribution of samples, averaging over the posterior uncertainty in $p$. What is the probability of observing 8 water in 15 tosses?

**3M4.** Using the posterior distribution constructed from the new (8/15) data, now calculate the probability of observing 6 water in 9 tosses.

**3M5.** Start over at **3M1**, but now use a prior that is zero below $p = 0.5$ and a constant above $p = 0.5$. This corresponds to prior information that a majority of the Earth's surface is water. Repeat each problem above and compare the inferences. What difference does the better prior make? If it helps, compare inferences (using both priors) to the true value $p = 0.7$.

**Hard.**

**Introduction.** The practice problems here all use the data below. These data indicate the gender (male=1, female=0) of officially reported first and second born children in 100 two-child families.

R code 3.28

```
birth1 <- c(1,0,0,0,1,1,0,1,0,1,0,0,1,1,0,1,1,0,0,0,1,0,0,0,1,0,
0,0,0,1,1,1,0,1,0,1,1,1,0,1,0,1,1,0,1,0,1,0,0,0,0,0,0,0,
1,1,0,1,0,0,1,0,0,0,1,0,0,1,1,1,1,0,1,0,1,1,1,1,1,0,0,1,0,1,1,0,
1,0,1,1,1,0,1,1,1,1)
birth2 <- c(0,1,0,1,0,1,1,1,0,0,1,1,1,1,1,0,0,1,1,1,0,0,1,1,1,0,
1,1,1,0,1,1,1,0,1,0,0,1,1,1,1,0,0,1,0,1,1,1,1,1,1,1,1,1,1,1,1,
1,1,1,0,1,1,0,1,1,0,1,1,1,0,0,0,0,0,1,0,0,0,1,1,0,0,1,0,0,1,1,
0,0,0,1,1,1,0,0,0,0)
```

So for example, the first family in the data reported a boy (1) and then a girl (0). The second family reported a girl (0) and then a boy (1). The third family reported two girls. You can load these two vectors into R's memory by typing:

R code 3.29

```
library(rethinking)
data(homeworkch3)
```

Use these vectors as data. So for example to compute the total number of boys born across all of these births, you could use:

R code 3.30

```
sum(birth1) + sum(birth2)
```

```
[1] 111
```

**3H1.** Using grid approximation, compute the posterior distribution for the probability of a birth being a boy. Assume a uniform prior probability. Which parameter value maximizes the posterior probability?

**3H2.** Using the `sample` function, draw 10,000 random parameter values from the posterior distribution you calculated above. Use these samples to estimate the 50%, 89%, and 97% highest posterior density intervals.

**3H3.** Use `rbinom` to simulate 10,000 replicates of 200 births. You should end up with 10,000 numbers, each one a count of boys out of 200 births. Compare the distribution of predicted numbers of boys to the actual count in the data (111 boys out of 200 births). There are many good ways to visualize the simulations, but the `dens` command (part of the `rethinking` package) is probably the easiest way in this case. Does it look like the model fits the data well? That is, does the distribution of predictions include the actual observation as a central, likely outcome?

**3H4.** Now compare 10,000 counts of boys from 100 simulated first borns only to the number of boys in the first births, `birth1`. How does the model look in this light?

**3H5.** The model assumes that sex of first and second births are independent. To check this assumption, focus now on second births that followed female first borns. Compare 10,000 simulated counts of boys to only those second births that followed girls. To do this correctly, you need to count the number of first borns who were girls and simulate that many births, 10,000 times. Compare the counts of boys in your simulations to the actual observed count of boys following girls. How does the model look in this light? Any guesses what is going on in these data?