

Bayesian Analysis

Justin Chin

Spring 2018

Abstract

We often think of the field of Statistics simply as data collection and analysis. While the essence of Statistics lies in numeric analysis of observed frequencies and proportions, recent technological advancements facilitate complex computational algorithms. These advancements combined with a way to update prior models and take a person's belief or beliefs into consideration have encouraged a surge in popularity of Bayesian Analysis. In this paper we will explore the Bayesian style of analysis and compare it to the more commonly known frequentist style.

Contents

1	Introduction	2
2	Terminology and Notation	2
3	Frequentist vs Bayesian Statistics	3
3.1	Frequentist Statistics	3
3.2	Bayesian Statistics	3
3.2.1	Derivation of Bayes' Rule	4
4	Derive Posterior for Binary Data	5
4.1	Bernoulli Likelihood	5
4.2	Beta Prior	5
4.3	Beta Posterior	5
4.4	Posterior is a Combination of Prior and Likelihood	6
5	Examples with Graphics	6
5.1	Bayesian Confidence Intervals	9
6	Markov Chain Monte Carlo	10
6.1	General Example of the Metropolis Algorithm with a Politician	11
6.2	A Random Walk	11
6.3	General Properties of a Random Walk	12
7	More About MCMC	12
7.1	The Math Behind MCMC	12
8	Generalize Metropolis Algorithm to Bernoulli Likelihood and Beta Prior	13
8.1	Metropolis Algorithm to Coin Flips	13

9	MCMC Examples	14
9.1	MCMC Using a Beta Distribution	14
9.2	MCMC Using a Normal Distribution	19
10	Conclusion	25

1 Introduction

The goal of statistics is to make informed, data supported decisions in the face of uncertainty. The basis of frequentist statistics is to gather data to test a hypothesis and/or construct confidence intervals in order to draw conclusions. The frequentist approach is probably the most common type of statistical inference, and when someone hears the term “statistics” they often default to this approach. Another type of statistical analysis that is gaining popularity is Bayesian Analysis. The big difference here is that we begin with a prior belief or prior knowledge of a characteristic of a population and we update that knowledge with incoming data.

As a quick example, consider we are investigating a coin to see how often it flips heads, and how often it flips tails. The frequentist approach would be to first gather data, then use this data to estimate the probability of observing a head. The Bayesian approach uses our prior belief about the fairness of the coin and the data to estimate the probability of observing a head. Our prior belief is that coins are fair, and therefore have 50/50 chance of heads and tails. But suppose we gathered data on the coin and observed 80 heads in 100 flips, we would use that new information to update our prior belief to one that the probability of observing a heads might fall somewhere between 50 and 80%, while the frequentist would estimate the probability to be 80%.

In this paper I will explore the Bayesian approach using both theoretical results and simulations.

2 Terminology and Notation

A **random variable**, Y , is a function that maps elements of a sample space to a real number. When Y is countable, we call it a discrete random variable. When Y is uncountable, we call it a continuous random variable. We use probability mass functions to determine probabilities associated with discrete random variables, and probability density function to determine probabilities associated with continuous random variables.

A **parameter**, θ or $\vec{\theta}$, is a numeric quantity associated with each random variable that is needed to calculate a probability.

We will use the notation $p(y|\theta)$ to denote either a probability mass function or probability density function, depending on the characterization of our random variable.

A **probability distribution** is a function that provides probabilities of different possible outcomes of an event.

We say denote the sentence “ X follows some distribution Y with parameters a_1, a_2, a_3, \dots ” as $X \sim Y(a_1, a_2, a_3, \dots)$. For example, if Y follows a beta distribution, then Y takes on real number values between 0 and 1, has parameters a and b , and has a probability density function

of $p(y|\vec{\theta} = (a, b))$. We simplify this by writing: $Y \sim \text{beta}(a, b)$.

3 Frequentist vs Bayesian Statistics

In statistics, parameters are unknown numeric characterizations of a population. A goal is to use data to estimate a parameter of interest.

In Bayesian statistics, **likelihood** is an indication of how much data contributes to the probability of the parameter. The likelihood of a parameter given data is the probability of the data given the parameter. Also, the likelihood is a function of the parameter(S) given the data, it is not a probability mass function or a probability density function. Likelihood is denoted with $L(\theta|x)$.

3.1 Frequentist Statistics

In the world of frequentist statistics, the parameter is fixed but still unknown. As the name suggests, this form of statistics stresses frequency. The general idea is to draw conclusions from repeated samples to understand how frequency and proportion estimates of the parameter of interest behave.

3.2 Bayesian Statistics

Bayesian statistics is motivated by Bayes' Theorem, named after Thomas Bayes (1701-1761). Bayesian statistics takes prior knowledge about the parameter and uses newly collected data or information to update our prior beliefs. Furthermore, parameters are treated as unknown random variables that have density of mass functions.

The process starts with a 'prior distribution' that reflects previous knowledge or beliefs. Then, similar to frequentist, we gather data to create a 'likelihood' model. Lastly, we combine the two using Bayes' Rule to achieve a 'posterior distribution'. If new data is gathered, we can then use our posterior distribution as a new prior distribution in a new model; combined with new data, we can create a new posterior distribution.

In short, the idea is to take a prior model, and then after data collection combine to a posterior model. Let us walk through a simple example where we try to estimate bias in a coin. We will break this process into four steps:

The first step is to identify the data. The sample space is heads or tails. The random variable, Y , the number of heads observed in a single coin toss is $\{0, 1\}$, where $y = 0$ indicates observing a 'tails' and $y = 1$ indicates observing a 'heads'. We note that Y is a random variable and y denotes an observed outcome of Y . The parameter, θ , is the probability of observing a head. Because Y is discrete in this case, it has the probability mass function:

$$p(y|\theta) = \begin{cases} \theta & \text{if } y = 1 \\ 1 - \theta & \text{if } y = 0 \end{cases}.$$

The second step is to create a descriptive model with meaningful parameters. Let $p(\text{heads})=p(y = 1)$. We will describe the underlying probability of heads, $p(\text{heads})$, directly as the value of the parameter θ . Or $p(y = 1|\theta) = \theta, 0 \leq \theta \leq 1$. $\theta = p(\text{heads})$
 It follows that $p(\text{tails})= p(y = 0|\theta) = 1 - \theta$. We can use the Bernoulli Distribution to get the likelihood of the data given the parameters. The likelihood is $L(\theta|y) = \theta^y(1 - \theta)^{(1-y)}$, where y is fixed and θ varies between 0 and 1.

The third step is to create a prior distribution. Suppose θ takes on values 0, 0.1, 0.2, 0.3, ..., 1 (an intuitive way to think about this is that a company makes 11 types of coins; note that we are simplifying θ to a discrete parameter). If we use a model where we assume $\theta = 0.5$ is the mean, it will look like a triangle with peak at 0.5 and goes down and out. Our graph here uses θ as the horizontal axis and $p(\theta)$ as the vertical axis. In fact, $p(\theta) = \frac{i}{10}$ where i takes on integers from 0 to 10.

The fourth and final step is to collect data and apply Bayes' Rule to reallocate credibility. First note that we define credibility as inference that uses newly observed past events to try to accurately predict uncertain future events. Suppose we flip heads once and use that as our data, D . If z is the number of heads and N is number of flips, $z = N = 1$. This gives us a graph with right triangle side at $\theta = 1$, and goes down to the left. Applying Bayes' rule, we combine to get a posterior distribution that is in between. It is important to note that even though our data had 100% heads does not mean our posterior model reflects only $\theta = 1$.

3.2.1 Derivation of Bayes' Rule

For two events r and c , we start with a fairly well known conditional probability $p(c|r) = p(r,c)/p(r)$ where $p(r,c)$ is the probability that r and c happen together, $p(r)$ is the probability of observing event r , and $p(c|r)$, read as "probability of 'c' given 'r'" is the probability of the event c occurring, with the knowledge that the event r happened. A little algebra leads to:

$$p(c|r) = \frac{p(r|c)p(c)}{p(r)} = \frac{p(r|c)p(c)}{\sum_{c^*} p(r|c^*)p(c^*)}$$

in the discrete case, or

$$\frac{p(r|c)p(c)}{\int_{c^*} p(r|c^*)p(c^*)}$$

in the continuous case. We note that c is fixed, but c^* takes on all possible values.

