A Zero-Math Introduction to Markov Chain Monte Carlo Methods

Ben Shaver Follow
Dec 22, 2017 · 11 min read

For many of us, Bayesian statistics is voodoo magic at best, or completely subjective nonsense at worst. Among the trademarks of the Bayesian approach, Markov chain Monte Carlo methods are especially mysterious. They're math-heavy and computationally expensive procedures for sure, but the basic reasoning behind them, like so much else in data science, can be made intuitive. That is my goal here.

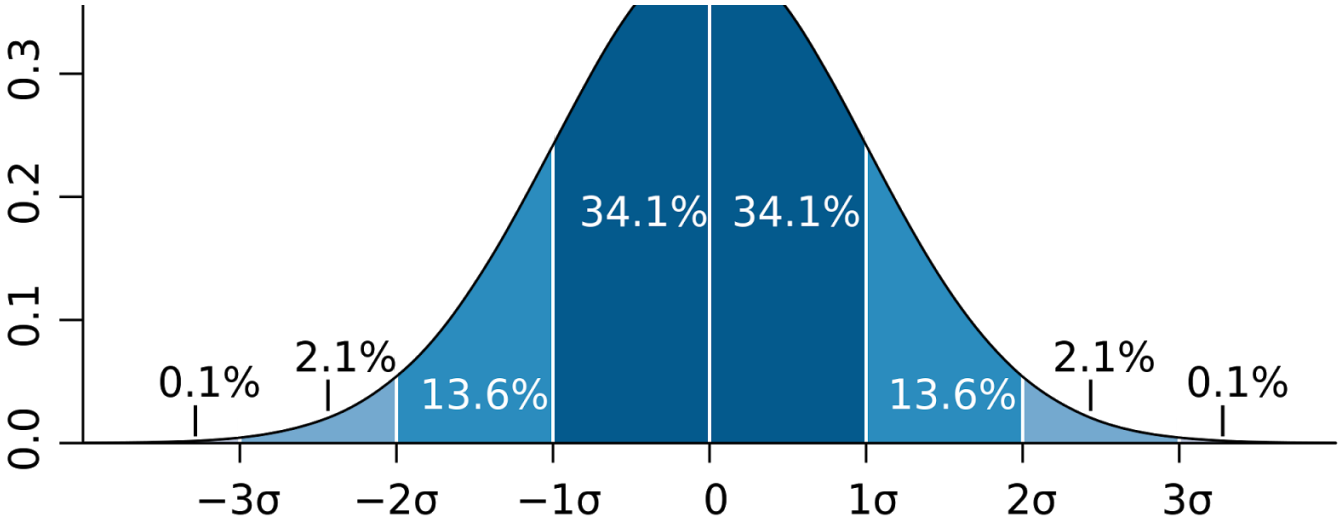
. . .

So, what are Markov chain Monte Carlo (MCMC) methods? The short answer is:

> *MCMC methods are used to approximate the posterior distribution of a parameter of interest by random sampling in a probabilistic space.*

In this article, I will explain that short answer, without any math.

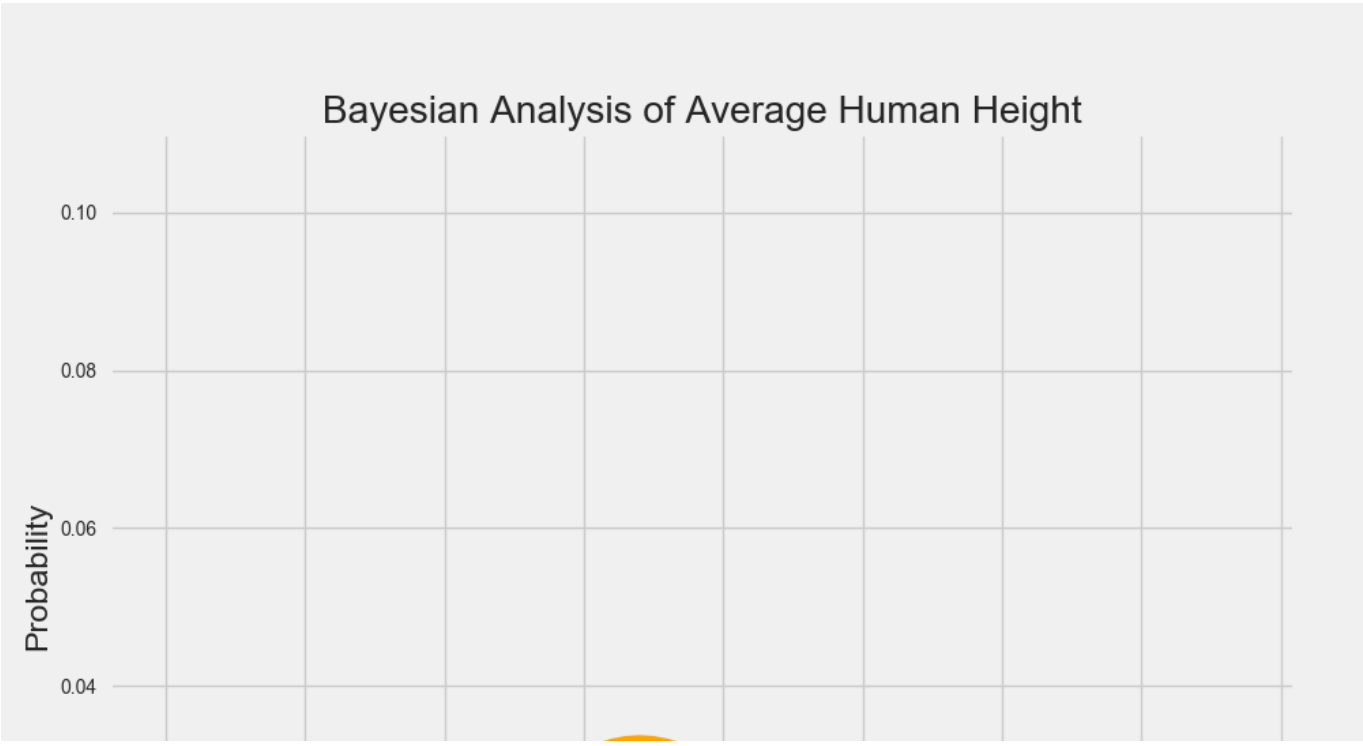
First, some terminology. A ***parameter of interest*** is just some number that summarizes a phenomenon we're interested in. In general we use statistics to estimate parameters. For example, if we want to learn about the height of human adults, our parameter of interest might be average height in in inches. A ***distribution*** is a mathematical representation of every possible value of our parameter and how likely we are to observe each one. The most famous example is a bell curve:

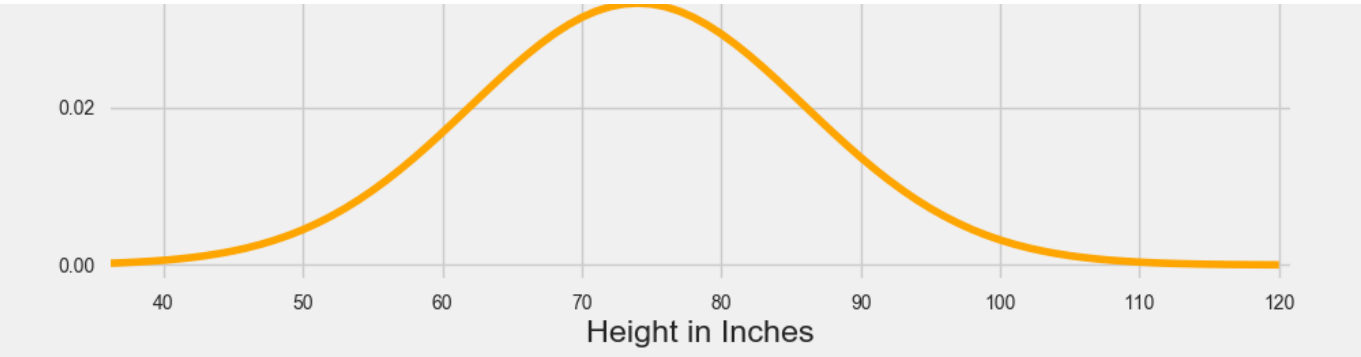


Courtesy M. W. Toews

In the ***Bayesian*** way of doing statistics, distributions have an additional interpretation. Instead of just representing the values of a parameter and how likely each one is to be the true value, a Bayesian thinks of a distribution as describing our *beliefs* about a parameter. Therefore, the bell curve above shows we're pretty sure the value of the parameter is quite near zero, but we think there's an equal likelihood of the true value being above or below that value, up to a point.

