Analyzing Radiocarbon Data with Temporal Order Constraints

A Thesis
Presented to
The Division of Mathematics and Natural Sciences
Reed College

In Partial Fulfillment
of the Requirements for the Degree
Bachelor of Arts

Reilly H. Villanueva

May 2016

Approved for the Division
(Mathematics)

Albyn Jones

Acknowledgements

First, thank you to my advisor, Albyn, without whom I could not have written this thesis. Your patience and flexibility with me on this project is immensely appreciated.

Next, thank you to Ilana and Simran for experiencing Reed along with me from the beginning. Here's to four years of late night talks and study sessions.

Finally, more than anyone else, I want to thank my parents, whose support made getting through this possible. I couldn't have done it without you guys.

Table of Contents

Abstract

In this thesis, a method for estimating the ages of multiple samples with radiocarbon data and known ordering is developed and then applied to data in an attempt to date the R tephra at Mount Rainier National Park. The method relies on Markov chain Monte Carlo methods, in particular, the Metropolis-Hastings algorithm.

Chapter 1

Introduction

Carbon-14 dating is a method to determine the age of artifacts pioneered by chemist Willard Libby in the mid-1940's (Bowman [1990]). The technique has been used to date many artifacts as well as archaeological and geological events. Estimates of how many carbon dates have been measured since 1953 range from 500000 to over 1 million (Taylor et al. [2014]). Carbon dating methods have been performed on many historic artifacts, including the Dead Sea scrolls and the Turin Shroud, which some believe had been used to wrap Christ's body, showing they date to around the turn of the eras (BC to AD) and the medieval period, respectively (Damon et al. [1989]; VanderKam [1994]). More broadly, carbon dating allows researchers access to a common time scale, even when studying sites where records of time were not kept. Thus carbon dating is especially useful when studying prehistoric events.

Carbon Dating Overview

All living things take in carbon from the atmosphere. Most carbon atoms are of the stable isotope carbon-12 (also written ^{12}C), which has six protons and six neutrons, but a small fraction of all carbon atoms are of the isotope carbon-14 (^{14}C) which, with six protons and eight neutrons, is radioactive. The ratio of carbon-14 to carbon-12 in living things is about the same as the ratio in the atmosphere, since both isotopes have the same chemical properties. However, when living things die, they stop taking in new carbon, and carbon-14 atoms continue to decay while the stable carbon-12 atoms do not.

The half-life of an isotope is the time it takes for half of the atoms in a sample of that isotope to decay. Carbon-14 has a half-life of 5730 years (Bowman [1990]). By comparing the ratio of carbon-14 to carbon-12 when the sample was living and some unknown time after it dies, we can calculate how much time has passed since the sample stopped taking in carbon from the atmosphere. To calculate this, first define the decay constant λ by

$$\lambda = \frac{\ln(2)}{t_{\frac{1}{2}}}$$

where $t_{\frac{1}{2}}$ is the half-life of carbon-14. Below, A is the measured ratio of carbon in a sample at present, A_0 is the ratio of carbon in the sample at the time it died, and t is the length of time the radioactive carbon has been decaying. Then we can find the time since the death of the specimen:

$$\begin{aligned} A &= A_0 e^{-\lambda t} \\ \frac{A}{A_0} &= e^{-\lambda t} \\ \ln \frac{A}{A_0} &= -\lambda t \\ -\frac{\ln \frac{A}{A_0}}{\lambda} &= t \end{aligned}$$

Thus the time since the specimen stopped taking in carbon from the atmosphere is

$$t = \frac{\ln(A_0) - \ln(A)}{\lambda}.$$

This points to a smooth exponential relationship between measured carbon dates and calendar years.

Unfortunately, carbon dating is not so straightforward. The ratio of carbon-14 to carbon-12 in the atmosphere is not constant over time, so measuring the carbon ratio in living things from the present is not a good estimate for the initial carbon ratio of a sample. Thus A_0 is unknown and finding a corresponding calendar year for a given carbon dated sample is not so simple (Reimer et al. [2013]). Scientists who want their samples carbon dated send them to a lab and receive a carbon date along with an estimated standard deviation (Naylor & Smith [1988]). While this carbon date is a function of the calendar year from which the sample originated, the constant A_0 is unknown, making it difficult to find the true age of a sample.

In order to make determining calendar years from carbon dates possible, calibration curves were created. The calibration curve used here, IntCal13 (Reimer et al. [2013]), spans from 0 to 50,000 BP, where BP stands for before present. The year 0 BP is defined as 1950 and larger values correspond to older ages. Carbon dating is very unreliable for dates more recent than 1950 due to nuclear testing, which has caused rapid fluctuations in the carbon ratio. The IntCal13 calibration curve was generated using data from samples that could be both carbon dated and more accurately dated using another method. For instance, approximately the first 14000 years BP of data came from tree ring samples that could be dated using dendrochronological techniques (Reimer et al. [2013]). The IntCal13 calibration curve is shown in Figure 1.1. The calibration curve is neither smooth nor monotonic, so having a carbon date will not necessarily produce an accurate calendar date estimate, as a single carbon date may be associated with multiple calendar years (Naylor & Smith [1988]).

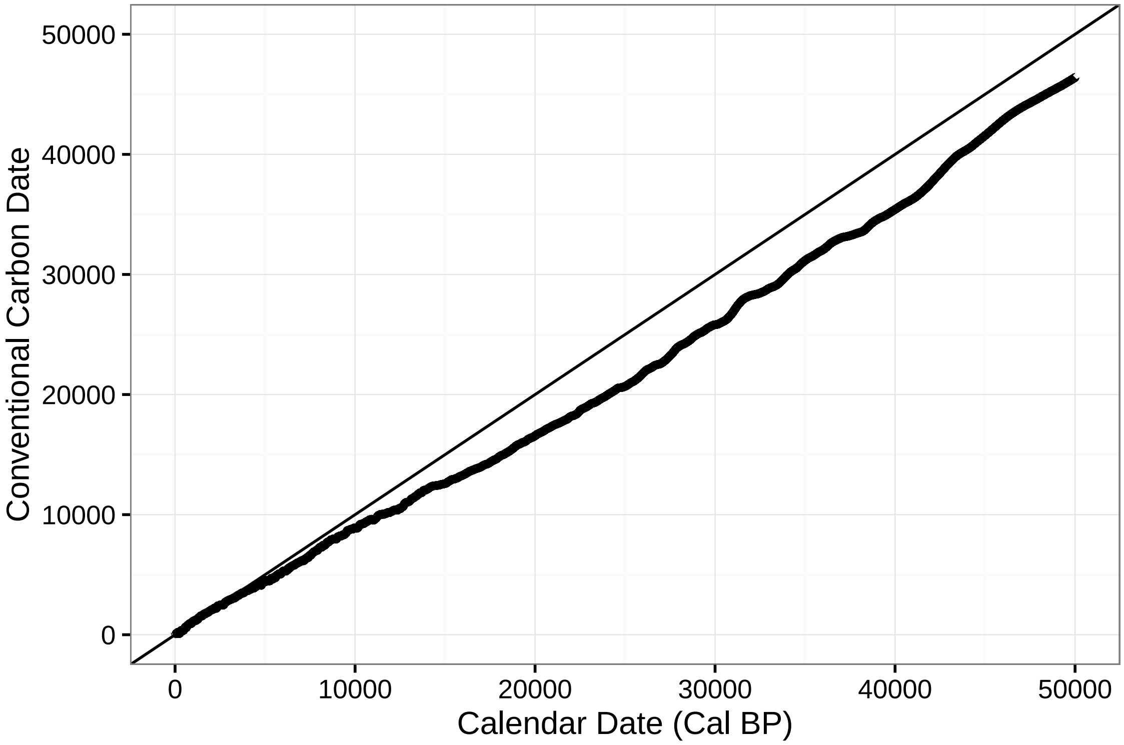

Figure 1.1: The most recent (2013) calibration curve relating conventional carbon dates to calendar year (BP) shown with the identity line.

Brief History of Radiocarbon Dating

In 1939, it was discovered that cosmic rays (atoms without electrons traveling at very high speeds) collided with atoms in the upper atmosphere to produce free neutrons. These neutrons would be absorbed by nitrogen atoms, which would release a proton, forming the isotope carbon-14. This discovery led to the development of carbon dating.

As previously mentioned, the idea of using this radioactive carbon to date archaeological and geological samples was first suggested in a paper published in 1946 by

Willard Libby, a chemist at the University of Chicago. Immediately, work began to verify the concept of carbon dating. In 1949, after the development of a procedure to measure carbon-14, the first carbon date was taken of a piece of wood from the Tomb of Egyptian King Zoser, known to have an age (in 1946) of 4650 ± 75 years. Libby's measured carbon-14 age was 3979 ± 350 years. A year later, Libby published his first "Curve of Knowns" relating historical age and percentage of carbon-14 left in a sample using seven data points of known age (Taylor et al. [2014]).

Throughout the 1950s, carbon dating continued to gain traction. Between 1949 and 1954, over 500 samples with unknown ages were carbon dated, most of them of archaeological significance. In 1953, a second, more widely distributed "Curve of Knowns" was published which incorporated three additional data points and a different half-life value for carbon-14. During this period, universities began to set up their own labs to measure carbon-14, and in 1959, the journal *Radiocarbon* was established. The acclaim of carbon dating reached its peak in 1960, when Libby won the Nobel Prize in chemistry for his work on carbon dating (Taylor et al. [2014]). In his Nobel lecture, Libby does not mention any problem in calibration. In fact, he discusses his second "Curve of Knowns" and states "the agreement with the predicted radiocarbon content seems to be satisfactory" despite the majority of samples older than 2000 years (in 1953) having a higher carbon-14 content than expected (Libby [1960]).

As early as 1958 there were concerns that carbon dates did not match up with the true ages of samples (Stuiver & Suess [1966]). In 1966, research performed at the University of Arizona was published claiming that the levels of carbon-14 in the atmosphere have fluctuated over time. In the paper, carbon-14 dates were taken from tree rings that had previously been dendrochronologically dated and found fluctuations in the carbon-14 to carbon-12 ratio over the past 6000 years (Damon et al. [1966]). Later that same year, another paper was published which argued convincingly that carbon ages are not calendar ages and included a primitive calibration table with carbon dates spanning 1000 years (Stuiver & Suess [1966]).

Conventional Carbon Dates to True Ages

Today, conventional carbon dates are calculated using the (inaccurate) half-life of 5570 years, the value used by Libby in his second "Curve of Knowns." Even though it is inaccurate, the value was so widely used in the early days of carbon dating that scientists decided it was better to continue using it rather than a more accurate value.