In Bayesian statistics, we often use θ for c and D for r . We use $p(c|r)$ as the posterior, the numerator as the likelihood times the prior, and the denominator as the evidence or marginal likelihood.

Bayes' Rule is the core of Bayesian Analysis, where θ is the unknown parameter, and D is the data. We use $p(\theta)$ as the prior distribution of θ , and $L(D|\theta)$ as the likelihood of the recorded data. The marginal likelihood is $p(D)$, and $p(\theta|D)$ is the posterior distribution. Altogether, rewriting Bayes' Rule in these terms yields the following:

$$p(\theta|D) = \frac{p(D|\theta)p(\theta)}{p(D)}$$

4 Derive Posterior for Binary Data

In this section I will explore how to analyze binary data using Bayesian techniques.

4.1 Bernoulli Likelihood

The Bernoulli distribution is used with binary data; for example, a coin flip is either heads or tails, a person's answer to a question might be yes or no, etc. Then our likelihood function of θ returns as:

$$L(y|\theta) = p(y|\theta) = \theta^y(1-\theta)^{(1-y)},$$

where $y = 1$ is denotes a head was observed and θ is the probability of observing a head.

If we have more than 1 flip of a coin, let $\vec{y} = \{y_i | i = 1, 2, \dots, n\}$ be the set of outcomes, where y_i is the outcome of the i th flip. Also, $z =$ the number of heads, $N =$ the number of flips, it follows that $L(\theta|\vec{y}) = \theta^z(1-\theta)^{(N-z)}$; this is useful for applying Bayes' rule to large data sets.

4.2 Beta Prior

Next we need to set up a model that represents our prior beliefs. If we are interested in estimating the probability of an outcome, which is often the case for binary data, we need a model whose values range from 0 to 1. We will use the Beta distribution. The beta prior distribution has parameters a and b with density:

$$\begin{aligned} p(\theta|a, b) &= \text{beta}(\theta|a, b) \\ &= \frac{\theta^{(a-1)}(1-\theta)^{(b-1)}}{B(a, b)} \end{aligned}$$

We use $B(a, b)$ to be a normalizing constant to ensure an area of 1.

$$B(a, b) = \int_0^1 \theta^{(a-1)}(1-\theta)^{(b-1)} d\theta.$$

Because θ is only defined on $[0, 1]$, and $a, b > 0$, we then can write the Beta Prior as:

$$p(\theta|a, b) = \frac{\theta^{(a-1)}(1-\theta)^{(b-1)}}{\int_0^1 \theta^{(a-1)}(1-\theta)^{(b-1)} d\theta}.$$

Also note that $\text{beta}(\theta|a, b)$ is referring to the beta distribution, but $B(a, b)$ is the beta function. The beta function is not in terms of θ because it has been integrated out.

4.3 Beta Posterior

Now, combining our prior belief with our data, we get a posterior distribution.

Suppose our data has N trials (we flip a coin N times) and have z successes (heads). We will substitute the Bernoulli likelihood and the Beta Prior into Bayes' rule to get the Posterior distribution.

$$\begin{aligned}
p(\theta|z, N) &= \frac{p(z, N|\theta)p(\theta)}{p(z, N)} && \text{Bayes' Rule} \\
&= \theta^z(1-\theta)^{(N-z)} \frac{\theta^{(a-1)}(1-\theta)^{(b-1)}}{B(a, b)} / p(z, N) && \text{Bernoulli and beta dist} \\
&= \theta^z(1-\theta)^{(N-z)} \theta^{(a-1)}(1-\theta)^{(b-1)} / [B(a, b)p(z, N)] \\
&= \theta^{((z+a)-1)}(1-\theta)^{((N-z+b)-1)} / [B(a, b)p(z, N)] \\
&= \theta^{((z+a)-1)}(1-\theta)^{((N-z+b)-1)} / B(z+a, N-z+b)
\end{aligned}$$

Where $p(\theta|z, N) \sim \text{beta}(z+a, N-z+b)$.

So if we use a beta distribution to model our prior belief for binary data, then the model for updated belief (posterior distribution) is also a beta distribution.

4.4 Posterior is a Combination of Prior and Likelihood

For this example, we see that the prior mean of θ is $\frac{a}{a+b}$, and the posterior mean is

$$\frac{z+a}{(z+a)+(N-z+b)} = \frac{z+a}{N+a+b}$$

Posterior Mean = (Data Mean x Weight1) + (Prior Mean x Weight2)

$$\frac{z+a}{N+a+b} = \left(\frac{z}{n} \cdot \frac{N}{N+a+b} \right) + \left(\frac{a}{a+b} \cdot \frac{a+b}{N+a+b} \right)$$

5 Examples with Graphics

Now let us explore some specific examples.

Example 1: Prior knowledge as a beta distribution

Suppose we have a regular coin, and out of 20 flips we observe 17 heads (85%). Our prior belief is a 0.5 chance of success, but the data is 0.85. Because we have a strong believe that the coin is fair, we might choose $a = b = 250$. Choosing a and b is where we have a bit of freedom. We chose these larger values of a and b because the resulting prior distribution has a mean of 0.5 which reflects our mean of 0.5. This is further emphasized by what we learned in Section 4.4: that the posterior mean is a function of a and b . Large a and b will impact the posterior more than the data.

Next we see that the posterior distribution is pushed a little towards the right, but not too much because of how large a and b are and a small sample size since the posterior mean is $z + a/N + a + b$. In other words, although our sample alone suggests the coin is very favored towards heads, we have such a strong prior belief in the coin's fairness that the data only slightly influences the posterior model. The mode will lie just over 50%. Notice how the mean of the distribution is 0.5, as can be seen in the graph. Looking at the same graph, we see that the likelihood is skewed slightly left with a mean around 0.85.

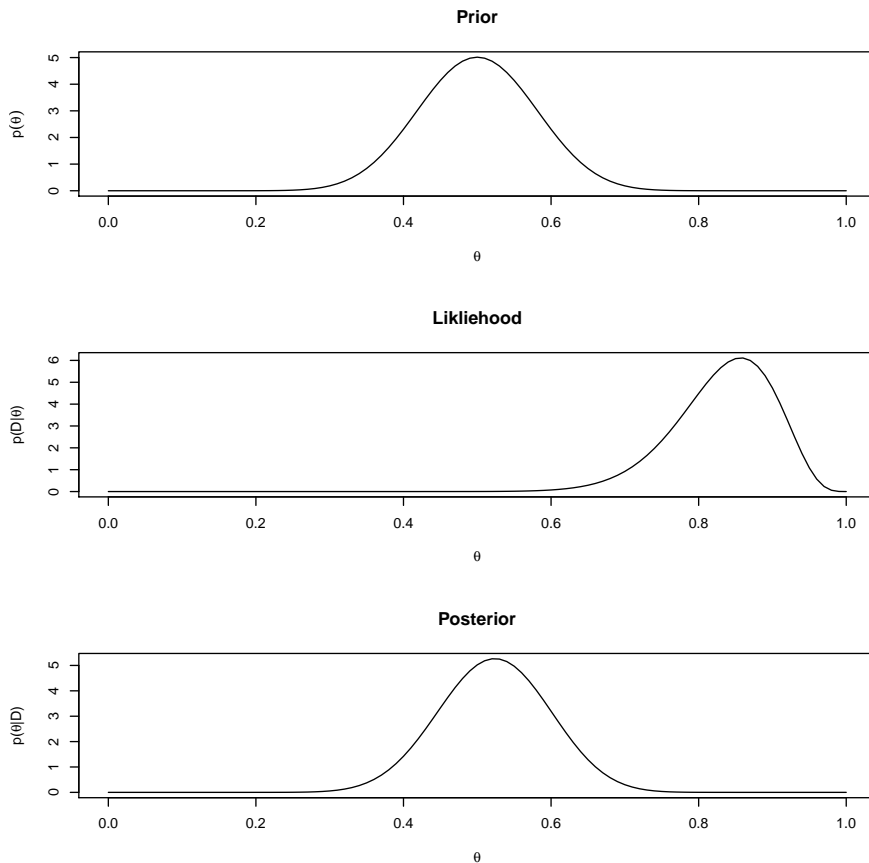
```

par(mfrow=c(3,1))
curve(dbeta(x,shape1=20,shape2=20),main="Prior",
xlab=expression(theta),ylab=expression(paste("p",theta)))

curve(dbeta(x,shape1=25,shape2=5),main="Likelihood",
xlab=expression(theta),ylab=expression(paste("p(D|",theta,")")))

curve(dbeta(x,shape1=23,shape2=21),main="Posterior",
xlab=expression(theta),ylab=expression(paste("p(",theta,"|D)")))

```



Example 2: Suppose we want to estimate a pro basketball player's free-throw making. Our prior knowledge comes from knowing that the average in professional basketball is around 75%. Our belief in this is much lower than example 1 so we will use a smaller a and b . We chose $a = 19$ and $b = 8$ such that the prior distribution is centered around 0.75, but also knowing we want the posterior to be less influenced by our prior belief. Next, we observe the basketball player makes 17 out of 20 (also 85%).

