Modern Model-Based Bayesian Causal Inference for Randomized Experiments with Hamiltonian Monte Carlo in Stan

Adam Rohde

Abstract

We explore the components of modern model-based Bayesian causal inferenece with a focus on randomized experiments. We discuss the potential outcomes framework, the Bayesian approach to causal inference, the MCMC sampling method Hamiltonian Monte Carlo, and the Stan probabilistic programming language. We also work through a simple example to illustrate how these components come together.

1 Introduction

Causal inference has experienced a renaissance over the last decade with redoubled efforts to expose the assumptions and limitations, as well as promise and importance, of casual inquiry. At the same time Hamiltonian Monte Carlo methods have dramatically improved the efficiency of sampling methods and has been built into probabilistic programming languages like Stan. These trends together promise to push the capabilities of Bayesian causal inference further than ever before. We follow Lee, Feller, and Rabe-Hesketh (2019) and Imbens and Rubin (2015) as guides to exploring how Hamiltonian Monte Carlo and probabilistic programming languages can be used in Bayesian causal inference. The focus here is on randomized experiments and model-based inference implemented in Stan, but the methods can be adapted for observational studies and other identification strategies. The primary goal of this paper is to explore the components of modern model-based Bayesian causal inferenece with a focus on randomized experiments. This includes the potential outcomes framework, the Bayesian approach to causal inference, the MCMC sampling method Hamiltonian Monte Carlo, and the Stan probabilistic programming language.

The potential outcomes framework for causal inference introduced by Rubin (1978) has become one of the most popular paradigms for the investigation of causal quantities. It allows us to seperate the treatment assignment mechanism from the model of the potential outcomes, which can clarify the modeling and assumptions required in causal inquiry. This framework can be applied in a variety of ways. One of the conceptually most simple versions of this is to take a model-based Bayesian approach. In this setting, the unobserved potential outcomes are viewed as random variables. We start with assumptions about the model for the joint distribution over the potential outcomes and the model for the treatment assignment mechanism. These can be used to derive a model for the posterior distribution over the unobserved potential outcomes, which allows us to impute them conditional on the observed data. From there, it is simple to estimate any causal quantities of interest. This analysis can be done analytically only in very simple situations. Therefore, we use a simulation strategy to build a simulated posterior distribution for the causal quantities of interest. This simulation strategy necessitates efficient sampling from the posterior distribution over the model parameters, given the observed data. Hence, we employ Markov Chain Monte Carlo (MCMC) methods to do the sampling. Given that modern statistical problems are increasingly high-dimensional, we focus on Hamiltonian Monte Carlo (HMC) methods as a way to avoid the issues that more traditional MCMC methods like random-walk Metropolis Hastings run into in high dimensions. HMC very efficiently explores the target distribution's state space. The Stan probabilistic programming language is a state-of-the-art language for implementing HMC applied to statistical problems. It is particularly well suited for the type of model-based Bayesian inference we discuss in this paper. Stan allows us to write code that looks like statistical notation to model and sample from the posterior distribution over our unobserved potential outcomes. Thus, we are very easily able to implement the simulation approach to Bayesian inference.

This paper will proceed by first touching on the conceptual foundations of the various topics and the details of how the topics relate and are used together. We then see a simple example application using the National Supported Work (NSW) experimental dataset.

2 Potential Outcomes Framework for Causal Inference

An important line of causal inquiry is focused on understanding the effects of some intervention on a system or population. We want to know the *causal* effect of this intervention or treatment on some measure related the system or population, which we'll call the outcome. The population could be Covid-19 patients, the treatment could be administration of some new theraputic, and the outcome could be mortality rate. Alternatively, we might be interested in how a certain economic policy effects some measure of inequality in the US. Many real-world statistical questions are causal in nature and interventions are an important type of causal question.[1]

How should we go about determining what the causal effect of the treatment was on the outcome? One answer is to view the problem within the potential outcomes framework. For every member of the population or observation of a system (call these units) there are two possibilities with respect to the treatment. The unit can be either treated or left untreated. In each case we assume that there is a constant value of the outcome for the unit. That is, if the unit recieves the treatment, we observe the *treatment potential outcome*. If the unit is not treated, we observe the *control potential outcome*. For every unit, we can only observe one of these states and, therefore, only one potential outcome. However, every unit still has two potential outcomes and the difference between the two is the effect of the treatment on that unit. Thus, there is a seperate treatment effect for every unit. Note that this is a *counterfactual* compairson. The fundamental problem of causal inference is that we can only ever observe one potential outcome for each unit, but we need both to understand the treatment effect. We are able to solve this problem only by introducing assumptions about the nature of the data (as in observational studies) or by imposing a design on the data generation process (as in randomized experiments). We can thus view causal inference as a missing data problem wherein we approximate the missing potential outcomes using assumptions and design. This is what is referred to as an *identification strategy*.

Note that a key feature of the potential outcomes model is that it allows us to treat the underlying model of the potential outcomes seperately from the treatment assignment mechanism. Treatment assignment is the mechanism through which some units end up being treated and some not. This can be designed or might occur through natural mechanisms. In many cases the ideal treatment assignment mechanism is random assignment. As we'll discuss further below, random assignment can help average out factors that might confound the treatment's effect on the outcome.

We'll now discuss potential outcomes a bit more formally. Say we have a sample of N units, $i = 1, ..., N$, and a treatment $D_i = \{0, 1\}$ for which we want to understand the effect on outcome $Y_i \in \mathbb{R}$. The potential outcomes framework holds that each unit has two potential outcomes Y_{0i} and Y_{1i}. Y_{0i} is the value of Y_i when unit i does not recieve the treatment, i.e., $D_i = 0$. Y_{1i} is the value of Y_i when unit i does recieve the treatment, i.e., $D_i = 1$. Again, note that in practice only one potential outcome for each unit can be observed. Hence, causal inference is about finding a way to *identify* something unobserved with something that is observed using assumptions or design as a way to solve the missing data problem. It is clear that the observed potential outcomes and missing potential outcomes can be written

$$Y_i^{\text{obs}} = Y_{1i}D_i + Y_{0i}(1 - D_i)$$
$$Y_i^{\text{mis}} = Y_{1i}(1 - D_i) + Y_{0i}D_i$$

We are usually primarily interested in estimating casual estimands like the average treatment effect (ATE), average treatment effect among the treated (ATT), or average treatment effect among the controls (ATC),

[1] This section draws Hazlett (2020b); Hazlett (2020a); Morgan and Winship (2014); and Imbens and Rubin (2015).

which aggregate the individual treatment effects across the entire popoulation under study or some subpopulation. We see that all of these quanties involve both potential outcomes for some units. Thus, in practice, they will involve some unobserved quantities. Our task will be to identify these unobserved quantities with some observed quantities or to impute them, through assumptions. We will then be able to use the observed and/or imputed quantities to estimate causal estimands like ATE to better understand the underlying causal relationship between the treatment and the outcome.

$$\begin{aligned}\text{ATE} &= \mathbb{E}[Y_{1i} - Y_{0i}]\\ \text{ATT} &= \mathbb{E}[Y_{1i} - Y_{0i}|D_i = 1]\\ \text{ATC} &= \mathbb{E}[Y_{1i} - Y_{0i}|D_i = 0]\end{aligned}$$

In what follows, we focus on randomized experiments, which is a specific identification strategy. The potential outcome framework, however, can be applied to a wide variety of other identification strategies (e.g., conditional on observables, instrumental variables, difference-in-difference, regression discontinuity). The key assumption in randomized experiments is that $\{Y_{0i}, Y_{1i}\} \perp D_i$. That is, the treatment assignment is independent of the potential outcomes. In randomized experiments, this is usually ensured in expectation based on the fact that treatments are assigned randomly. However, it it possible that in finite samples, the assumption is not exactly true. There are strategies to address this problem, for example conditioning on observable covariates. However, there is always the possibility that the randomized treatment assignment is not completely random in any given finite sample. (Note that this is a seperate issue from *selection bias* that comes up in non-randomized designs, in which teratment assignment might have systematic bias to treat certain types of units.) Another important assumption is the stable unit treatment value assumption (SUTVA), which states that one unit being treated does not effect the potential outcomes of another unit. This may not be the case in practice but is often assumed for simplicity. Our discussion will likewise assume stable unit treatment values.

We'll now discuss the independence of potential outcomes assumption and what it buys us. Armed with this assumption, we can show how to identify the ATE with observed quantities. As mentioned above,

$$\text{ATE} = \mathbb{E}[Y_{1i} - Y_{0i}].$$

We've also assumed independence of potential outcomes:

$$\{Y_{0i}, Y_{1i}\} \perp D_i.$$

Thus we can identify the ATE as follows.

$$\begin{aligned}\text{ATE} &= \mathbb{E}[Y_{1i} - Y_{0i}]\\ &= \mathbb{E}[Y_{1i}] - \mathbb{E}[Y_{0i}]\\ &= \mathbb{E}[Y_{1i}|D_1 = 1] - \mathbb{E}[Y_{0i}|D_1 = 0] && \text{by random assignment of treatment}\\ &= \mathbb{E}[Y_{1i}D_i + Y_{0i}(1 - D_i)|D_1 = 1] - \mathbb{E}[Y_{1i}D_i + Y_{0i}(1 - D_i)|D_1 = 0] && \text{becasue of conditionals}\\ &= \mathbb{E}[Y_i^{\text{obs}}|D_1 = 1] - \mathbb{E}[Y_i^{\text{obs}}|D_1 = 0] && \text{by definition of } Y_i^{\text{obs}}\\ &= \text{observed difference in means}\end{aligned}$$

So we've shown that our independence of potential outcomes assumption allows us to identify the ATE with the simple observed difference in means (i.e., the difference in the observed average outcome for treated units and the observed average outcome for control units). As usual, we can estimate the observed expectations with the sample averages.

$$\begin{aligned}\text{observed difference in means} &= \mathbb{E}[Y_i^{\text{obs}}|D_1 = 1] - \mathbb{E}[Y_i^{\text{obs}}|D_1 = 0] \\ &= \frac{1}{N_1}\sum_{i=1}^{N_1} Y_{i,\text{treated}}^{\text{obs}} - \frac{1}{N_0}\sum_{i=1}^{N_0} Y_{i,\text{control}}^{\text{obs}}\end{aligned}$$

Thus far, we've only discussed how to get a point estimate of caual quantities like ATE, ATT, and ATC. In the next section we'll explore the Bayesian perspective on causal inference which will also allow us to estimate such quantities of interest, in addition to giving us imputations of the unit-level missing potential outcomes. Thus, in the Bayesian framework, we are able to estimate any causal quantity of interest as well as evaluate simulate from posterior distributions that will allow us to better undersand the distribution of estimates.

