Thursday, July 20, 2023 Working paper. Please do not distribute without author consent.
Replication materials are stored at https://github.com/DamonCharlesRoberts/mice-imputation-psci

Seeing the leaves through the forest*

A primer on using random forest models for missing data problems

Damon Charles Roberts †
University of Colorado Boulder
damon.roberts-1@colorado.edu

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

*Word Count (including references): 5705. I would like to thank Madeline Mader, Alexandra Siegel, Andrew Q. Philips, and Jennifer Wolak for their advice and many conversations during the development of this project as well as Andy Baker for offering me the space to write the manuscript and for his feedback. I would also like to thank the discussants and panelists at MPSA for their useful feedback and encouragement.

†Corresponding author.

Introduction

Missing values are common in social science data. King et al. (2001) estimate that political scientists lose about one-third of their data in their complete case regression analyses due to missing data. This manuscript has two goals. First, to convince political scientists that the consequences of missing data should be avoided. Without the first goal, the second goal of the manuscript is likely to not be of use. The second goal of the manuscript is to provide an accessible introduction to the use of random forest models to fix a common pattern of missing data.

While implementations of random forest models to solve missing data problems are not new, discussions of these tools are not readily accessible to most quantitative political scientists [1] as these procedures are popular in fields such as Statistics and Bioinformatics.[2] In the following sections, I review the consequences of incomplete data in one's analysis, then I review existing techniques, then I describe how random forest models work and describe the intuition behind their implementation to solve familiar missing data problems. I finish the manuscript with simulations comparing different implementations of random forest models for imputation alongside common approaches to handling missing data in political science.

[1] Though see Marbach (2022) for an example in political science.

[2] Using JSTOR's text mining tool on journals in political science reports that there were only about 40 published articles since 2010 that discuss "MICE" – the common imputation framework that random forest models are implemented in – and "imputation". Link to search.

Types of missing data: MCAR, MAR, and MNAR

Missing data are not equal. The first form missing data takes is Missing Completely At Random (`MCAR`). This means that the data generating process for the missingness is random. The second form, Missing at Random (`MAR`), refers to data that are missing to some observed cause that can be accounted for. This type of missing data is argued to be the most likely type of data we encounter given factors like the interdependence of outcomes in our social world as well as the size of our datasets that leave fewer unobserved variables on the table (Schunk 2008). The third form is called Non-ignorable (`NI`), or sometimes Missing Not at Random (`MNAR`). `MNAR` refers to a pattern of missingness that includes both observable and unobservable causes. **?@tbl-summary-missing-processes** in the Supplementary Materials includes a summary of these, along with their consequences.

Dealing with missingness

One estimate suggests that 94% of political scientists employ to handle missing data of all types is referred to as Listwise Deletion (`LWD`) (King et al. 2001). `LWD` is a common default for statistical software that we use for fitting statistical models and essentially excludes observations from our analysis if the dependent variable or any of the covariates contain missing values. In cases of `MCAR` this approach is appropriate as the exclusion of an observation from an analysis would be random.

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; Azur et al. 2011). 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). Figure 1 visualizes this with a DAG.

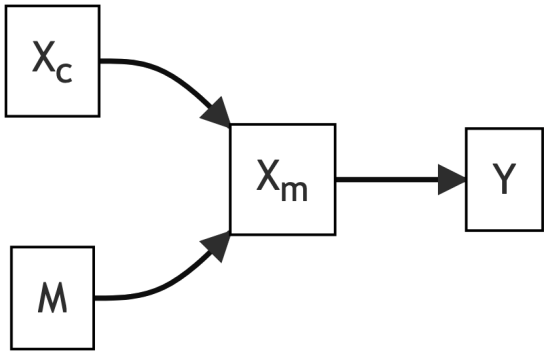


Figure 1: MAR data as confound

One step researchers take is to take the mean or median values of the column in cross-sectional settings or in panel settings to take the mean of that variable over time. These interpolation and extrapolation approaches are an improvement over `LWD` in that it uses information from complete cases to provide an estimate of what that missing value should be. In cross-sectional and panel data, these approaches still have some limitations in that it makes strong assumptions about the comparability of that particular missing value for that particular observation.

