

Friday, June 2, 2023

Seeing the leaves through the forest

A primer on using random forest models for missing data problems

Anonymized for review

ABSTRACT Political scientists often struggle with decisions about what to do with incomplete cases for their regression analyses, and one can often make several decisions that influence one's ultimate substantive conclusions. In many areas of research outside of political science and the social sciences, scholars take advantage of an extension of multiple imputation, which offers the choice to leverage machine learning models for predicting values in missing data. This manuscript provides a summary of missing data and its consequences for our regression models along with providing an explanation of how to implement random forest models with an expanded form of the multiple imputation procedure, called multiple imputation with chained equation to handle complex causes for non-random missingness in our data. After providing a primer on standard missing data procedures in political science and random forest with multiple imputation with chained equations, I examine its performance on simulated data. I conclude by providing recommendations for dealing with missing data in practice.

KEYWORDS Missing data; Multiple imputation; Machine learning

Introduction

Missing values are common in social science data. **king_et-al_2001_apsr**<empty citation> estimate that political scientists lose about one-third of their data in their complete case regression analyses due to missing data. There are many reasons why missing values arise in our data. In surveys, these are referred to as item-nonresponse and pose threats to obtaining unbiased estimates for public opinion research when researchers utilize listwise deletion (LWD) in their regression analyses (**weisberg_2005_chicago**); mainly when the researcher can predict the cause of the missingness with an observed cause - which scholars refer to the data as “missing at random” (**king_et-al_2001_apsr**).¹ More generally, missing values come about in other types of data due to the unit’s (e.g., country, respondent, politician) attempt to obfuscate information, data collector error, or researcher error. When common and not from a stochastic data generating process (DGP), missing data pose threats to causal inference through omitted variable bias where the missing value in the variables included in a statistical model can be predicted by another variable in your model.

As this poses a serious threat to our ability to make conclusions about causal processes, several political scientists generate tools and learn from other fields to deal with these challenges due to the severity of the consequences for inference. There are four primary approaches that political scientists use in dealing with missing data: (1) listwise deletion or complete case analysis (LWD), (2) simple mean or median based imputation, (3) single regression-based imputation, or (4) multiple imputation (MI). Many of the most popular approaches to imputation carry assumptions that scholars have to satisfy. This manuscript aims to provide a primer to political scientists on

1. Which is a common occurrence, more so than missingness caused entirely by random chance - which, if one meets this condition, gives unbiased estimates with LWD.

the use of machine learning models in the Multiple Imputation with Chained Equations (MICE) framework - an extension of the familiar MI approach.

Machine learning with MICE is not a new approach. While a JSTOR search of Political Science journals reports a handful of articles that discuss this particular approach, it is certainly not widely used by political scientists nor has it received much accessible discussion of how it works under the hood.²

As the present article discusses, MICE allows the researcher to choose various models to aid the imputation process. Selecting a model with its underlying assumptions provides many benefits for researchers in choosing a procedure that is *most* appropriate for their particular circumstances. Choosing models that allow for flexibility and models that do not have a number of restrictive assumptions benefit those unsure about the exact data-generating process of the missingness. The specific emphasis in this paper on using random forest models in MICE (RF-MICE) results from the strong preference of those who use machine learning models for predictive regressions to use random forest models due to their flexibility as a result of their fully non-parametric nature and for their performance under many different, and complex, circumstances (montgomery_olivella_2018_ajps; james_et-al_2013_springer). Fields like the biomedical sciences treat missing values as out-of-sample predictions that random forest models predict; they see the purpose of imputation as aligned with a task that random forest models are optimal. As other fields use this RF-MICE procedure, there are implementations in STATA, R, and Python. All of which are relatively easy in terms of knowledge to code in either of these languages (buuren_goothuis-oudshoorn_2011_jss). To be clear, this manuscript argues that we should not only consider using the MICE framework for imputation but also should consider

2. Though, see marbach_2022_pa<empty citation>.

using Random Forest models.

This manuscript compares the utility of random forest models in the **MICE** procedure for imputation to other common imputation tools used in political science. In doing this, the manuscript encourages political scientists to consider this procedure when faced with conditions where missing data are present, and the other common tools seem unsuitable. It is not to paint these models as superior in absolute terms to other procedures for imputation. To do that is a waste of time given the variety of circumstances where some methodological tools are suitable in some circumstances and not for others. This manuscript provides a primer encouraging political scientists to consider this tool when faced with missing data and to give them enough background so they may be comfortable using it. Furthermore, the manuscript agrees with the common recommendation that practitioners should recognize the value of reducing one's dependence on a single procedure and contends that one must consider using multiple procedures to reduce the dependency of one's results on any given procedure. This article provides code snippets that readers can use to implement all of these procedures in R.

The next section reviews common approaches handling missing data that political scientists currently use. The next section describes machine learning and random forest models and links this to my claim of their utility for predicting missing data when used in the **MICE** procedure. I then move into applied examples where I examine the performance of these random forest models to other common approaches to dealing with missing data on simulated data. I then discuss recommendations for when one should use the reigning popular techniques or the random forest application in the **MICE** procedure.

Types of missing data, imputation, and MI in Political Science

MCAR, MAR, and MNAR

Missing data arise in different forms. Researchers describe missing data in three ways - often using somewhat unintuitive acronyms. The first form missing data takes is Missing Completely At Random (MCAR). This means that the data generating process for the missingness is random - there are no observed or unobserved causes of missingness. The second form missing data takes is Missing At Random (MAR). In MAR, these data are missing due to some observed cause. However, they are “Missing at Random” once you account for that observed cause of missingness. Some argue that MAR is much more common given the state of how large most contemporary social science data sets are (schunk__2008__atsa). The third form that missing data take is Missing Not At Random (MNAR).³ MNAR happens when observed and unobserved causes explain the missingness. What distinguishes MNAR from MAR is that the researcher does not have a clear path forward to handle the cause of missingness. This occurs either because the variable where there is missingness is, itself, a cause of the missingness, or data explaining the cause of missingness is unobserved by the researcher. Since missing data take different forms, researchers use a few different approaches to deal with these challenges. I summarize the types of missing data problems there are and their potential ramifications in Table 1 and the common tools used to solve these problems in Table 2.