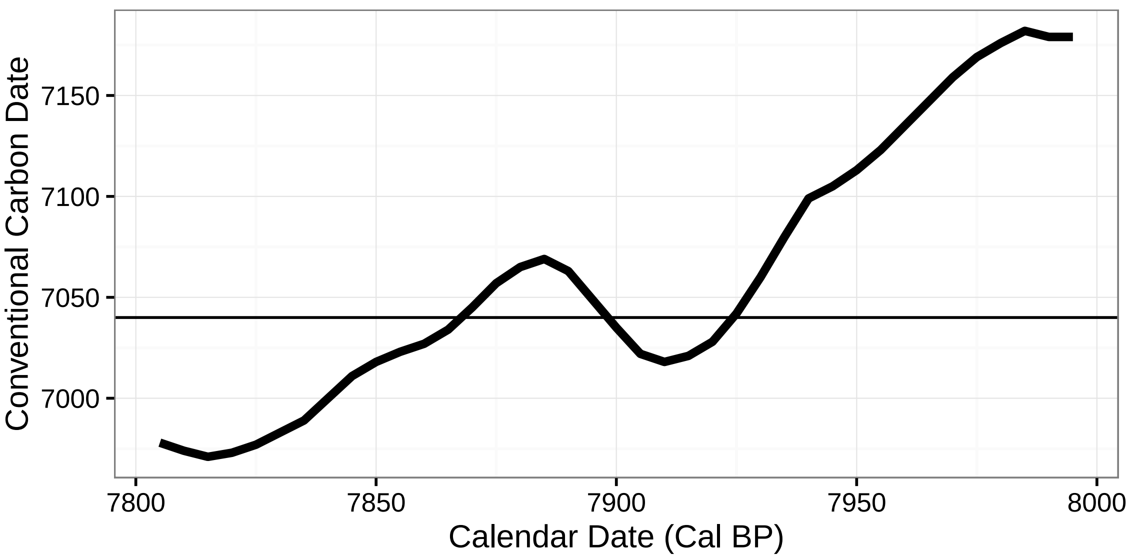

Figure 1.2: A segment of the calibration curve shown with three possible ages for a single carbon date.

Additionally, all conventional carbon dates are calculated by comparing the amount of carbon-14 in a sample to the amount in the reference standard, the reference standard being 0.95 times the activity in oxalic acid derived from sugar beets grown in France in 1950 (Hedman [2007]).

Using the calibration curve, we can calculate a probability distribution on calendar years for a single conventional carbon date. However, scientists often have information about their samples beyond the carbon dates measured in a lab, which can help to narrow the distribution of possible ages. This thesis deals with getting more accurate approximations of age by incorporating this additional information into the data analysis. Specifically, it tackles the problem where multiple samples are found at a single site with some samples found above or below others in the sediment, where samples found nearer to the surface of the earth are assumed to be younger compared to samples found deeper in the sediment. This analysis was originally suggested by Buck et al. [1992].

Using a Bayesian analysis with the temporal ordering of multiple samples as a prior condition, this thesis uses a Markov Chain Monte Carlo method to come up with age approximations for all samples, including one sample for which age is determined by temporal ordering alone.

Chapter 2

Bayesian Inference

2.1 Bayes Theorem

Imagine we are trying to estimate the probability of an outcome A. In a frequentist framework, there is some parameter or set of parameters which control this outcome. Data is collected and analyzed with the goal of estimating these parameters. Once estimates are found, they can be used to estimate the probability of outcome A.

Bayesian analysis works differently. A Bayesian begins the analysis with a prior probability for the outcome A. After data is collected, the Bayesian uses the data to update their belief about the probability of outcome A. Thus it is the data together with outside knowledge that was used to estimate the probability of outcome A.

This process of using both outside knowledge and data can be stated mathematically as Bayes theorem. Let B be an event and let $A_1, A_2, \ldots , A_n$ be disjoint outcomes which form a partition of the sample space. Then

$$P(A_i|B) = \frac{P(B|A_i)P(A_i)}{P(B)} = \frac{P(B|A_i)P(A_i)}{\sum_{i=1}^{n} P(B|A_i)P(A_i)}$$

where $P(A_i)$ is the prior probability of outcome A_i, $P(B|A_i)$ is the probability of event B occurring given that outcome A_i has occurred, $\sum_{i=1}^{n} P(B|A_i)P(A_i)$ is the total probability of event B occurring (it sums to the total probability of event B occurring). The result is the updated probability of outcome A_i given event B, $P(A_i|B)$, which is known as the posterior probability of event A_i.

Generalizing to Probability Densities

In this analysis, we are trying to determine the age of a sample (a single fossil or artifact) based on the its carbon date(s), the calibration curve, and its position relative to other samples. To understand how this fits into a Bayesian framework, we will describe Bayes theorem for probability densities.

Let x be the data that was gathered, and let θ be the parameters of the distribution we believe the data was generated from where either x or θ may be vectors. Let $\pi(\theta)$ be the prior distribution, the distribution we think our parameters have before incorporating the data. Then $f(x|\theta)$ is the likelihood function (which should be thought of as a function of θ, not x), and Bayes theorem for probability densities states

$$\pi(\theta|x) = \frac{f(x|\theta)\pi(\theta)}{\int f(x|\theta)\pi(\theta)d\theta}.$$

Bayes theorem yields the posterior distribution $\pi(\theta|x)$ which as before incorporates both the data and our prior beliefs about the parameters to produce an updated belief about θ. For every possible value of θ, the joint distribution $f(x|\theta)\pi(\theta)$ measures how well the value θ fits the data we have collected (Robert & Casella [1999]). The denominator is a normalizing constant which produces a posterior distribution that integrates to one.

2.2 Prior Information

A Bayesian analysis relies on information in addition to data in order to perform well. We will now discuss the information incorporated in this analysis.

Calibration Curve

The main source of prior information here is the calibration curve. Recall from Chapter 1 that a conventional carbon date is the age of a sample (in Cal BP) given a constant ratio of carbon-12 to carbon-14 in the atmosphere (a false assumption). The calibration curve allows us to construct a function roughly relating our conventional carbon dates to calendar years. A portion of the calibration curve is reproduced in Figure 2.1.

The calibration curve is reported as a data set including the variables: calibrated calendar year (Cal BP), conventional carbon-14 date, and standard deviation. The

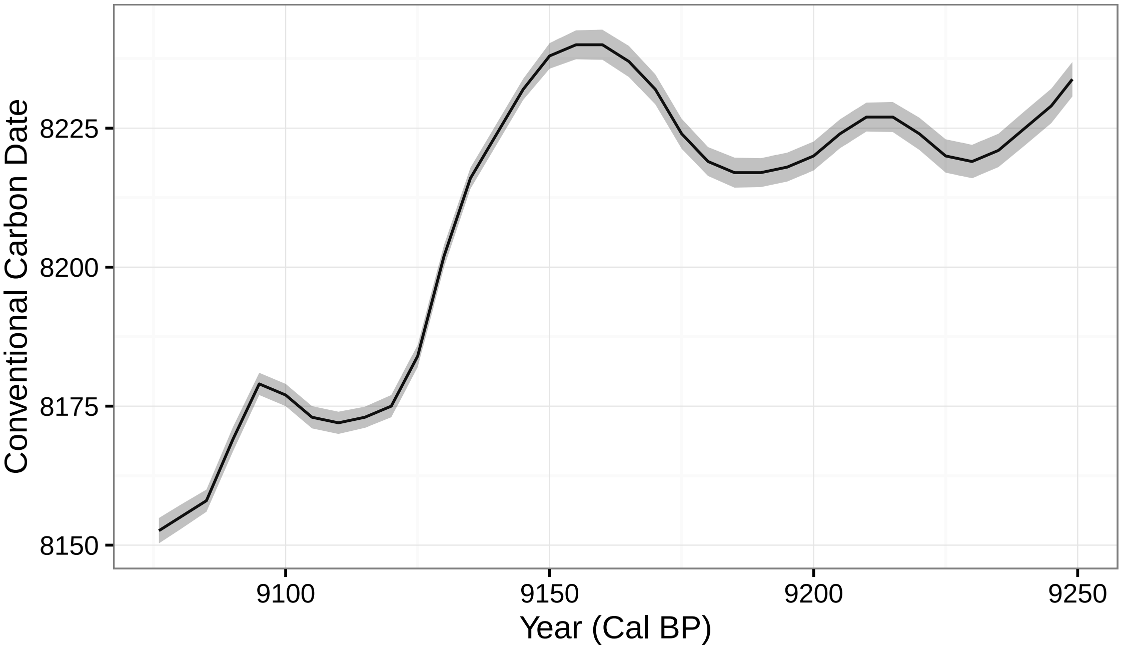

Figure 2.1: A segment of the calibration curve shown with ±1 standard deviation.

idea is that for a given calendar year, there is an expected value for the conventional carbon date. However, there is some uncertainty in the calibration curve itself, so the standard deviation in the conventional carbon dates is also reported. Thus, the calibration curve can be modeled as a series of normal distributions where for each year BP in the data set, a normal distribution is given with a mean conventional carbon date and standard deviation (Reimer et al. [2013]).

Years BP are not reported for every consecutive number between 0 and 50,000, and the gaps between years are not evenly spaced throughout the 50,000 year time span. Rather, as accurate data for different time periods became more difficult for the creators of calibration curve to find, the intervals between years BP reported become wider. Thus for the first few thousand years BP, values for conventional carbon dates and standard deviations are reported at intervals of 5 years BP; however, for the oldest years BP, these values are reported at 100 year intervals. Additionally, the reported values for standard deviation increase as years BP increase. In fact, the standard deviations are an order of magnitude larger for the oldest years BP compared with the youngest. This is all to say that we expect dating older samples to be less precise than dating more recent ones.