With Hot Deck imputation, you replace a missing value for a column with that of an observation that has similar observed characteristics (Andridge and Little 2010). As you

are deciding the imputed values based on observed data, this theoretically it is desirable for situations where you have data that are MAR. Despite how straight-forward the procedure sounds, like matching methods, there is significant debate about the best way to determine whether another observation is similar enough to the one with the missing data (Andridge and Little 2010). Also in datasets where missing data are common, it becomes tricky to find complete observations that are similar to those that are incomplete.

MI seeks to solve these issues by imagining the task of imputation in a prediction framework (Rubin 1996). This approach often uses some implementation of Multi-Chain Monte Carlo (MCMC) to leverage the other variables in the dataset to generate a joint posterior distribution of all possible missing values for that particular observation. A popular implementation of MI in political science is the AMELIA II software (James Honaker, King, and Blackwell 2011; J. Honaker, King, and Blackwell 2011; Lall 2016). 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 (J. Honaker, King, and Blackwell 2011; Kropko et al. 2014). MI, however, often requires a set of distributional assumptions for the joint distribution - often the multivariate normal (J. Honaker, King, and Blackwell 2011). 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.

Multiple Imputation through Chained Equations (MICE) is a variant of MI that seeks more computational efficiency and loosens some requirements. 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). `MICE` tries to get around this limitation in a few steps, as described by Azur et al. (2011). 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.

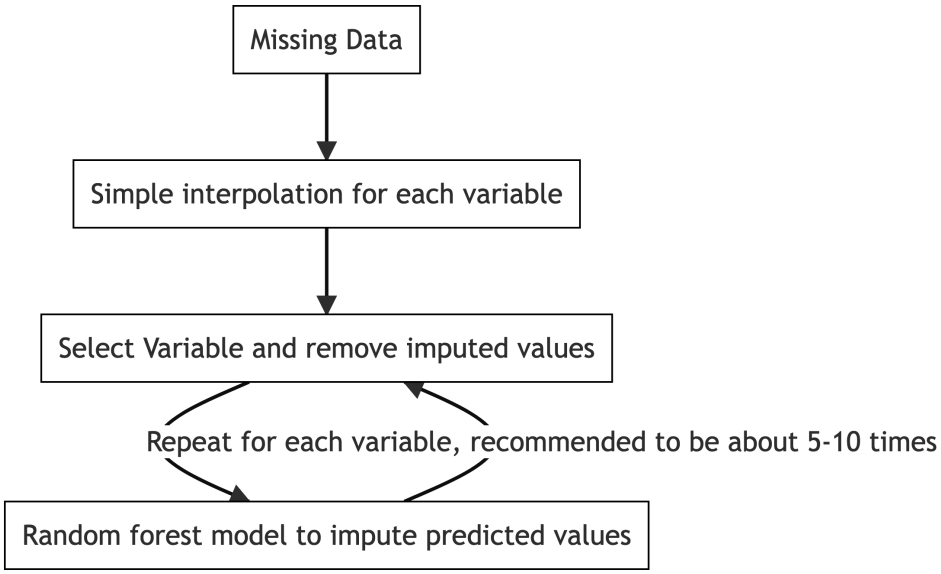


Figure 2: Steps of RF-MICE procedure

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. As `MICE` is regression-based, options for the underlying algorithm to estimate those imputed values are as numerous as our choices for regular statistical analysis. The focus of this manuscript is to examine an extremely flexible regression technique called random forest models.

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) 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 in terms of explanation of the `MAR` process, but on the model's ability to predict a fixed (true) value (Rubin 1996). 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, Tibshirani, and Friedman 2009).

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; Gelman, Hill, and Vehtari 2021) 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). 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, Tibshirani, and Friedman 2009), 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 `DGP`s that fall neatly within the optimal realm for `GLM`s. 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 and Olivella 2018). 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 performince of `RF-MICE` with simulations