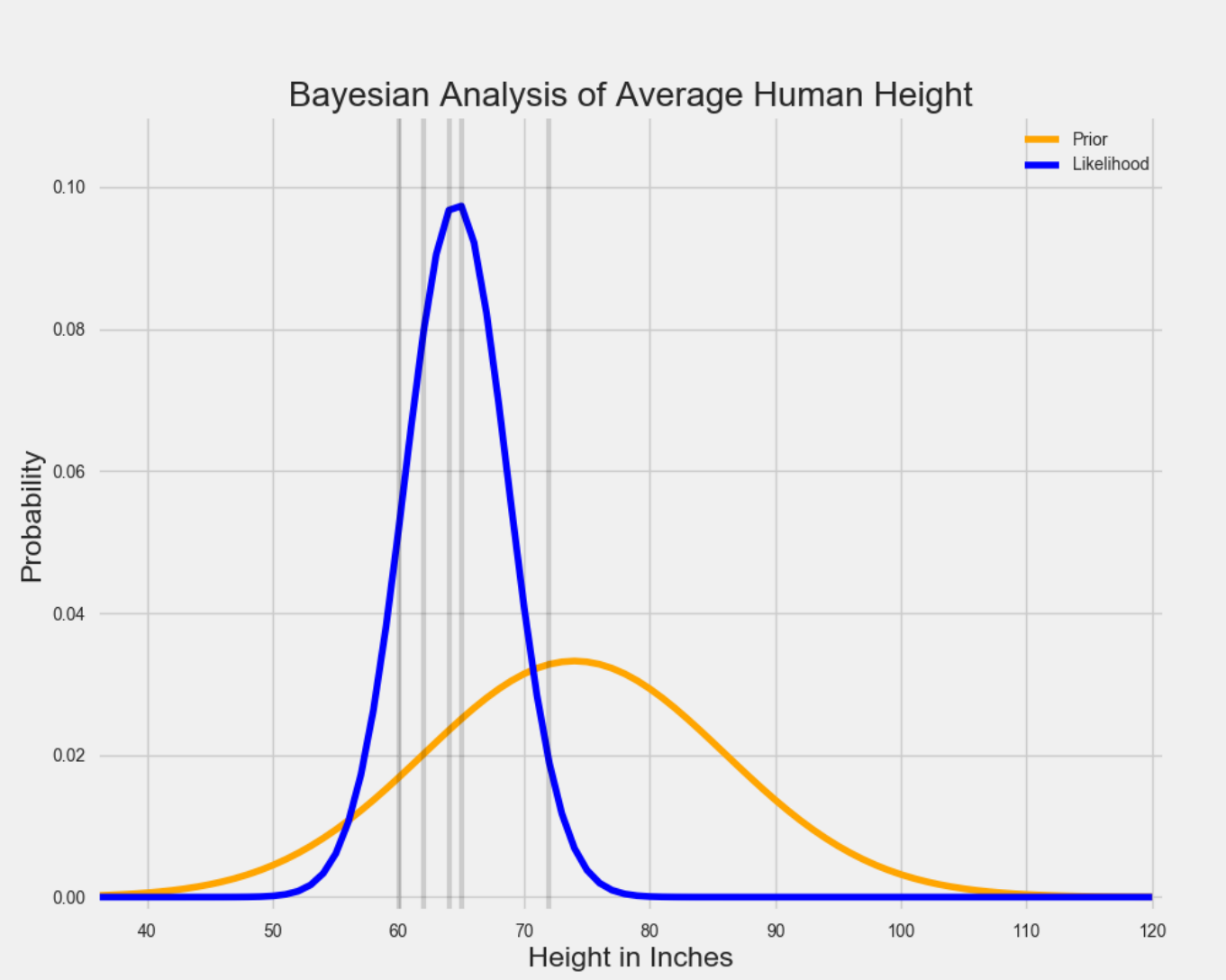
As it happens, human heights do follow a normal curve, so let's say we believe the true value of average human height follows a bell curve like this:





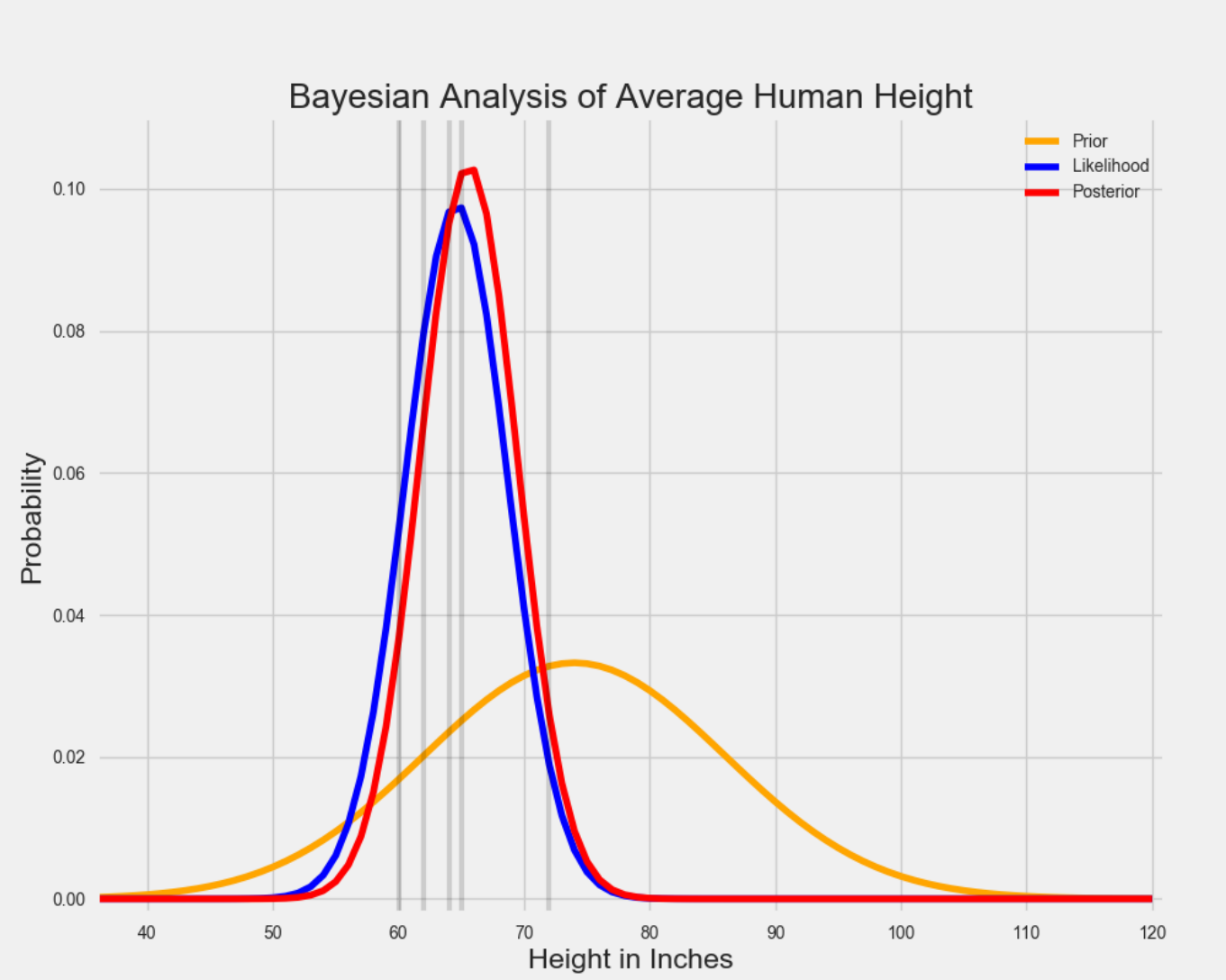
Clearly, the person with beliefs represented by this graph has been living among giants for years, because as far as they know, the most likely average adult height is 6'2" (but they're not super confident one way or another).