3 Bayesian Causal Inference

For each unit, there are two observed quantities (D_i, Y_i^{obs}) and one missing quantity (Y_i^{mis}). The Bayesian causal inference views these three quantities as random variables which are drawn from an underlying joint probability distribution. The Bayesian perspective views the missing potential outcomes as unobserved random variables. The main goal of Bayesian inference here is to create a model for posterior distribution of the missing potential outcomes, given the observed potential outcomes and the treatment assignments.[2]

$$\Pr(Y^{\text{mis}}|Y^{\text{obs}}, D)$$

Such a model will allow us to derive the distribution for the causal estimand of interest, $\tau(Y_0, Y_1, D)$.[3] $\Pr(Y^{\text{mis}}|Y^{\text{obs}}, D)$ depends, crucially, on the joint distribution of potential outcomes, $\Pr(Y_0, Y_1)$, and on the treatment assignment mechanism, $\Pr(D|Y_1, Y_0)$. $\Pr(Y_0, Y_1)$ usually requires subject matter expertise and can be very difficult to choose. $\Pr(D|Y_1, Y_0)$ is a probabilistic rule that determines which units are treated. In randomized experiments, this mechanism is known.[4] That is, randomization makes the specification of $\Pr(D|Y_1, Y_0)$ unnecessary. In fact, for our purposes, $\Pr(D|Y_1, Y_0) = 1/\binom{N}{N_t}$, where $\sum_{i=1}^{N} D_i = N_t$. In observational studies, we would also have to model the treatment assignment mechanism, $\Pr(D|Y_0, Y_1, \theta)$. Note that we can also allow for observed covariates, X_i, which might allow us to correct for any imbalance after treatment assignment is carried out on our finite sample. For now, we omit such covariates for simplicity. As we've seen, the Bayesian model-based approach to causal inference within the potential outcomes framework allows us to consider $\Pr(Y_0, Y_1)$ and $\Pr(D|Y_1, Y_0)$ seperately.

We'll now step through how to move from specifying a model for $\Pr(Y_0, Y_1)$ to arriving at a model for $\Pr(Y^{\text{mis}}|Y^{\text{obs}}, D)$ and, finally, for $\Pr(\tau|Y^{\text{obs}}, D)$. This consists of five main parts.

- Choose a joint distribution over potential outcomes, $\Pr(Y_0, Y_1)$, or $\Pr(Y_0, Y_1|\theta)$ and priors for θ.
- Derive $\Pr(Y^{\text{mis}}|Y^{\text{obs}}, D, \theta)$.
- Derive $\Pr(\theta|Y^{\text{obs}}, D)$.
- Combine $\Pr(Y^{\text{mis}}|Y^{\text{obs}}, D, \theta)$ and $\Pr(\theta|Y^{\text{obs}}, D)$ to get $\Pr(Y^{\text{mis}}|Y^{\text{obs}}, D)$, by integrating over θ.
- Use $\tau(Y_0, Y_1)$ and $\Pr(Y^{\text{mis}}|Y^{\text{obs}}, D)$ to get $\Pr(\tau|Y^{\text{obs}}, D)$.

[2] This section draws on Li (2019), Imbens and Rubin (2015), and Lee, Feller, and Rabe-Hesketh (2019).

[3] Note that τ could be the ATE, ATT, ATC, or percentile treatment effects or the standard deviation of treatment effects, among a variety of other possible quantities of interest.

[4] In randomized experiments, this mechanism is controlled in large part by the researchers. However, issues like noncompliance and attrition can lead to non-random assignment even in randomized experiments. But, generally, these issues have straightforward solutions.

3.1 Joint Distribution Over Potential Outcomes, Pr(Y_0, Y_1)

The joint distribution of potential outcomes, $\Pr(Y_0, Y_1)$, is an input to the Bayesian causal inference framework. Under some relatively weak restrictions,[5] we can write $\Pr(Y_0, Y_1)$ as an integral over the product of N IID unit-level distributions, with common parameter vector θ and prior distribution over these parameters $\Pr(\theta)$.

$$\Pr(Y_1, Y_0) = \int \prod_{i=1}^{N} \Pr(Y_{1i}, Y_{0i}|\theta)\Pr(\theta)d\theta$$

Again, $\Pr(Y_{1i}, Y_{0i}|\theta)$ is a modeling degree of freedom. This can, again, be difficult to model and usually requires subject matter expertise. Results can be robust to this choice, however, for randomized experiments. How to choose the prior, $\Pr(\theta)$, also requires care; but results are also generally robust to this choice for randomized experiments.

3.2 Conditional Distribution over Missing Potential Outcomes, Pr$(Y^{\text{mis}}|Y^{\text{obs}}, D, \theta)$

We can combine the conditional dsitribution over treatment assignments, $\Pr(D|Y_0, Y_1, \theta)$, and the model for the joint distribution over potential outcomes, conditional on θ, $\Pr(Y_1, Y_0|\theta)$, to get the the joint distribution over treatment assignment and the potential outcomes, given the parameters, $\Pr(D, Y_0, Y_1|\theta)$.[6]

$$\Pr(D, Y_0, Y_1|\theta) = \Pr(D|Y_0, Y_1, \theta)\Pr(Y_1, Y_0|\theta)$$

We can then use $\Pr(D, Y_0, Y_1|\theta)$ to get the conditional distribution of the potential outcomes given treatment assignments and parameters. Note that randomization means that the treatment is independent of the potential outcomes, $\{Y_0, Y_1\} \perp D$. So this expression simplifies.

$$\Pr(Y_0, Y_1|D, \theta) = \frac{\Pr(Y_0, Y_1, D|\theta)}{\Pr(D|\theta)} = \frac{\Pr(Y_0, Y_1, D|\theta)}{\int \Pr(y_0, y_1, D|\theta)dy_0 dy_1} = \Pr(Y_0, Y_1|\theta)$$

Now we must transform $\Pr(Y_0, Y_1|D, \theta)$ to $\Pr(Y^{\text{mis}}|Y^{\text{obs}}, D, \theta)$. It is easy to write Y^{mis} and Y^{obs} as functions of Y_0, Y_1, and D. And so we can write $(Y^{\text{mis}}, Y^{\text{obs}}) = g(Y_0, Y_1, D)$, where g is the transformation.

$$Y_i^{\text{obs}} = \begin{cases} Y_{0i}, & \text{if } D_i = 0 \\ Y_{1i}, & \text{if } D_i = 1 \end{cases}, \quad Y_i^{\text{mis}} = \begin{cases} Y_{0i}, & \text{if } D_i = 1 \\ Y_{1i}, & \text{if } D_i = 0 \end{cases}$$

So we can transform $\Pr(Y_0, Y_1|D, \theta)$ to $\Pr(Y^{\text{mis}}, Y^{\text{obs}}|D, \theta)$. This then allows us to get $\Pr(Y^{\text{mis}}|Y^{\text{obs}}, D, \theta)$.

$$\Pr(Y^{\text{mis}}|Y^{\text{obs}}, D, \theta) = \frac{\Pr(Y^{\text{mis}}, Y^{\text{obs}}|D, \theta)}{\Pr(Y^{\text{obs}}|D, \theta)} = \frac{\Pr(Y^{\text{mis}}, Y^{\text{obs}}|D, \theta)}{\int \Pr(y^{\text{mis}}, Y^{\text{obs}}|D, \theta)dy^{\text{mis}}}$$

3.3 Conditional Distribution over Parameters, Pr$(\theta|Y^{\text{obs}}, D)$

To get the posterior conditional distribution over the parameters, $\Pr(\theta|Y^{\text{obs}}, D)$, we combine the prior over the parameters, $\Pr(\theta)$, and the likelihood of the observed data given the parameters, $\Pr(Y^{\text{obs}}, D|\theta)$. But we first need to get the likelihood by marginalizing out the missing potential outcomes from $\Pr(Y^{\text{mis}}, Y^{\text{obs}}|D, \theta) = \Pr(Y^{\text{mis}}, Y^{\text{obs}}, D|\theta)$, which we derived above.

[5] See Imbens and Rubin (2015) page 152.