Dealing with missingness

At the time of writing, king_et-al_2001_apsr<empty citation> estimated that 94% of political scientists use LWD to deal with missing data. In short, LWD does not seek to impute

3. Sometimes called Non-ignorable (NI).

missing values. Instead, if the dependent variable or any of the covariates in a regression model for a given observation are missing, the researcher does not include that observation in the analysis. Traditionally, scholars argue that LWD performs best (in terms of reducing the resulting bias in the researcher’s subsequent regression models) when the data are MCAR.

If the data are MAR or MNAR, deleting observations with missing data introduces bias in one’s regression estimates through a failure to account for correlation between the independent variable(s) and the error (king_et-al_2001_apsr; weisberg_2005_chicago; schunk_2008_atas; azur_et-al_2011_ijmpr). Furthermore, it has the potential to decrease statistical power. A meta-analysis of comparative and international political economy papers that use LWD demonstrates that political scientists have much, upwards of 50%, more Type I error - an incorrect rejection of the null hypothesis - than we would expect as a result of how we implement LWD (lall_2016_pa). To visualize this, we can draw a directed acyclic graph; Figure 1.

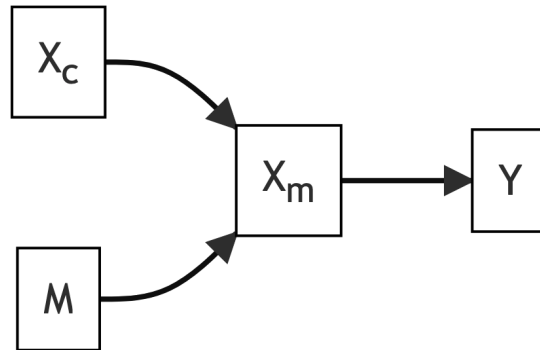


Figure 1: MAR data as confound

Others push against this claim and instead argue that LWD does not inherently generate bias for non-MCAR data but that researchers neglect to control for the cause of MAR or MNAR

(arel-bundock_pelc_2018_pa). This is still dependent on the researcher's grasp of theory and ability to identify the DGP leading to the missingness. Though the onus is on the researcher to do this, it is often a relatively high standard given the complexity to which social phenomena relate and the tendency for our datasets to be highly-dimensional. Furthermore, this also increases the number of parameters one must include in their statistical models - which runs the risk of increasing collinearity (schrodt_2014_jopr).

Like LWD, simple imputation techniques like mean-and-median-based imputation do not reduce the chances of biased regression estimates. These approaches, called interpolation and extrapolation, are common for panel data and cross-sectional time series data. If you have missing data for an observation in one panel, you can take the same observations' responses in a previous panel and a latter panel. You then take the mean or the median of that particular observation for that variable. Other approaches seek to reduce this MAR-based bias through conditioning on other variables.

Hot Deck approaches to imputing missingness are regression-based in that they define the dependent variable as the one the researcher is attempting to impute and use variables thought to predict the cause of missingness in MAR contexts (schunk_2008_atsea). In this approach, you often use a few variables to condition on. In many cases, the precise mechanism generating missingness is often tricky to triangulate. As a result, if you fail to provide the correct model specification when the data are MAR, you often end up with biased regression estimates.

MI seeks to solve this issue by using the entire dataset for imputing missing values (rubin_1996_jasa). This approach uses the other variables in the dataset to generate a joint posterior distribution of all possible missing values for that particular observation. Many assume that most social science data sets are sufficiently large enough to condition on the mechanism

generating MAR ([schunk_2008_atsa](#)). Unlike the other approaches, MI also generates uncertainty around the imputed values - via its construction of the joint posterior distribution ([rubin_1996_jasa](#)) - which enables the researcher to be more transparent about the validity of those imputed values and to include that uncertainty in the researcher's subsequent statistical analyses ([king_et-al_2001_apsr](#); [von-hippel_2015_semmj](#)). A prevalent implementation of MI in political science is the AMELIA II software ([amelia](#); [honaker_et-al_2011_jss](#); [lall_2016_pa](#)). This useful tool provides a computationally fast and simple process for imputation by taking advantage of bootstrapping with the EMB algorithm. Compared to the other approaches to missing data, AMELIA II performs quite well ([honaker_et-al_2011_jss](#); [kropko_et-al_2014_pa](#)). MI, however, often requires a set of distributional assumptions for the joint distribution - often the multivariate normal ([honaker_et-al_2011_jss](#)). Another challenge with this tool is that it runs up with the curse of dimensionality – if you are asking for more information by using more variables than you have observations, many non-regularized models will provide inaccurate estimates.

There is a variant to MI that seeks more computational efficiency and loosens some requirements. This variant is called Multiple Imputation through Chained Equations (MICE). MICE performs quite well for large imputation tasks. MI struggles to impute values when there is missingness in the other variables of the dataset as it estimates imputed variables based on the joint distribution as opposed to focusing each imputation by optimizing predictions for each variable ([kropko_et-al_2014_pa](#)). MICE tries to get around this limitation in a few steps, as described by [azur_et-al_2011_ijmpr](#)<empty citation>. Figure 2 provides a visual representation of the procedure for a form of MICE used in this manuscript. I include more details about random forest models in the following subsection.