Lets imagine this person went and collected some data, and they observed a range of people between 5' and 6'. We can represent that data below, along with another normal curve that shows which values of average human height *best explain the data:*



In Bayesian statistics, the distribution representing our beliefs about a parameter is called the ***prior distribution,*** because it captures our beliefs *prior* to seeing any data. The ***likelihood distribution*** summarizes what the observed data are telling us, by representing a range of parameter values accompanied by the likelihood that each each parameter explains the data we are observing. Estimating the parameter value that maximizes the likelihood distribution is just answering the question: what parameter value would make it most likely to observe the data we have observed? In the absence of prior beliefs, we might stop there.

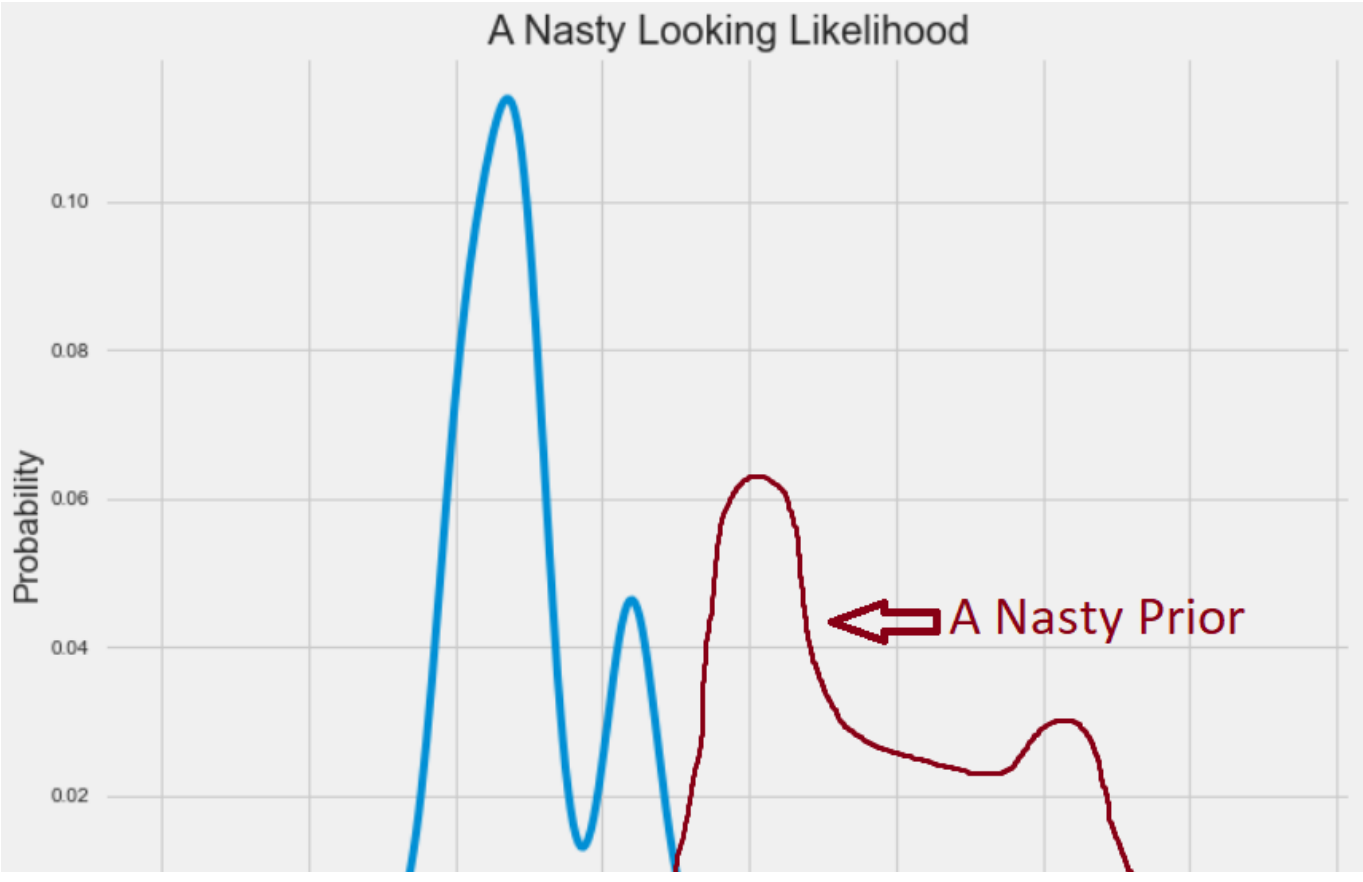
The key to Bayesian analysis, however, is to combine the prior and the likelihood distributions to determine the ***posterior distribution***. This tells us which parameter values maximize the chance of observing the particular data that we did, taking into account our prior beliefs. In our case, the posterior distribution looks like this:

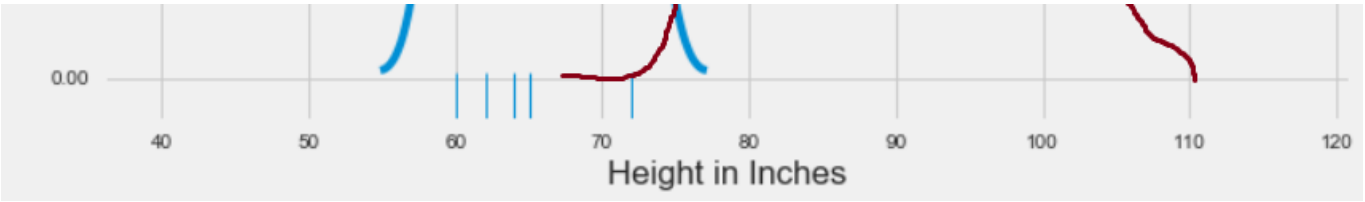


Above, the red line represents the posterior distribution. You can think of it as a kind of average of the prior and the likelihood distributions. Since the prior distribution is shorter and more spread out, it represents a set of belief that is 'less sure' about the true value of average human height. Meanwhile, the likelihood summarizes the data within a relatively narrow range, so it represents a 'more sure' guess about the true parameter value.