The posterior model is moved over a reasonable chunk as a result of our smaller value of a and b .

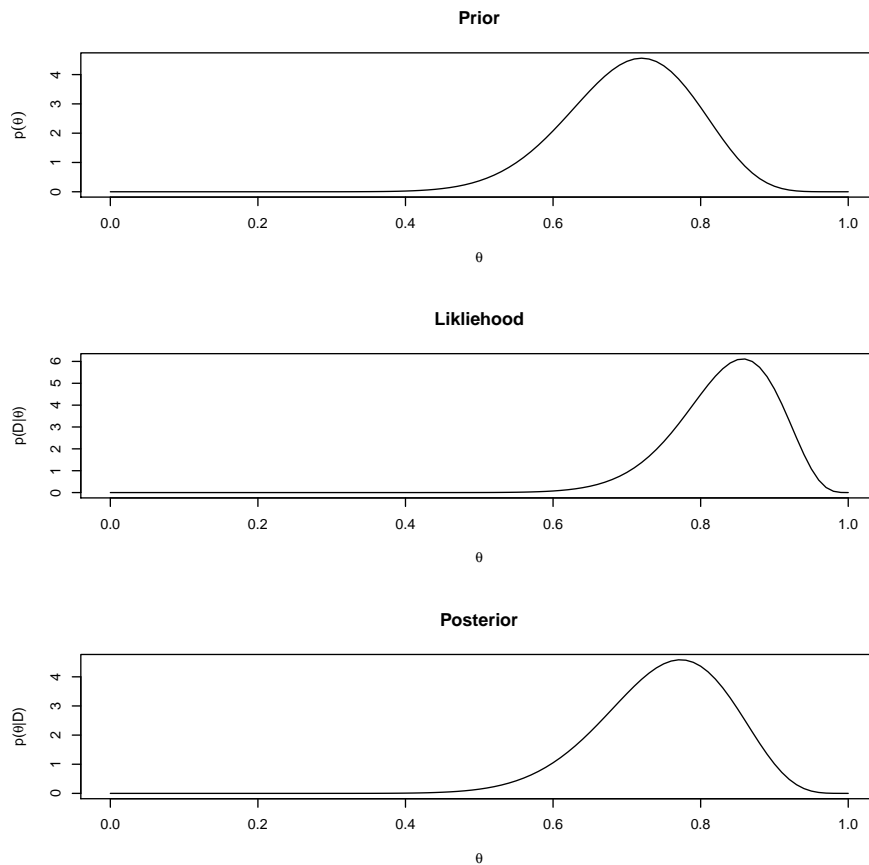
As a result, the mode is just under 80%. The likelihood is the same since we have the same values of N and z . Notice how our posterior mean is more influenced by our data than in Example 1,

this is because a and b are much smaller.

```
par(mfrow=c(3,1))
curve(dbeta(x,shape1=19,shape2=8),main="Prior",
xlab=expression(theta),ylab=expression(paste("p",theta)))

curve(dbeta(x,shape1=25,shape2=5),main="Likelihood",
xlab=expression(theta),ylab=expression(paste("p(D|",theta,")")))

curve(dbeta(x,shape1=18,shape2=6),main="Posterior",
xlab=expression(theta),ylab=expression(paste("p(",theta,"|D)")))
```



Example 3: Suppose we are trying to observe some colorful rocks on a distant planet, and each rock is either blue or green. We want the probability that we will grab a blue versus green rock (and call blue success). Suppose we have absolutely no knowledge beforehand, so our prior model will not be informative. To reflect this, we have chosen $a = b = 1$. If the robots find 17 out of 20 blue, we see that the prior has very little influence on the posterior while the data is essentially identical to the posterior.

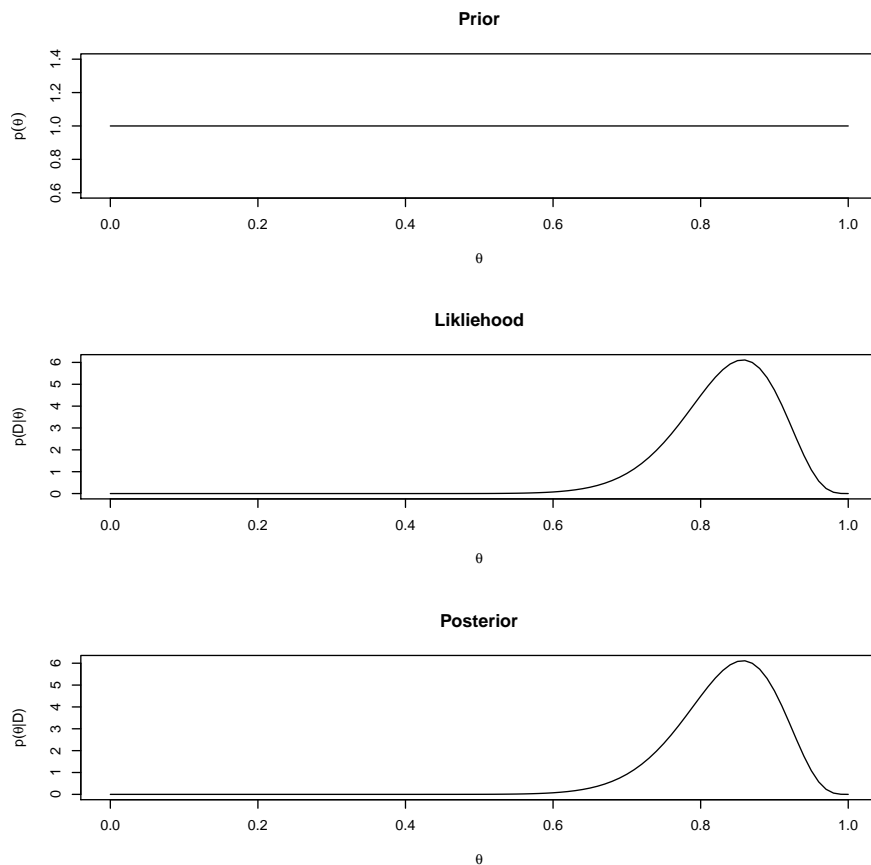

```

par(mfrow=c(3,1))
curve(dbeta(x,shape1=1,shape2=1),main="Prior",
xlab=expression(theta),ylab=expression(paste("p",theta)))

curve(dbeta(x,shape1=25,shape2=5),main="Likelihood",
xlab=expression(theta),ylab=expression(paste("p(D|",theta,")")))

curve(dbeta(x,shape1=25,shape2=5),main="Posterior",
xlab=expression(theta),ylab=expression(paste("p(",theta,"|D)")))

```



5.1 Bayesian Confidence Intervals

The Bayesian credible interval is a range of values that the unknown parameter lies in with a given probability.

In Bayesian Statistics, we treat the boundaries as fixed, and the unknown parameter as the variable. For example if a 95% credible interval for some parameter θ is (L, U) , then we say L and U are fixed lower and upper bounds, and θ varies. That is a 95% probability that θ falls between L and U . This contrasts that of frequentist confidence intervals which are reversed.

In frequentist statistics, a confidence interval is an estimate of where a given fixed parameter lies (with some amount of confidence). In other words, 95% confidence means that if one was to gather data 100 times, at least 95 out of those 100 times, the parameter will lie in the interval constructed. Note, there will be 100 different confidence intervals with different l and u . “We are 95% confident that the true parameter lies in the interval (l, u) .”

A credible interval is actually quite natural compared to a confidence interval. An interpretation of a credible interval is: “There is a 95% chance that the parameter lies in our interval.” This makes sense to most people because hearing “a 95% chance” has a more intuitive meaning than the frequentist confidence interval explanation.