Having a value for every year BP makes estimating the ages of the samples simpler. Here we use linear interpolation to estimate the calibration curve for calendar

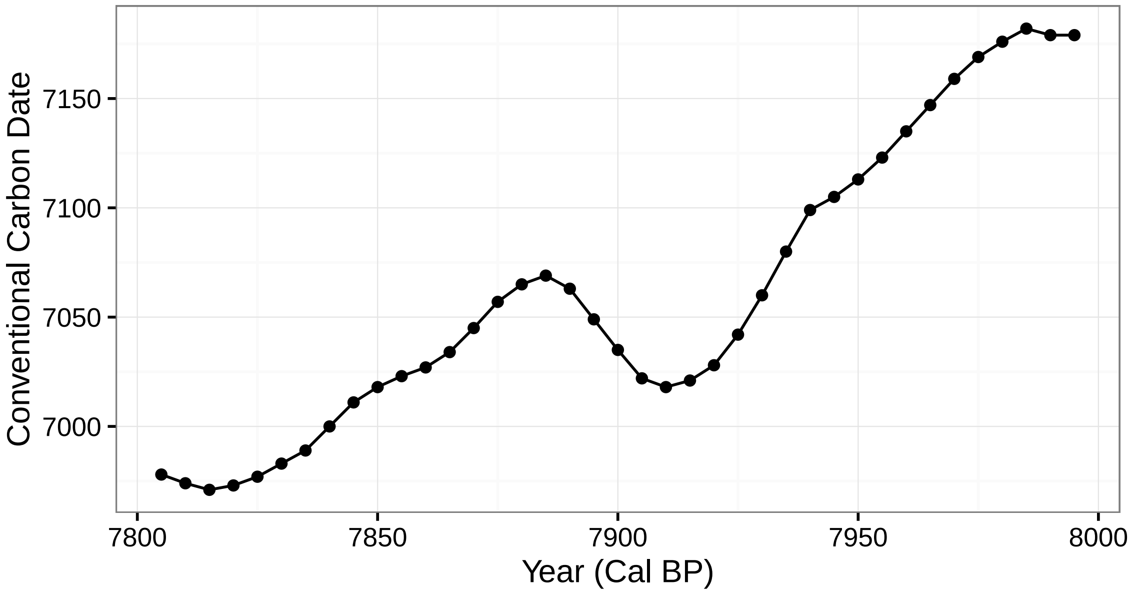

Figure 2.2: Linear interpolation of a small segment of calibration curve

years between 0 and 50,000 years BP that are not included in the data set. Linear interpolation is the simplest method for doing this, and can be thought of as "connecting" the data points. An example of linear interpolation can be seen in Figure 2.2.

Temporal Order Constraints

The other important piece of prior knowledge we have in this analysis is the depth at which each sample was found, relative to those of other samples. For example, suppose we have three samples, $\alpha_1, \alpha_2, \alpha_3$ where each sample is found in a deeper stratum than the previous one, as shown in Figure 2.3.

Then we can add the following constraint to our age BP estimate for each sample:

$$0 < \alpha_1 < \alpha_2 < \alpha_3 < 50000$$

This ordering will be enforced even if the associated conventional carbon date values do not follow this pattern. In effect, this constraint narrows the posterior distribution and increases the probability density within the interval. However, note that within the acceptable range for a given sample (i.e. $[0, \alpha_2]$ for α_1), the probabilities remain the same relative to each other since we have no additional knowledge about these dates to make relative changes to the prior on this interval (Buck et al. [1992]). We

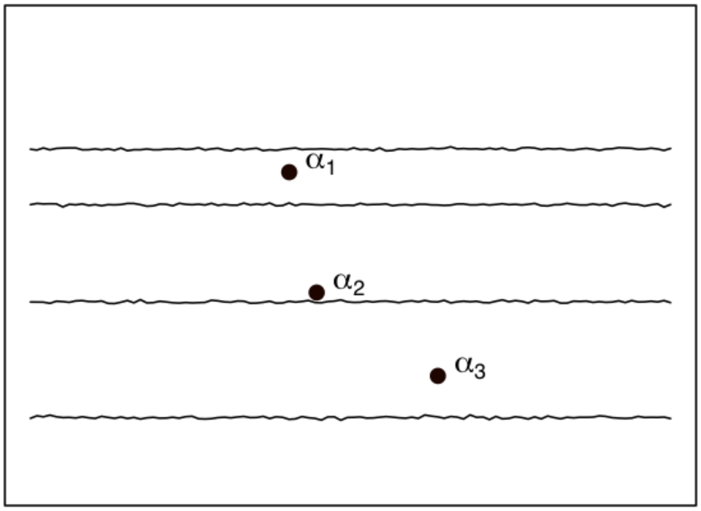

Figure 2.3: Three samples from the same site. Each sample is found in a different layer of sediment, allowing us to infer their relative ages.

call this a uniform prior.

Additionally, a single sample might yield multiple carbon date values. If there is no order or structure to these multiple values, then constraining all carbon dates to a single interval is probably appropriate. Sometimes though, there is structure to these multiple values, for instance tree ring samples. Trees grow new rings seasonally, and each of the rings captures the carbon-14 level when the ring was formed. Using dendrochronology, we can study these rings and estimate the number of years between rings. We can then use these multiple carbon dates with known temporal offsets to better estimate the age of the sample.

Finally, sometimes we want to find the age of samples which were not previously living, such as the tephra sample dated in this analysis. In this case, we rely entirely on the temporal order constraints to arrive at age estimates.

Chapter 3

Finding the Posterior

In order to estimate the age of a sample, we have to evaluate the posterior distribution. Recall from chapter 2 the equation for the posterior distribution:

$$\pi(\theta|x) = \frac{f(x|\theta)\pi(\theta)}{\int f(x|\theta)\pi(\theta)d\theta}$$

where x and θ may be vectors. Let θ be the true age for the sample x, and let $c(\theta)$ be the approximated ages from the carbon date by the calibration curve. Let σ^2 be the estimated standard error of the sample. Then the likelihood function $f(x|\theta)$ is defined as the density of the sample x in a normal distribution with mean $c(\theta)$ and variance σ^2.

In order to use this equation, we need a method for evaluating the integral in the denominator of this equation. For the calibration problem, however, the integral cannot be evaluated analytically, since the function relating carbon dates to ages is unknown; our calibration curve is only an approximation of this function. Instead we must find a way to estimate the integral, or directly estimate the posterior distribution.

3.1 Numerical Integration

The most straightforward method for approximating the integral is numerical integration. Let the interval $[a, b]$ be the range of years over which we are integrating. Then we can approximate this integral using boxes under the curve, which is essentially Riemann sums. In the rectangle method, we break the interval into subintervals, find the value of the function at some point in each subinterval, multiply these values by the lengths of each of the subintervals, and then add all of these values together.

Then the rectangle method can be written as

$$\int_a^b f(x|\theta)\pi(\theta)d\theta \approx \sum_{i=a}^{b} f(x|\theta_i)\pi(\theta_i)\Delta\theta_i$$

where $\Delta\theta_i$ is the length of the subintervals (Robert & Casella [1999]). For this application, we use $\Delta\theta_i = 1$ since we have an interpolated value for the calibration curve at intervals of one calendar year.

This is simple to compute for a single sample, where a sample is either a single carbon date or a set of carbon dates with gaps of known length between them, such as a tree with dates from multiple rings. However, we want to be able to evaluate posterior distributions where the bounds of integration for one sample are dependent on the ages of its neighbors, as is the case with samples with unknown temporal offsets. For a given set of ages $\theta_1, \ldots, \theta_m$, the likelihood that these are the true ages for a set of $x_1, \ldots, x_m$ carbon dates is the product of the likelihoods for the individual ages

$$f(x_1, \ldots, x_m|\theta_1, \ldots, \theta_m) = f(x_1|\theta_1) \times \cdots \times f(x_m|\theta_m)$$

since we can assume that the ages of the samples are independent of each other.

Numerical integration works well when the number of samples with known ordering but unknown temporal offset is small. However, when there are more than two samples, doing this calculation is slow.

For a single carbon date, there is one calculation to be done for every possible year the sample could originate from. When there are two samples, one calculation has to be done for every pair of years the two samples could originate from. Similarly for every possible combination of three samples, one calculation is required for every triple of possible years. This trend continues as we evaluate more samples. The result is an polynomial increase in the number of calculations required for a linear increase in the number of samples ($O(n^d)$ where d is the dimension of the array we are creating equal to the number of samples), resulting in longer running times to perform the computations and requiring more memory. Beyond two samples, numerical integration is not an efficient method for finding the posterior since creating arrays with more than two dimensions requires significant memory.

3.2 Markov Chain Monte Carlo

In higher dimensions with unknown offset, it is computationally more efficient to simulate the posterior distribution rather than evaluate it explicitly. One way to simulate the posterior is to use Markov chain Monte Carlo methods. Here we specifically use the Metropolis-Hastings algorithm. We will start by examining the simulation of the posterior distribution for a single sample.

Let Θ_i be a random variable, and let θ_i be a single draw from the random variable Θ_i. A *Markov chain* is a sequence $\Theta_1, \Theta_2, \ldots$ of random variables where the conditional distribution of Θ_{n+1} given $\theta_1, \ldots, \theta_n$ depends only on θ_n. A sequence of random variables $\Theta_1, \Theta_2, \ldots$ is *stationary* if for every positive integer k, the distribution of the k-tuple $(\theta_{n+1}, \ldots, \theta_{n+k})$ does not depend on n (Robert & Casella [1999]).

The goal of Markov chain Monte Carlo is to generate a sequence of values $\theta_1, \theta_2, \theta_3, \ldots$ from a Markov chain such that the stationary distribution of the Markov chain is the posterior distribution. Thus, these methods take an initial value of the parameter being estimated and use some update mechanism to generate the next value in the sequence. This process is then repeated many times (millions is not uncommon) until the stationary distribution is thoroughly explored.

Specific Markov Chain Monte Carlo algorithms implement different methods for moving to from one position to the next. The basic update mechanism for the algorithm used here, Metropolis-Hastings, is below:

Metropolis-Hastings Update

Let h be the (not necessarily normalized) probability density of the stationary distribution we are constructing.

1. While at position θ_i, propose a move to position ϕ having probability density conditioned on θ_i, written as $q(\phi|\theta_i)$.

2. Let $h(\theta)$ be the density at the sample when h is parameterized by θ. Compute the Hastings ratio:

$$r(\theta_i, \phi) = \frac{h(\phi)q(\phi|\theta_i)}{h(\theta_i)q(\theta_i|\phi)}.$$

3. Accept the proposed move with probability $a(\theta_i, \phi) = \min(1, r(\theta_i, \phi)$. In other words,

$$\theta_{i+1} = \begin{cases} \phi, & \text{with probability } a(\theta_i, \phi) \\ \theta_i, & \text{with probability } 1 - a(\theta_i, \phi). \end{cases}$$

With this update, if the proposed move is to a point of higher density, we will move to the new position. If the proposed move is to a point of lower density, we will move to that point with probability $a(\theta_i, \phi) \leq 1$ and stay at the current position with probability $1 - a(\theta_i, \phi)$ (Brooks et al. [2011]).

Another commonly used Markov Chain Monte Carlo algorithm is the Gibbs sampler. Let $\Theta = (\Theta_1, \dots, \Theta_p)$ be a random variable with p components. Suppose we can simulate from corresponding conditional densities $f_1, \dots, f_p$ where

$$\Theta_i | \theta_1, \dots, \theta_{i-1}, \theta_{i+1}, \dots, \theta_p \sim f_i(\theta_i | \theta_1, \dots, \theta_{i-1}, \theta_{i+1}, \dots, \theta_p).$$

Then we can use the Gibbs sampler:

Gibbs Update

Begin with the Markov chain at state t, written as $\theta^t = (\theta_1^t, \dots, \theta_p^t)$.