When the prior the likelihood are combined, the data (represented by the likelihood) dominate the weak prior beliefs of the hypothetical individual who had grown up among giants. Although that individual still believes the average human height is slightly higher than just what the data is telling him, he is mostly convinced by the data.

In the case of two bell curves, solving for the posterior distribution is very easy. There is a simple equation for combining the two. But what if our prior and likelihood distributions weren't so well-behaved? Sometimes it is most accurate to model our data or our prior beliefs using distributions which don't have convenient shapes. What if our likelihood were best represented by a distribution with two peaks, and for some reason we wanted to account for some really wacky prior distribution? I've visualized that scenario below, by hand drawing an ugly prior distribution:





Visualizations rendered in Matplotlib, enhanced using MS Paint

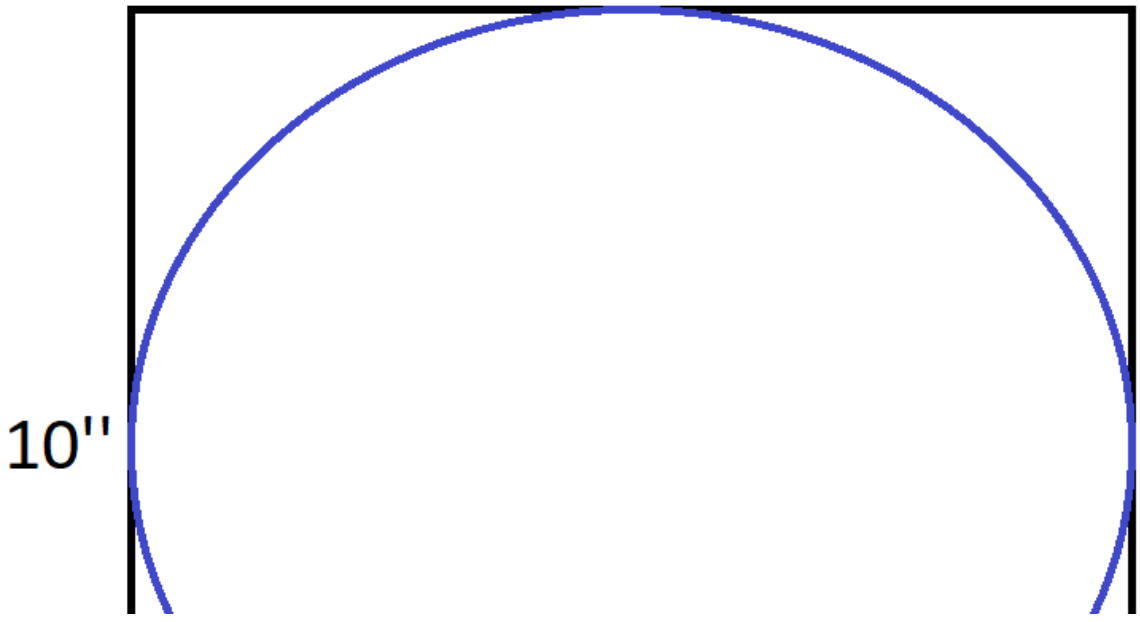
As before, there exists *some* posterior distribution that gives the likelihood for each parameter value. But its a little hard to see what it might look like, and it is impossible to solve for analytically. Enter MCMC methods.

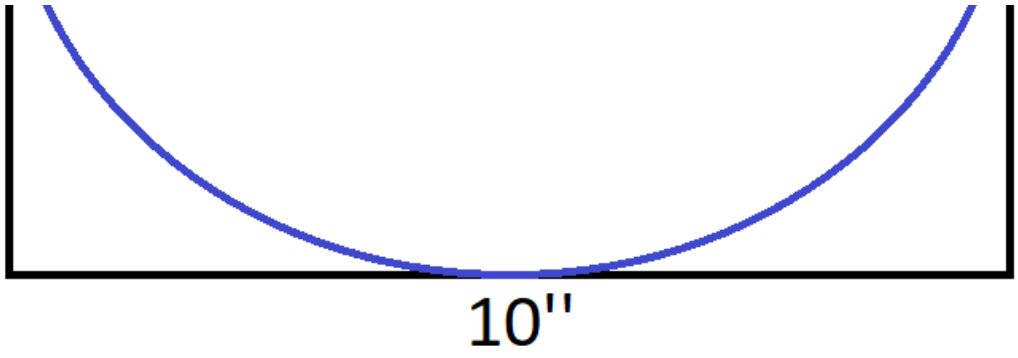
MCMC methods allow us to estimate the shape of a posterior distribution in case we can't compute it directly. Recall that MCMC stands for Markov chain Monte Carlo methods. To understand how they work, I'm going to introduce Monte Carlo simulations first, then discuss Markov chains.

. . .

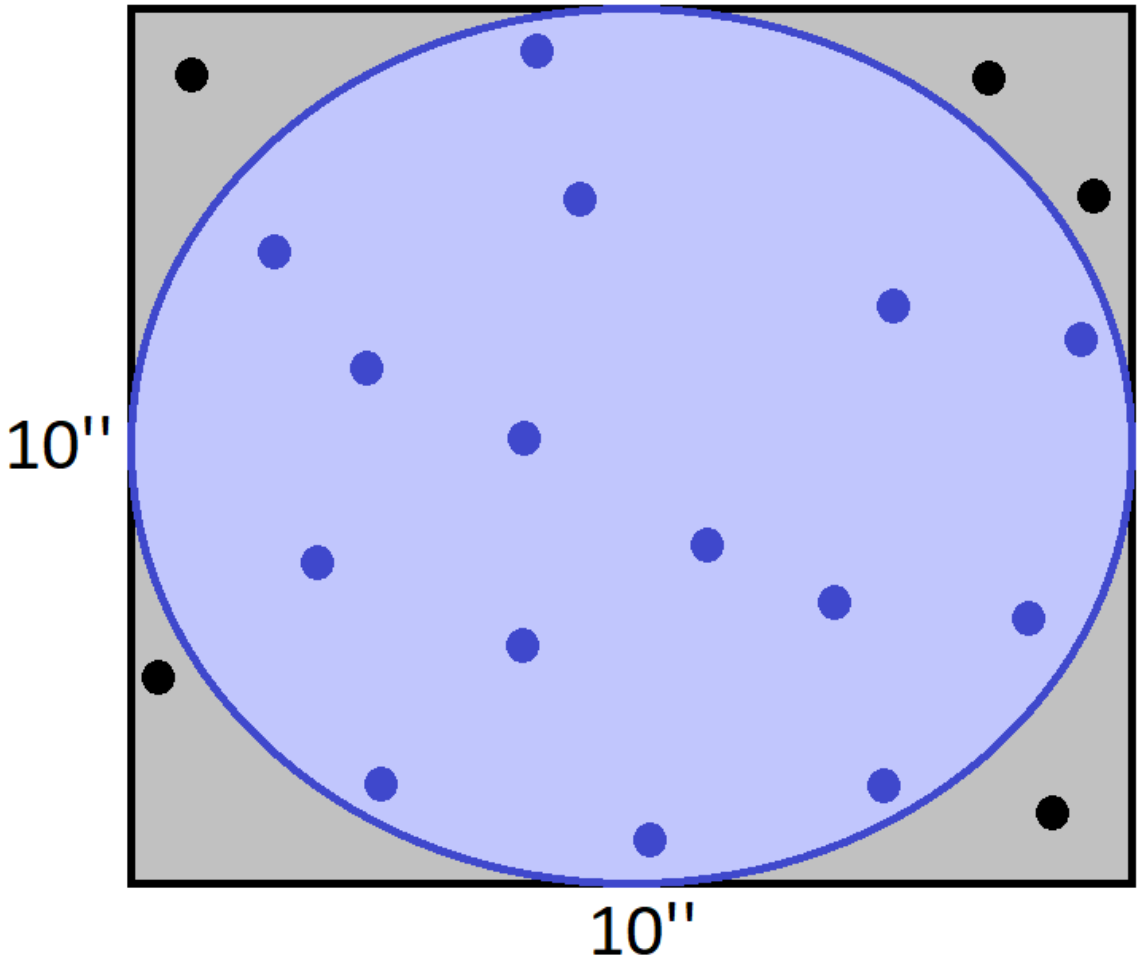
Monte Carlo simulations are just a way of estimating a fixed parameter by repeatedly generating random numbers. By taking the random numbers generated and doing some computation on them, Monte Carlo simulations provide an approximation of a parameter where calculating it directly is impossible or prohibitively expensive.