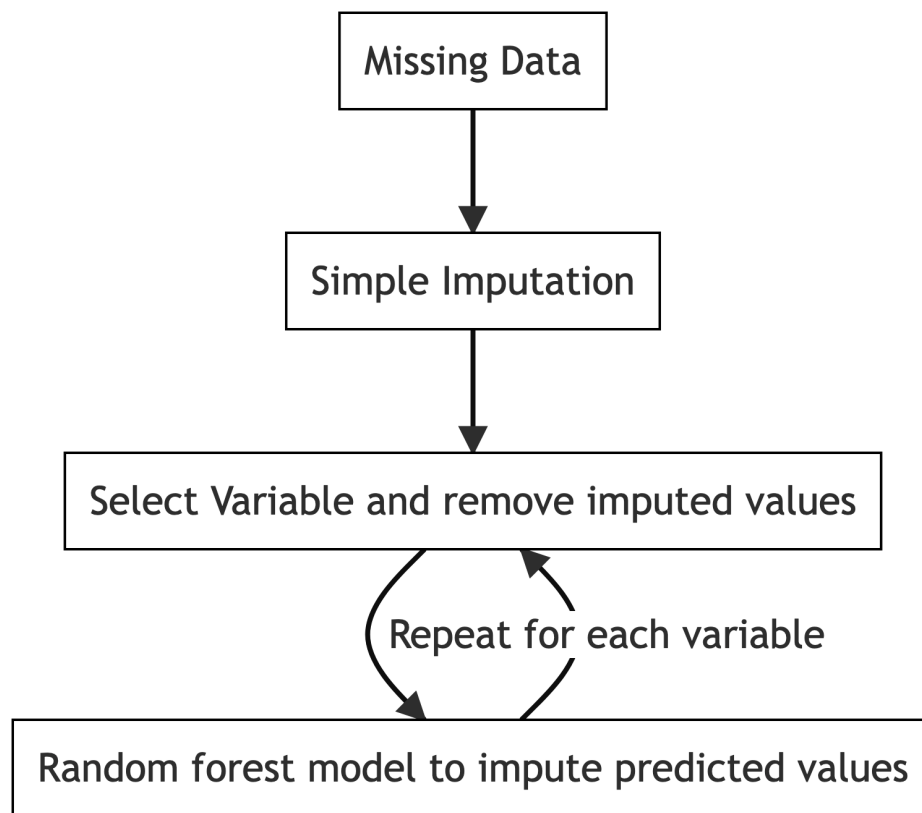


Figure 2: Steps of RF-MICE procedure

First, MICE performs a simple imputation, or interpolation, for every missing value in the entire dataset. These are the placeholder values. The second step in the general MICE paradigm involves identifying one variable to impute. Once complete, it then removes those placeholder values. The third step then involves regressing the observed values of the variable on the other variables in the model and replacing the predicted values generated from the regression model for the missing values. The fourth step is to repeat steps two and three for each variable in the data set with missing values - this constitutes a single iteration. As a fifth step, you perform between five and ten iterations.⁴

The advantage of this chained equation procedure is to estimate each variable as an outcome with its own regression model that is most appropriate for it. This means that the imputation task optimizes on each variable containing MAR data as opposed to optimizing the task for the whole dataset. Though MI as implemented in AMELIA is regression based, the regression models that one may use in MICE are as numerous as those a researcher may choose from when engaged in statistical analysis. This means that the assumptions and the performance of the model one uses for the imputation are the same as in standard statistical analyses. Though it decreases some of the requirements for modeling the MAR process, it is not entirely atheoretical. We can, however, reduce the dependence that the imputed data have on a researcher's ability to theorize about the MAR process by selecting models that are accustomed to dealing with a large number of parameters without increasing inefficiency.

One valuable model for allowing one to include a large number of parameters without losses to efficiency, is a form of ensemble machine learning model called Random Forests. As the

4. Though, the exact number of recommended iterations used in MICE are still up for debate (buuren_gothuis-oudshoorn_2011_jss; azur_et-al_2011_ijmpr). The recommendation is that you elect to go with more iterations if not constrained by computational limitations.

following discussion highlights, random forest models are optimal for engaging in predictive tasks (montgomery_olivella_2018_ajps), which appears appropriate for the task of predicting missing values. These models have the additional benefit of not requiring a multivariate normal distribution, not requiring one to specify a potentially incorrect model of which variables are included in the MAR process, nor is it in the form of a GLM. That is, these models reduce some of the dependence of an imputation task on the researcher’s beliefs about the source of the missingness. As researchers apply these random forest models within the MICE framework, they also benefit from the advantages that MI provide over the hot deck framework. Furthermore, machine learning models like ensemble procedures apply regularization to deal with highly dimensional datasets. These three features suggest that this procedure offers much more flexibility to the researcher.

Table 1: Types of missing data processes, problems, and solutions

Type	Cause	Problem
MCAR	Missing data patterns are stochastic	Does not cause bias in estimates
MAR	Missing data patterns are not stochastic; however, once accounting for observed causes of missingness, any remaining missing data are as-if random.	Generates a type of omitted variable bias without accounting for it.
MNAR	Missing data patterns are not stochastic; caused by unobserved sources.	Generates a type of omitted variable bias; hard to correct for as the source is unobserved.

Table 2: Common solutions for missing data

Procedure	Assumptions	What it does	Fixes
LWD	Sources of missingness are random.	Removes rows that have a missing value for <i>any</i> variable in the statistical model	MCAR
Simple	Sources of missingness may be explained by some type of autocorrelation (spatial, temporal, non-independence of observations).	Takes the average or median value of observations that occur either before (if temporal data), proximate to (if spatial data), or similar to (if in the same sample), and fills that value into all rows with missingness for that variable.	Data with autocorrelation or are not i.i.d.
Hotdeck	Regression-based. So assumptions made are dependent on the particular regression you choose for this procedure.	Fits a regression model (specified by the researcher) where the researcher regresses the variable with missing values on other columns in the dataset. Fills empty rows with their predicted values from the regression model. Assumes that the regression model is properly specified.	MAR

Procedure	Assumptions	What it does	Fixes
MI	Assumes a normally distributed joint posterior distribution. Assumes that a GLM is appropriate for describing the non-stochastic sources of missingness.	Fits a Bayesian Linear Regression. Iteratively regresses each variable containing missing values onto all other variables in the dataset. Completes this process multiple times to account for uncertainty in the particular construction of the posterior distribution.	MAR