For our three examples earlier with coins, free-throws, and random rocks, we see the frequentist 95% confidence interval for $N = 20$ and $z = 17$ is

$$\frac{17}{20} \pm 1.96 \sqrt{\frac{(17/20)(3/20)}{20}} = (0.6935, 1.01) = (0.69, 1).$$

Note that though our calculation gave us an upper bound of 1.01, our greatest possible value is 1, so we adjust.

In contrast, let us look at the credible intervals for the three examples. The credible interval for Example 1 is: (0.377, 0.667), for 2 is: (0.563, 0.898), and for 3 is: (0.683, 0.942). Notice how the frequentist is the same for all three examples, but the credible intervals depend on our prior knowledge and data so they are different.

6 Markov Chain Monte Carlo

The Markov chain Monte Carlo technique is used for producing accurate approximations of a posterior distribution for realistic applications. As it turns out, the mathematics behind creating posterior distributions can be challenging to say the least. The class of methods used to simplify this is called the Markov chain Monte Carlo (or MCMC for short). With technological advances, we are able to use Bayesian data analysis of realistic application which would not have been possible 30 years ago.

Sometimes we have multiple parameters instead of just θ , so we have to adjust. For example, suppose we have 6 parameters. The parameter space is a six-dimensional space involving the joint distribution of all combinations of parameter values. If each parameter can take on 1,000 values, then the six-dimensional parameter space has $1,000^6$ combinations of parameter values. This is extremely impractical.

For the rest of this section, assume that the prior distribution is specified by an easily evaluated function. So for a specified θ , $p(\theta)$ is easily calculated. We also assume that the likelihood function $p(D|\theta)$ can be computed for specified D and θ . The MCMC will give us an approximation for the posterior: $p(\theta|D)$.

For this technique, we see the population from which we sample from as a mathematically defined distribution (like we do with the posterior probability distribution).

6.1 General Example of the Metropolis Algorithm with a Politician

The Metropolis algorithm is a type of MCMC. Here are the rules of the game:

- Suppose that there is a long chain of islands. A politician wants to visit all the islands, but spend more time on island with more people.
- At the end of each day he can stay, go one island east, or one island west.
- He does not know of the populations or how many islands there are.
- When he is on an island, he can figure out how many people there are on that island.
- When the politician proposes a visit to an adjacent island, he can figure out the population of that island as well.

The politician plays the game in this way:

- He first flips a fair coin to decide whether to propose moving east or west.
- Then, if the proposed island has a larger population than his current island, then he definitely goes; but if the proposed island has a smaller population, then he goes to the proposed island only probabilistically.

If the population of the proposed island is only half as big, he goes to the island with a 50% chance. In detail, let $P_{proposed}$ be the population of the proposed island, and $P_{current}$ be the population of current island. Notice that these capitals P's are not probabilities. Then, his probability of moving is

$$p_{move} = \frac{P_{proposed}}{P_{current}}.$$

He does this by spinning a fair spinner marked on its circumference with uniform values from 0 to 1. If the pointed to value is between 0 and p_{move} , then he moves.

6.2 A Random Walk

We will continue with the island hopping but use a more concrete example. Suppose there is a chain of 7 islands. We will index the islands with θ such that $\theta = 1$ is the west-most island, and $\theta = 7$ is the east-most island. Population increases linearly, such that the relative population (not absolute population) $P(\theta) = \theta$.

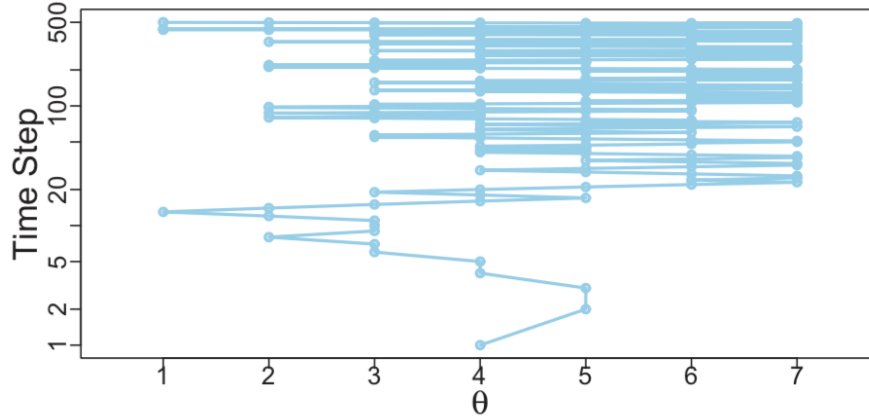
Also it may be helpful to pretend that there are more islands on either end of the 7, but with a population of 0 so the proposal to go there will always be rejected.

Say the politician starts on island $\theta_{current} = 4$ at day 1: $t = 1$. In order to decide where to go on day 2 he flips a coin to decide whether to propose moving east or west. Say the coin proposes moving east, then $\theta_{proposed} = 5$. Let us say that $P(5) > P(4)$ or the relative population of island 5 is greater than that of 4, then the move is accepted and thus at $t = 2$, $\theta = 5$.

Next, we have $t = 2$ and $\theta = 5$. Say the coin flip proposes a move west. Probability of accepting this proposal is $p_{move} = \frac{P(\theta_{proposed})}{P(\theta_{current})} = \frac{4}{5} = 0.8$. Then he spins a fair spinner with circumference marked from 0 to 1. Suppose he gets a value greater than 0.8, then the proposed

move is rejected; thus $\theta = 5$ when $t = 3$. This is repeated thousands of times.

The picture below is from page 148 of Kruschke's book and shows 500 steps of the process above. [2]



6.3 General Properties of a Random Walk

We can combine our knowledge above to get the probability of moving to the proposed position:

$$P_{move} = \min\left(\frac{P(\theta_{proposed})}{P(\theta_{current})}, 1\right).$$

This is then repeated (using technology) thousands of times as a simulation.

7 More About MCMC

MCMC is extremely useful when our target distribution, the posterior distribution, $P(\theta)$ is proportional to $p(D|\theta)p(\theta)$, the likelihood times the prior. By evaluating $p(D|\theta)p(\theta)$ without normalizing $p(D)$, we can generate random representative values from the posterior distribution. Though we will not use the mathematics directly, here is kind of a pseudo proof of why it works.

Suppose there are two islands. The relative transition probabilities between adjacent positions exactly match the relative values of the target distribution.

If we do this across all the positions, in the long run, the positions will be visited proportionally to the target distribution.

7.1 The Math Behind MCMC

We are on island θ . The probability of moving to $\theta+1$ is $p(\theta \rightarrow \theta+1) = 0.5 \cdot \min(P(\theta+1)/P(\theta), 1)$. Notice that this is the probability of proposing the move times the probability of accepting the proposed move. Recall that we first flipped a coin to see if we will even propose the move (0.5), and then continued the move probabilistically.

Similarly, if we are on island $\theta + 1$, then the probability of moving to θ is:

$$p(\theta + 1 \rightarrow \theta) = 0.5 \cdot \min(P(\theta)/P(\theta + 1), 1)$$

The ratio of the two is:

$$\begin{aligned} \frac{p(\theta \rightarrow \theta + 1)}{p(\theta + 1 \rightarrow \theta)} &= \frac{0.5 \cdot \min(P(\theta + 1)/P(\theta), 1)}{0.5 \cdot \min(P(\theta)/P(\theta + 1), 1)} \\ &= \begin{cases} \frac{1}{P(\theta)/P(\theta + 1)} & \text{if } P(\theta + 1) > P(\theta) \\ \frac{P(\theta + 1)/P(\theta)}{1} & \text{if } P(\theta + 1) < P(\theta) \end{cases} \\ &= \frac{P(\theta + 1)}{P(\theta)} \end{aligned}$$

This tells us that when going back and forth between adjacent islands, the relative probability of the transition is the same as the relative values of the target distribution.

Though we have applied this Metropolis algorithm to only a discrete case in one dimension with proposed moves only east and west, this can be extended to continuous values on any number of dimensions and with a more general proposal distribution.

The method is still the same for a more complicated case:

We must have a target distribution $P(\theta)$ over a multidimensional continuous parameter space from which we generate representative sample values.

We must be able to compute the values of $P(\theta)$ for any possible θ .