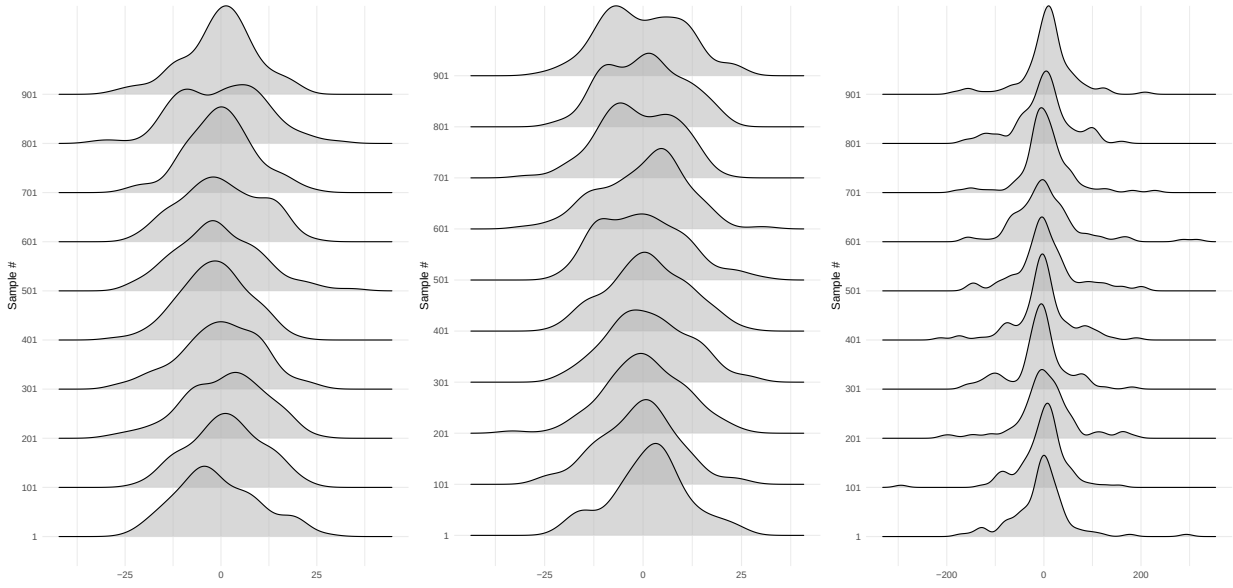
Using `R` (R Core Team 2022), I simulate a population where $N = 1000000$. The population has 5 variables (excluding a row index variable). The data generating process (DGP) of these variables are presented in Equation 1. I then use the `infer` (Couch et al. 2021) package to take 1000 random samples from that population where the size of each sample is $n = 100$.

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

For each sample, I use the `miceRanger` (Wilson 2021) 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` (van Buuren and Groothuis-Oudshoorn 2011) package's amputation function, which forms the `MAR` pattern for each variable with a linear regression, as it provides a more complex `MAR` pattern for the tools to solve. 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; though, it should not cause dramatically more complex processes.

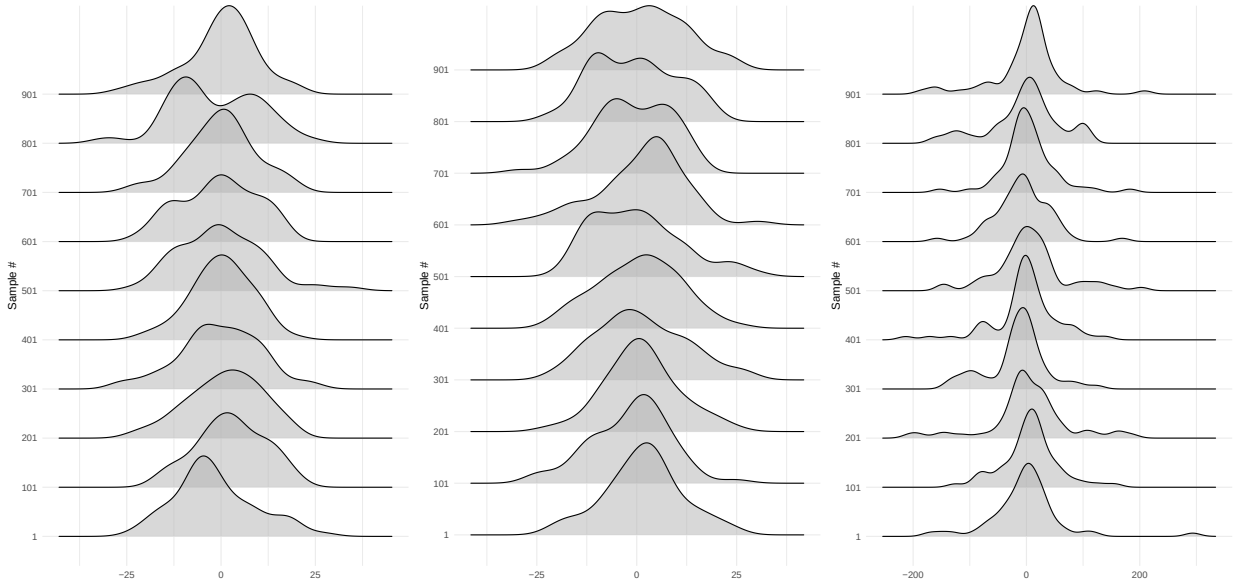
I use the `ggplot2` (Wickham 2016) and `ggridges` (Wilke 2022) packages to display the distributions of the original sample data and the amputed data. As I have 1000 samples, I try to simplify the plots by presenting the density distributions of the X, Y, and Z variables for 10 samples. The density distributions for the original data are presented in Figure 3 and the density distributions for the amputed data are presented in Figure 4.