Procedure	Assumptions	What it does	Fixes
RFMICE	<p>MICE, in general, is constrained by the assumptions of the particular estimator one chooses. RFMICE, however, is designed to loosen a number of assumptions. Still in the Frequentist framework so it does not produce posterior distributions to reflect uncertainty. Uncertainty is reflected by variation within and between iterations.</p>	<p>First performs simple imputation on each variable that has missingness. Then fits a random forest model where it attempts to predict each variable containing missingness using all other variables present in the dataset. Within each iteration, Random Forest models perform their own iterative procedures to maximize predictive capacity using cross-validation. It performs this for each variable and produces some user specified number of datasets which are the result of those maximally predicted values for each missing value.</p>	MAR

The utility of random forest models for imputing political science data

Random forest models are concerned with calculating a fixed out-of-sample prediction under the supervised machine learning framework. They are popular among data scientists and researchers primarily interested in providing predictions instead of engaging in causal inference. In non-imputation applications of supervised machine learning models - a broader category of machine learning models that includes random forests, these models take partial, observed information about an outcome and estimate the relationship between the units with observed outcomes and several other observed features for those units. Then for the units without an observed outcome, we generalize that relationship to predict an outcome for them.

To get an intuition of the fundamental goal of a supervised machine learning model, I will provide an analogy. If we are looking to sort beans into a good or bad pile before we toss them into a pot, we often want to collect information about them. Features like color, size, and plumpness can all be good indicators of whether a bean will taste good or bad. Say we have over 5,000 beans, and we are a chef at a restaurant approaching the dinner hour, and we do not have time to sort all these beans. To save time, we look at these features like color, size, and plumpness and make two piles - good or bad for a subset of our 5,000 beans; say, in this instance, about 25%. We then get one of our employees to sort the rest of the beans for us. This employee may know less than us about what features matter more and how to identify a bean that will taste good or bad. Nevertheless, they can look at the two piles we have already made and try to pick up on patterns that make good and bad beans different from one another. With this information, the employee can “learn” these patterns so that even without the same knowledge as the chef, they

can still make predictions about whether a bean will taste good or bad.

We have some units with recorded observations for a particular variable for imputation. Leveraging this, we can treat these documented observations as information so that we can “train” our computer to find a relationship between the observed information in that variable and information from other variables for that unit. We can then generalize this relationship to predict what that value would be for a missing unit. This seems like a reasonable approach to thinking about imputing missing values. We are not making naive imputations by taking the mean value. As we are using an expanded form of MI, we can also use all variables in the dataset to clarify that pattern. As a MICE procedure, we can specify a machine learning model, and we do this iteratively so that if we have more than one variable that we want to impute, we are not limited to accurate imputed values only for the units that are complete except for that particular variable to be imputed. I will elaborate more on how this works in a few paragraphs. Before we go there, however, I want to take the time to provide a few more details about how random forest models work, as they do not represent the whole of supervised machine learning.

Practitioners refer to random forests as tree-based ensemble models. As a supervised machine learning model, they start with the basic intuition described above; but in the process of “learning” or “training,” these models follow a few distinct steps. Decision tree models split regions of the predictive space. For example, say we want to predict vote choice by one’s partisanship. We would split the predictive space into regions. These regions could be different degrees of partisanship, like strong Republicans and weak Republicans. When we split this predictive space into regions, we essentially are subsetting our dataset of partisans into these different areas of the predictive space and trying to optimize a model to provide the best predictions in each region in our predictive space. That is, we try to find a model that maximizes the predictive performance of vote choice

for strong Republicans, weak Republicans, independents, and so on. We examine performance by comparing how well we are predicting the unobserved outcomes relative to the observed outcomes within those predictive regions. We can “bag” our trees. When we “bag” the decision trees, we bootstrap the training sample and build a tree for each bootstrapped sample and average across them. Doing this allows for a decrease in the variance of the predictions coming from the model, which helps with reducing the chances of generating a trained model that will fail to adequately generalize to our data set not included in the training step.

As this is a primer for the applied researcher, I note that this discussion simplifies decision trees and random forest models. For those interested in more details about these concepts, **james_et-al_2013_springer**<empty citation> provide a helpful discussion of these concepts. The main point is that random forest models specialize in generating predictions by optimizing at the *value* and *not* the *variable* level. This characteristic suggests that these models have the potential to be powerful tools for many different applications.

We can discuss their applicability to imputation in the MICE framework with a foundation in how random forest models work. As random forest models specialize in generating fixed out-of-sample predictions, these models have a joint goal with multiple imputation in that it should not be evaluated on the model’s correctness but on the model’s ability to predict a fixed (true) value (**rubin_1996_jasa**). Here, one might think of missing data as the out-of-sample predictions intended to be estimated. With random forest models, you train the model on a training data set (often randomly generated through cross-validation), a randomly selected portion of your data that you train the model on, and then fit the model on the testing set, the remaining data not used for the training stage of your model (**hastie_et-al_2009_springer**).

OLS models often perform relatively poorly on making out-of-sample predictions as they are

BLUE, assuming the data on hand are relatively representative of the population. As we have MAR data, this assumption likely fails, and any out-of-sample predictions are likely to be biased due to overfitting. Further, OLS may also generate out-of-bounds predictions for non-continuous data (long_1997_sage; gelman_et-al_2021_cup) which is particularly troublesome in settings of prediction.

Other models provide within-bounds predictions, such as logistic models; however, as generalized linear models, they still assume a linear functional form and often produce biased interpretations of the likelihood function when presented with unobserved systematic processes (mood_2010_esr). Though OLS and Logistic regression underlie a lot of machine learning as tools, many consider them to have limited applicability to complicated settings requiring prediction. Random forest models provide within-bounds predictions, and they are fully non-parametric (hastie_et-al_2009_springer), meaning they do not assume a functional form and consequently a joint distribution. This means that we have more flexibility in terms of what variables we have in our datasets that we need to impute. As social scientists, we rarely have datasets that contain DGPs that fall neatly within the optimal realm for GLMs. This added flexibility by using MICE and random forests makes the researcher's job easier.