[6] Both of the distributions we use here are either chosen by the researcher or come from the randomized experiment design.

$$\Pr(Y^{\text{obs}}, D|\theta) = \int \Pr(y^{\text{mis}}, Y^{\text{obs}}, D|\theta) dy^{\text{mis}}$$

Then combine the priors and likelihood to get the desired result.

$$\Pr(\theta|Y^{\text{obs}}, D) = \frac{\Pr(\theta)\Pr(Y^{\text{obs}}, D|\theta)}{\Pr(Y^{\text{obs}}, D)} = \frac{\Pr(\theta)\Pr(Y^{\text{obs}}, D|\theta)}{\int \Pr(\theta)\Pr(Y^{\text{obs}}, D|\theta)d\theta}$$

3.4 Conditional Distribution over Missing Potential Outcomes, Pr$(Y^{\text{mis}}|Y^{\text{obs}}, D)$

Next we'll combine $\Pr(Y^{\text{mis}}|Y^{\text{obs}}, D, \theta)$ and $\Pr(\theta|Y^{\text{obs}}, D)$ to get $\Pr(Y^{\text{mis}}, \theta|Y^{\text{obs}}, D)$. We can then marginalize to get $\Pr(Y^{\text{mis}}|Y^{\text{obs}}, D)$.

$$\Pr(Y^{\text{mis}}, \theta|Y^{\text{obs}}, D) = \Pr(Y^{\text{mis}}|Y^{\text{obs}}, D, \theta)\Pr(\theta|Y^{\text{obs}}, D)$$

$$\Pr(Y^{\text{mis}}|Y^{\text{obs}}, D) = \int \Pr(Y^{\text{mis}}, \theta|Y^{\text{obs}}, D)d\theta$$

$\Pr(Y^{\text{mis}}|Y^{\text{obs}}, D)$ is the "posterior predictive distribution" of Y^{mis}.

3.5 Conditional Distribution over Causal Estimand, Pr$(\tau|Y^{\text{obs}}, D)$

Now we can use $\Pr(Y^{\text{mis}}|Y^{\text{obs}}, D)$ and the observed data to get $\Pr(\tau|Y^{\text{obs}}, D)$. We do this by transforming the estimand $\tau(Y_0, Y_1, D)$ to $\tau(Y^{\text{mis}}, Y^{\text{obs}}, D)$, in a manner similar to the transformation to $Y^{\text{mis}}, Y^{\text{obs}}$ above. Thus, we've arrived at the posterior distribution over the causal estimand of interest. Stated differently, we are able to impute the missing potential outcomes and use these to estimate the causal quantity of interest. That is, we can impute the control potential outcomes for treated units and we can impute the treatment potential outcomes for control units. We can then calculate the posterior distribution of causal estimands of the form $\tau(Y^{\text{mis}}, Y^{\text{obs}}, D)$. This is because Y^{obs} and D are known and we can predict Y^{mis} using $\Pr(Y^{\text{mis}}|Y^{\text{obs}}, D)$. This approach incorporates two types of uncertainty. First, this incorporates uncertainty from parameter estimation, which is captured in the distribution $\Pr(\theta|Y^{\text{obs}}, D)$. Second, this incorporates unccertainty from imputation, which is captured in the distribution $\Pr(Y^{\text{mis}}|\theta, Y^{\text{obs}}, D)$. So we see uncertainty coming in from sampling the model parameters from the posterior over the model parameter and we se uncertainty coming in from sampling the imputed potential outcomes from the posterior over the missing potential outcomes, given the observed potential outcaomes, the treatment assignements, and the parameters.

3.6 Estimating Causal Quantities of Interest

Now that we have an imputed value for the missing potential outcome for each unit, we are able to estimate causal quantities of interest like average treatment effect (ATE), average treatment effect among treated (ATT), average treatment effect among controls (ATC), and others. This is because we have not only the imputed, formerly missing, potential outcomes for each unit but we also have the original observed potential outcomes for each unit. So we are easily able to estimate the treatment effect for each unit. These individual unit-level treatment effects can be aggregated in a variety of ways to estimate the previously mentioned causal quantities of ineterest. These are calculated as follows.

$$\text{ATE} = \mathbb{E}[Y_{1i} - Y_{0i}] = \frac{1}{N}\sum_{i=1}^{N}[Y_{1i} - Y_{0i}]$$

$$\text{ATT} = \mathbb{E}[Y_{1i} - Y_{0i}|D_i = 1] = \frac{1}{N_1}\sum_{i=1}^{N_1}[Y_{1i} - Y_{0i}]$$

$$\text{ATC} = \mathbb{E}[Y_{1i} - Y_{0i}|D_i = 0] = \frac{1}{N_0}\sum_{i=1}^{N_0}[Y_{1i} - Y_{0i}]$$

These are similar to the estimates of these causal quantities that we saw before. However, now we have found a way to impute a value for both potential outcomes for each unit. So we have access to the individual unit-level treatment effects, as well as any agregation of these one might be interested in.

We are able to view the treatment effect as either for the specific finite sample that we've observed or for a random sample from an infinte super-population. This means that the random sample is antoher source of uncertainty, in addition to uncertainty from imputation and uncertainty from parameter estimation. The super-population average treatment effect is the expectation over all distribution of finite sample average treatment effects that come from sampling from the infinite super-population. This can be written as

$$\text{ATE}_{sp} = \mathbb{E}_{sp}[\text{ATE}_{fs}] = \mathbb{E}_{sp}[\overline{Y}_1 - \overline{Y}_0]$$

We'll show how the super-population and finite-sample ATE's compare in our example below, but here we just note that the super-popualtion ATE is less precise and has a wider posterior distribution due to the additional source of uncertainty.

3.7 Simulation Approach

In many settings analytical solutions to the above are not feasible or are very difficult due to having to evaluate a complicated intergral. A simulation approach, however, is easy to implement and more broadly applicable than analytic methods for deriving the posterior. This approach, thus, explicitly involves imputation of the missing potential outcomes. To do this, we use $\Pr(Y^{\text{mis}}|Y^{\text{obs}}, D, \theta)$ and $\Pr(\theta|Y^{\text{obs}}, D)$ to repeatedly impute the missing potential outcomes.

That is, we repeat the following steps many times to get a simulated posterior for τ.

- First, we draw $\theta \sim \Pr(\theta|Y^{\text{obs}}, D)$.
- Second, we input this value for θ into $\Pr(Y^{\text{mis}}|Y^{\text{obs}}, D, \theta)$ to impute all of the missing potential outcomes. Note that we do not impute each missing potential outcome seperately. We impute all of them together without redrawing values for θ.
- These can be used to get an estimate of τ.

We can then take the average or other summary statistics of the simiulated posterior of τ to estimate our causal quantities of interest.

4 Hamiltonian Monte Carlo

We now need a way to efficiently sample from the distributions discussed above in order to impute the missing potential outcomes for each unit and then estimate our causal quantities of interest. This can be done with various techniques, including Markov Chain Monte Carlo (MCMC) sampling methods like Gibbs Sampling or Metropolis Hastings.[7] However, such methods do not make as effeccient use of computation as we might like and can fail in high dimensions, as we'll discuss below. Hamiltonian Monte Carlo (HMC) methods have grown in popularity recently because they address this problem in a suprising and powerful manner. "Instead of relying on fragile heuristics, the method is built upon a rich theoretical foundation that makes it uniquely suited to the high-dimensional problems of applied interest."[8] HMC is an auxulliary variables MCMC method that introduces "momentum" variables to facilitate more efficient movement through the original state space by moving in the joint space and then marginalizing back to the original state space. It's fully realized form follows the Metropoolis-Hasting algorithimic proposal-acceptance framework. In this section, we'll develop some intuition for how HMC works and why it can be an improvement over other sampling methods. We'll also discuss some of the mathematical details of how the algorithm works.[9]

Often we are interested in sampling from a target distribution as a means to calculate the Monte Carlo estimate of some function of that target distribution. In many cases, what we're primarily interested in are the expectations of target distributions. For the purposes of this discussion (and as a requirement of HMC), we'll focus on smooth or continuous densities.[10] Thus, evaluating expectations means evaluating integrals. For complex distributions this can be exceedingly difficult. Hence, Monte Carlo approximation is often used. For our current purposes, we are interested in understanding posterior distributions over model parameter values and in particular the expected values of these posterior distributions. Our discussion related to the intuition for HMC will follow some of the language and set up of Betancourt (2017) and will focus on expectations. We'll clarify which elements are most important to our present uses as we carry on.

4.1 Typical Set

Clearly, it is most efficient to focus on sampling from the regions of the target distribution that play the largest role in the expectations, rather than regions that contribute negligabily to the expectations of interest. Expectations are calculated by evaluating an integral over a volume in the parameter space. There are two components to this: the volumne we're integrating over and the density at each location in the space. There is a tension between these two in terms of contribution to the integral / expectations. Areas with high density are necessarily small in volume; conversely, areas with low density are necessarily large in volume. We must consider the dynamics between density and volume in determining which portions of the target distribution are most important for evaluating expectations. In high-dimensional spaces, the regions that contribute most to expectations concentrate distant from either extreme of the distribution, the mode or the tails. That is, the regions near the mode end up having too small of volume for these regions to contribute significantly to expectations; while the density in regions very far from the mode is too small for these regions to contribute significantly to expectations. These conflicting forces are balanced in the region called the *typical set*, whose contribution to expectations is significant. The typical set in very high dimensions narrows as the volume near the mode shrinks and the density of more dsitant regions vanishes. Thus, in high dimensions significant contributions to expectations only come from the typical set.[11] It's easy to see that it would be very useful to be able to sample from the typical set only and not waste computation exploring and sampling regions of the target distribution that are not in the typical set. This will be the focus of MCMC in general and HMC in particular.