The distribution $P(\theta)$ does not need to be normalized (just nonnegative).

Usually $P(\theta)$ is the unnormalized posterior distribution on θ .

8 Generalize Metropolis Algorithm to Bernoulli Likelihood and Beta Prior

In the island example, the islands represent candidate parameter values, and the relative populations represent relative posterior probabilities. Transitioning this to coin flipping: the parameter θ takes on values from the continuous interval 0 to 1, and the relative posterior probability is computed as likelihood times prior.

It is as if there are an infinite number of tiny islands, and the relative population of each island is its relative posterior probability density. Furthermore, instead of only proposing jumps to two possible islands (namely the adjacent islands), we can propose jumps to further islands.

We need a proposal distribution that will let us visit any parameter value on the continuum 0 to 1. We will use a uniform distribution.

8.1 Metropolis Algorithm to Coin Flips

We will as usual assume we flip a coin N times and observe z heads. We use our Bernoulli likelihood function $p(z, N|\theta) = \theta^z(1 - \theta)^{(N - z)}$ and start with a prior $p(\theta) = \beta(\theta|a, b)$.

For the proposal jump in the Metropolis algorithm, we use a standard uniform distribution that is $\theta_{prop} \sim \text{unif}(0, 1)$.

Denote the current position as θ_c , and the proposed parameter as θ_p .

We then have 3 steps.

Step 1: randomly generate a proposed jump, $\theta_p \sim \text{unif}(0, 1)$.

Step 2: compute the probability of moving to the proposed value/position. The following is a quick derivation:

$$\begin{aligned} &= p_{move} = \min\left(1, \frac{P(\theta_p)}{P(\theta_c)}\right) \\ &= \min\left(1, \frac{p(D|\theta_p)p(\theta_p)}{p(D|\theta_c)p(\theta_c)}\right) \\ &= \min\left(1, \frac{\text{Bernoulli}(z, N|\theta_p)\text{beta}(\theta_p|a, b)}{\text{Bernoulli}(z, N|\theta_c)\text{beta}(\theta_c|a, b)}\right) \\ &= \min\left(1, \frac{\theta_p^z(1-\theta_p)^{N-z}\theta_p^{a-1}(1-\theta_p)^{b-1}/B(a, b)}{\theta_c^z(1-\theta_c)^{N-z}\theta_c^{a-1}(1-\theta_c)^{b-1}/B(a, b)}\right). \end{aligned}$$

The above follows through the generic Metropolis form, P as likelihood times prior, and finally the Bernoulli likelihood and beta prior. If the proposed value θ_p happens to be outside of the bounds of θ , then the prior or likelihood is set to 0, thus $p_{move} = 0$.

Step 3: accept the proposed parameter value if a random value sampled from $[0, 1]$ uniform distribution is less than p_{move} . If not, then reject the proposed parameter value and "stay on the same island for another day".

The above steps are repeated until there is a sufficiently representative sample generated.

9 MCMC Examples

9.1 MCMC Using a Beta Distribution

The main idea of this code is to simulate a posterior distribution. First we will show the code as a whole with its outputs, and then we will break our code into four chunks to talk about separately. Note that the “#” indicates a section that is commented out, purely there for understanding or notes. We are simulating our posterior model, so will look at the predicted mean as well as a histogram and simple plot.

```
#t=theta
#N=flips
#z=heads
#likelihood=(1-t)^(N-z)*t^z
#proportion as parameter

N=20
z=17
a=25
b=5
```

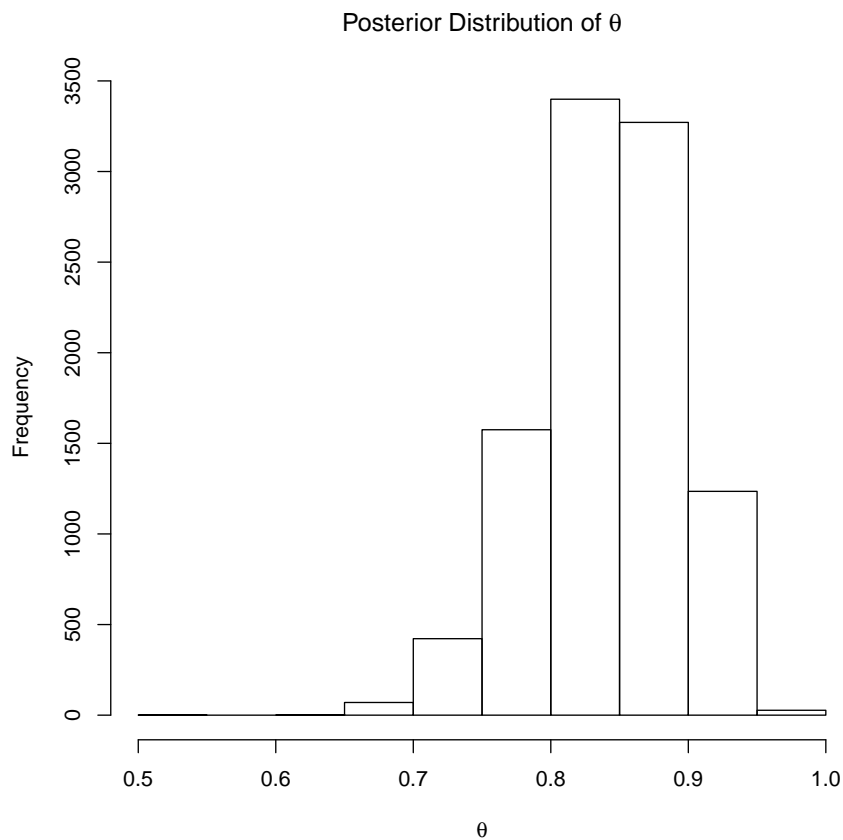
```

bern.like=function(t,N,z){((1-t)^(N-z))*t^(z)}
#bern.like(0.5,20,17) to test our function
post.theta=c()
#.5 as starting point
post.theta[1]=0.5

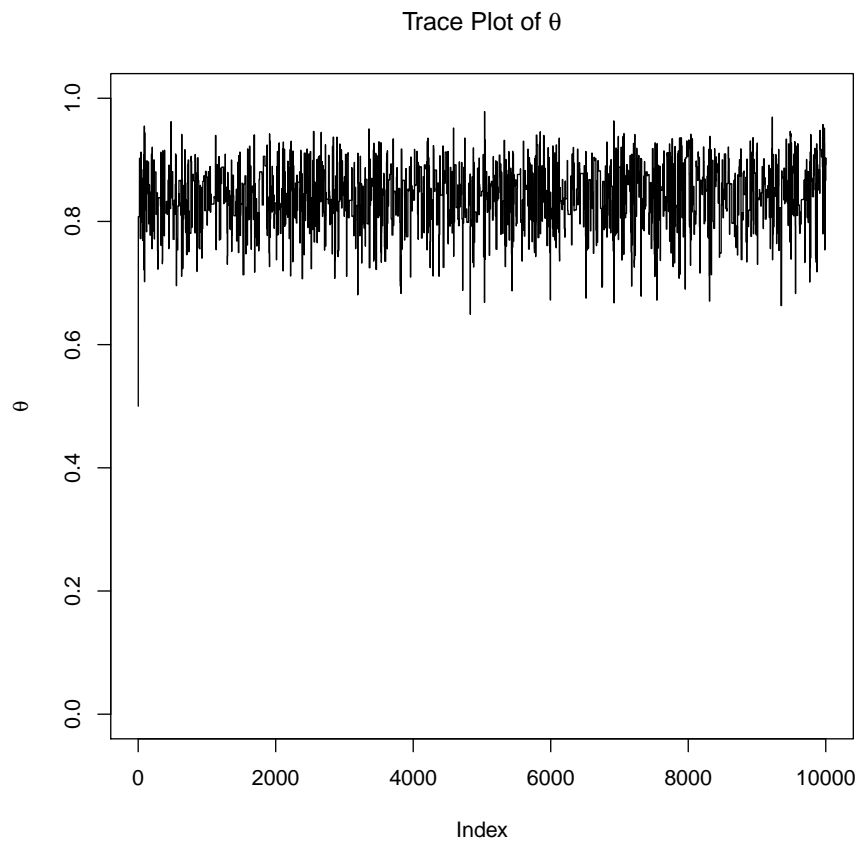
for(i in 1:10000){
prop.theta=runif(1,0,1)
#like*prior for current value
temp.num=bern.like(prop.theta,N,z)*dbeta(prop.theta,a,b)
temp.den=bern.like(post.theta[i],N,z)*dbeta(post.theta[i],a,b)
temp=temp.num/temp.den
R=min(temp,1)
A=rbinom(1,1,R)
post.theta[i+1]=ifelse(A==1,prop.theta,post.theta[i])
}

hist(post.theta, main=expression(paste("Posterior Distribution of ", theta)),
xlab=expression(paste(theta)))

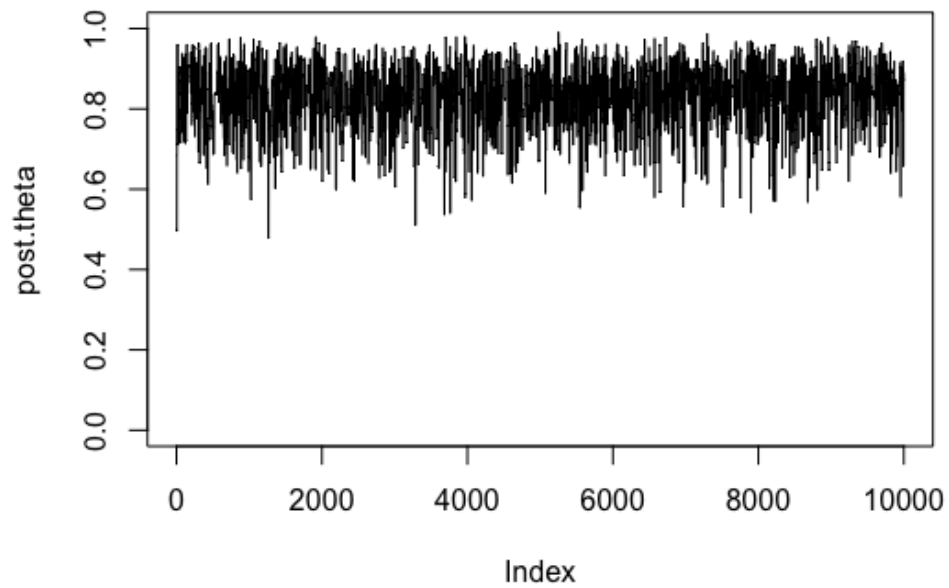
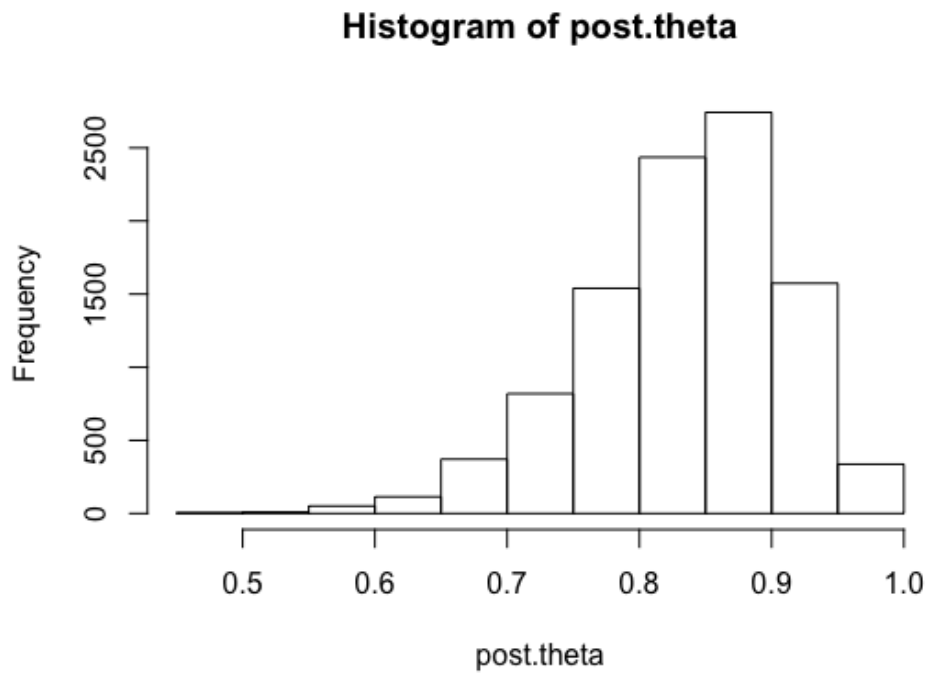
```



```
plot(post.theta,type="l",ylim = c(0,1), main=expression(paste("Trace Plot of ",
theta)),ylab=expression(paste(theta)))
```



```
mean(post.theta)
## [1] 0.8404051
```