On other performance metrics, random forest models, as an ensemble method, provide much more accurate predictions than single-tree alternatives in machine learning, such as CART (montgomery_olivella_2018_ajps). As discussed above, rather than generating a single estimate from a single model, ensemble models, like random forests, calculate multiple models and learn from their performance; this is the purpose of using the bootstrapped samples.

In the context of using MICE, I argue that political scientists should *consider* using random forest models to make accurate predictions with fewer assumptions and be more lenient in terms

of conditioning on the cause of MAR in the data set. Recall that within each MICE iteration, one performs a model predicting (imputing) a missing value based on the other variables in the dataset. Explicitly, this means you are running a model per variable with missing data.

Given that random forests are non-parametric, within each MICE iteration, the relationship between the variable to be imputed and those used to make the predictions can be non-linear and can take many different multivariate distributional forms. This is a significant advancement on traditional MI, which assumes a multivariate normal distribution. Additionally, this is an advancement on other MICE models that political scientists use, which may not inherently conceptualize missing data as these unobserved values to predict from a generalized relationship of the observed variable with the other variables in the dataset. These relationships are also not assumed to be linear. Furthermore, using RF-MICE has the advantage over hot-deck procedures for those unsure about a variable's precise DGP for MAR.

In the next section, I illustrate the use of random forest models for political scientists by demonstrating a simulated application of a random forest implementation of MICE. The following section also compares this implementation's performance to other common approaches to handling missing data in political science in terms of our ability to reduce unobserved bias in our data and in computational costs.

The performance of RF-MICE with simulations

Using R (**R**), I simulate a population where $N = 1000000$. The population has 5 variables (excluding a row index variable). The data generating process (DGP) of these variables are presented in Equation 1. I then use the **infer** (**infer**) package to take 1000 random samples from that population where the size of each sample is $n = 100$.

$$\begin{aligned}
a_i &= \text{Gamma}(2, 2) \\
b_i &= \text{Binomial}(1, 0.6) \\
x_i &= 0.2 \times a_i + 0.5 \times b_i + \text{Normal}(0, 1) \\
z_i &= 0.9 \times a_i \times b_i + \text{Normal}(0, 1) \\
y_i &= 0.6 \times x_i + 0.9 \times z_i + \text{Normal}(0, 1)
\end{aligned} \tag{1}$$

For each sample, I use the **miceRanger** (**miceranger**) package to “ampute” the data to introduce missingness for 40% of the observations that follows a **MAR** pattern. The documentation of this package suggests that the amputation utilizes a logistic regression to generate a **MAR** pattern for each variable. This is advantageous to the **mice** (**mice**) package’s amputation function which forms the **MAR** pattern for each variable with a linear regression. As tools such as AMELIA assume a MVN for the imputation, I would not expect that there would be many differences in performance between different imputation procedures. Complicating the **MAR** process should help distinguish the limitations and benefits of the imputation procedures.

I use the **ggplot2** (**ggplot2**) and **ggridges** (**ggridges**) packages to display the distributions of the original sample data and the amputed data. As I have 1000 samples, I try to simplify the plots by presenting the density distributions of the X, Y, and Z variables for 10 samples. The density distributions for the original data are presented in Figure 3 and the density distributions for the amputed data are presented in Figure 4.

With these amputed datasets, I then apply some of the procedures I have discussed to impute these values. Interpolation is a quite simple procedure where I can fill in missing values by using the mean value of that particular variable for the non-missing observations. I perform this interpolation with the **mice** package (**mice**) and iterate over it to provide 10 datasets. I also