[7] Li (2019)

[8] Betancourt (2017)

[9] This section draws on Betancourt (2017) and Handcock (2020).

[10] We require this because we will be taking derivatives of the energy function.

[11] Note that, even if expectations are not of interest, the typical set in such high-dimensional settings is still important for understanding the distribution for the same reasons as for expectations. The volume near the mode vanishes and the density far from the mode vanishes. Thus most samples from the distribution will come from the typical set.

4.2 MCMC

The goal of Markov Chain Monte Carlo methods is to randomly explore the typical set, sampling as we go along, as a way to calculate Monte Carlo approximations of expectations. MCMC methods are designed such that they will eventually explore the typical set and yeild useful estimates. It is possible that typical MCMC methods will not be able to explore sufficiently in finite time, however. We'll discuss some of the principles of MCMC and the Metropolis-Hastings approach and then show how these can fail in high dimensions.

"A Markov chain is a progression of points in [state] space generated by sequentially applying a random map known as a Markov transition."[12] The Markov transition defines the probability of moving from one state to another. When a Markov transition ensures that a sample of states following a specific distribution will move these sample states to a new set of states that also follow the same distribution, we have that the Markov chain is balanced and will continue to produce samples from the distribution of interest. We can use this fact to construct a Markov transition that explores the states of a target distribution of interest. Crucually, it can be shown that, given a properly defined Markov transition, any randomly initialized markov chain will eventually converge to the target distribution and, thereafter, only produce samples from the target distribution.[13] In other words, the Markov chain will eventually drift into the typical set and then begin to move through the typical set. With enough iterations, the Markov chain will serve as a useful approximation of the typical set. We can then use the resulting sequence of states from such a Markov chain as a sample from the target distribution and, hence, calculate Monte Carlo approximations of quantities of interest like expectations.

Markov chains, under ideal conditions, exhibit three stages of behavior. First, the chain moves toward the typical set from its initial starting position in state space. During this stage, the chain does not represent a useful sample from the target distribution. This pahse is called the "burn in." The second stage starts when the chain first reaches the typical set and makes it's first journey through the typical set. During this stage, the chain starts to resemble a sample from the target distribution and estimation based off the chain improves greatly. The third stage follows the first traversal of the typical set and constitutes the chain exploring the finer details of the typical set. This stage sees subtler distributional details being picked up and gradual improvement in estimation based off the chain. It is important to acknowledge that there are "pathological" distribution behaviors that can lead to important parts of the distribution not being captured well.[14] However, for our current purposes, we'll omit a discussion of these issues. We will note that the $\hat{R}$ statistic quantifies the variation across simultaneous Markov several chains initialized from various locations in the state space. This can be used to identify when some chains are struggling with potential pathelogical portions of the distribution.[15]

4.3 Metropolis-Hastings

So how might we construct a Markov transition that suits our needs (i.e., is balanced and leads to exploration of the typical set)? Likely the most popular solution is the Metropolis-Hastings (MH) Algorithm.[16] The MH algorithm works as follows. Start at some initial state. First, stochastically select a proposal state as the next state. The proposal is usually chosen from some proposal distribution, $h(s'|s)$, where s is the current state and s' is the proposed state. Second, choose to move to the proposal state with probability defined by the MH acceptance probability. The MH acceptance probability is carefully designed so that proposals that are too far from the typical set are not accepted. Below $\pi(s)$ is the target distribution.

$$\text{MH Acceptance Probability} = \min\left[1, \frac{h(s|s')\pi(s')}{h(s'|s)\pi(s)}\right]$$

[12] Betancourt (2017)

[13] Betancourt (2017)

[14] See Betancourt (2017) for more discussion.

[15] If $\hat{R}$ is not 1 then there are likely pathologies.

[16] Metropolis et al. (1953), Hastings (1970)

Note that when the proposal distribution is symmetric, the MH acceptance probability formula simplifies some. An important portion of implementing this algorithm in practice is carefully chosing the proposal distribution so that we are able to accept enough proposals to be efficient.[17] The proposal distribution will propose states across a wide volume of the state space, but the acceptance rate ensures that we usually reject proposals that are too far from the typical set. Thus, MH will explore the typical set when given enough iterations, with relatively few deviations from this.

```
Metropolis-Hastings Algorithm

Initialize Markov chain at s.
For 1 to n:
    Draw s' ~ h(s'|s).
    Draw u ~ Uniform[0,1].
    Calculate MH Acceptance Probability a = min(1,(h(s'|s)pi(s'))/(h(s|s')pi(s))).
    If u<a, accept s'.
    Else stay at s.
```

Again, the beauty of MH is that a well chosen proposal distribution will help explore the entire target distribution state space, while the accpetance probability ensures that the states that are accepted in the chain are exactly prorportional to their density in the target distribution. However, in high dimensions this starts to breakdown. In these settings, the volume outside the typical set becomes exponentially larger relative to the volume within. So the algorithm will end up proposing states in the tails of the target distribution, where the density is extremely low. Thus, the acceptance probability will be very small as well and most proposals will be rejected. So MH becomes very inefficient in high-dimensions, with many proposals going unused.[18] Thus, we end up exploring the typical set very slowly.

HMC is a significant improvement over these methods in that it is able to directly explore just the area around the typical set. Therefore, we sample only from the most impactful regions of the target distribution and do so without wasting computation and time.

4.4 HMC, Newtonian Mechanics, & Efficient Exploration of High-Dimensional State Space

As mentioned above, Hamiltonian Monte Carlo methods improve on other MCMC methods by more directly targeting the typical set and more efficiently drawing samples from the typical set. The mechanisms through which HMC achieves this have simple but suprising intuitions. The goal of HMC is to define a Markov transition kernel that closely follows the contours of the typical set, without falling into low acceptance situations like we saw for MH. To do this, we want to define a vector field that follows the typical set in the state space. When at a given state, we want to follow the vector field for some amount of time and then arrive at a new point that is also in the typical set. Moving through the state space thus, we efficiently explore the typical set as opposed to moving around at random and accepting and rejecting based on whether a proposal is near the typical set. This is the key point that distinguishes HMC. It only proposes states that are in or very close to the typical set. So we efficiently explore the state space, minimizing computation but still concentrating on the typical set.

This efficient movement is achieved by leveraging the geometry of the target distribution. Specifically, we use the gradient of the probability density function to move through the state space on contours of the density that have equal probability. It turns out that the differential geometry to achieve this movement through state space is mathematically equivalent to the dynamics of physical systems under Newtonian Mechanics. That is, staying in the typical set in our state space is mathematically equivalent to keeping a satellite in

[17] As might be intuitive, the best proposal distribution is the target distribution. So in practice, we want the proposal distribution to be as similar as possible to the target distribution. This can be difficult to achieve.

[18] Note that, given infinite time, MH in high-dimensions will sitll converge. But we dont have infinite time.

orbit around a planet. In particular we must balance momentum and gravity. In the metaphor of the planet and satellite, as the satellite falls towards the planet, due to the pull of gravity, momentum grows until the satellite is pushed away from the planet. As the satellite starts to move further from the planet, momentum decreases and gravity is able to pull the satellite back towards the planet. These counterbalancing behaviors keep the planet in orbit perpetually.

Our goal is to find a way to mimic this behaviour in our exploration of the target distribution's state space, where each "orbit" is a contour of equal probability. "[T]he key to twisting the gradient vector field into a vector field aligned with the typical set, and hence one capable of generating efficient exploration, is to expand our original probabilistic system with the introduction of auxiliary momentum parameters."[19] We will need to ensure that these momentum auxillary variables have a distribution that ensures conservation of energy in the way described above. To add momentum, we expand the target distribution state space to "joint space" that includes position, q, and momentum, p. We also then "lift" the target distribution up into to a joint distribution on the joint space with the choice of a conditional distribution over the momentum variable.

$$\pi(q,p) = \pi(p|q)\pi(q)$$

This garauntees that trajectories exploring the typical set of the phase space project down to trajectories exploring the typical set of the target distribution. Hence, the addition of the auxiliary variables allow us to efficiently explore typical set of the target distribution state space by exploring the typical set of the joint space. We can write the joint distribution in terms of a *Hamiltonian*, $H(q,p)$, which captures the geometry of the joint space, including the typical set of the joint space, because it captures both position and momentum. The value of the hamiltonian at any point in joint space is called the *energy* at that point.[20] We'll use the gradients of this to identify exactly how we can move in the joint space to preserve total energy and thus efficiently move through the typical set.

$$\begin{aligned}\pi(q,p) = e^{-H(q,p)} \iff H(q,p) &= -\log \pi(q,p) \\ &= -\log[\pi(p|q)\pi(q)] \\ &= -\log \pi(p|q) - \log \pi(q) \\ &\equiv K(p,q) + V(q) \\ &= \text{kinetic energy} + \text{potential energy}\end{aligned}$$

The Hamiltonian can be decomposed into two terms that can be called kenetic and potential energy. Kinetic energy is a distribution over the joint space and potential energy corresponds to the target distribution. Kinetic energy must be sepcified for the actual implementation.