1. Generate $\Theta_1^{t+1} \sim f_1(\theta_1 | \theta_2^t, \dots, \theta_p^t)$.
2. Generate $\Theta_2^{t+1} \sim f_2(\theta_2 | \theta_1^{t+1}, \theta_3^t, \dots, \theta_p^t)$.

   $\vdots$

p. Generate $\Theta_p^{t+1} \sim f_p(\theta_p | \theta_1^{t+1}, \dots, \theta_{p-1}^{t+1})$.

The Gibbs Sampler differs from Metropolis-Hastings in accepting every proposed move; there is no accept/reject step. While this algorithm is simpler than Metropolis-Hastings, Gibbs sampling cannot always be used as it requires being able to sample directly from these conditional distributions, which are not always available (Robert & Casella [1999]). However, the Gibbs sampler is actually a special case of Metropolis-Hastings where the acceptance probability is equal to 1, as we will now show.

Let $\theta = (u, v)$ where u is the component that is being updated, such that we are updating v conditional on u. The unnormalized density $h(\theta) = h(u, v)$ can be factored as $h(u, v) = g(v)q(v, u)$ where $q(v, u)$ is the normalized conditional distribution of u given v, and $g(v)$ is an unnormalized marginal distribution of v. Then the proposal distribution is q as before and the proposed move is to $\phi = (u^*, v)$. The Hastings ratio is

$$r(\theta, \phi) = \frac{h(u^*, v)q(u, v)}{h(u, v)q(v, u^*)} = \frac{g(v)q(v, u^*)q(u, v)}{g(v)q(v, u)q(v, u^*)} = 1$$

such that the proposed move is always accepted. We've shown this holds with two components, but this could be extended to p components by replacing v in the equation above with some $p-1$ other variables that will be held constant while u is updated (Robert & Casella [1999]). Thus, Gibbs sampling can be reduced to Metropolis-Hastings.

There are many other variations on the Metropolis-Hastings and Gibbs samplings algorithms, including Metropolis-within-Gibbs, where each component of the random variable Θ is updated individually using the Metropolis-Hastings update. Another is Block Gibbs, where multiple components of Θ are updated simultaneously using the Gibbs update (Brooks et al. [2011]).

Compared to the general Metropolis-Hastings algorithm, there are fewer parameters to select in the Gibbs sampler. When using the Metropolis-Hastings algorithm, there is the choice of the proposal density, and along with it, obtaining a reasonable rejection-rate, the rate at which proposed moves are rejected. With Gibbs sampling, the proposal distribution is already selected and without an accept/reject stage, the rejection-rate is 0. However, it is not always feasible to use Gibbs sampling since full conditionals are not always available.

Symmetry of Normal Distribution

The Hastings ratio requires us to compute the conditional density of moving both from θ_i to ϕ and from ϕ to θ_i, where θ_i is the current position of the Markov chain and y is the proposed move. However, all of the proposal distributions we are using are normal, and thus symmetric. Let σ be the standard error of the sample in question, and let Θ_i and N be random variables distributed

$$\begin{aligned} \Theta_i &\sim Normal(x_i, \sigma^2) \\ \Phi &\sim Normal(y, \sigma^2). \end{aligned}$$

These are the distributions we are constructing to measure the probability of moving between positions in the Markov chain. Since these are normal distributions, the densities are equal:

$$f(\theta_i|\eta,\sigma) = \frac{1}{\sigma\sqrt{2\pi}}e^{-\frac{(\theta_i-\eta)^2}{2\sigma^2}} = \frac{1}{\sigma\sqrt{2\pi}}e^{-\frac{(\eta-\theta_i)^2}{2\sigma^2}} = f(\eta|\theta_i,\sigma).$$

Thus the probabilities of moving in either direction are identical, and we can ignore the conditional density term when calculating the Hastings ratio since

$$\frac{f(\eta|\theta_i)}{f(\theta_i|\eta)} = 1.$$

Considerations in Implementing Metropolis-Hastings

In order to implement the Metropolis-Hastings algorithm, we need to select the proposal distribution, the distribution from which we select a proposed move at each iteration. Since in this case our calibration curve is modeled as a series of normal distributions, it is convenient for our proposal densities to be normal distributions as well, though in general proposal distributions do not need to be related to the likelihood function. Thus, the proposal distributions will be normal with mean at the current state of the Markov chain. Selecting the variance for this normal distribution is less obvious.

There are a couple of considerations when selecting the variance. If the variance is too small, then the Markov chain will be slow to mix and more iterations will be necessary to reach the stationary distribution. On the other hand, if the variance is too large, then more proposed moves will be rejected and the Markov chain will again be slow to mix. Thus the variance needs to be somewhere between these extremes. Here the variance was chosen by examining plots of Markov chain values against iterations to see how well the chain is mixing, as well as by considering the rejection rate for the chain overall (Brooks et al. [2011]).

Another consideration is the starting value for the Markov chain. The starting value must of course adhere to the conditions we set for each subsequent step of the Markov chain. For instance, if we require the random variables Θ_1, Θ_2 to be strictly ordered such that $\theta_1^i < \theta_2^i$ for all i, then the starting points should also obey this condition, $\theta_1^0 < \theta_2^0$, in order for the starting points to be in the state space of the Markov chain (the set of values we allow the Markov chain to contain). However, since we do not know what the stationary distribution is when we begin, the starting points are not generally considered part of the stationary distribution. Instead, we start the Markov chain at any point and expect to eventually reach the stationary distribution.

After we have an initial value for the Markov chain and a proposal distribution, we have to decide how long to let the algorithm run. The more trials we simulate, the more likely we are to have reached and more thoroughly explored the stationary distribution, and the more confidence we have in our results. On the other hand, runs with larger numbers of trials take longer to generate. We can decide whether or not a simulation has enough trials by examining how well the Markov chain has explored the sample space. This can be done by looking at plots of the values in the simulation versus trial number to try to determine if part of the state space has not been adequately explored. Ultimately though, it is impossible to know for certain whether or not the simulation has adequately explored the state space.

Analyzing the Markov Chain

In order to analyze values that were generated from the stationary distribution (since these are the values we are interested in), a number of iterations are commonly thrown away at the beginning of a simulation in a process known as burn-in. In practice, it is difficult to impossible to know whether or not the stationary distribution has been reached, so the number of iterations to throw out is determined by looking at statistics computed on the simulation.

Another common practice applied to Markov chains before analysis is performed is called thinning. Since each value in the chain is generated from a distribution centered at the previous value, there is correlation between subsequent values. If the chain is slow to mix, then it is likely the correlation between subsequent values, or autocorrelation, within the chain, is very high. In this case, instead of using every value, only every k^{th} value is sampled to estimate the posterior distribution, with k determined by the severity of the autocorrelation. By removing much of the autocorrelation within the Markov chain, we can assume that we have approximately independent samples. Additionally, if the chain has high autocorrelation, then not much information is lost by using only every k^{th} value. In removing the majority of values in the Markov chain, we may lose precision in our estimation of the posterior distribution, but we do simplify the analysis by having approximately independent samples (Brooks et al. [2011]).

After performing thinning and burn-in as required, we will be left with samples from a Markov chain that are distributed according to the desired stationary distribution. Thus we need to estimate the probability density function of the unknown parameters. Since our variable of interest is discrete, we will do this by counting the number of times a particular value appears in our Markov chain, and then dividing by the sample size to normalize (we want the sum of all the probability values to be one).

Let $q_1, q_2, \ldots, q_n$ be the sequence generated after burn-in and thinning are performed, and let Θ be the set of possible parameter estimates. Then the density of $\theta \in \Theta$ is

$$P(\theta) = \frac{\sum_{i=1}^{n} \mathbb{1}_\theta(q_i)}{n}$$

where $\mathbb{1}_\theta$ is the indicator function defined as

$$\mathbb{1}_\theta(q_i) = \begin{cases} 1, & \text{if } q_i = \theta \\ 0, & \text{otherwise.} \end{cases}$$

This is our posterior distribution.

Credible Regions

To more precisely describe our estimates for θ, we use credible regions, which are the Bayesian equivalent of frequentist confidence intervals. Like their frequentist counterparts, credible regions are a set of parameter values that describe the uncertainty of our estimates. However, confidence intervals are random intervals which, under repeated trials, have some specified probability of covering the true value, since in the frequentist view the true value as fixed. A Bayesian frames the parameter as a random variable (which is thus not fixed) and the Bayesian credible region captures uncertainty in the parameter as a probability distribution.

The credible region will be the region with the highest posterior density, also called a highest posterior density or HPD region. To find this region, we want a union of intervals which contains some percentage of the posterior density, typically 95%. Furthermore, we want this union of intervals to be as narrow (contain as little of the parameter space) as possible. An example of a credible region is shown in Figure 3.1.

In the case where the HPD region is contained in a single interval $[a, b]$, we can state these conditions as the following minimization problem, where P is the desired amount of density contained in the interval:

$$\text{Choose } (a, b) \text{ to minimize: } b - a$$
$$\text{Subject to: } \int_a^b f(x|\theta)d\theta = P$$

More generally for unions of intervals, we want the sum of the integrals over each of the intervals to sum to P, subject to the sum of the lengths of the intervals being minimized.

More simply, the problem of finding a credible region can be thought of as placing a horizontal line through the density curve and including regions where the density is higher than the line in the credible interval and excluding all other values. Then we can increase or decrease the density contained in the credible interval by moving this horizontal line lower or higher (respectively) on the posterior density until the desired amount of probability is contained in the interval.

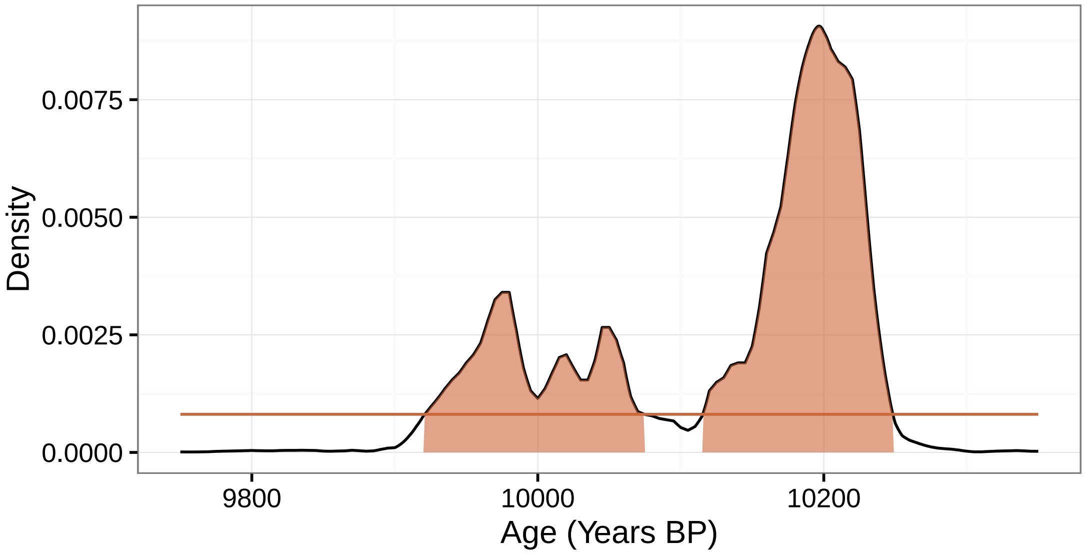

Figure 3.1: An example of a posterior distribution with a non-contiguous 95% credible region.