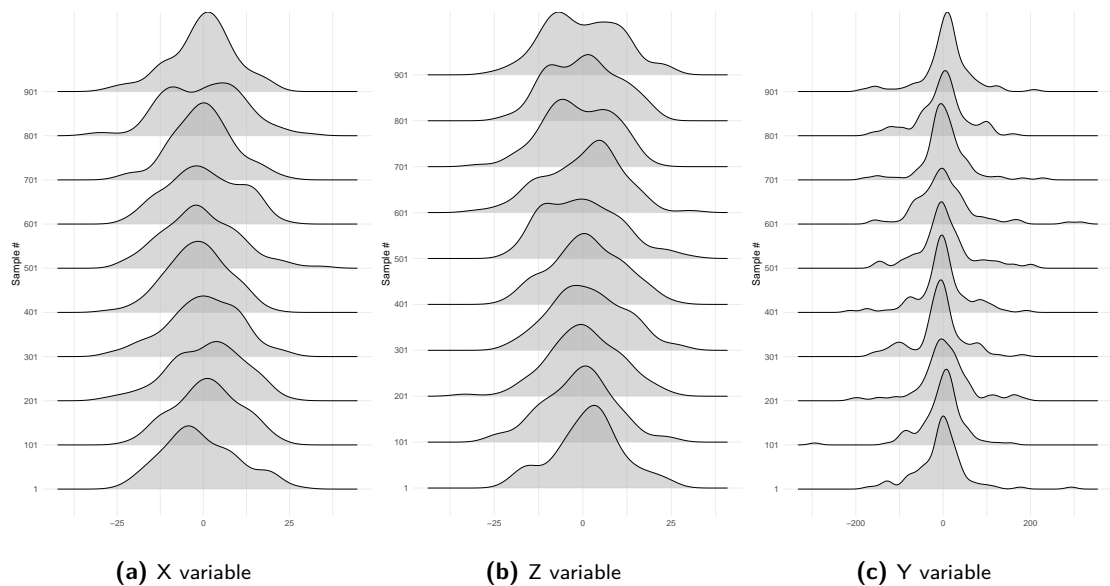


Figure 3: Distributions of complete sample data

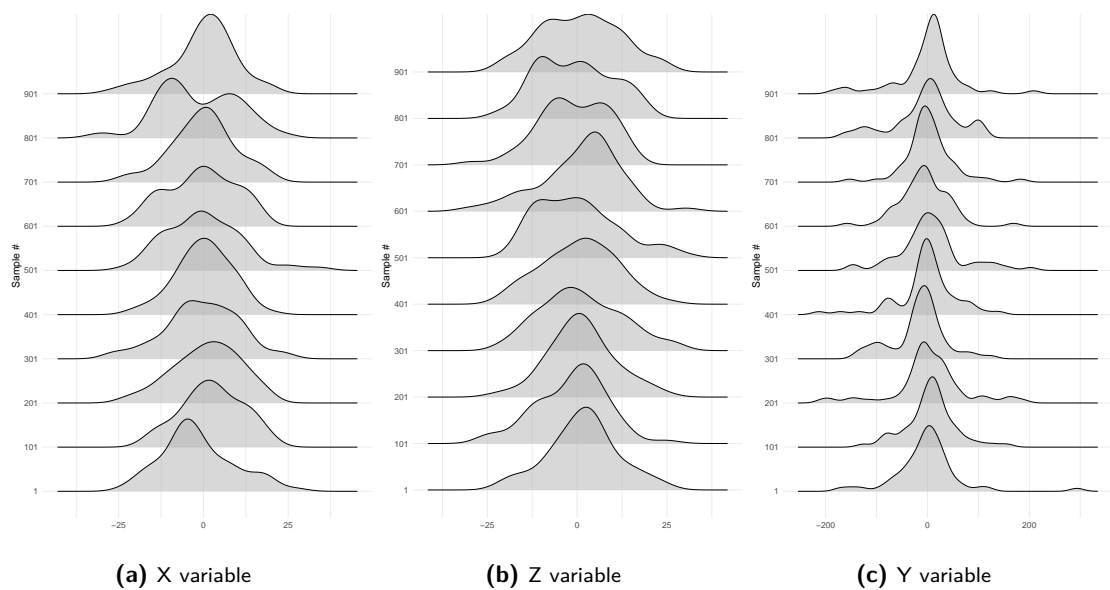


Figure 4: Distributions of imputed sample data

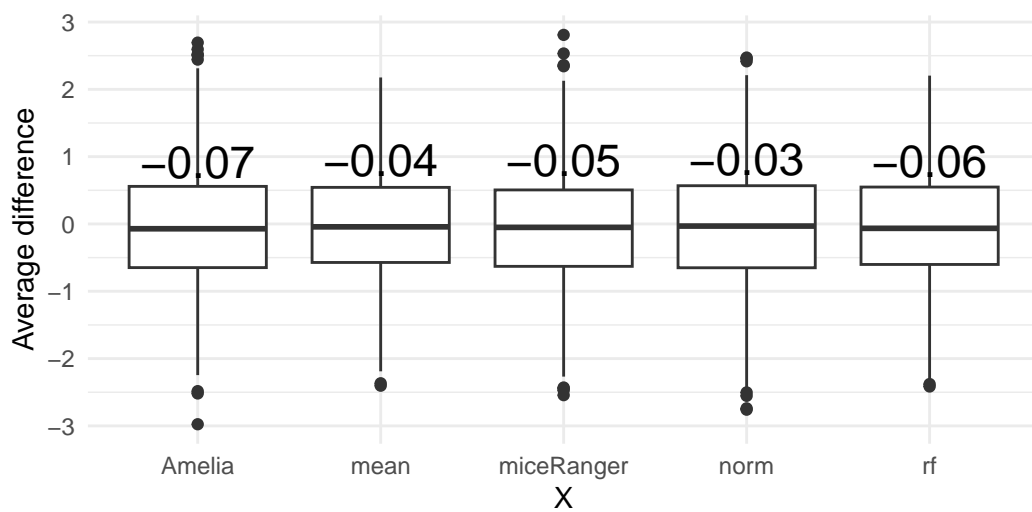
use the `AMELIA` II package ([honaker_et-al_2011_jss](#)) to perform standard MI and also store 10 datasets from the iterations. I use a standard Bayesian linear model in the MICE framework with the `mice` package ([mice](#)). Bayesian linear models with uniform distributions or a weak prior distribution are similar to the familiar Ordinary Least Squares ([gelman_et-al_2021_cup](#)). As discussed before, the final procedure I use is a random forest in the MICE framework. I perform the RF-MICE procedure using the `mice` package ([mice](#)) and an alternative package `miceRanger` ([miceRanger](#)). I provide an example code block to do this in the supplementary information.

When producing the imputed datasets, I use the `tictoc` package to record the amount of time each procedure takes to complete the task on the 1000 samples as a measure of computational cost.⁵ As a number of factors may affect the absolute computational costs for these procedures (e.g., hardware, whether other applications or software are running, whether one uses parallelization, etcetera), I am primarily going to focus on the relative computational costs of each procedure.