(a) X variable **(b)** Z variable **(c)** Y variable

Figure 3: Distributions of complete sample data

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 (van Buuren and Groothuis-Oudshoorn 2011) and iterate over it to provide 10 datasets. I also use the `AMELIA II` package (J.



(a) X variable **(b)** Z variable **(c)** Y variable

Figure 4: Distributions of amputed sample data

Honaker, King, and Blackwell 2011) 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 (van Buuren and Groothuis-Oudshoorn 2011). Bayesian linear models with uniform distributions or a weak prior distribution are similar to the familiar Ordinary Least Squares (Gelman, Hill, and Vehtari 2021). 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 (van Buuren and Groothuis-Oudshoorn 2011) and an alternative package `miceRanger` (Wilson 2021). 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.[3] As a number of factors may affect the absolute computational costs

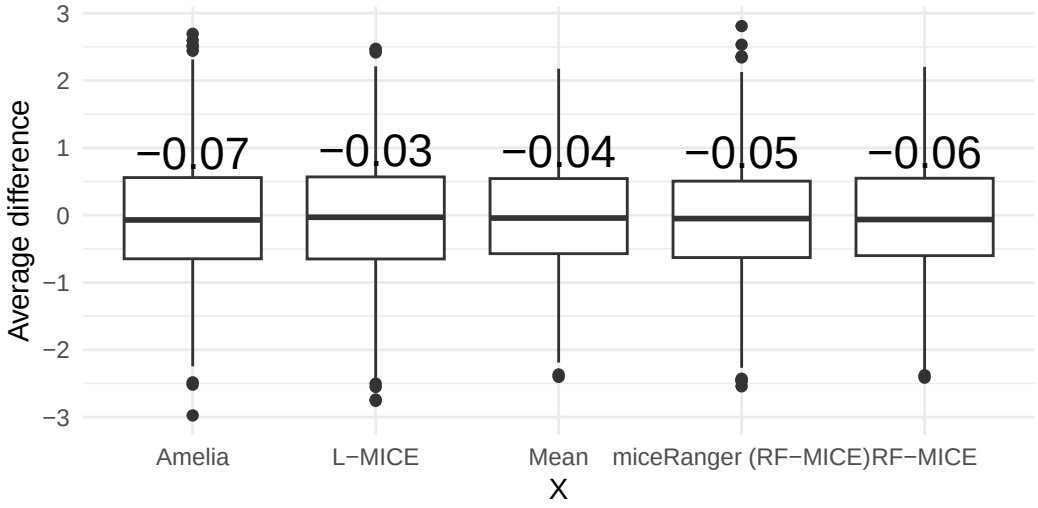
[3]It is important to note that these benchmarks are based on a computer 16 GB of RAM, with a Apple Silicon M2 Pro Processor and a 10-core graphics card.

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.