3.3 Multiple Samples with Temporal Ordering

Now that we have discussed basic Markov chain Monte Carlo methods, let's examine how they apply to our problem of ordered samples with unknown temporal offsets. Recall that we are modeling these m samples as

$$0 < \theta_1 < \theta_2 < \cdots < \theta_m < 50000$$

where θ_j is the age of sample j and larger subscripts indicate (older) samples found in deeper strata.

We can use this knowledge to better inform the proposal density for each of our samples. Let θ_j^i be the i-th estimate for the j-th sample's age and let x_j be the radiocarbon date of the j-th sample. Then for each iteration i and sample j, we restrict θ_j^i by

$$f(x_j|\theta_{j-1}^{i-1}, \theta_{j+1}^{i-1}) = \begin{cases} N(\theta_j^{i-1}, \sigma^2), & \text{if } \frac{\theta_j^{i-1} - \theta_{j-1}^{i-1}}{2} < \theta_j^i < \frac{\theta_{j+1}^{i-1} - \theta_j^{i-1}}{2} \\ 0, & \text{otherwise} \end{cases}$$

where $N(p, q)$ is a normal distribution with mean p and variance q (Buck et al. [1992]).

In other words, we will require that our i-th estimate for sample x_j be between our previous estimates for x_{j-1} and x_{j+1}, specifically between the midpoints of these neighboring points and the previous estimate for x_j. Thus within every iteration i, the components of the Markov chain will conform to the condition $\theta_1^i < \ldots < \theta_m^i$. Note that this condition does not necessarily hold when comparing θ values between iterations.

We are restricting the values θ_j^i can take based only on values from the previous iteration $i - 1$ rather than using θ_{j-1}^i since it is possible for $\theta_{j-1}^i > \theta_{j+1}^{i-1}$, leaving no possible values for θ_j^i to take on. With this restriction set, every value of θ_j^i is estimated using the Metropolis-Hastings update with θ_j^{i-1} as the initial value and moving to θ_j^i. After constructing the Markov chain, each component can be analyzed individually since they were conditioned on all other components while being generated.

Dating Samples without Carbon Dates

Some artifacts were not previously living and thus cannot be carbon dated. However, we can still use carbon-14 data from nearby samples to estimate their age.

Suppose we have a sample θ_k which cannot be carbon dated but we have carbon data for nearby samples with the following temporal ordering on their ages:

$$0 < \theta_1 < \ldots < \theta_k < \ldots < \theta_m < 50000$$

Since this sample does not have a carbon date, we will z_k to represent the known data we do have about it, namely its location in the strata. Then the proposal density for each iteration i and this sample k is given by

$$f(z_k|\theta^i_{k-1}, \theta^i_{k+1}) = \begin{cases} c, & \text{if } \theta^i_{k-1} < \theta^i_k < \theta^i_{k+1} \\ 0, & \text{otherwise} \end{cases}$$

where c is some constant.

Note that we are restricting the values θ_k can take on only by trials in the current iteration. Thus in a single iteration, we simulate ages for all of the samples that have carbon dates before simulating the age of samples without a carbon date.

Chapter 4

Application to R tephra Data

4.1 The Data

The data for this thesis comes from the Mount Rainier National Park in Washington state and is an attempt to date a tephra deposit found in the park known as the R tephra. Tephra are the ash and rock fragments produced by volcanoes while they are erupting. There several tephra deposits found within the park. The R tephra in particular is the oldest of many tephra associated with Mount Rainier during the Holocene era (the current geological epoch). Since it is the oldest tephra in an era, it is useful to geologists who can use it to roughly gauge the boundary between the Pleistocene era, which ended approximately 11,700 years ago, and Holocene eras in the region near Mount Rainier, since the R tephra was produced by an eruption near the end of the post-glacial period (Samolczyk et al. [2016]).

Tephra cannot be carbon-dated directly. Since tephra are essentially volcanic ash, they do not have a carbon-14 to carbon-12 ratio in equilibrium with the atmosphere when first formed, as they are neither living nor produced by a living thing. Instead, tephra are dated by measuring the carbon ages of samples near the tephra. This dataset includes five samples, two above and three below the R tephra (above means more recent). The data is summarized below in Table 4.1. Note that the carbon-14 dates do not adhere to the ordering implied by the stratigraphic evidence.

4.2 Comparison of Numerical Integration and MCMC

To begin with, we will check to ensure that the Markov Chain Monte Carlo method yields results comparable to the those given by numerical integration. This test is

Table 4.1: The data in the R tephra dataset (Samolczyk et al. [2016]).

Sample Number	Description	Carbon-14 age (years BP)	Location relative to R tephra
1	Coniferous needle	8905±20	20.5cm above
2	Twig	8920±60	directly above
3	Charcoal	8890±40	directly under
4	Peat	8760±80	50cm below
5	Small twig	8990±60	100cm+ below

performed on a single sample with a uniform prior. The sample used is the coniferous needle sample with carbon-14 age 8905 ± 20 years BP.

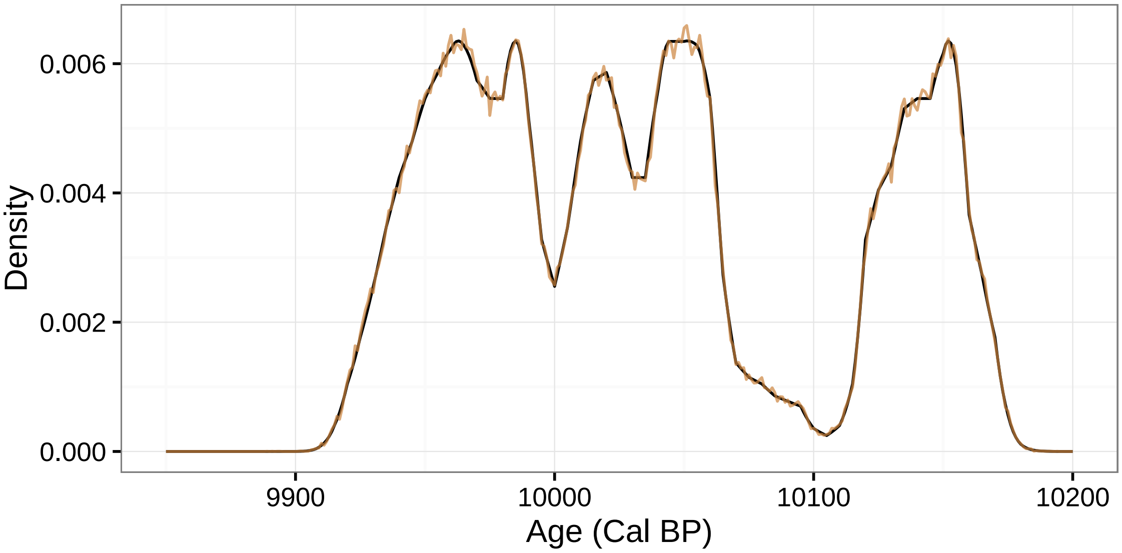

Figure 4.1: The posterior density for the age of the coniferous needle sample. The tan line was calculated using Markov chain Monte Carlo, and the black line with numerical integration.

The results of the two methods match closely, as can be seen in Figure 4.1. Moreover, the 95% credible intervals are nearly identical: numerical integration resulted in the credible interval of $(9923, 10069) \cup (10117, 10171)$ while Markov chain Monte Carlo resulted in a credible interval of $(9923, 10069) \cup (10117, 10170)$. The credible intervals are shown as shaded regions in Figure 4.3.

The Markov chain Monte Carlo results were obtained from a simulation of two million trials. The proposal distribution was normal with mean at the current position and variance 1600. The rejection rate was approximately 33%. The initial value for

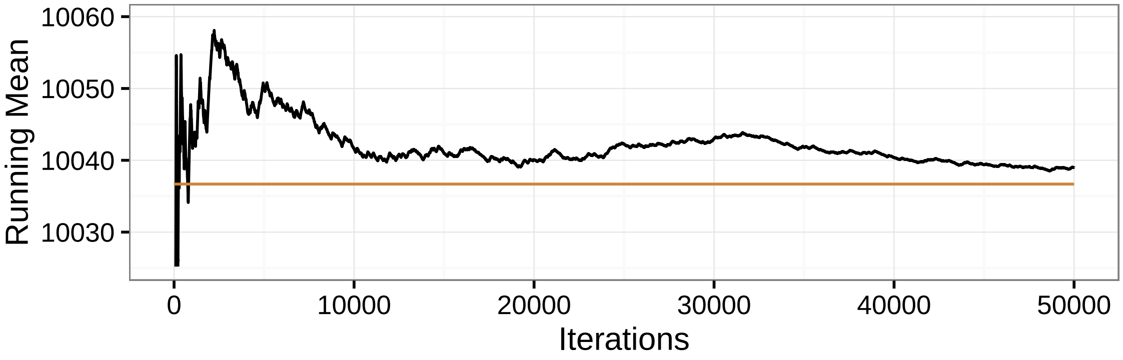

Figure 4.2: The mean value of the Markov chain plotted against the number of elements in the chain. The tan line shows the mean for the entire Markov chain.

the chain was taken by selecting at random a year with an associated carbon age within 170 years of the sample carbon age. The first 5000 elements of the Markov chain were discarded as burn-in and every 5th value in the simulation was sampled. The running average for the first several thousand iterations is shown in Figure 4.2.

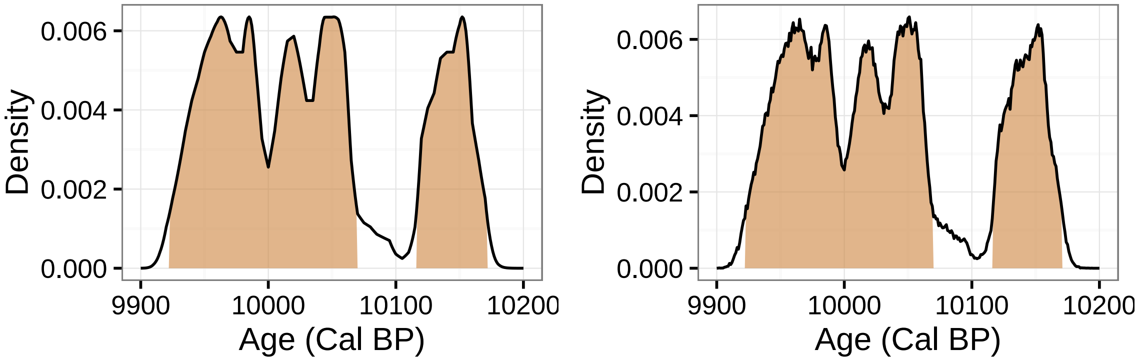

Figure 4.3: Densities of sample shown with 95% credible interval. Left density is numerical integration and right is Markov chain Monte Carlo.