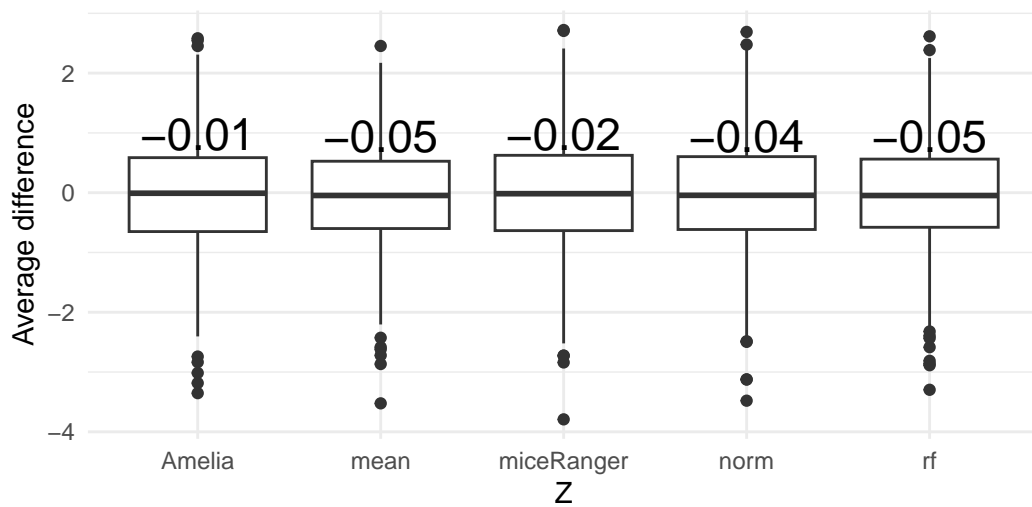
Each imputation procedure produced $m = 10$ datasets per simulated dataset, $s = 1000$. I have a total of $s \times m$ datasets. For each s dataset, I took the difference the values for each m dataset from the values in the complete dataset and took the average of these differences across the 10 imputed, m , datasets for each sample to give me a mean score of the discrepancy for each s dataset. Using the `ggplot2` package ([ggplot2](#)) I produce Figure 5, which represents these mean discrepancy scores for the three variables that were originally imputed. The text on each boxplot represents the median sample’s average discrepancy for that particular procedure.

Overall, we see that the procedures do quite well in that the average difference between the actual data and the imputed data are quite small across the datasets. We see that RF-MICE

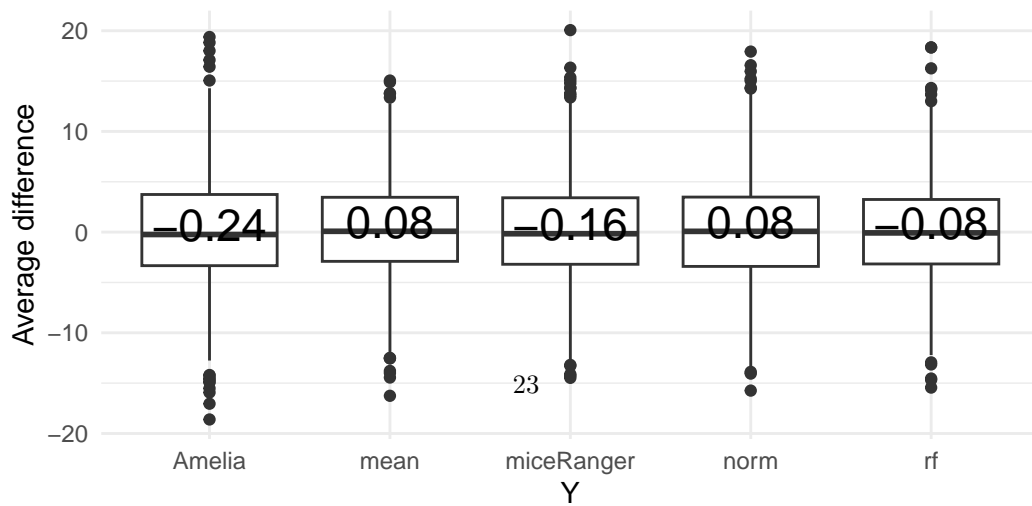
5. It is important to note that these benchmarks are based on a computer 16 GB of RAM, with a Apple Silicon M2 Pro Processor and a 10-core graphics card.



(a) X variable



(b) Z variable



(c) Y variable

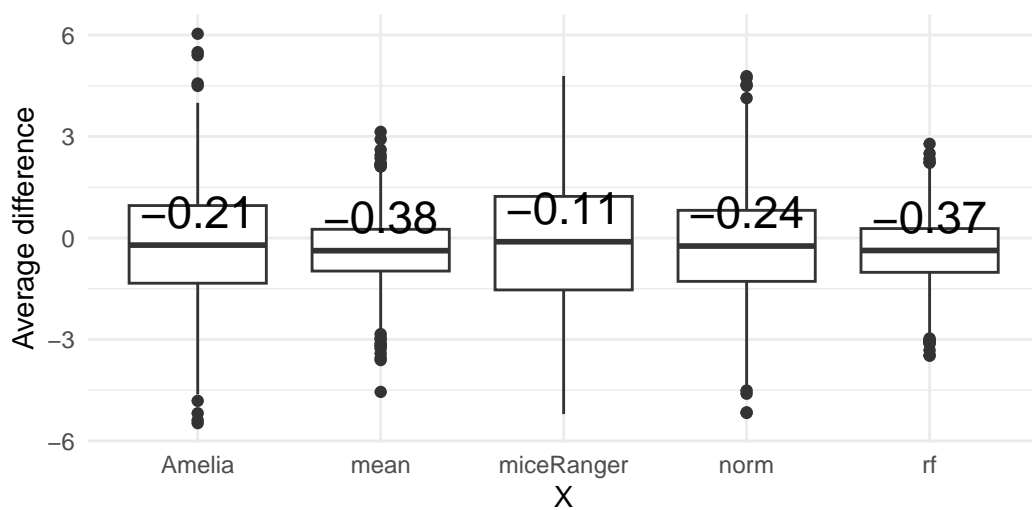
Figure 5: Discrepancies between original and imputed data

when implemented with `miceRanger` (**miceranger**) consistently does a good job at coming closer to the correct value than the other procedures do. RF-MICE when implemented in the `mice` (**mice**) package does a poorer job at this, however. In terms of speed to execute the imputation, interpolation with the `mice` package took an average of 0.085 seconds; Amelia took an average of 0.022 seconds; Linear-MICE took an average of 0.101 seconds; RF-MICE, as implemented by `mice` (**mice**), took an average of 0.397 seconds; and RF-MICE, as implemented by `miceRanger` (**miceRanger**), took an average of 1.597 seconds.