Given that the Hamiltonian captures the geometry of the joint space and the typical set of the joint space, we can use it to generate the vector field aligned with the typical set of the joint space that we can use to efficiently traverse the typical set in the original state space. This is achieved by using Hamilton's equations.[21] These prescribe how to balance kinetic and potential energy for a given total energy as we move through time.

$$\begin{aligned}\frac{dq}{dt} &= \frac{\partial H}{\partial p} = \frac{\partial K}{\partial p} \\ \frac{dp}{dt} &= -\frac{\partial H}{\partial q} = -\frac{\partial K}{\partial q} - \frac{\partial V}{\partial q}\end{aligned}$$

Importantly, note that $\frac{\partial V}{\partial q}$ is the gradient of the log of the target distribution. So by incorporating the momentum variables and using Hamilton's equations, we are able to move through the joint space following

[19] Betancourt (2017)

[20] The notation and description in this section draw on Betancourt (2017).

[21] Handcock (2020)

contours of equal probability. This allows us to move directly through the typical set in the joint space, once we find it initially. We are then able to project down onto the original state space and also move directly around the target distribution typical set.

In short, HMC boils down to the following steps. Given some point in the original state space, we sample from the conditional distribution of momentum given position, $\pi(p|q)$, to get a point in the joint space. Once in the joint space, we explore the typical set of the joint space for some time by integrating Hamilton's equations. Since the joint diststrubition and space is chosen according to the Hamiltonian, the trajectory on the typical set in joint space projects down onto the typical set in the original state space.

It is important to note that, while we are traversing a Hamiltonian trajectory in the joint space, we are in reality exploring a "level set" of constant energy in a deterministic manner. However, the transition up from the original state space to the joint space involves a stochastic selection of momentum from the conditional distribution $\pi(p|q)$, which corresponds to a stochastic choice of energy level set in the joint space. So HMC involves stochastic elevations into energy level sets in joint space, deterministic traversal of these level sets, and projection back down into the target distribution state space. The choice of $\pi(p|q)$ is what allows us to stay in the typical set in both spaces, assuming that the intial point was in the typical set of the target distribution. This is similar to the choice of proposal distribution in Metropolis Hastings. This will be discussed further in the next section. Of practical importance are the choices of how long to traverse the Hamiltonian trajectories, how to actually integrate these, and specifically how to choose $\pi(p|q)$. We will discuss how Stan approaches these issues below.

As we've seen, we can build a Markov chain that becomes an efficiently drawn sample from the target distribution state space that is concentrated on the target distribution's typical set, even in high dimensions.

4.5 HMC in Practice

There are a number of "algorithmic degrees of freedom" that must be delt with in HMC, as well as other practicalities that alter the form of fully realized implementations of HMC. We will discuss these issues here following how they are presented in Betancourt (2017). However, details specific to the Stan implementation of HMC are left for below.

Among the algorithmic degrees of freedom in HMC is the conditional distribution over momentum, given position or, alternatively, the kinetic energy function. There are an infinite number of possible distributions over momentum. A family of often used distributions is that of Euclidean-Gaussian distributions. These are of the following form and are independent of position.[22] Such choices for kinetic energy can simplify derivation and calculation. Other alternatives include Riemannian-Gaussian kinetic energy and various non-Gaussian kinetic energies.

$$\begin{aligned} \pi(p|q) &= N(0, \Sigma) \\ \implies K(p,q) &= \frac{1}{2} p^\top \Sigma^{-1} p + \log|\Sigma| + \text{constant} \end{aligned}$$

Another choice that must be made to implemnt HMC is the choice of integration time. That is, how long to traverse Hamiltonian trajectories in the joint space before projecting back down to the target distribution state space. There is an inherent tradeoff in this choice. Short integration times might not take advantage fully of the main benefits of HMC that allow for efficient and fast exploration of the typical set. However, long integration times might lead us back close to where we started in addition to increasing computation. Note that no single integration time will be optimal everywhere. Thus, dynamic selection of integration time is best. How this might be done is beyond the scope of this section but below we'll touch on the choices made for Stan below (use of the *No-U-Turn termination criterion*).

When running HMC in practice, a problem we encouter is that we cannot solve the Hamiltonian equations (and hence follow the Hamiltonian trajectories) exactly. Numerical approximation is necessary and numerical

[22] Betancourt (2017), Handcock (2020)

inaccuracies can compound as we traverse the Hamiltonian trajectory. This error pushes us from the true trajectory. The typical solution to this is to use *symplectic integrators*, which oscillate near the true trajectory without diverging too much in any direction. A common choice of symplectic inegrator is the *leapfrog integrator*, which works with Euclidean-Gaussian kinetic energies. The idea behind this integrator is fairly simple. We essentially break the discretized update steps wherein we update momentum and position into a half step for momentum, a full step for position, followed by a half step for momentum. Note that regions of high curvature in the target distribution are particularly difficult for these sorts of integrators, which will diverge quickly towards infinity in such regions. This can actually be viewed as a feature, rather than a bug in that divergences can make it easier to detect problematic geometries.

```
Leapfrog Integrator

1. Initialize q and p.
2. For 0 <= n < floor(T/eps):
3.    p(n+1/2) = p(n)     - eps/2 * dV/dq * q(n)
4.    q(n+1)   = q(n)     + eps   * p(n+1/2)
5.    p(n+1)   = p(n+1/2) - eps/2 * dV/dq * q(n+1)
```

While sympletic integrators do well, they still introduce some bias that we need to correct for. This is done by treating the Hamiltonian transition state as a proposal for a Metropolis-Hastings approach on the joint space that we accept or reject to correct for the bias. We alter the Hamiltonian transition to this end. Suppose we integrate forward from an initial state for L steps and propose the state at the last step. The MH acceptance probability is what we'd expect, the minimum of 1 and the ratio of the proposal densities and the target densities (which here are the joint space densities). We also make the Hamiltonian transition reversible so that we cann get non-zero ratio of proposal densities. We do this by flipping the sign of momentum for the proposal state.[23]

$$\begin{aligned}
\text{Acceptance Probability} &= \min\left[1, \frac{\mathbb{Q}(q, p|q', -p')\pi(q', -p')}{\mathbb{Q}(q', -p'|q, p)\pi(q, p)}\right] \\
&= \min\left[1, \frac{\pi(q', -p')}{\pi(q, p)}\right] \\
&= \min\left[1, \frac{\exp(-H(q', -p'))}{\exp(-H(q, p))}\right] \\
&= \min\left[1, \exp(-H(q', -p') + H(q, p))\right]
\end{aligned}$$

For optimal performance you actually have to average the proposals for all states along the Hamiltonian trajectory into one proposal. However we do not go into the details of this here. There are also ways of diagnosing poorly chosen kinetic energies, regions of high curvature, and understanding the limitations of these diagnostics. These are left to Betancourt (2017).

4.6 HMC Algorithm

As mentioned above, Hamiltonian Monte Carlo is a form of Metropolis-Hastings MCMC sampling method. Unsuprisingly, HMC follows a similar progression as the MH approach discussed above. Given that we're at a certain state in the target distribution state space, we fisrt propose a new value for momentum. Next we update the state in the joint space by traversing the Hamiltonian trajectory in joint space. We then accept or reject the proposed updated state based on the acceptance probability. We then marginalize to get the new state in the original state space. Below are the details of how a HMC algorithm might be implemented.[24]

[23] See Betancourt (2017) for more details. See Handcock (2020) a similar formulation.

[24] Handcock (2020)

```
Hamiltonian Monte Carlo Algorithm

1. Initialize position at q.
2. For 1 to n:
3.    Draw p ~ pi(p|q).
4.    Draw u ~ Uniform[0,1].
5.    For 1 to L:
6.        p(l+1/2) = p(l)     - eps/2 * dV/dq * q(l)
7.        q(l+1)   = q(l)     + eps   * p(l+1/2)
8.        p(l+1)   = p(l+1/2) - eps/2 * dV/dq * q(l+1)
9.    Proposal State is (q(L),p(L)).
10.   Calculate Acceptance Probability a = min(1,exp(-H(q(L),-p(L)) + H(q,p))).
11.   If u<a, accept (q(L),p(L)).
12.   Else stay at (q,p).
13.   Discard momentum.
```

5 Probabilistic Programming Languages & Stan

Probabilistic programminng languages are tools for statistical modeling.[25] They facilitate and abstract away details of building many probability models, allowing users to quickly and easily work with such models. The goal of probabilitsc programming languages is to make the analysis of these models easy and the focus of the users time, rather than just constructing and executing them. Importantly, probabilistic programming languages are particularly well suited to Bayesian inference, hence their discussion here.

Stan[26] is a popular C++ based probabilistic programming language with interfaces in R, Python, Julia, and Matlab. We will be using the `rstan` R library below. Stan makes Bayesian inference and modeling straightforward with code written in Stan looking like statistical notation. In particular, Stan makes sampling from posterior probability distributions very easy and efficient. It does this by implementing a variant of Hamiltonian Monte Carlo called the no-U-turn sampler that very efficiently explores the states of the posterior distribution of interest.[27]

Stan's implementation of HMC addresses the practical optimization choices mentioned above. That is, Stan uses a multivariate normal distribution for the conditional distribution of momentum that does not depend on position. This is the Euclidean-Gaussian distribution mentioned above. Stan uses the Leapfrog integrator and introduces the Metropolis-Hastings acceptance step. Stan also uses the no-U-turn termination criterion for determining integration time.[28] These optimizations make Stan extremely fast. However, all the details of HMC as used by Stan are hidden away from the user. Instead, the user is able to utilize this state-of-the-art sampling algorithm for a wide range of applications without actually understanding HMC or its variants. Thus, application based case studies like Lee, Feller, and Rabe-Hesketh (2019) don't need to mention what Stan is doing "under the hood." This makes for a very focused user modeling experience.

As such, we'll now discuss some of the details of Stan from the users perspective. We focus on the structure and components of a Stan program in R.[29] Writing data generating models in Stan is straignforward. A Stan program is composed of blocks that each have a specific task.