This analysis is evidence that Markov chain Monte Carlo gives similar results to numerical integration in one dimension, so we know Markov chain Monte Carlo performs as anticipated. Now we will move onto to the higher dimensional problem.

4.3 Stratigraphically Ordered Samples

Now we will apply this Markov chain Monte Carlo method to the five temporally ordered samples. After running five million trials, the results are shown in Figures 4.4 and 4.7.

As is shown in Figure 4.4, the ordered prior greatly affects the posterior distribution in this instance. We are able to narrow the posterior distributions by constraining the ages we allow each sample by their stratigraphic order. This is especially evident in the case of Sample 3, where the inclusion of this prior shifted most of the posterior density to older ages, where the posterior density had previously been very low.

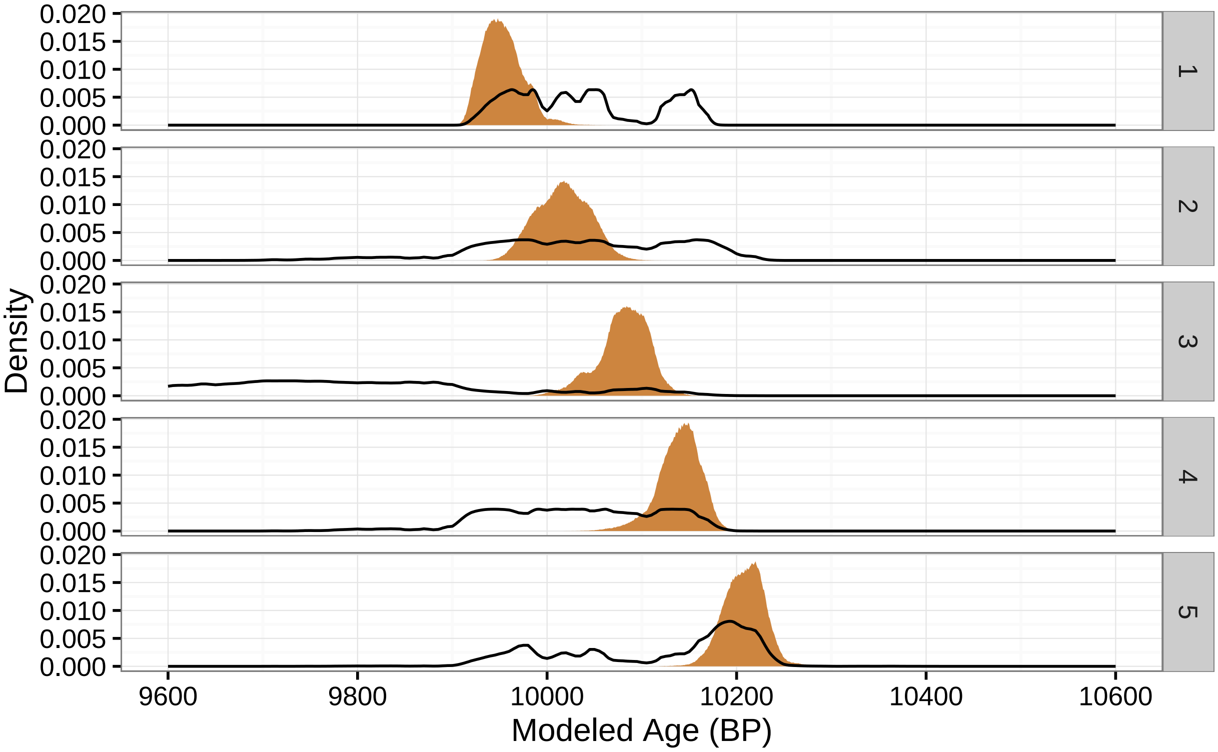

Figure 4.4: The results of the analysis of the five samples. The tan filled densities are the posterior densities after MCMC analysis with the ordered prior. The line plots are densities based on individual numerical integrations without the ordered prior.

The posterior distributions were generated by a Markov chain of five million iterations where the first 15000 iterations were discarded as burn-in. The running mean over the first 20000 values of the Markov chain are shown in Figure 4.5. After burn-in, the chain was then thinned with $k = 15$, such that only every 15th value was kept and used for analysis. Figure 4.6 shows the effect of thinning on the autocorrelation for two of the samples. The effect was similar for the other three samples. The variance of the proposal distribution was 1600. The rejection rate for the Markov chain was about 20%.

Credible regions containing 95% of the posterior density as well as the mean and median age for each sample are reported in Table 4.2 and Figure 4.7. All of these 95% HPD regions are unimodal and more narrow than those produced without the

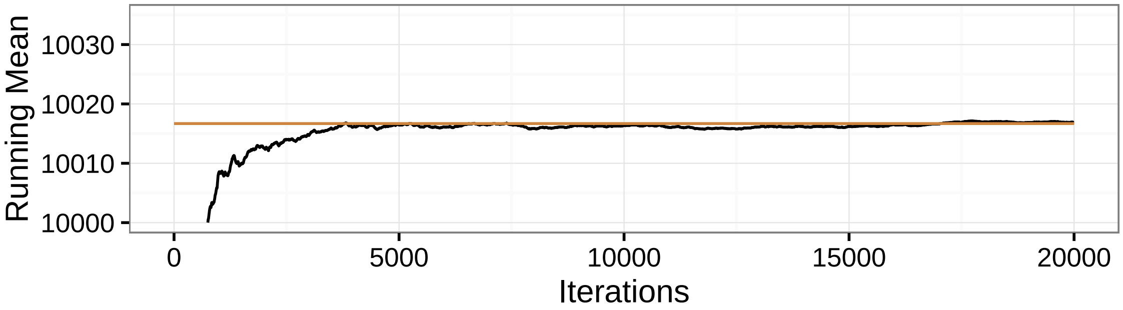

Figure 4.5: The running mean over the first 20000 iterations of the Markov chain.

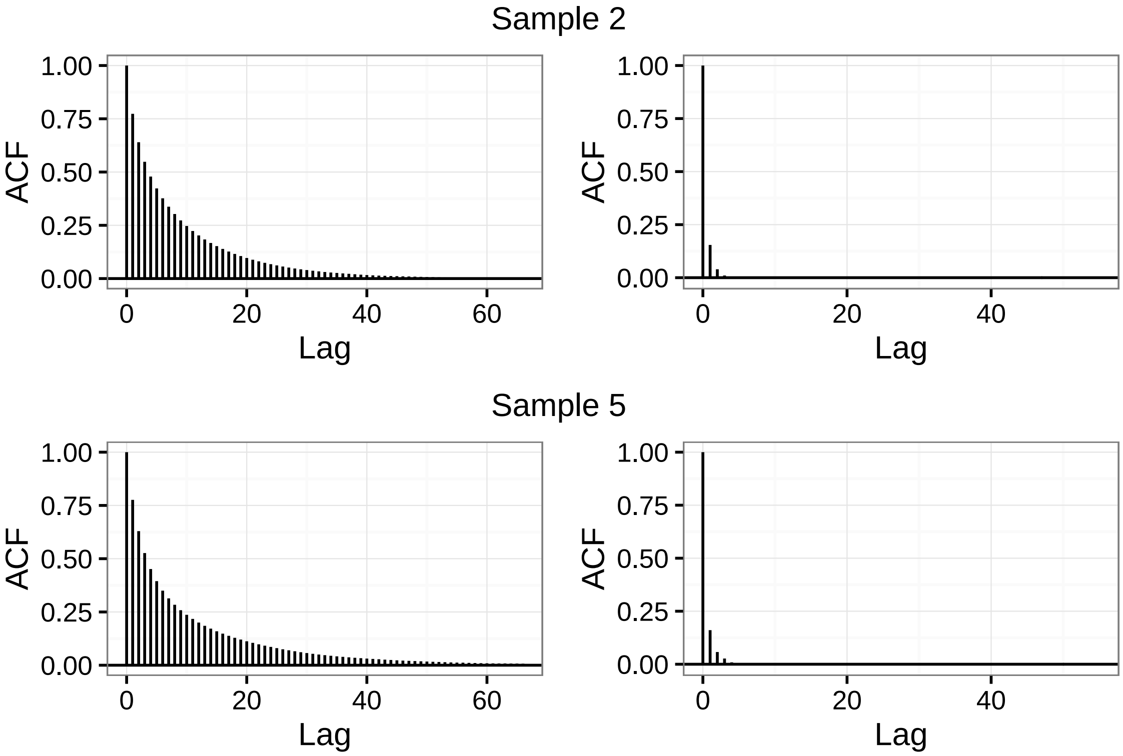

Figure 4.6: The autocorrelation function on two of the samples before and after thinning.

Table 4.2: The 95% credible intervals as well as the means and medians of the posterior distribution for the five samples in years BP.

Sample	95% Credible Interval	Mean	Median
1	(9916, 9992)	9953	9951
2	(9963, 10068)	10017	10017
3	(10022, 10128)	10081	10083
4	(10093, 10181)	10139	10141
5	(10163, 10248)	10210	10209

stratigraphically ordered prior.

4.4 Dating the R tephra

To date the R tephra, the simulation was run for two million trials. The proposal distribution was normal with a variance of 625. The variance was chosen to be lower here than in the previous applications since at every iteration, including the R tephra constraint already restricts the values each component can take on. The rejection rate was approximately 20%. The first 10000 trials were discarded as burn-in and then every fifth trial was sampled.

The posterior densities of the five carbon dated samples along with the R tephra are shown in Figure 4.8. Credible regions along with means and modes for each of the posterior densities are listed in Table 4.3. Since the effect of the temporal constraints depends on the samples used in the analysis, the posterior densities for the five carbon dated samples are slightly different now than when the R tephra was not included. The posterior density for the R tephra is shown in more detail in Figure 4.9.

As opposed to the analysis without the R tephra, the densities for the carbon dated samples are slightly flatter (the 95% credible intervals are wider). The samples found above the R tephra have more density in younger ages than before while samples found below the R tephra have more density in older ages. This is as expected since, in any given iteration of the simulation, we required the ages of the samples to be strictly ordered.

Overall, the mean and median ages (BP) of the R tephra were found to be 10048 and 10050, respectively. The values found by Samolczyk et al. [2016] were 10044 BP and 10047 BP as well as a 95.4% credible region of (9958,10127). The 95.4% credible region calculated here was found to be (9980, 10110).