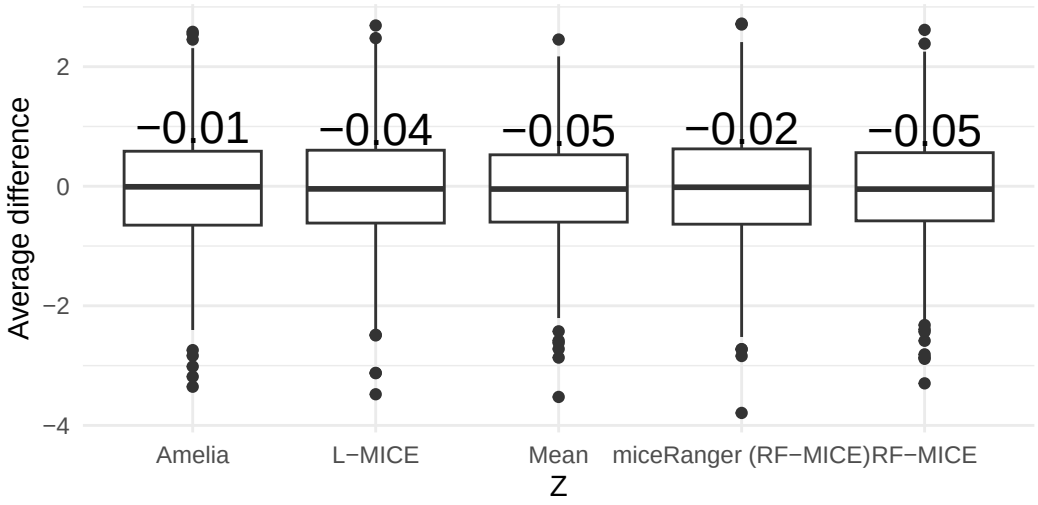
Each imputation procedure produced $m = 10$ datasets per simulated dataset, $s = 1000$. I have a total of $s \times m$ datasets. For each s dataset, I took the difference the values for each m dataset from the values in the complete dataset and took the average of these differences across the 10 imputed, m, datasets for each sample to give me a mean score of the discrepancy for each s dataset. Using the `ggplot2` package (Wickham 2016) 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` when implemented with `miceRanger` (Wilson 2021) consistently does a good job at coming closer to the correct value than the other procedures do. `RF-MICE` when implemented in the `mice` (van Buuren and Groothuis-Oudshoorn 2011) 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 0.086 seconds; Amelia took an average of 0.022 seconds; `Linear-MICE` took an average of 0.101 seconds; `RF-MICE`, as implemented by `mice` (van Buuren and Groothuis-Oudshoorn 2011), took an average of 0.347 seconds; and `RF-MICE`, as implemented by `miceRanger` (Wilson 2021), took an average of 6.857 seconds.

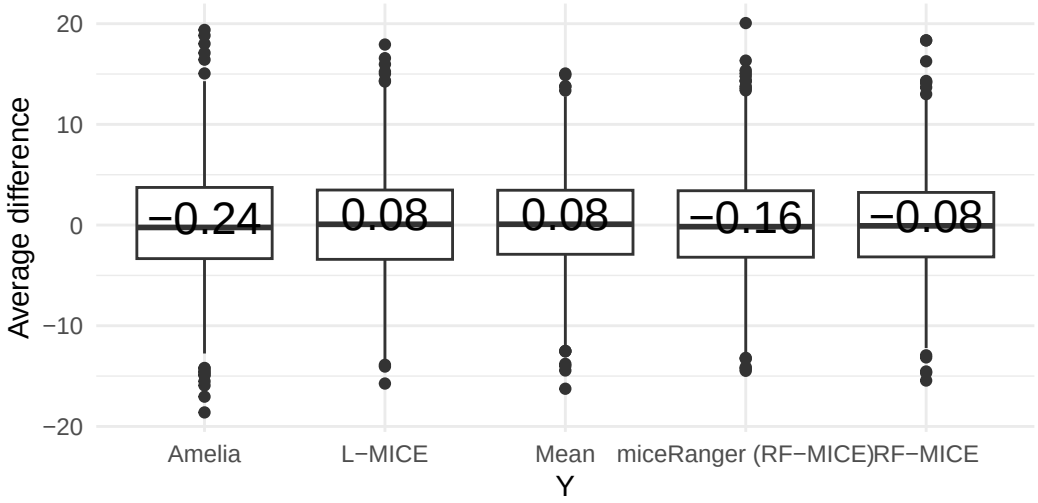
Though it is not a novel claim, I argue that in situations where we have missingness due to a `MAR` pattern, our regression models suffer from bias due to the systematic process generating



(a) X variable



(b) Z variable



(c) Y variable

Figure 5: Discrepancies between original and imputed data

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.