[25] Sampson (2016)

[26] Stan (n.d.a)

[27] Gelman, Lee, and Guo (2015)

[28] Stan (n.d.b)

[29] This section draws on Savage (2020).

The blocks that typically appear in Stan programs include:

- `functions` - define functions to be used later
- `data` - identify data to be used
- `transformed data` - transform data
- `parameters` - identify the unknowns to be estimated, including any restrictions on their values
- `transformed parameters` - transform parameters and/or data
- `model` - define probability model
- `generated quantities` - generate outputs from the model including posterior predictions

As we'll see in the next section, Stan programs look like statistical notation and are easy to interpert, with the focus on the model and analysis and not on syntax or how to efficiently sample from the posterior distributions of interest.

6 Example Application

In this section, we will work through an example of model-based Bayesian causal inference from a randomized experiment using Stan. This example follows Chapter 8 "Model-Based Inference for Completely Randomized Experiments" of Imbens and Rubin (2015)[30] and uses the well-studied National Supported Work (NSW) experimental dataset.

We will work in the Bayesian causal inference framework described above and will revisit some of the key ideas here. In particular, the potential outcomes are considered random vsariables and, thus, so are any functions of them like the average treatment effet or median treatment effect. Since the causal quantities of interest are random variables, we can think about their probability distribution.

We build a model of these potential outcomes that depends on some unknown model parameters. We use the observed data to learn the distribution of these unknowns. We then draw values from the distribution of these unknowns and use these to impute the missing potential outcomes from the hypothesized model. We can then do inference on any causal estimand of interest, $\tau = \tau(Y_0, Y_1, D, X)$, where each of the arguments here are vectors of the potential outcomes, the treatment assignments, and covariates, respectively. As before, we require SUTVA or row interchangability of τ.

Since any dataset can only ever have half of the potential outcomes for a population, there will never be direct empirical information on the dependence between treated and control potential outcomes. So modeling this dependency is a key focus here. Note, however, that the potential outcomes (and causal estimands) are well defined regrdless of how they or treatment assignment are modeled.

As discussed above, the goal is to get the conditional distribution over the vector of missing potential outcomes given the observed potential outcomes and treatment assignments: $\Pr(Y^{\text{mis}}|Y^{\text{obs}}, D)$. Then we can infer the distribution of causal quantities of interest by writing them as functions of the missing and observed data.

[30] This example is also discussed in Lee, Feller, and Rabe-Hesketh (2019) with a description of how Stan can be used for such an application.

6.1 NSW Data

Before getting into the modeling, we first quickly discuss the data in its own right. The NSW data are from an experiment on a job training program for male workers disadvantaged in the job market in the 1970's. The data consist of information on each individual like age, years of education, whether they have been married, whether they finished high school, and ethnicity. The data also has measures of pre-training earnings in 1975 and in the one to two years before training. There is also a dummy for whether the individuals had zero earnings in 1975 or in the two years before training (called 1974 earnings). Finally, the outcome we're interested in is earnigns in 1978. There are 445 men in the data; 260 were placed in the control group; 185 were placed in the treatment group. Earnings are in thousands of dollars. In general, annual earnings are very low for the individuals. There is a slight imbalance between the treatment and control groups on ethnicity, degrees, and individuals with zero income in 1975. Summary statistics for the information in the data are presented below.[31] This includes balance t-tests for differences in means and KS-tests for differences in distribution. Note that we also run this balance test on the outcome of interest. We discuss this result in more detail below.

Table 1: NSW Summary Statistics

	Covariate	Mean	SD	Average Controls	Average Treated	T pval	KS pval
Age	age	25.37	7.10	25.05	25.82	0.27	0.52
Black	black	0.83	0.37	0.83	0.84	0.15	0.01
Education	educ	10.20	1.79	10.09	10.35	0.65	NA
Hispanic	hisp	0.09	0.28	0.11	0.06	0.06	NA
Married	married	0.17	0.37	0.15	0.19	0.33	NA
No Degree	nodegr	0.78	0.41	0.83	0.71	0.00	NA
Earnings 1974	re74	2.10	5.36	2.11	2.10	0.98	0.56
Earnings 1975	re75	1.38	3.15	1.27	1.53	0.39	0.05
Earnings 1978	re78	5.30	6.63	4.55	6.35	0.01	0.04
Treatment Indicator	treat	0.42	0.49	NA	NA	NA	NA
Zero Earnings 1974	u74	0.73	0.44	0.75	0.71	0.33	NA
Zero Earnings 1975	u75	0.65	0.48	0.68	0.60	0.07	NA

We calcualte the ATE using the simple difference in means estimator mentioned in the initial potential outcomes section above, which is identified here because of randomization. We also calculate the corresponding standard error and 95% confidence interval. This will be useful for comparison with the ATE estimate we get from the Bayesian approach and the posterior distribution for the ATE. We see that the treatment effect of the jobs program on earnings in thousands of dollars is 1.79, with a standard error of 0.67.

Table 2: Difference in Means Estimate of ATE

Estimate	SE	CI Low	CI High
1.79	0.67	1.73	1.86

[31] This replicates Table 8.1 in Imbens and Rubin (2015).

6.2 Review of Bayesian Modeling and Causal Inference

The Bayesian approach starts with the joint distribution of the potential outcomes.[32] Under some relatively unrestrictive restrictions[33], we can model this as the integral over the product of IID individual-level distributions, where θ are unknown parameters and $\Pr(\theta)$ is our prior over these parameters.

$$\Pr(Y_0, Y_1) = \int \prod_{i=1}^{N} \Pr(Y_{0i}, Y_{1i}|\theta) \Pr(\theta) d\theta$$

How to model $\Pr(Y_{0i}, Y_{1i}|\theta)$ is difficult and usually requires subject matter expertise. Though results can be robust to this modeling decision in randomized experiments as we are discussing here. Below we'll use a multivariate normal for this. How to choose the prior, $\Pr(\theta)$, also requires care. But results are generally pretty robust to this choice. We'll also use normals for this below.

In observational studies, we would also need to model the treatment assignment mechanism, $\Pr(D|Y_{0i}, Y_{1i})$, but randomization makes this unnecessary here. For our purposes, $\Pr(D|Y_{0i}, Y_{1i}) = 1/\binom{N}{N_t}$.[34]

The Bayesian approach consists of four parts to move from our observed data to a posterior distribution of causal quantity of interest given the observed data.

- Derive $\Pr(Y^{\text{mis}}|Y^{\text{obs}}, D, \theta)$.
- Derive $\Pr(\theta|Y^{\text{obs}}, D)$.
- Combine $\Pr(Y^{\text{mis}}|Y^{\text{obs}}, D, \theta)$ and $\Pr(\theta|Y^{\text{obs}}, D)$ to get $\Pr(Y^{\text{mis}}|Y^{\text{obs}}, D)$, by integrating over θ.
- Use $\tau(Y_0, Y_1)$ and $\Pr(Y^{\text{mis}}|Y^{\text{obs}}, D)$ to get $\Pr(\tau|Y^{\text{obs}}, D)$.

These steps follow the steps outlined in greater detail above. Once we have the posterior over τ (which could be ATE or ATT or any causal quantity of interest) we can then calculate the posterior mean or any other summary of the posterior.

In many settings analytical solutions to the above are not feasible or are very difficult due to having to evaluate a complicated intergral. A simulation approach, however, is easy to implement and more broadly applicable than analytic methods for deriving the posterior. This approach involves imputation of the missing potential outcomes. To do this, we use $\Pr(Y^{\text{mis}}|Y^{\text{obs}}, D, \theta)$ and $\Pr(\theta|Y^{\text{obs}}, D)$ to repeatedly impute the missing potential outcomes.

That is, we repeat the following steps many times to get a simulated posterior for τ.

- First, we draw $\theta \sim \Pr(\theta|Y^{\text{obs}}, D)$.
- Second, we input this value into $\Pr(Y^{\text{mis}}|Y^{\text{obs}}, D, \theta)$ to impute all of the missing potential outcomes. Note that we do not impute each missing potential outcome seperately. We impute all of them together without redrawing values for θ.
- This can be used to get an estimate of τ.

We can then take the average or other summary statistics of the simiulated posterior of τ to estimate our causal quantities of interest.

[32] This section draws on Imbens and Rubin (2015) Section 8.4-8.5.

[33] These include row-interchangability and appealing to Finetti's theorem. See Imbens and Rubin (2015) page 152.

[34] Note that $\sum_{i=1}^{N} D_i = N_t$.

6.3 Model Choices

We'll outline the modeling choices we've made here. We assume an underlying distribution for the joint distribution of the potential outcomes that is a multivariate normal distribution with parameter vector θ. In our analysis, we assume that the potential outcomes are independent (i.e., that correlation, ρ, is zero); we don't use the covariates; and we we assume different variances for treatment and control units.

$$\Pr(Y_{0i}, Y_{1i}|\theta) \sim \mathsf{Normal}\left(\begin{pmatrix}\mu_c \\ \mu_t\end{pmatrix}, \begin{pmatrix}\sigma_c^2 & \rho\sigma_c\sigma_t \\ \rho\sigma_c\sigma_t & \sigma_t^2\end{pmatrix}\right)$$

The model we assume for the unknown parameters, conditional on the observed data, $\Pr(\theta|Y^{\text{obs}}, D)$, is comprised of a prior over the unobesrved parameters and the likelihood of the observed data, given these parameters. That is, $\Pr(\theta|Y^{\text{obs}}, D) \propto \Pr(\theta)\Pr(Y^{\text{obs}}|D, \theta)$. Note that due to randomization we don't need to worry about modeling the treatment assignment.