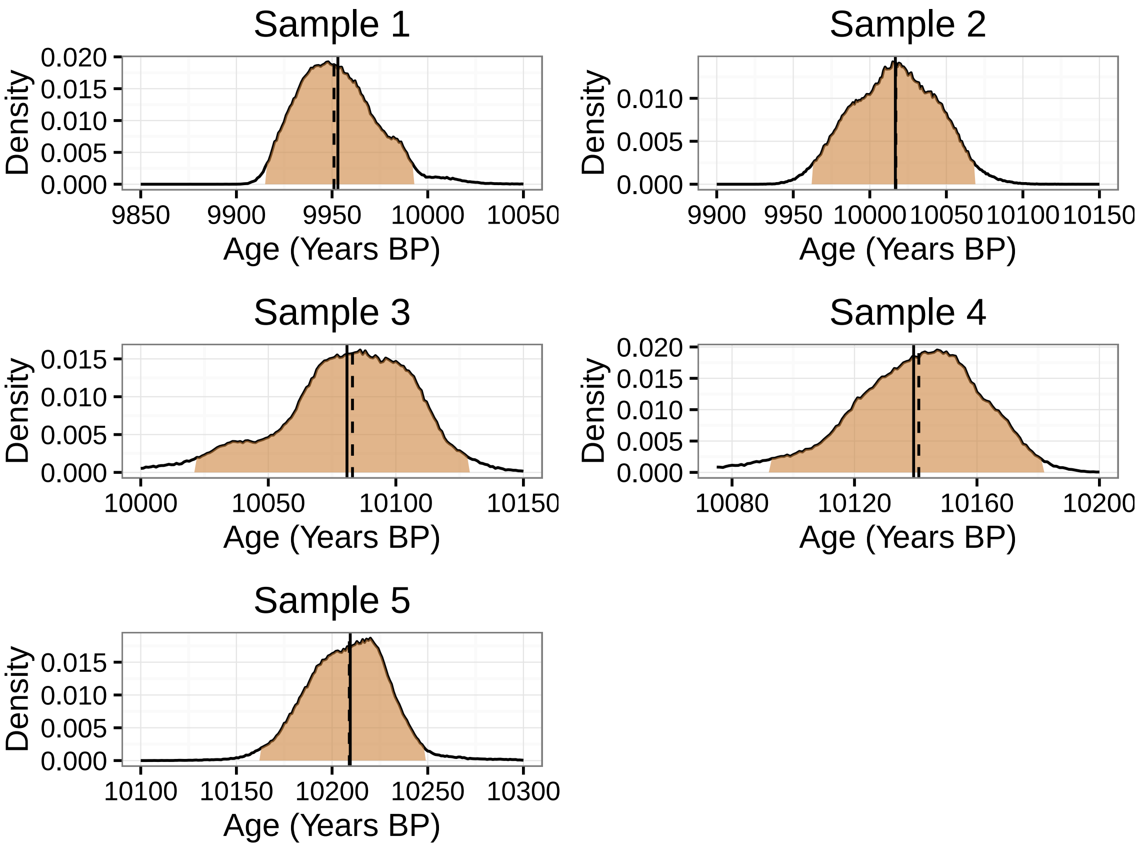

Figure 4.7: Posterior distributions for the samples shown with 95% credible intervals. The solid line is the mean of each density and the dashed line is the mode.

Table 4.3: The 95% credible intervals as well as the means and medians of the posterior distribution for the five samples in years BP.

Sample	95% Credible Interval	Mean	Median
1	(9915, 9992)	9953	9950
2	(9956, 10072)	10014	10013
R Tepha	(9981,10110)	10048	10050
3	(10018, 10137)	10083	10087
4	(10092, 10183)	10143	10140
5	(10163, 10247)	10209	10209

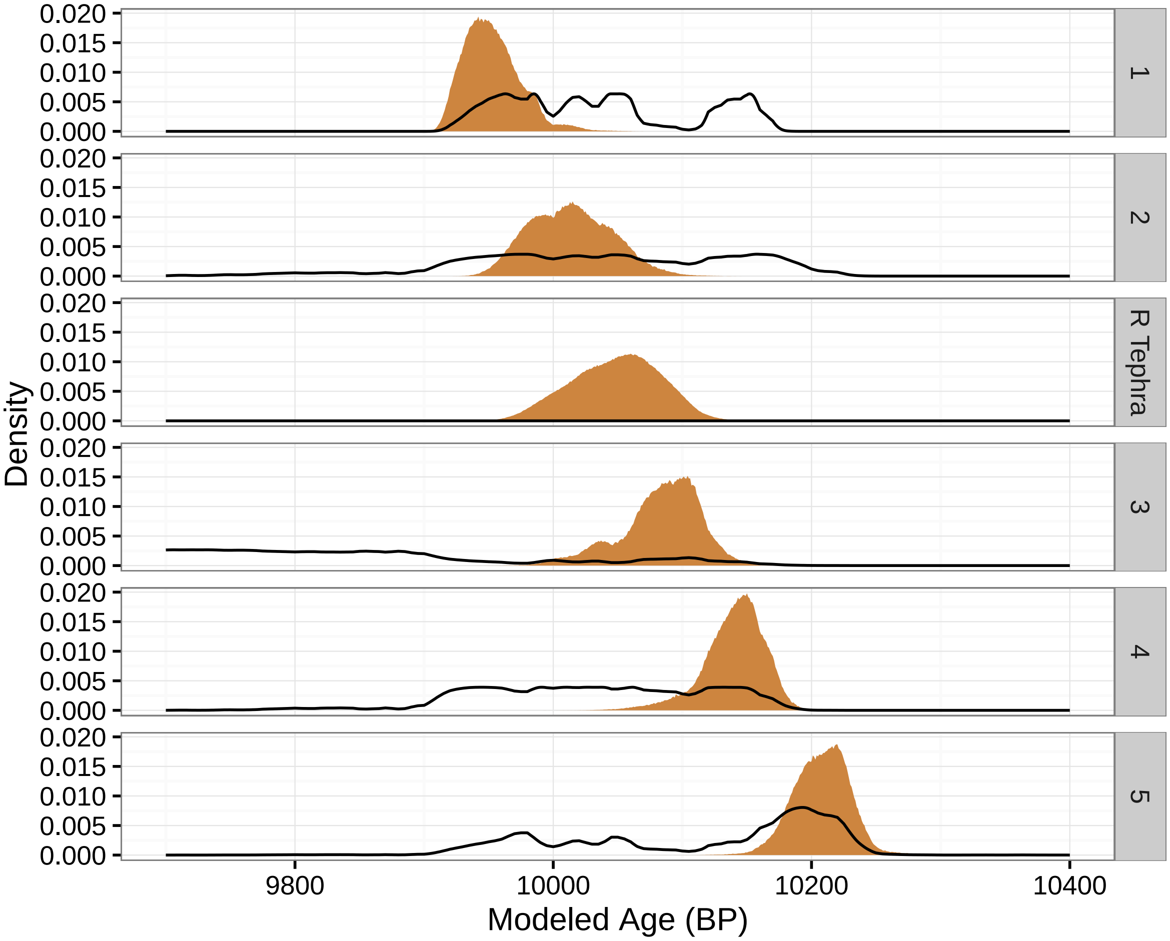

Figure 4.8: Posterior densities for the analysis with the R tephra boundary included. The line plots are densities based on individual numerical integrations without the ordered prior.

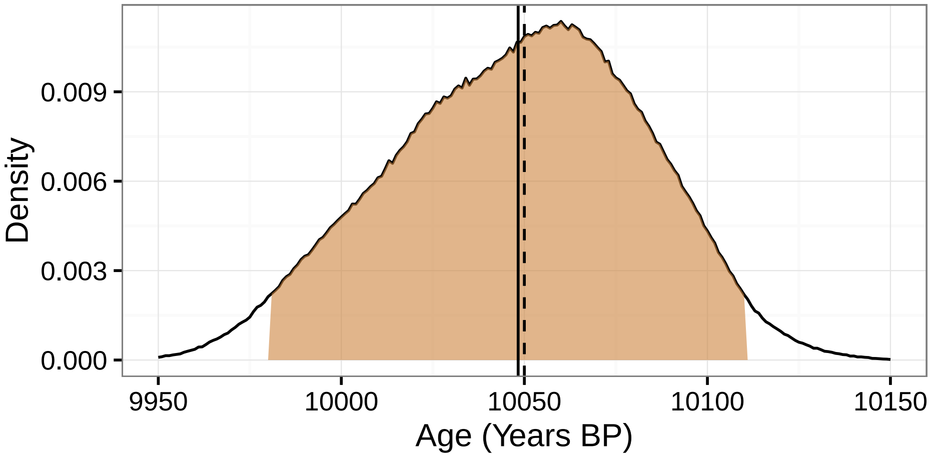

Figure 4.9: The posterior density for the R tephra, shown with a 95% credible region. The solid and dashed lines are, respectively, the mean and median of the density.

Conclusion

In this analysis, we have implemented a computational Bayesian method, Markov chain Monte Carlo, to approximate the ages of five carbon-14 dated samples, as well as a sample from the R tephra, with a temporally ordered prior. The results were in agreement with other work which had previously dated the R tephra.

Further Work

This line of study could continue by looking at time periods of interest, such as the boundaries of phases of pottery development in an ancient society. This analysis would still rely on Markov chain Monte Carlo, but dating phase boundaries differs from dating a single sample without a carbon date. This is because when studying phases, we want to weight the effect each sample has on the selection of the phase boundary by how many of the samples we believe belong in a phase. This approach is laid out by Buck et al. [1992].