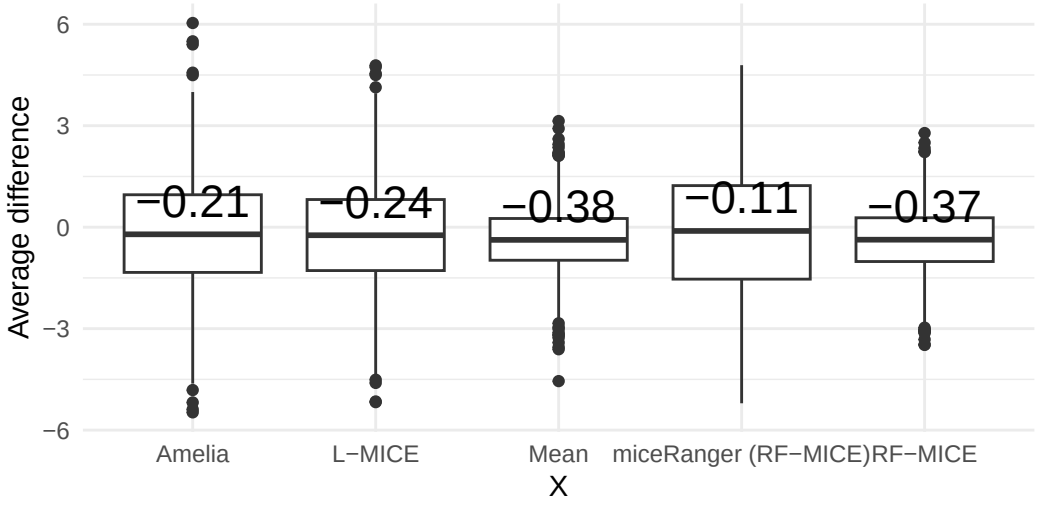
To examine this claim, I take the amputed and imputed datasets (10) and use Rubin's rule (Rubin 1996) to pool across the regression models performed on each amputed and imputed sample. I then calculate the discrepancy by subtracting the parameter value from the point estimate.

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 (Wilson 2021), does a relatively good job at reducing levels of bias in my eventual statistical analyses. RF-MICE through the mice (van Buuren and Groothuis-Oudshoorn 2011) package, however, performs worse than AMELIA and Linear-MICE.

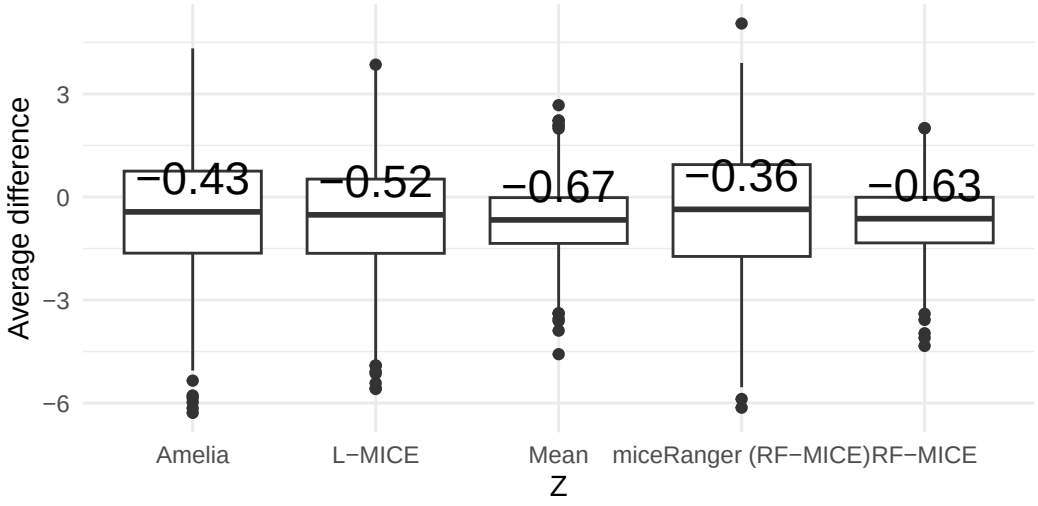
Conclusions

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



(a) X variable



(b) Z variable

Figure 6: Discrepancies of estimates between original and imputed data

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, RF-MICE implementationed with miceRanger does 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 (Wilson 2021), 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 and Plümper 2017). 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). 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.

References

Andridge, Rebecca R., and Roderick J. A. Little. 2010. "A Review of Hot Deck Imputation for Survey Non-Response." *International Statistical Review* 78 (1): 40–64. https://doi.org/10.1111/j.1751-5823.2010.00103.x.

Azur, Melissa J., Elizabeth A. Stuart, Constantine Frangakis, and Philip J. Leaf. 2011. "Multiple Imputation by Chained Equations: What Is It and How Does It Work?" *International Journal of Methods in Psychiatric Research* 20 (1): 40–49. https://doi.org/10.1002/mpr.329.