We assume normal priors for $\mu_c, \mu_t, \sigma_c, \sigma_t$. Note that we actually assume that $\mu_c = \alpha$ and that $\mu_c = \alpha + \tau$ and place normal priors over α and τ.

$$\begin{aligned}\alpha &\sim \mathsf{Normal}(0, 100) \\ \tau &\sim \mathsf{Normal}(0, 100) \\ \sigma_c &\sim \mathsf{Normal}(0, 100) \\ \sigma_t &\sim \mathsf{Normal}(0, 100)\end{aligned}$$

The likelihood, $\Pr(Y^{\text{obs}}, D|\theta)$, can be derived[35] and is also normal. Note that because we only observe one potential outcome per unit, the likelihood does not depend on ρ.

$$\Pr(Y^{\text{obs}}|D, \theta) \sim \mathsf{Normal}(\alpha + D\tau, D\sigma_t + (1 - D)\sigma_c)$$

With the priors and likelihood, we are able to draw from $\Pr(\theta|Y^{\text{obs}}, D)$. We can then use the values we get for the parameters in the model for missing potential outcomes, conditional on the unknown parameters, $\Pr(Y^{\text{mis}}|Y^{\text{obs}}, D, \theta)$, which will also be normal. We can write down two seperate equations. One for imputing missing control potential outcomes; and one for imputing missing treatment potential outcomes.

$$\begin{aligned}\Pr(Y_{1i}|Y_{0i}, \theta, D_i = 0) &\sim \mathsf{Normal}\left(\mu_t + \rho\frac{\sigma_t}{\sigma_c}(Y_{0i} - \mu_c), \sigma_t^2(1 - \rho^2)\right) \\ \Pr(Y_{0i}|Y_{1i}, \theta, D_i = 1) &\sim \mathsf{Normal}\left(\mu_c + \rho\frac{\sigma_c}{\sigma_t}(Y_{1i} - \mu_t), \sigma_c^2(1 - \rho^2)\right)\end{aligned}$$

In the above, τ is the super-population average treatment effect and is our causal quantity of interest. We'll also calculate a finite population ATE.

6.4 Analysis with R and Stan

We now turn to analyzing the NSW data in R using Stan for sampling from the posterior distributions. Our goal is to replicate results from Table 8.6 in Imbens and Rubin (2015).[36] First, we define our Stan program. We do this in R, rather than creating an additional Stan program file for ease of presentation here. However, the Stan documentation suggests using a seperate file. The program inlcudes 4 blocks that correspond to the block types discussed above: `data`, `parameters`, `model`, and `generated quantities`. As mentioned above, Stan code reads like statistical notation and the details of the syntax should be intuitive. We define what

[35] See pages 158-159 of Imbens and Rubin (2015) for an example how such a likelihood is derived.

[36] This section follows Lee, Feller, and Rabe-Hesketh (2019).

the data will look like (and bounds for them), what the unknown parameters are (and bounds for them), what our priors are, what the likelihood is, and what the posterior looks like for missing potential outcomes. We also calculate the finite-sample ATE. The program is annotated for further elucidation.

```
sp =
"data {
  int<lower=0> N;                   // sample size
  vector[N] y;                      // observed outcomes
  vector[N] d;                      // treatment assignments
  real<lower=-1,upper=1> rho;       // assumed correlation between the potential outcomes
}
parameters {
  real alpha;                       // super-population control average
  real tau;                         // super-population average treatment effect
  real<lower=0> sigma_c;            // residual SD for the control
  real<lower=0> sigma_t;            // residual SD for the treated
}
model {
   alpha ~ normal(0, 100);          // prior over super-population control average
   tau ~ normal(0, 100);            // prior over super-population average treatment effect
   sigma_c ~ normal(0, 100);        // prior over residual SD for the control
   sigma_t ~ normal(0, 100);        // prior over residual SD for the treated
   y ~ normal(alpha + tau*d, sigma_t*d + sigma_c*(1 - d)); // likelihood of observed outcomes
}
generated quantities{
  real tau_fs;                      // finite-sample ATE
  real y0[N];                       // potential outcome if D = 0
  real y1[N];                       // potential outcome if D = 1
  real tau_unit[N];                 // unit-level treatment effect
  for(n in 1:N){
    real mu_c = alpha;              // super-population control average
    real mu_t = alpha + tau;        // super-population treatment average
    if(d[n] == 1){
      y0[n] = normal_rng(mu_c + rho*(sigma_c/sigma_t)*(y[n] - mu_t),
      sigma_c*sqrt(1 - rho^2));     // imputed missing control potential outcome for treated
      y1[n] = y[n];                 // observed treated potential outcome for treated
    }else{
      y0[n] = y[n];                 // observed control potential outcome for control
      y1[n] = normal_rng(mu_t + rho*(sigma_t/sigma_c)*(y[n] - mu_c),
      sigma_t*sqrt(1 - rho^2));     // imputed missing treated potential outcome for control
    }
    tau_unit[n] = y1[n] - y0[n];    // unit-level treatment effect
  }
  tau_fs = mean(tau_unit);          // finite-sample ATE
}"
```

We combine the data into a list to give to the Stan program. As mentioned above, the outcome variable is 1978 earnings. The treatment indicator is whether an individual participated in the jobs program. N is the number of individuals in the study. ρ is the correlation between potential outcomes; here we assume $\rho = 0$.

```
y = lalonde$re78
d = lalonde$treat
N = nrow(lalonde)
rho = 0
```

```
data = list(N = N, y = y, d = d, rho = rho)
```

Now we run the Stan program from above on the NSW data. This will create Markov chains that sample from the posterior over the model parameters (using the priors and likelihood) which are then used as inputs to the distribution over missing potential outcomes which allows us to impute the missing potential outcomes and therefore calculate the finite sample ATE. We choose to run with 4000 Markov chain iterations on four seperate Markov chains each with warm-up period of 500 iterations. Note that the results of running the Stan program will aggregate over all 4 chains and running seperate chains allows us to compare the behavior of the chains which allows us to check the robustness of the procedure.

```
model = stan(model_code = sp,data = data,iter = 4000, chains = 4,warmup = 500)
```

After running the Stan program, we print a summary of the results. First, it is important to note that the $\hat{R}$ statistic for all model parameters is 1. This indicates that there was not much variation across the four Markov chains and no evidence of pathological geometries in the state space. We see that all the chains converged. Note that the output of running the Stan program is not a single estimate for each parameter but a Markov chain that represents a simulated posterior distribution over each parameter. We are able to look at the mean value, the standard error of the mean value, the standard deviation of the distribution, as well as percentiles and effective degrees of freedom.

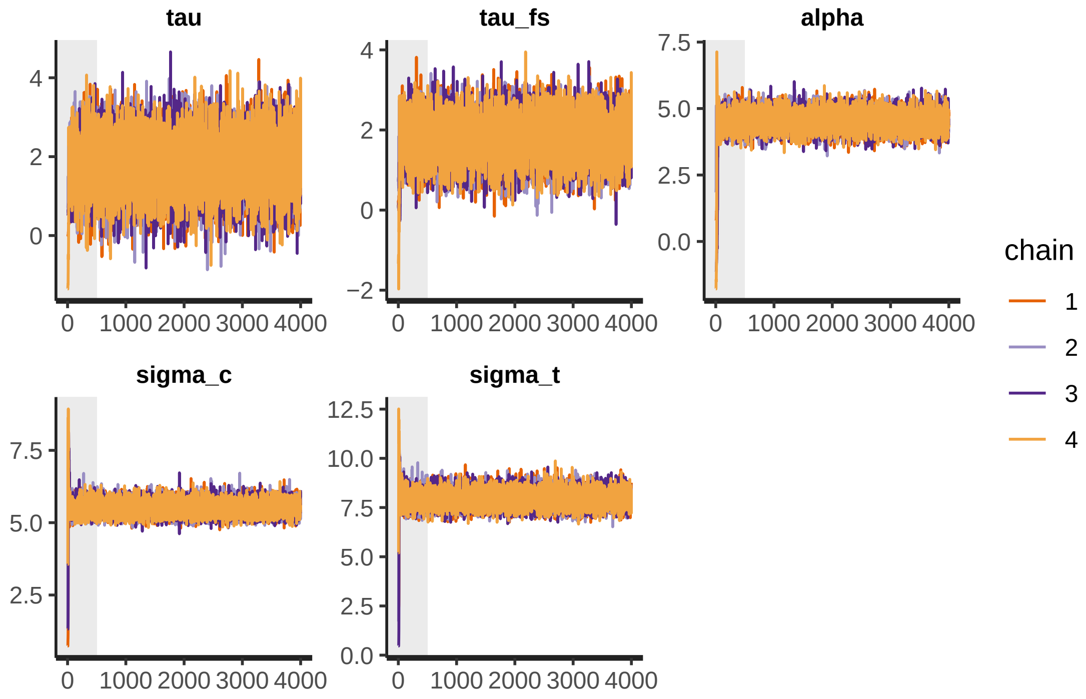
We see that our super-population estimate is similar to our difference in means estimate of ATE. The finite sample estimate is also similar. These also are comparable to the estimates in Table 8.6 in Imbens and Rubin (2015) and Lee, Feller, and Rabe-Hesketh (2019).