The output for the predicted theta value is 0.834. This is reflected in the graphs above: the horizontal axis on the histogram, and the vertical axis on the walk plot.

First we define our variables that we use in the code. The variables we use are t , N , and Z . We also specify our equation for the likelihood.

```
#t=theta
#N=flips
#z=heads
#likelihood=(1-t)^(N-z)*t^(z)
#proportion as parameter
```

Second, we define our Bernoulli likelihood with the below function. We then state our prior value of θ , the number of flips (20), and the number of heads (17). We then define our posterior theta as an empty vector that we will fill in with our results from MCMC. As our prior belief is that theta is around 0.5, we decide to start our walk at $\theta = 0.5$. This could really start at any value from zero to one.

```
bern.like=function(t,N,z){((1-t)^(N-z))*t^(z)}
bern.like(0.5,20,17)
post.theta=c()
#0.5 as starting point
post.theta[1]=0.5
```

Third, we need to tell R how many times to run through our simulation; we have chosen 10,000. Then we define our numerator and denominator and complete the calculation for the value of “temp”, our probability of moving to our theoretical “proposed island”. If the value is greater than 1, then we move, if not (if else), we move with a probability of “temp”, we do this by letting R be the minimum of temp and 1. We then use A to sample from a binomial distribution with $n=1$, $size=1$, and probability R . The next line simply loops our “if, then, if else” code.

```
for(i in 1:10000){prop.theta=runif(1,0,1)
#like*prior for current value
temp.num=bern.like(prop.theta,N,z)*dbeta(prop.theta,a,b)
temp.den=bern.like(post.theta[i],N,z)*dbeta(post.theta[i],a,b)
temp=temp.num/temp.den
R=min(temp,1)
A=rbinom(1,1,R)
post.theta[i+1]=ifelse(A==1,prop.theta,post.theta[i])}
```

Fourth and finally, we ask for our histogram of theta values, a trace plot to follow our walk, and the mean that we received for that given walk. We should note that every time we run this, we will receive a slightly different result. If we were to insert a set seed command, we could theoretically save our results.

```
hist(post.theta)
plot(post.theta,type="l",ylim = c(0,1))
mean(post.theta)
```