Couch, Simon P., Andrew P. Bray, Chester Ismay, Evgeni Chasnovski, Benjamin S. Baumer, and Mine Çetinkaya-Rundel. 2021. "infer: An R Package for Tidyverse-Friendly Statistical Inference." *Journal of Open Source Software* 6 (65): 3661. https://doi.org/10.21105/

joss.03661.

Gelman, Andrew, Jennifer Hill, and Aki Vehtari. 2021. *Regression and Other Stories*. New York: Cambridge University Press.

Hastie, Trevor, Robert Tibshirani, and Jerome Friedman. 2009. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. 2nd ed. Springer Series in Statistics. New York: Springer.

Honaker, James, Gary King, and Matthew Blackwell. 2011. "Amelia II: A Program for Missing Data." *Journal of Statistical Software* 45 (7): 1–47. https://doi.org/10.18637/jss.v045.i07.

Honaker, J, G King, and M Blackwell. 2011. "Amelia II: A program for missing data, R package version 1.5., 2012." *Journal of Statistical Software* 45 (7): 1–3.

James, Gareth, Daniela Witten, Trevor Hastie, and Robert Tibshirani. 2013. *An Introduction to Statistical Learning: With Applications in r*. New York: Springer.

King, Gary, James Honaker, Anne Joseph, and Kenneth Scheve. 2001. "Analyzing Incomplete Political Science Data: An Alternative Algorithm for Multiple Imputation." *American Political Science Review* 95 (1).

Kropko, Jonathan, Ben Goodrich, Andrew Gelman, and Jennifer Hill. 2014. "Multiple imputation for continuous and categorical data: Comparing joint multivariate normal and conditional approaches." *Political Analysis* 22 (4): 497–519. https://doi.org/10.1093/pan/mpu007.

Lall, Ranjit. 2016. "How Multiple Imputation Makes a Difference." *Political Analysis* 24 (4): 414–33. https://doi.org/10.1093/pan/mpw020.

Long, Scott J. 1997. *Regression Models for Categorical and Limited Dependent Variables*.

Advanced Quantitative Techniques in the Social Sciences Series. Thousand Oaks, CA: Sage Publications.

Marbach, Moritz. 2022. "Choosing Imputation Models." *Political Analysis* 30 (4): 597–605.

Montgomery, Jacob M, and Santiago Olivella. 2018. "Tree-Based Models for Political Science Data." *American Journal of Political Science* 62 (3): 729–44. https://doi.org/10.1111/ajps.12361.

Mood, Carina. 2010. "Logistic regression: Why we cannot do what We think we can do, and what we can do about it." *European Sociological Review* 26 (1): 67–82. https://doi.org/10.1093/esr/jcp006.

Neumayer, Eric, and Thomas Plümper. 2017. *Robustness Tests for Quantitative Research*. New York: Cambridge University Press.

R Core Team. 2022. "R: A Language and Environment for Statistical Computing." Vienna, Austria: R Foundation for Statistical Computing. https://www.R-project.org/.

Rubin, Donald B. 1996. "Multiple Imputation After 18+ Years." *Journal of the American Statistical Association* 91 (434): 473–89.

Schunk, Daniel. 2008. "A Markov Chain Monte Carlo Algorithm for Multiple Imputation in Large Surveys." *AStA* 92 (1): 101–14. https://doi.org/10.1007/s10182-008-0053-6.

van Buuren, Stef, and Karin Groothuis-Oudshoorn. 2011. "mice: Multivariate Imputation by Chained Equations in r." *Journal of Statistical Software* 45 (3): 1–67. https://doi.org/10.18637/jss.v045.i03.

Wickham, Hadley. 2016. *Ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York. https://ggplot2.tidyverse.org.

Wilke, Claus O. 2022. *Ggridges: Ridgeline Plots in 'Ggplot2'*. https://CRAN.R-project.org/

package=ggridges.

Wilson, Sam. 2021. "miceRanger: Multiple Imputation by Chained Equations with Random Forests." https://CRAN.R-project.org/package=miceRanger.

Andridge, Rebecca R., and Roderick J. A. Little. 2010. "A Review of Hot Deck Imputation for Survey Non-Response." *International Statistical Review* 78 (1): 40–64. https://doi.org/10.1111/j.1751-5823.2010.00103.x.