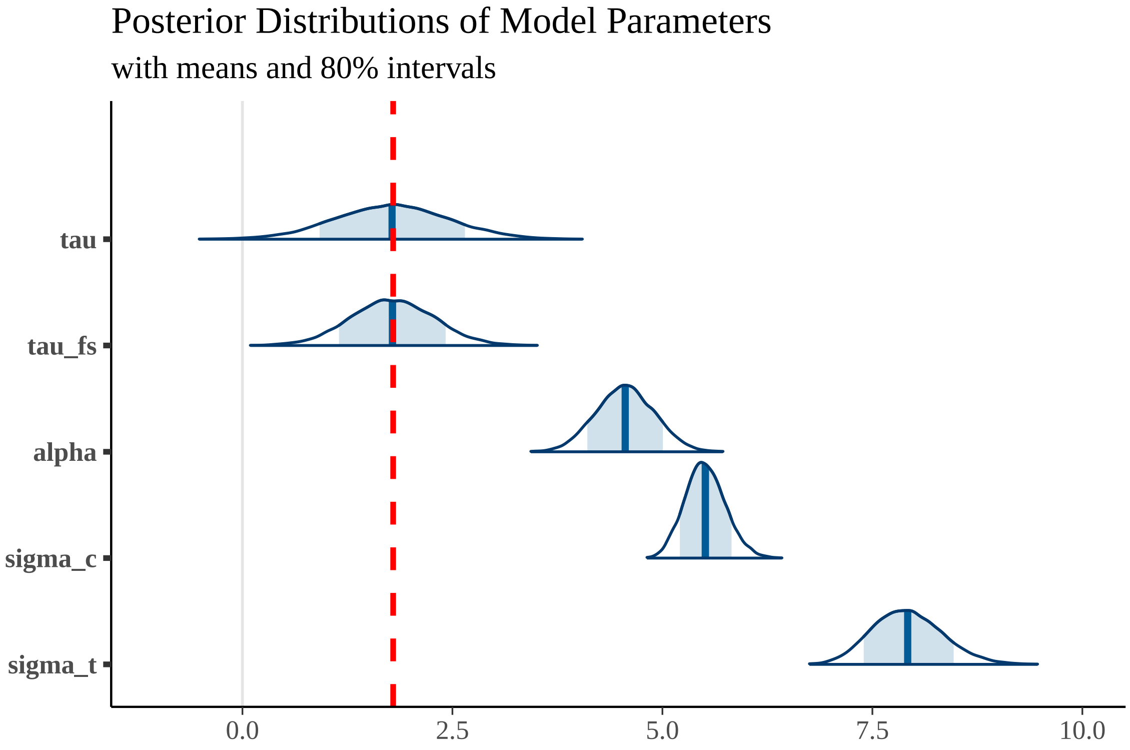
```
print(model, pars = c("tau", "tau_fs","alpha", "sigma_c", "sigma_t"),
      probs = c(0.1, 0.5, 0.9), digits = 3)
```

```
## Inference for Stan model: 14455e842847e41fd9a53e7834530fa5.
## 4 chains, each with iter=4000; warmup=500; thin=1;
## post-warmup draws per chain=3500, total post-warmup draws=14000.
##
##          mean se_mean    sd   10%   50%   90% n_eff  Rhat
## tau     1.782   0.006 0.680 0.919 1.784 2.649 12683 1.000
## tau_fs  1.786   0.004 0.502 1.150 1.782 2.419 14095 1.001
## alpha   4.557   0.003 0.348 4.106 4.556 5.006 12311 1.000
## sigma_c 5.511   0.002 0.241 5.208 5.502 5.823 12429 1.000
## sigma_t 7.921   0.004 0.419 7.397 7.903 8.468 12795 1.000
##
## Samples were drawn using NUTS(diag_e) at Thu Jun 11 18:33:43 2020.
## For each parameter, n_eff is a crude measure of effective sample size,
## and Rhat is the potential scale reduction factor on split chains (at
## convergence, Rhat=1).
```

We plot the actual samples across iterations, including the warm-up period. We can see that the Markov chains exhibit the expected behaviour: first exploration of the state space outside the typical set, then, once the typical set is found, exploration only within the typical set.

We also plot the posterior distributions of the model parameters. These show the same mean point estimates as discussed above, but make clear that what we actually are looking at is a full simulated posterior distribution over the model parameters. Hence, we are able to calculate any summary statistic of interest for the causal quantities of interest. We also plot a red line at the simple difference in means estimate from above. Note that here we see that the distribution for the finite-sample ATE is narrower than that for the super-popualtion ATE.





There are many other versions of this analysis that can be done. We could incorporate covariates, alter the assumptions we make, or alter the models we choose. See Imbens and Rubin (2015) and Lee, Feller, and Rabe-Hesketh (2019) for examples of some of the alternatives. Similar approaches can also be applied to observational studies.

7 Conclusion

This paper has explored the the conceptual foundations of and connections between the potential outcomes framework, Bayesian causal inference, Hamiltonian Monte Carlo, and probabilistic programming languages like Stan. We also saw how these ideas can be used together in a simple example application using the National Supported Work (NSW) experimental dataset. We followed Lee, Feller, and Rabe-Hesketh (2019) and Imbens and Rubin (2015) in exploring how Hamiltonian Monte Carlo and probabilistic programming languages can be used in model-based Bayesian causal inference, focusing on randomized experiments. The potential outcomes framework for causal inference can be applied in a model-based Bayesian approach, in which unobserved potential outcomes are viewed as random variables. Assumptions about the model for the joint distribution over the potential outcomes and the model for the treatment assignment mechanism can be used to derive a model for the posterior distribution over the unobserved potential outcomes. This allows us to impute the missing potential outcomes conditional on the observed data. We then estimate whichever causal quantities are of interest. We use a simulation approach to build a posterior distribution for the causal quantities of interest. We employ Hamiltonian Monte Carlo to do the required sampling to avoid issues more traditional sampling method can have in high dimensions. Specifically, the probabilistic programming language Stan allows us to write code that looks like statistical notation to model and sample from the posterior distribution over our unobserved potential outcomes. Thus, we are very easily able to implement the simulation approach to Bayesian inference. Causal inference and Hamiltoniam Monte Carlo have both become more and more influential over recent years. Innovations like probabilistic programming languages like Stan promise to push the capabilities of causal inference further than ever before.

References

Betancourt, Michael. 2017. “A Conceptual Introduction to Hamiltonian Monte Carlo.” http://arxiv.org/abs/1701.02434.

Gelman, Andrew, Daniel Lee, and Jiqiang Guo. 2015. “Stan: A Probabilistic Programming Language for Bayesian Inference and Optimization.” *Journal of Educational and Behavioral Statistics*. http://www.stat.columbia.edu/~gelman/research/published/stan_jebs_2.pdf.

Handcock, Mark. 2020. “Lecture 10 Hamiltonian Monte Carlo.” Department of Statistics, UCLA; Lecture Slides for Statistics 202C "Monte Carlo Methods".

Hastings, W. K. 1970. “Monte Carlo Sampling Methods Using Markov Chains and Their Applications.” *Biometrika* 57 (1): 97–109. http://www.jstor.org/stable/2334940.

Hazlett, Chad. 2020a. “Experiments.” Department of Political Science, UCLA; Lecture Slides for Political Science 200C "Causal Inference for Social Science".

———. 2020b. “The Potential Outcome Framework.” Department of Political Science, UCLA; Lecture Slides for Political Science 200C "Causal Inference for Social Science".

Imbens, Guido W., and Donald B. Rubin. 2015. *Causal Inference for Statistics, Social, and Biomedical Sciences: An Introduction*. Cambridge University Press. https://doi.org/10.1017/CBO9781139025751.

Lee, JoonHo, Avi Feller, and Sophia Rabe-Hesketh. 2019. “Model-Based Inference for Causal Effects in Completely Randomized Experiments.” *Stan Case Studies* 6.

Li, Fan. 2019. “Bayesian Causal Inference: A Tutorial.” Department of Statistical Science, Duke University; Bayesian Causal Inference Workshop, Ohio State University.

Metropolis, Nicholas, Arianna W. Rosenbluth, Marshall N. Rosenbluth, Augusta H. Teller, and Edward Teller. 1953. “Equation of State Calculations by Fast Computing Machines.” *The Journal of Chemical Physics* 21 (6): 1087–92. https://doi.org/10.1063/1.1699114.

Morgan, Stephen L., and Christopher Winship. 2014. *Counterfactuals and Causal Inference: Methods and Principles for Social Research*. 2nd ed. Analytical Methods for Social Research. Cambridge University Press. https://doi.org/10.1017/CBO9781107587991.

Rubin, Donald B. 1978. “Bayesian Inference for Causal Effects: The Role of Randomization.” *Ann. Statist.* 6 (1): 34–58. https://doi.org/10.1214/aos/1176344064.

Sampson, Adrian. 2016. “Probabilistic Programming.” Department of Computer Science, Cornell University; Lecture Slides for Computer Science 4110 "Programming Languages and Logics". http://adriansampson.net/doc/ppl.html.

Savage, Jim. 2020. “A Quick-Start Introduction to Stan for Economists.” QuantEcon; QuantEcon Notebook Library. https://nbviewer.jupyter.org/github/QuantEcon/QuantEcon.notebooks/blob/master/IntroToStan_basics_workflow.ipynb.

Stan, Development Team. n.d.a. “Stan.” https://mc-stan.org/.

———. n.d.b. “Stan Reference Manual: Hamiltonian Monte Carlo.” https://mc-stan.org/docs/2_23/reference-manual/hamiltonian-monte-carlo.html.