Suppose that we'd like to estimate the area of the follow circle:



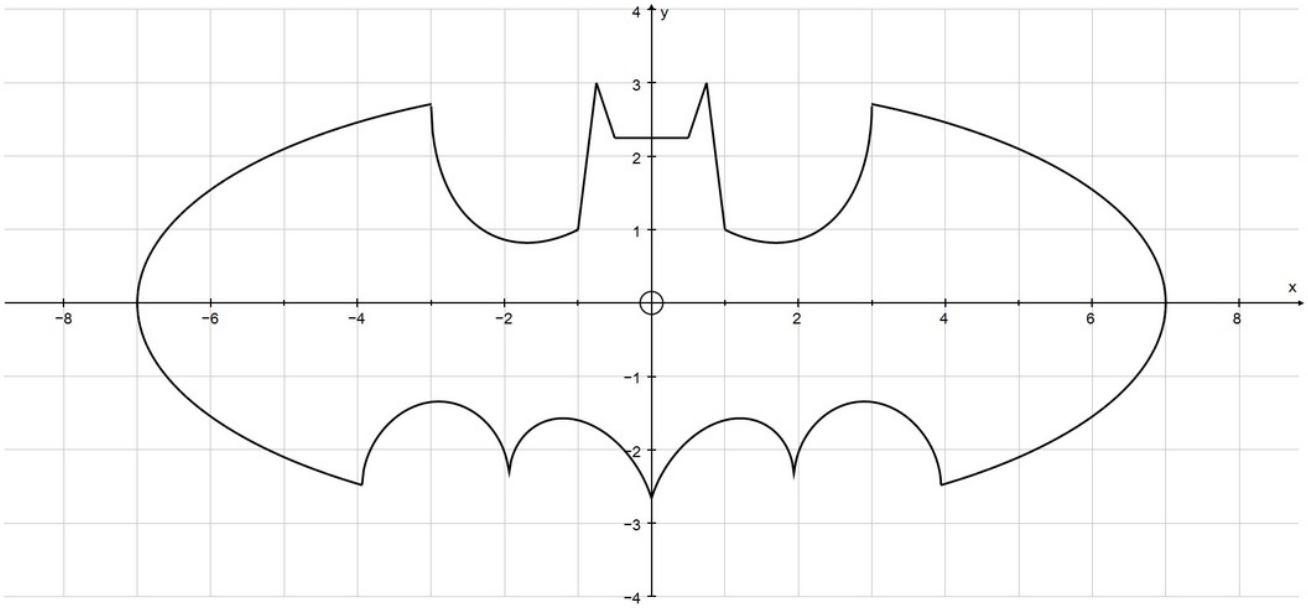


Since the circle is inside a square with 10 inch sides, the area can be easily calculated as 78.5 square inches. Instead, however, we can drop 20 points randomly inside the square. Then we count the proportion of points that fell within the circle, and multiply that by the area of the square. That number is a pretty good approximation of the area of the circle.



Since 15 of the 20 points lay inside the circle, it looks like the circle is approximately 75 square inches. Not too bad for a Monte Carlo simulation with only 20 random points.

Now, imagine we'd like to calculate the area of the shape plotted by the Batman Equation:



Here's a shape we never learned an equation for! Therefore, finding the area of the bat signal is very hard. Nevertheless, by dropping points randomly inside a rectangle containing the shape, Monte Carlo simulations can provide an approximation of the area quite easily!

Monte Carlo simulations aren't only used for estimating the area of difficult shapes. By generating a lot of random numbers, they can be used to model very complicated processes. In practice, they're used to forecast the weather, or estimate the probability of winning an election.

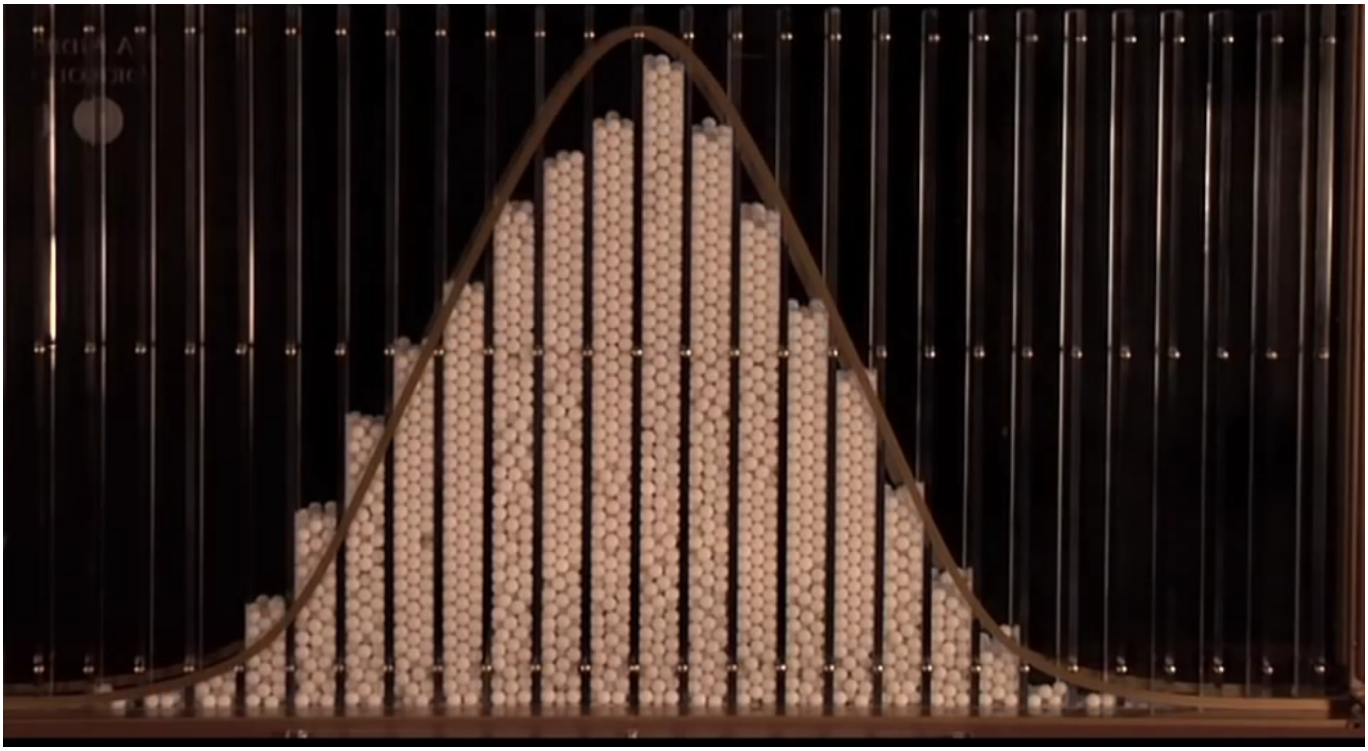
. . .