Azur, Melissa J., Elizabeth A. Stuart, Constantine Frangakis, and Philip J. Leaf. 2011. "Multiple Imputation by Chained Equations: What Is It and How Does It Work?" *International Journal of Methods in Psychiatric Research* 20 (1): 40–49. https://doi.org/10.1002/mpr.329.

Couch, Simon P., Andrew P. Bray, Chester Ismay, Evgeni Chasnovski, Benjamin S. Baumer, and Mine Çetinkaya-Rundel. 2021. "infer: An R Package for Tidyverse-Friendly Statistical Inference." *Journal of Open Source Software* 6 (65): 3661. https://doi.org/10.21105/joss.03661.

Gelman, Andrew, Jennifer Hill, and Aki Vehtari. 2021. *Regression and Other Stories*. New York: Cambridge University Press.

Hastie, Trevor, Robert Tibshirani, and Jerome Friedman. 2009. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. 2nd ed. Springer Series in Statistics. New York: Springer.

Honaker, James, Gary King, and Matthew Blackwell. 2011. "Amelia II: A Program for Missing Data." *Journal of Statistical Software* 45 (7): 1–47. https://doi.org/10.18637/jss.v045.i07.

Honaker, J, G King, and M Blackwell. 2011. "Amelia II: A program for missing data, R

package version 1.5., 2012." *Journal of Statistical Software* 45 (7): 1–3.

James, Gareth, Daniela Witten, Trevor Hastie, and Robert Tibshirani. 2013. *An Introduction to Statistical Learning: With Applications in r*. New York: Springer.

King, Gary, James Honaker, Anne Joseph, and Kenneth Scheve. 2001. "Analyzing Incomplete Political Science Data: An Alternative Algorithm for Multiple Imputation." *American Political Science Review* 95 (1).

Kropko, Jonathan, Ben Goodrich, Andrew Gelman, and Jennifer Hill. 2014. "Multiple imputation for continuous and categorical data: Comparing joint multivariate normal and conditional approaches." *Political Analysis* 22 (4): 497–519. https://doi.org/10.1093/pan/mpu007.

Lall, Ranjit. 2016. "How Multiple Imputation Makes a Difference." *Political Analysis* 24 (4): 414–33. https://doi.org/10.1093/pan/mpw020.

Long, Scott J. 1997. *Regression Models for Categorical and Limited Dependent Variables*. Advanced Quantitative Techniques in the Social Sciences Series. Thousand Oaks, CA: Sage Publications.

Marbach, Moritz. 2022. "Choosing Imputation Models." *Political Analysis* 30 (4): 597–605.

Montgomery, Jacob M, and Santiago Olivella. 2018. "Tree-Based Models for Political Science Data." *American Journal of Political Science* 62 (3): 729–44. https://doi.org/10.1111/ajps.12361.

Mood, Carina. 2010. "Logistic regression: Why we cannot do what We think we can do, and what we can do about it." *European Sociological Review* 26 (1): 67–82. https://doi.org/10.1093/esr/jcp006.

Neumayer, Eric, and Thomas Plümper. 2017. *Robustness Tests for Quantitative Research*.

New York: Cambridge University Press.

R Core Team. 2022. "R: A Language and Environment for Statistical Computing." Vienna, Austria: R Foundation for Statistical Computing. https://www.R-project.org/.

Rubin, Donald B. 1996. "Multiple Imputation After 18+ Years." *Journal of the American Statistical Association* 91 (434): 473–89.

Schunk, Daniel. 2008. "A Markov Chain Monte Carlo Algorithm for Multiple Imputation in Large Surveys." *AStA* 92 (1): 101–14. https://doi.org/10.1007/s10182-008-0053-6.

van Buuren, Stef, and Karin Groothuis-Oudshoorn. 2011. "mice: Multivariate Imputation by Chained Equations in r." *Journal of Statistical Software* 45 (3): 1–67. https://doi.org/10.18637/jss.v045.i03.

Wickham, Hadley. 2016. *Ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York. https://ggplot2.tidyverse.org.

Wilke, Claus O. 2022. *Ggridges: Ridgeline Plots in 'Ggplot2'*. https://CRAN.R-project.org/package=ggridges.

Wilson, Sam. 2021. "miceRanger: Multiple Imputation by Chained Equations with Random Forests." https://CRAN.R-project.org/package=miceRanger.