Appendix A

R Code

Reading in Data and Linear Interpolation

The calibration is read from a website, linearly interpolated with the **approx** function, then put into a data frame.

```
IntCal13 <- read.table("http://www.radiocarbon.org/
   IntCal13%20files/intcal13.14c", skip=8,sep=",")[,c
   (1,2,5)]

CalC <- approx(IntCal13$V1, IntCal13$V2, n=50000, xout=c
   (1:50000))
CalS <- approx(IntCal13$V1, IntCal13$V5, n=50000, xout=c
   (1:50000))

Reimer13 <- data.frame("Year"=CalC$x, "C14␣Age"=CalC$y, "
   Sigma"=CalS$y)
```

Numerical Integration Function

This function takes a single carbon date and its estimated standard error and outputs a normalized posterior distribution for that carbon date.

```
post <- function (carb, sigma, data = Reimer13){
    year <- 1:50000
    prior <- rep(50000^-1, 50000)
    poster <- dnorm(carb,data$C14.Age, sigma) * prior
    poster/sum(poster)
}
```

MCMC Function for a Single Carbon Date

This implements the Metropolis-Hastings algorithm for a single carbon date, its standard error, and a given number of simulation. It also keeps track of the rejection rate of the simulation.

```
mcmcSing <- function(carb, sigma, iterations, data =
   Reimer13){
    reject <- 0
    init <- sample(singInit(carb), 1 )
    # initialize vector
    arr <- rep(0,iterations)
    for (i in 1:iterations) {
        # create density for initial guess
        postOrg <- dnorm(carb,data$C14.Age[init],sigma)
        # pick a new year from a normal dist centered at
           the initial guess
        newYear <- rnorm(1, init, 40)
        # make sure the new year is positive integer & in
           the support
        while (newYear > 500000 | newYear < 0){
            newYear <- rnorm(1, init, 40)
        }
        newYear <- round(newYear)
        # create density for proposed new year
        postNew <- dnorm(carb, data$C14.Age[newYear],sigma
           )
        if (postNew/postOrg >= 1) {
            init <- newYear
        } else {
            #accept/ reject
            draw <- runif(1,0,1)
            if (postNew/postOrg >= draw){
                init <- newYear
            } else reject = reject + 1
        }
        # put the year into the appropriate part of vector
        arr[i] <- init
    }
    # output vector
    acceptanceRate <- (iterations-reject)/iterations
    print(acceptanceRate)
    arr
}
```

MCMC output to Posterior Density

This function takes the output of a MCMC simulation and returns the posterior distribution.

```
postMCMC <- function(vector, data = Reimer13){
    out <- rep(0 , length(data[[1]]) )
    for (i in (min(vector)):max(vector)) {
        out[i] <- length(vector[which(vector==i)])
    }
    out/sum(out)
}
```

Finding Initial Age Guesses for Carbon Dates

Two functions which together return a vector of initial guesses to use in MCMC functions. The first is a helper function which takes a single carbon date returns a vector of ages that approximately fit a given carbon date. The second takes a vector of carbon dates and returns a vector of strictly ordered ages such that $c_1 < c_2 < \ldots < c_n$.

```
singInit <- function(carb, data = Reimer13){
    # take all calendar years in ball centered on carb w/
       radius 5
    data$Year[which( carb - 150 < data$C14.Age & data$C14.
       Age < carb + 150 )]
}

multInit <- function(carb){
    # initialize vector
    inits <- rep( 0, length(carb) )
    for (i in seq_along(carb) ) {
        # take smallest year for first element
        if (i == 1){
            inits[i] <- min( singInit( carb[i] ) )
            # take largest year for last element
        } else if (i == length(carb) ) {
            inits[i] <- max( singInit( carb[i] ) )
            # otherwise take smallest year greater than
               previous year
        } else {
            years <- singInit(carb[i])
            inits[i] <- min( years[ which(years > inits[i
               -1] + 15) ])
        }
    }
    # return the vector of years
    inits
}
```

MCMC for Multiple Samples without R Tephra

Two function which together perform the Metropolis-Hastings algorithm on a vector of carbon dates. They take a vector of carbon dates, a vector of standard errors, a vector of initial values, a prior distribution, and the length of the simulation as input. The first function is a helper function which performs one iteration on one component of the simulation. The second function keeps track all the simulated trials in a matrix as well as the rejection rate of the simulation.

```
genUnknownLag <- function(carb,init,sigma, prior, lowBound
   =0,upBound=50000, data = Reimer13){
    reject <- 0
    carbpost <- dnorm(carb,data$C14.Age[init],sigma)*prior
       [init]
    # draw new year
    newYear1 <- rnorm(1,init,40)
    # make sure it's between appropriate bounds
    while (newYear1 > (upBound-2) | newYear1 < (lowBound
       +2) ) {
        newYear1 <- rnorm(1, init, 40)
    }
    newYear1 <- round(newYear1)
    newPost1 <- dnorm(carb,data$C14.Age[newYear1],sigma)*
       prior[newYear1]
    # check if newYear1 has a higher probability than init
    if (newPost1/carbpost >= 1) {
        init <- newYear1
    } else {
            draw <- runif(1,0,1)
            # accept/ reject
            if (newPost1/carbpost >= draw) {
                init <- newYear1
            } else reject <- 1
    }
    df <- c(init,reject)
    df
}

unknownLagMult <- function(carb, sigma, inits, iterations
   = 50000, prior = rep(50000^-1,50000)) {
    reject <- 0
    len <- length(carb)
    results <- matrix(0, nrow = iterations, ncol = len)
    results[1,] <- inits
    bounds <- rep(0,len-1)
    for (i in 2:iterations){
        for (j in 1:(len-1)){
            bounds[j] <- ( results[i-1,j+1] + results[i-1,
               j] ) / 2
        }
        for (j in 1:len){
            if (j == 1 ){
                k <- genUnknownLag( carb[j], init =
                   results[i-1,j], sigma[j], upBound =
                   bounds[j], prior = prior)
            } else if (j == len){
                k <- genUnknownLag(carb[j], init = results
                   [i-1,j], sigma[j], lowBound = bounds[j
                   -1],prior = prior)
            } else {
                k <- genUnknownLag(carb[j], init = results
                   [i-1,j], sigma[j], lowBound = bounds[j
                   -1],  upBound = bounds[j], prior =
                   prior)
            }
            reject <- reject + k[2]
            results[i,j] <- k[1]
        }
    }
    print( reject / (5 * (iterations-1)) )
    results
}
```

MCMC for Multiple sample with R Tephra

Function very similar to the previous function except it also simulates data for the R Tephra sample.

```
unknownLagR <- function(carb, sigma, inits, iterations =
   50000, prior = rep(50000^-1,50000)) {
    reject <- 0
    len <- length(carb)
    results <- matrix(0, nrow = iterations, ncol = len)
    results[1,] <- inits
    bounds <- rep(0,len-1)
    tephra <- rep(0,iterations)
    for (i in 2:iterations){
        print(i)
        for (j in 1:(len-1)){
            bounds[j] <- ( results[i-1,j+1] + results[i-1,
               j] ) / 2
        }
        bounds[2] <- round(runif(1, results[i-1,2]+2,
           results[i-1,3]-2))
        tephra[i-1] <- bounds [2]

        for (j in 1:5){
            if (j == 1 ){
                k <- genUnknownLag( carb[j], init =
                   results[i-1,j], sigma[j], upBound =
                   bounds[j], prior = prior)
            } else if (j == 5){
                k <- genUnknownLag(carb[j], init = results
                   [i-1,j], sigma[j], lowBound = bounds[j
                   -1],prior = prior)
            } else {
                k <- genUnknownLag(carb[j], init = results
                   [i-1,j], sigma[j], lowBound = bounds[j
                   -1],  upBound = bounds[j], prior =
                   prior)
            }
            reject <- reject + k[2]
            results[i,j] <- k[1]
        }
    }
    print( reject / (5 * (iterations-1)) )
    out <- data.frame(results,tephra)
    out
}
```

Finding Credible Regions

This functions uses R’s **uniroot** function to find the credible region of a density function to a specified confidence level.

```
hpdRegion <- function(density, CI = .95){
    uniroot( function(density,CI,h) {sum(density[which(
       density>h)]) - CI},
             c(0,1), density = density, CI = CI)$root
}
```

References

Bowman, S. (1990). *Radiocarbon Dating*. Berklee, California: University of California Press.

Brooks, S., Gelman, A., Jones, G. L., & Meng, X.-L. (2011). *Handbook of Markov Chain Monte Carlo*. Boca Raton, Florida: CRC Press.

Buck, C., Kenworthy, J., Litton, C., & Smith, A. (1991). Combining archaeological and radiocarbon information: A bayesian approach to calibration. *Antiquity*, (pp. 801–821).

Buck, C., Litton, C., & Smith, A. (1992). Calibration of radiocarbon results pertaining to related archaeological events. *Journal of Archaeological Science*, (pp. 497–512).

Damon, P. E., Donahue, D. J., Gore, B. H., Hatheway, A. L., Jull, A. J. T., Linick, T. W., Sercel, P. J., Toolin, L. J., Bronk, C. R., Hall, E. T., Hedges, R. E. M., Housley, R., Law, I. A., Perry, C., Bonani, G., Trumbore, S., Woelfli, W., Ambers, J. C., Bowman, S. G. E., Leese, M. N., & Tite, M. S. (1989). Radiocarbon dating of the shroud of turin. *Nature*, *337*(6208), 611–615. `http://dx.doi.org/10.1038/337611a0`

Damon, P. E., Long, A., & Grey, D. C. (1966). Fluctuation of atmospheric c ^{14} during the last six millennia. *Journal of Geophysical Research*.

Hedman, M. (2007). *The Age of Everything: How Science Explore the Past*. The University of Chicago Press.

Libby, W. (1960). Nobel lecture: Radiocarbon dating. Available at: `http://www.nobelprize.org/nobel_prizes/chemistry/laureates/1960/libby-lecture.pdf`.

Naylor, J. C., & Smith, A. F. M. (1988). An archaeological inference problem. *Journal of the American Statistical Association*, (pp. 588–595).

Nicholls, G., & Jones, M. (2001). Radiocarbon dating with temporal order contraints. *Applied Statistics*, (pp. 503–521).

R Core Team (2015). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. `https://www.R-project.org/`

Reimer, P. J., Baillie, M. G. L., Bard, E., Bayliss, A., Beck, J. W., Bertrand, C. J. H., Blackwell, P. G., Buck, C. E., Burr, G. S., Cutler, K. B., Damon, P. E., Edwards, R. L., Fairbanks, R. G., Friedrich, M., Guilderson, T. P., Hogg, A. G., Hughen, K. A., Kromer, B., McCormac, G., Manning, S., Ramsey, C. B., Reimer, R. W., Remmele, S., Southon, J. R., Stuiver, M., Talamo, S., Taylor, F. W., van der Plicht, J., & Weyhenmeyer, C. E. (2013). Intcal13 and marine13 radiocarbon age calibration curves 0-50,000 years cal bp. *Radiocarbon*, (pp. 1029–1058).

Robert, C. P., & Casella, G. (1999). *Monte Carlo Statistical Methods*. Springer texts in statistics.

Samolczyk, M. A., Vallance, J. W., Cubley, J. F., Osborn, G. D., & Clark, D. H. (2016). Geomchemical characterization and dating of r tephra, a postglacial marker bed in mount rainier national park, washington, usa. *Canadian Journal of Earth Science*.

Stuiver, M., & Suess, H. E. (1966). On the relationship between radiocarbon dates and true sample ages. *Radiocarbon*.

Taylor, R. E., Bar-Yosef, O., & Renfrew, C. (2014). *Radiocarbon Dating*. Left Coast Press, second ed.

VanderKam, J. C. (1994). *The Dead Sea scrolls today*. Grand Rapids, Michigan: Wm. B. Eerdmans Publishing Co.

Wickham, H. (2009). *ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York. `http://had.co.nz/ggplot2/book`