The second element to understanding MCMC methods are Markov chains. These are simply sequences of events that are probabilistically related to one another. Each event comes from a set of outcomes, and each outcome determines which outcome occurs next, according to a fixed set of probabilities.

An important feature of Markov chains is that they are ***memoryless***: everything that you would possibly need to predict the next event is available in the current state, and no new information comes from knowing the history of events. A game like Chutes and

Ladders exhibits this memorylessness, or ***Markov Property,*** but few things in the real world actually work this way. Nevertheless, Markov chains are powerful ways of understanding the world.

In the 19th century, the bell curve was observed as a common pattern in nature. (We've noted, for example, that human heights follow a bell curve.) Galton Boards, which simulate the average values of repeated random events by dropping marbles through a board fitted with pegs, reproduce the normal curve in their distribution of marbles:



Pavel Nekrasov, a Russian mathematician and theologian, argued that the bell curve and, more generally, the law of large numbers, were simply artifacts of children's games and trivial puzzles, where every event was completely independent. He thought that interdependent events in the real world, such as human actions, did not conform to nice mathematical patterns or distributions.

Andrey Markov, for whom Markov chains are named, sought to prove that non-independent events may also conform to patterns. One of his best known examples required counting thousands of two-character pairs from a work of Russian poetry. Using those pairs, he computed the conditional probability of each character. That is, given a certain preceding letter or white space, there was a certain chance that the next letter would be an A, or a T, or a whitespace. Using those probabilities, Markov was ability to

simulate an arbitrarily long sequence of characters. This was a Markov chain. Although the first few characters are largely determined by the choice of starting character, Markov showed that in the long run, the distribution of characters settled into a pattern. Thus, even interdependent events, if they are subject to fixed probabilities, conform to an average.

For a more useful example, imagine you live in a house with five rooms. You have a bedroom, bathroom, living room, dining room, and kitchen. Lets collect some data, assuming that what room you are in at any given point in time is all we need to say what room you are likely to enter next. For instance, if you are in the kitchen, you have a 30% chance to stay in the kitchen, a 30% chance to go into the dining room, a 20% chance to go into the living room, a 10% chance to go into the bathroom, and a 10% chance to go into the bedroom. Using a set of probabilities for each room, we can construct a chain of predictions of which rooms you are likely to occupy next.

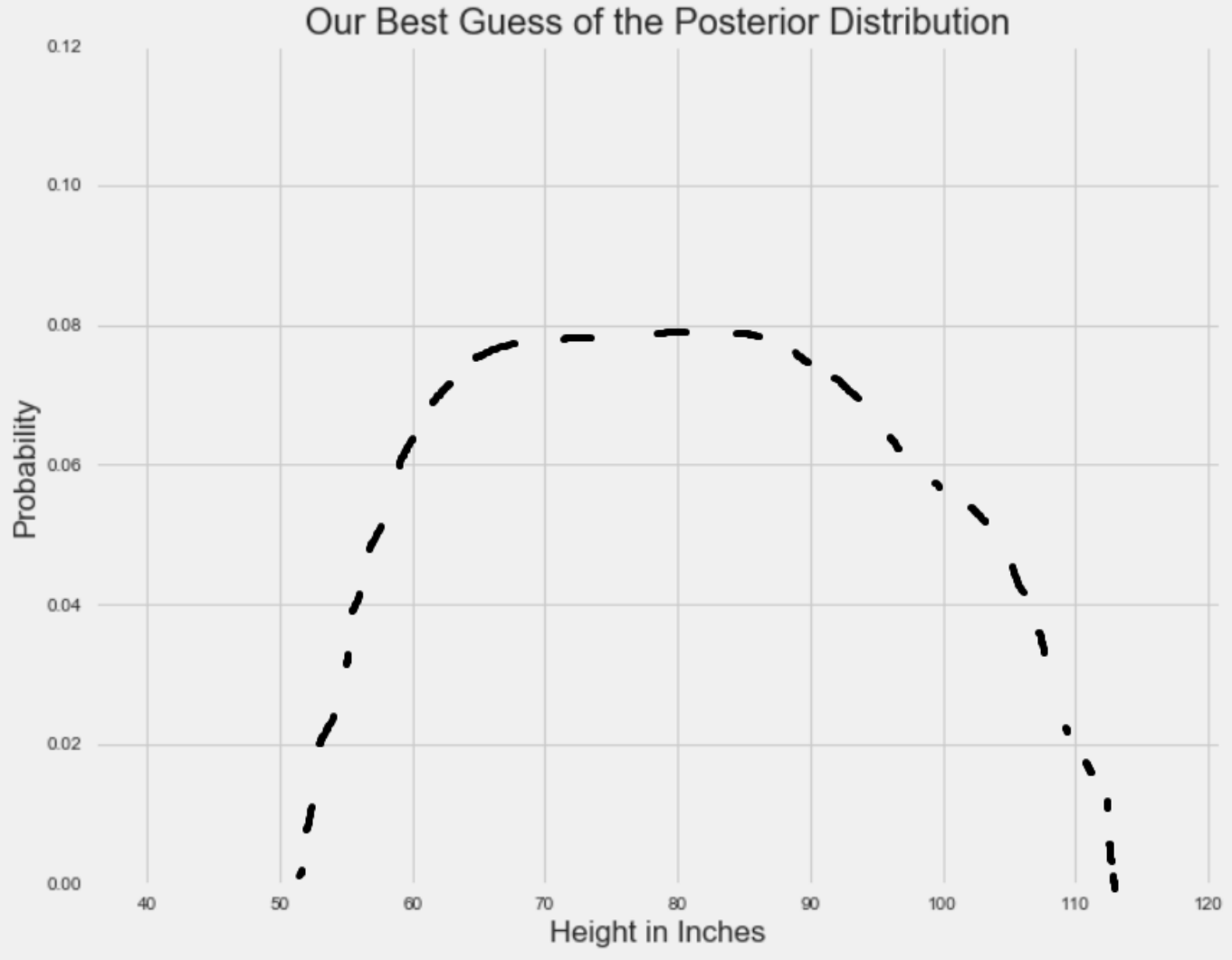
Making predictions a few states out might be useful, if we want to predict where someone in the house will be a little while after being in the kitchen. But since our predictions are just based on one observation of where a person is in the house, its reasonable to think they won't be very good. If someone went from the bedroom to the bathroom, for example, its more likely they'll go right back to the bedroom than if they had come from the kitchen. So the Markov Property doesn't usually apply to the real world.

Running the Markov chain for thousands of iterations, however, does give the long-run prediction of what room you're likely to be in. More importantly, this prediction isn't affected at all by which room the person began in! Intuitively, this makes sense: it doesn't matter where someone is in the house at one point in time in order to simulate and describe where they are likely to be in the long-term, or *in general*. So Markov chains, which seem like an unreasonable way to model a random variable over a few periods, can be used to compute the long-run tendency of that variable if we understand the probabilities that govern its behavior.

• • •

With some knowledge of Monte Carlo simulations and Markov chains, I hope the math-free explanation of how MCMC methods work is pretty intuitive.