9.2 MCMC Using a Normal Distribution

Similar to the above, we are using this code to take our prior knowledge and combine it with our data in order to simulate a posterior distribution. That said we are using a normal distribution with mean of 25. Though we know the mean, we are trying to get our model to predict the mean of 25. First we will show the code as a whole with its outputs, and then we will break our code into four chunks to talk about separately. We are simulating our posterior model, so, again, we will look at the predicted mean as well as a histogram and simple plot.

```
#n=trials, m=mu, s=sigma
#code to simulate data from a normal distribution with mean 20 and sd 3

n=65
set.seed(1)
data=rnorm(65,25,3)
data

## [1] 23.12064 25.55093 22.49311 29.78584 25.98852 22.53859 26.46229
## [8] 27.21497 26.72734 24.08383 29.53534 26.16953 23.13628 18.35590
## [15] 28.37479 24.86520 24.95143 27.83151 27.46366 26.78170 27.75693
## [22] 27.34641 25.22369 19.03194 26.85948 24.83161 24.53261 20.58774
## [29] 23.56555 26.25382 29.07604 24.69164 26.16301 24.83858 20.86882
## [36] 23.75502 23.81713 24.82206 28.30008 27.28953 24.50643 24.23991
## [43] 27.09089 26.66999 22.93373 22.87751 26.09375 27.30560 24.66296
## [50] 27.64332 26.19432 23.16392 26.02336 21.61191 29.29907 30.94120
## [57] 23.89834 21.86760 26.70916 24.59484 32.20485 24.88228 27.06922
## [64] 25.08401 22.77018

#data.input: data that is needed to calculate likelihood
#m.input mean value that is needed to calculate likelihood

N=20
z=17
a=25
b=5

#using the simulated data to estimate the mean
norm.like=function(data.input, m.input){prod(dnorm(data.input,m.input,3))}

post.mu=c()
prop.mu=c()
#19 arbitrary; our "first island"/ first guess
post.mu[1] = 19

#again fairly arbitrary\ prior belief about behavior of center and spread
m=20
s=7

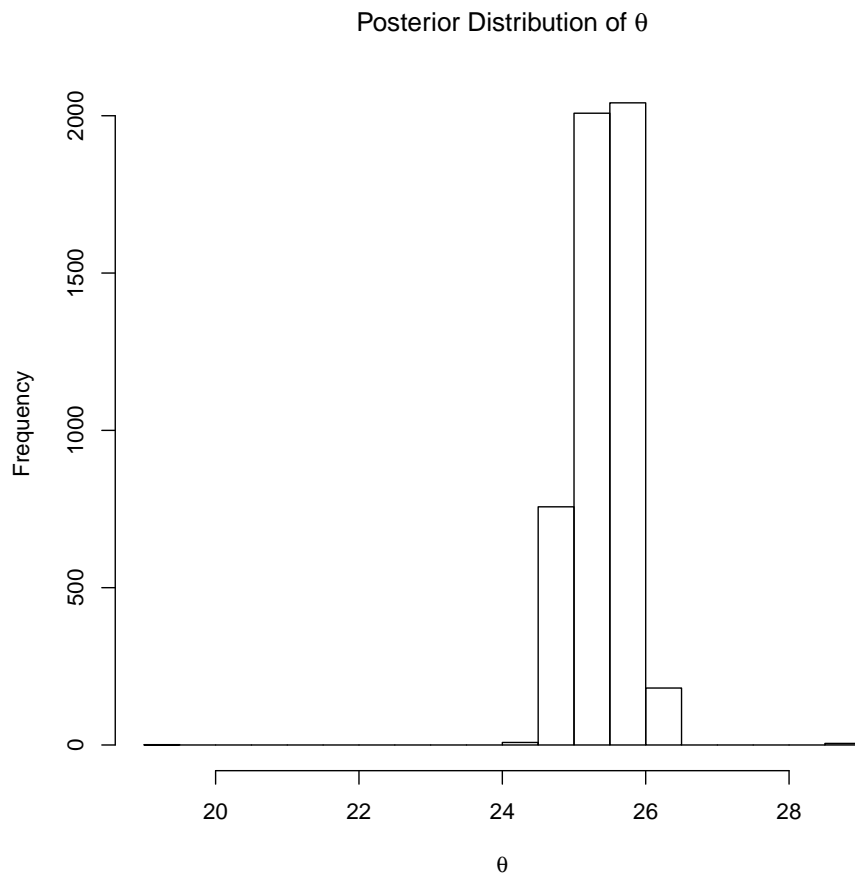
for(i in 1:5000){
  prop.mu[i]=runif(1,0,50)
```

```

num=norm.like(data,prop.mu[i])*dnorm(prop.mu[i],m,s)
den=norm.like(data,post.mu[i])*dnorm(post.mu[i],m,s)
temp=num/den
R=min(temp,1)
A=rbinom(1,1,R)
post.mu[i+1]=ifelse(A==1,prop.mu[i],post.mu[i])
}

hist(post.mu, main=expression(paste("Posterior Distribution of ", theta))
,xlab=expression(paste(theta)))

```

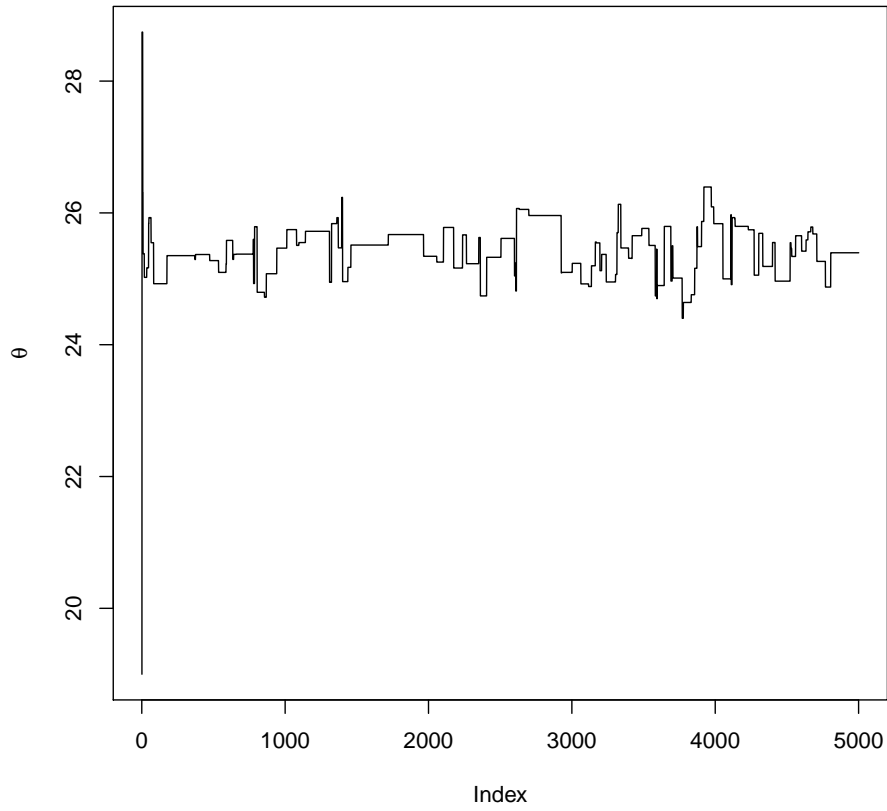


```

plot(post.mu,type="l",main=expression(paste("Trace Plot of ", theta)),
ylab=expression(paste(theta)))

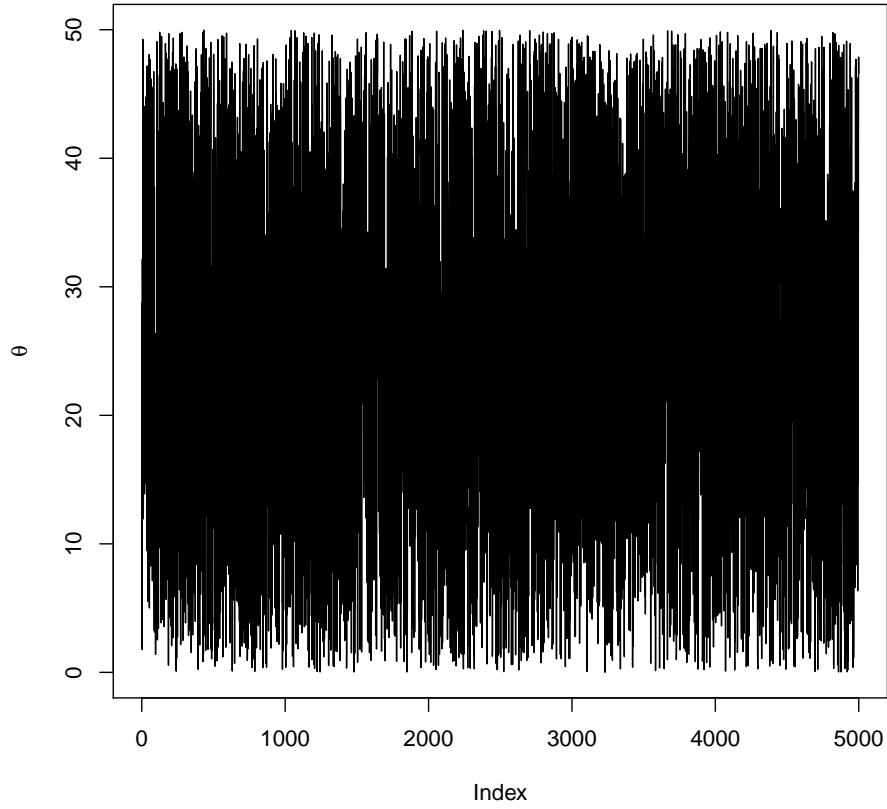
```