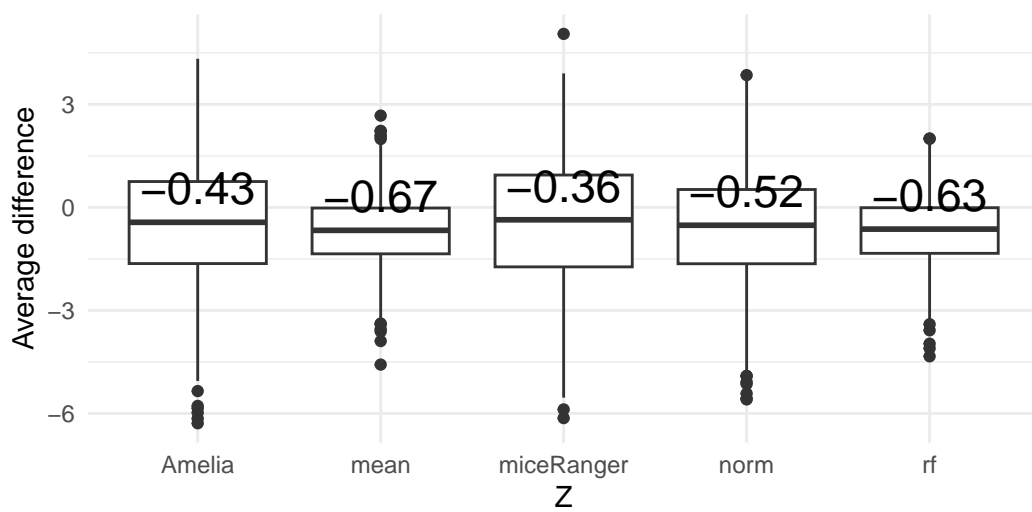
Though it is not a novel claim, I argue that in situations where we have missingness due to a MAR pattern, our regression models suffer from bias due to the systematic process generating that missingness. What I have argued so far is that we should consider using RF-MICE as it is, relative to other MI tools, a flexible tool that may be able to model a number of MAR processes which would help with reducing bias in our regression models.

To examine this claim, I take the amputed and imputed datasets (10) and use Rubin’s rule (`rubin_1996_jasa`) to pool across the regression models performed on each amputed and imputed sample. I then calculate the discrepancy by subtracting the parameter value from the point estimate.

Figure 6 present the distribution of differences between my point estimate and of my parameter value for the β coefficient for X and for Z respectively. This figure demonstrates, again, that RF-MICE, as implemented in `miceRanger` (**miceRanger**), does a relatively good job at reducing levels of bias in my eventual statistical analyses. RF-MICE through the `mice` (**mice**) package, however, performs worse than AMELIA and Linear-MICE.



(a) X variable



(b) Z variable

Figure 6: Discrepancies of estimates between original and imputed data

Examining these procedures with the 2022 CES

How would I implement these procedures with a common political science dataset? This section demonstrates how to do this with the common content from the 2022 Cooperative Election Study (CES). While I am unable to compare the ability for these procedures to reduce bias as I do not have complete data, the next section provides computational benchmarks for each of these procedures on a common dataset in political science as well as estimates the Z-score difference between these procedures to demonstrate whether resulting differences between these procedures' performance has any meaningful difference that the practitioner should be concerned with.

Conclusions

In theory, RF-MICE is a quite flexible tool that can operate in a number of circumstances to not only discover systematic processes leading to missingness in one's data, but to also use such information to recover the values. These expectations rely on the claim that Random Forest models are optimized for discovering patterns and to use that information to make out-of-sample predictions. With RF-MICE Random forests are coupled with MICE to produce significant improvements at retrieving the true values when a MAR process is present.

Using simulated data, I demonstrate two implementations of RF-MICE in R and compare it to implementations of more common procedures for dealing with MAR processes in political science. Overall, my simulated data and a number of measures of performance favor those theoretical expectations.

When comparing the distribution of imputed samples to the true samples, as if the MAR process was not present, both RF-MICE implementations do quite well in recovering the true values that

were missing. When examining the discrepancy between what the imputed value is and the true value, the two **RF-MICE** implementations provide quite small discrepancies on average.

Measuring the performance of **RF-MICE** in terms of reducing bias in statistical estimation, I find that **RF-MICE**, when implemented with the **miceRanger** package (**miceRanger**), stands out as a valuable tool for researchers to use to reduce bias generated with **MAR** processes.

While in theory, it is nice for the applied researcher to hear about a new and powerful tool or procedure, they face many constraints. When examining how much time each implementation of a procedure took, I observed that both implementations of **RF-MICE** were not significantly worse in time it took to complete the procedure relative to other multiple imputation procedures.

Altogether, the simulated data suggest that **RF-MICE** is a valuable procedure for the applied researcher's toolbox. When dealing with missing data that one suspects may not be the result of a **MCAR** process, one should consider the flexibility that **RF-MICE** provides. It does not require the researcher to specify the variables involved in the **MAR** process, but also does not introduce significant computational costs nor introduce so much complexity as a procedure that it is impossible to anticipate or diagnose problems the procedure may introduce.

It is important to remind the reader that all tools have their limitations and that their value are quite dependent on the context for which they are to be applied. It is useful, however, to include tools that vary in the assumptions they make (**neumayer_plumper_2017_cup**). The capabilities of **RF-MICE** are no different. While **RF-MICE** is quite flexible for addressing a number of **MAR** processes, it is important to note that it is not and should not be seen as a default or the sole tool for one to use when dealing with missing data. For example, simulations demonstrate that **RF-MICE** performs poorly when the missing data pattern arises from a moderating relationship (**marbach_2022_pa**). With the number of tools and their implementations being so available to

the applied researcher, one should not shy away from using multiple implementations of these tools to ensure that one's substantive conclusions are not dependent on one tool or implementation.