Recall that we are trying to estimate the posterior distribution for the parameter we're interested in, average human height:



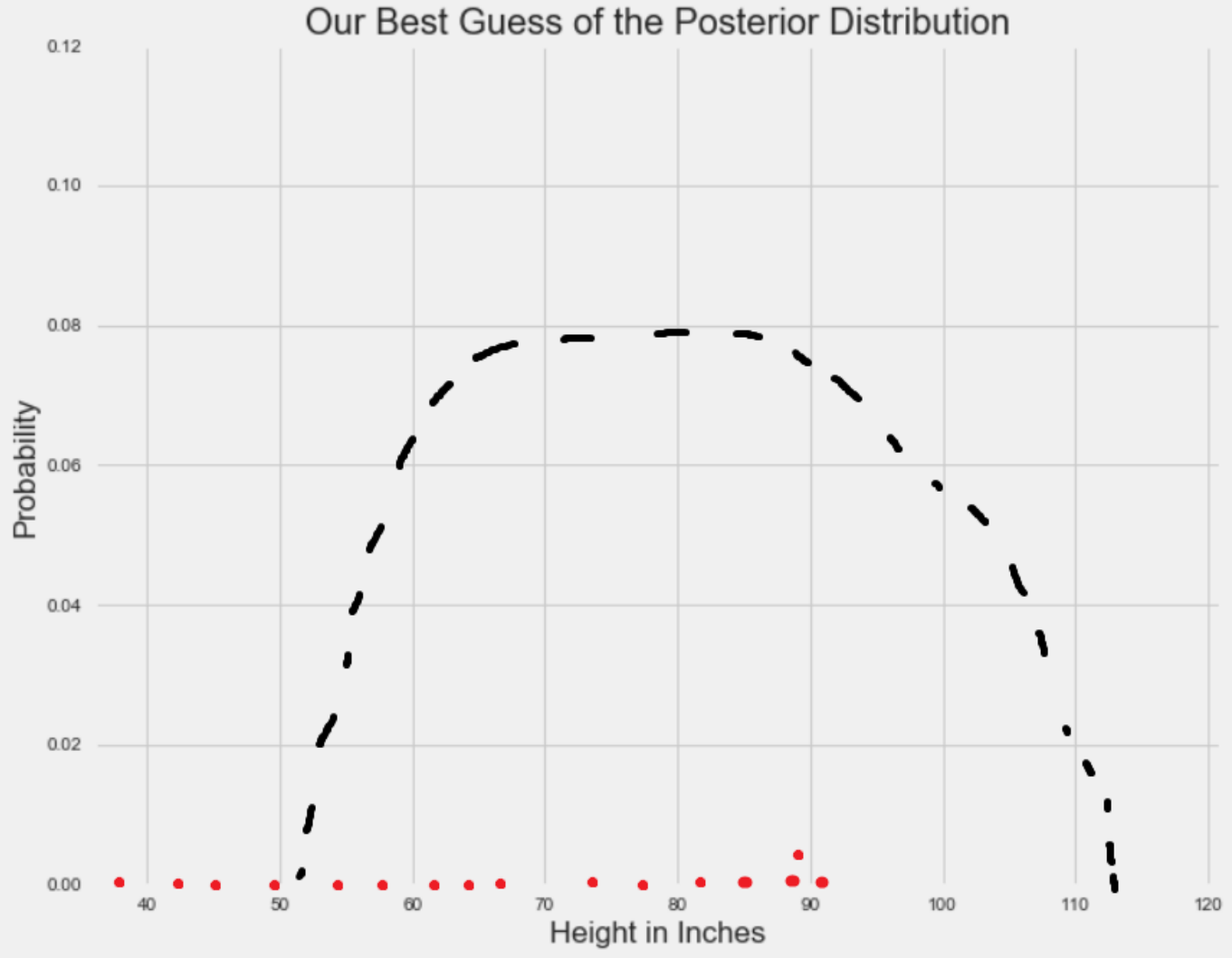
I am not a visualization expert, nor apparently am I any good at keeping my example within the bounds of common sense: my example of the posterior distribution seriously overestimates average human height.

We know that the posterior distribution is somewhere in the range of our prior distribution and our likelihood distribution, but for whatever reason, we can't compute it directly. Using MCMC methods, we'll effectively *draw samples from the posterior* distribution, and then compute statistics like the average on the samples drawn.

To begin, MCMC methods pick a random parameter value to consider. The simulation will continue to generate random values (this is the Monte Carlo part), but subject to

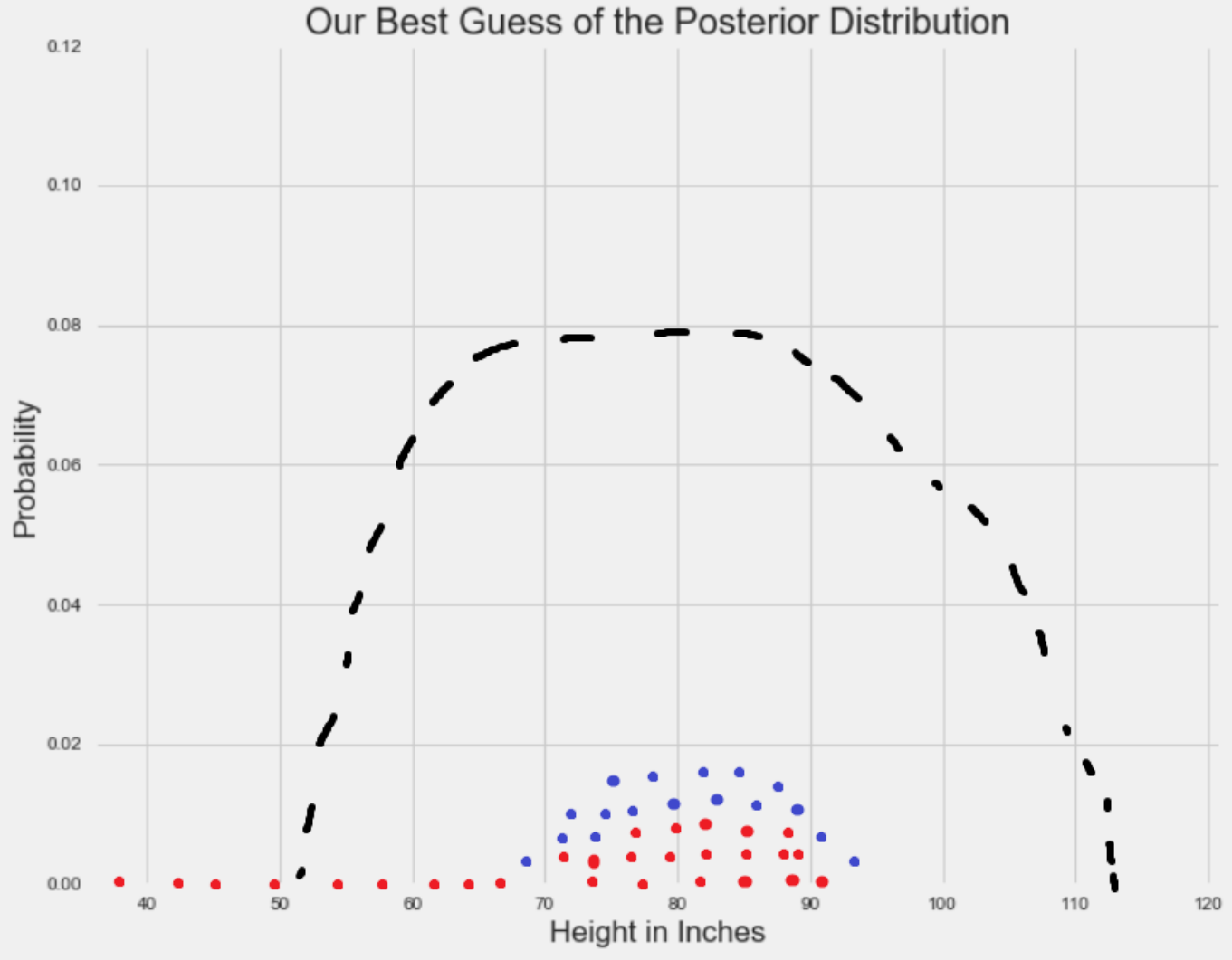
some rule for determining what makes a good parameter value. The trick is that, *for a pair of parameter values,* it is possible to compute which is a better parameter value, by computing how likely each value is to explain the data, given our prior beliefs. If a randomly generated parameter value is better than the last one, it is added to the chain of parameter values with a certain probability determined by *how much* better it is (this is the Markov chain part).