Trace Plot of θ



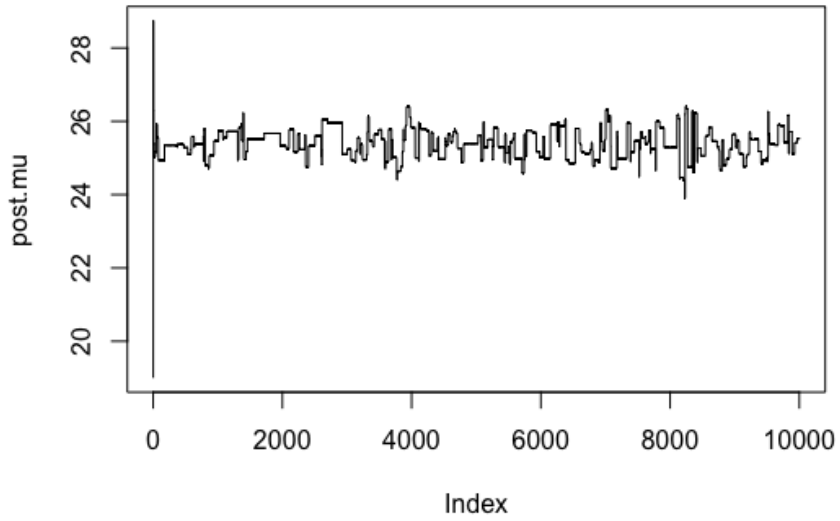
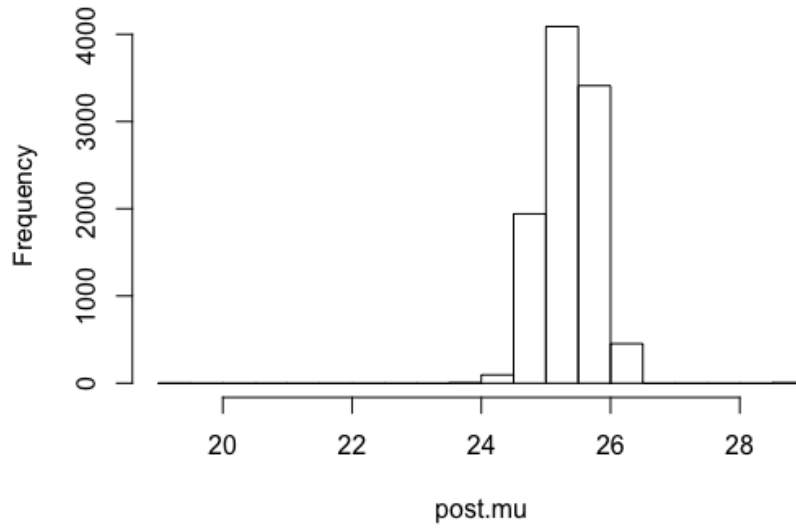
```
plot(prop.mu,type="l",main=expression(paste("Trace Plot of ", theta)),  
ylab=expression(paste(theta)))
```

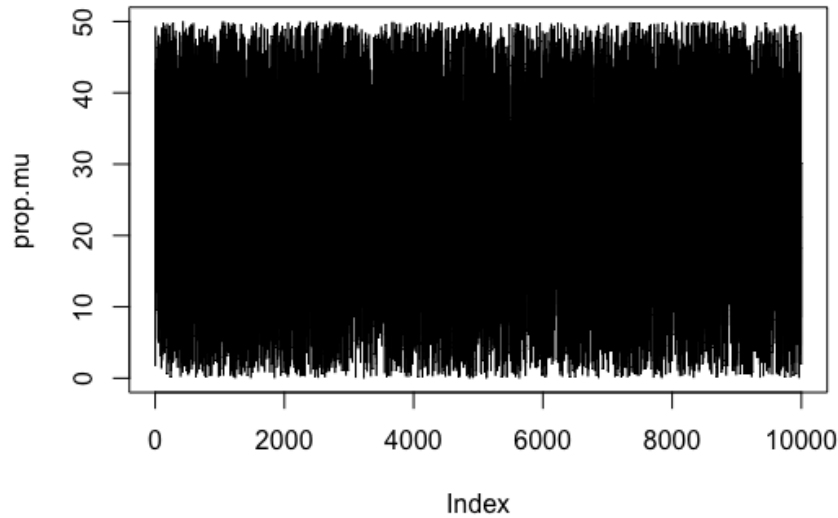
Trace Plot of θ



```
mean(post.mu)
## [1] 25.42862
```

Histogram of post.mu





The output for the predicted mean is 25.3873. This is reflected in the graphs above: the horizontal axis on the histogram, and the vertical axis on the walk plots. The last graph is a bit hard to interpret because of how dense it is, this is because of how many iterations we ran through.

First we specify that our data has 65 samples, starting at one, and with a mean of 25. We then set the seed so we can replicate our results. We then produce data that follows a normal distribution with a mean of 25 and standard deviation of 3.

```
n=65
set.seed(1)
data=rnorm(65,25,3)
data
data.input
m.input
```

Second, we define our likelihood in the case of a normal distribution. The norm.like function calculates the normal likelihood for any given data set and mean value. After that, we create empty vectors for post.mu and prop.mu. After this we pick a starting point for our walk. This is essentially choosing which island to start our random walk on. We have chosen 19. Lastly for this section, we include a guess of what the mean and standard deviation could be.

```
likelihood=cumprod
norm.like=function(data.input, m.input){prod(dnorm(data.input,m.input,3))}

post.mu=c()
prop.mu=c()
```



```
#6 arbitrary; our "first island"/ first guess
post.mu[1] = 19

#again fairly arbitrary\ prior belief about behavior of mean
m=20
s=7
```

Third, we choose to simulate 10,000 steps. We first calculate the like.prior for the proposed value and the like.prior for the current value. Then we define our numerator and denominator and complete the calculation for the value of “temp”, our probability of moving to our theoretical “proposed island”. If the value is greater than 1, then we move, if not (if else), we move with a probability of “temp”, we do this by setting R (the probability of accepting the proposed value) to be the minimum of temp and 1. We then use A to sample from a binomial distribution with $n=1$, $size=1$, and probability R . The next line simply loops our “if, then, if else” code.

```
#runif used for proposing new parameter value
for(i in 1:10000){prop.mu[i]=runif(1,0,50)
num=norm.like(data,prop.mu[i])*dnorm(prop.mu[i],m,s)
den=norm.like(data,post.mu[i])*dnorm(post.mu[i],m,s)
temp=num/den
R=min(temp,1)
A=rbinom(1,1,R)
post.mu[i+1]=ifelse(A==1,prop.mu[i],post.mu[i])
}
```

Fourth and finally, we simply ask for a histogram of our posterior mu, include plots of both post.mu and prop.mu, and ask for our predicted mean.

```
hist(post.mu)
plot(post.mu,type="l")
plot(prop.mu,type="l")
mean(post.mu)
\end{lstlisting}
```

10 Conclusion

We have explored two ways to make inferences using the Bayesian approach: the theoretical model and simulation. The theoretical model uses our prior belief or knowledge of an unknown parameter to influence our collected data to create a posterior estimate that we draw inference from. Sometimes the math behind calculating the posterior is too difficult and/or impractical. The simulation method is similar in that we use prior beliefs to influence our data, but instead of directly calculating our posterior, we use MCMC to simulate a posterior.

References

- [1] Barry Evans
Bayes, Mammograms and False Positives
<https://www.northcoastjournal.com/humboldt/bayes-mammograms-and-false-positives/Content?oid=3677370> 2016
- [2] John Kruschke Doing Bayesian Data Analysis: A Tutorial Introduction with R
2011 Elsevier Inc.
- [3] Rafael Irizarry I declare the Bayesian vs. Frequentist debate over for data scientists
<https://simplystatistics.org/2014/10/13/as-an-applied-statistician-i-find-the-frequentists-versus-bayesians-debate-completely-inconsequential/>
- [4] The Cthaeh What is Bayes' Theorem?
<https://www.probablisticworld.com/what-is-bayes-theorem/>
Probabilistic World 2016
- [5] Wikipedia Andrey Markov
<https://en.wikipedia.org/wiki/AndreyMarkov>
- [6] Wikipedia Markov chain
<https://en.wikipedia.org/wiki/Markovchain#History>
- [7] Wikipedia Thomas Bayes
<https://en.wikipedia.org/wiki/ThomasBayes>, <https://en.wikipedia.org/wiki/Bayes%27theorem>