To explain this visually, lets recall that the height of a distribution at a certain value represents the probability of observing that value. Therefore, we can think of our parameter values (the x-axis) exhibiting areas of high and low probability, shown on the y-axis. For a single parameter, MCMC methods begin by randomly sampling along the x-axis:



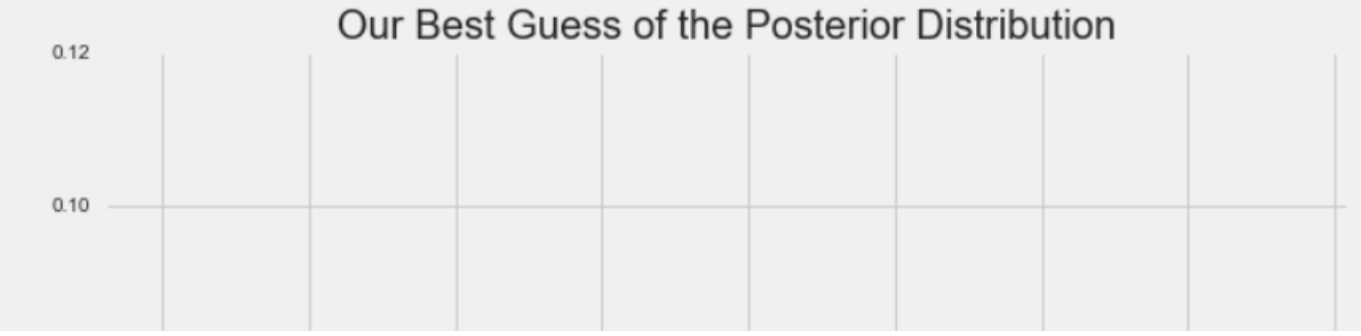
Red points are random parameter samples

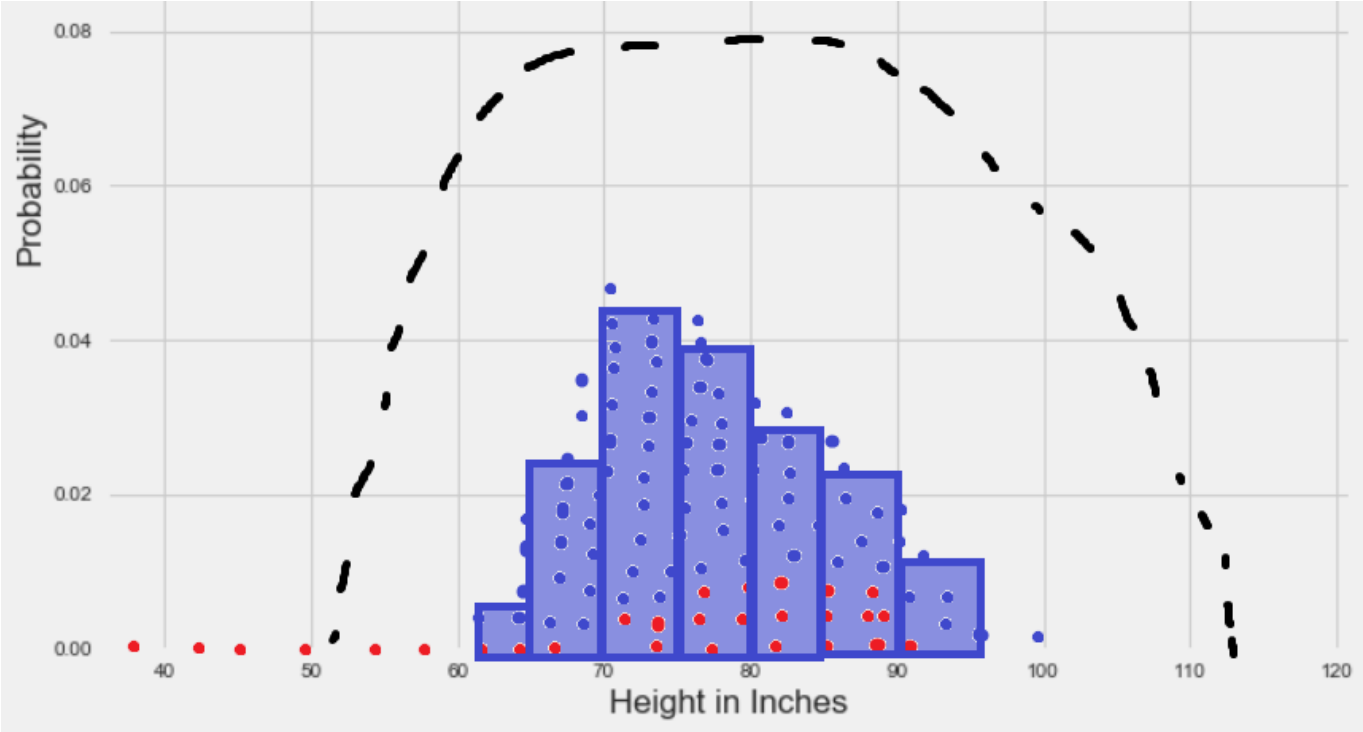
Since the random samples are subject to fixed probabilities, they tend to converge after a period of time in the region of highest probability for the parameter we're interested in:



Blue points just represent random samples after an arbitrary point in time, when convergence is expected to have occurred. Note: I'm stacking point vertically purely for illustrative purposes.

After convergence has occurred, MCMC sampling yields a set of points which are samples from the posterior distribution. Draw a histogram around those points, and compute whatever statistics you like:





Any statistic calculated on the set of samples generated by MCMC simulations is our best guess of that statistic on the true posterior distribution.

MCMC methods can also be used to estimate the posterior distribution of more than one parameter (human height *and* weight, say). For *n* parameters, there exist regions of high probability in n-dimensional space where certain sets of parameter values better explain observed data. Therefore, I think of MCMC methods as randomly sampling inside a *probabilistic* space to approximate the posterior distribution.

. . .

Recall the short answer to the question 'what are Markov chain Monte Carlo methods?' Here it is again as a TL;DR:

> *MCMC methods are used to approximate the posterior distribution of a parameter of interest by random sampling in a probabilistic space.*

I hope I've explained that short answer, why you would use MCMC methods, and how they work. The inspiration for this post was a talk I gave as part of General Assembly's Data Science Immersive course in Washington, DC. The goals of that talk were to explain Markov chain Monte Carlo methods to a non-technical audience, and I've tried to do the

same here. Leave a comment if you think this explanation is off the mark in some way, or if it could be made more intuitive.

Thanks to Matt Brems.

Data Science Bayesian Statistics Markov Chains Monte Carlo Towards Data Science