Friday, July 21, 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 Though missing data is pervasive in political science datasets, attempting to regain information from it remains a relatively uncommon step in data pre-processing. While there are many options out there, the benefits and drawbacks each provide can make it difficult to discern which to use. This note has two goals. First, to provide a review of the consequences of missing data and to provide a reference for common options used by political scientists. The second goal of the note is to advocate for the uptake of using random forest models in the Multiple Imputation with Chained Equations framework. In doing so, it lays out the intuition of these models and how that fits with the task of imputing missing data while also comparing the use of this implementation to other common approaches used in political science with simulated data that are representative of political science data.

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 (`RF`) 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. Table 1 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.

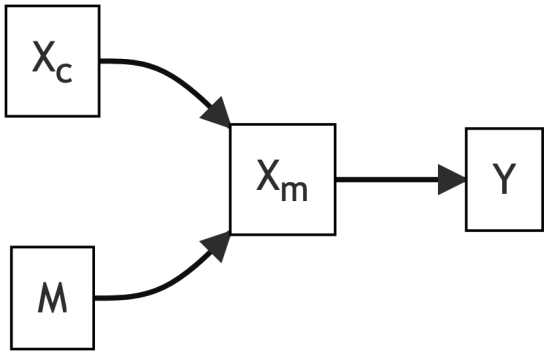


Figure 1: MAR data as confound

An improvement calculates 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 as it makes strong assumptions about the comparability of that cell with the others.

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, theoretically it is desirable for situations

where you have data that are MAR. 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. 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 (Honaker, King, and Blackwell 2011). 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, King, and Blackwell 2011). MI, however, often requires a set of distributional assumptions for the joint distribution - often the multivariate normal (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 RF in the following subsection.

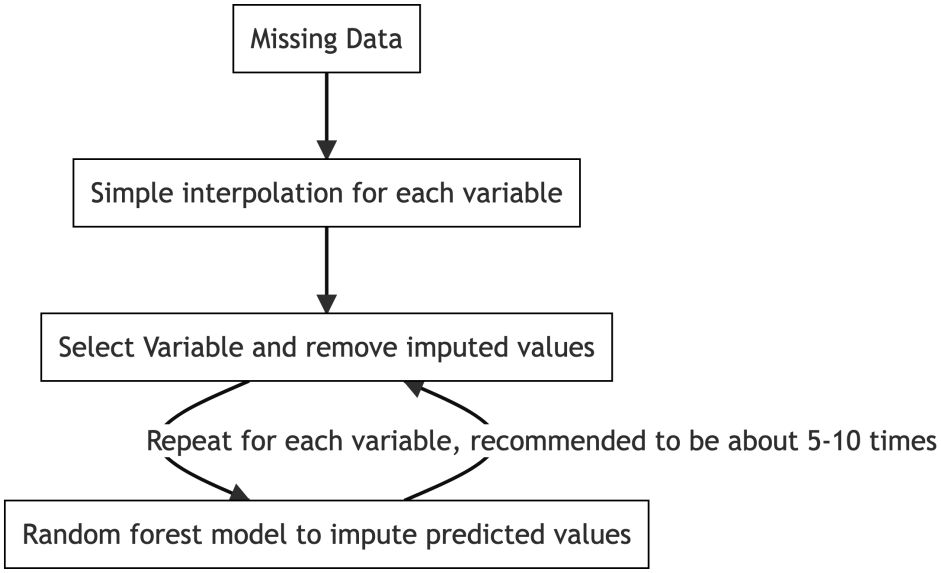


Figure 2: Steps of RF-MICE procedure

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 RF.

The utility of random forest models for imputing political science data

`RF` are a special type of ensemble supervised machine learning models. For those who would like an in-depth introduction of random forests and supervised machine learning, I recommend James et al. (2013). Here, I will provide a simple analogy that outlines the intuition behind what supervised machine learning is to then follow-up with a discussion about how we can apply this intuition to solve missing data problems.

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.

While this is a very simplistic analogy of how the broader class of supervised machine

learning models work, this describes the same task that `RF`, a special type of supervised machine learning, are optimized to perform. These `RF` are optimized through ensemble-based methods to do complete this task recursively and to optimize each iteration by learning from the other iterations. Once this model is trained to determine these patterns, if the model was trained well, it will provide an optimal prediction of what a fixed, but unobserved, observation should be based on the data provided after training.

In a missing data scenario, we can think of the task as training the random forest model on our complete observations in the dataset. In the `MICE` framework, we will fit a random forest model to predict the the values that are originally coded as missing for each variable. We use observations with complete data to train these models so that the weights placed on each predictor variable lead to a model that provides predicted values close to the observed values. Once we have trained this random forest model, we will need to use interpolation on the predictor variables to give us a reasonable data point so that we can use the full information of our data to come up with a predicted value for the column that we are imputing. We do these steps for each variable in our dataset. However, once we have done a full loop through our dataset, we now have predicted values that are better than those interpolated values that we first had to plug-in for the predictor variables for a given imputation model. With `MICE` we do multiple loops through the dataset to reduce the bias that the interpolated starting values may have generated in our imputed value. Intuitively, the more times we impute the whole dataset, the more accurate our imputed values will become. However, with many simulation studies on different types of data and under different circumstances, the common recommendation is that you only need to produce between 5-10 versions of your imputed dataset before the bias generated from those interpolated initial values become

relatively inconsequential (see Azur et al. 2011 for a discussion). Though the exact number of recommended imputed datasets is up for debate and is heavily dependent on a user's computational resources.

This is not the only option at our disposal, however. We can use a Bayesian implementation of a linear regression in the `MICE` framework as well. The processes to impute the data are mostly the same. However, linear regression and `RF` are designed for different tasks. Linear regression models are optimized to help us determine whether a predictor explains an outcome once we account for other variables. `RF`, however, are optimized to learn patterns from existing data, then to use new information to predict some outcome. Furthermore, linear regression has a number of practical limitations. Linear regression is designed to provide predicted values that can range from $-\infty$ to ∞. `RF`, on the other hand are extremely flexible in that they can produce predicted values for continuous, categorical, and binary outcomes. An additional practical advantage that I'll mention here is that `RF` are non-parametric models which provide a distinct degree of flexibility to estimate the underlying DGP of the missingness over Generalized Linear Models. With all these considerations together, we may expect that random forests implemented in the `MICE` framework (`RF-MICE`) are extremely flexible and useful tools to solve missing data problems.

The performance of `RF-MICE` with simulations

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 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. For those interested, distributions of the original sample data and the amputed sample data are included in the Supplementary Materials.

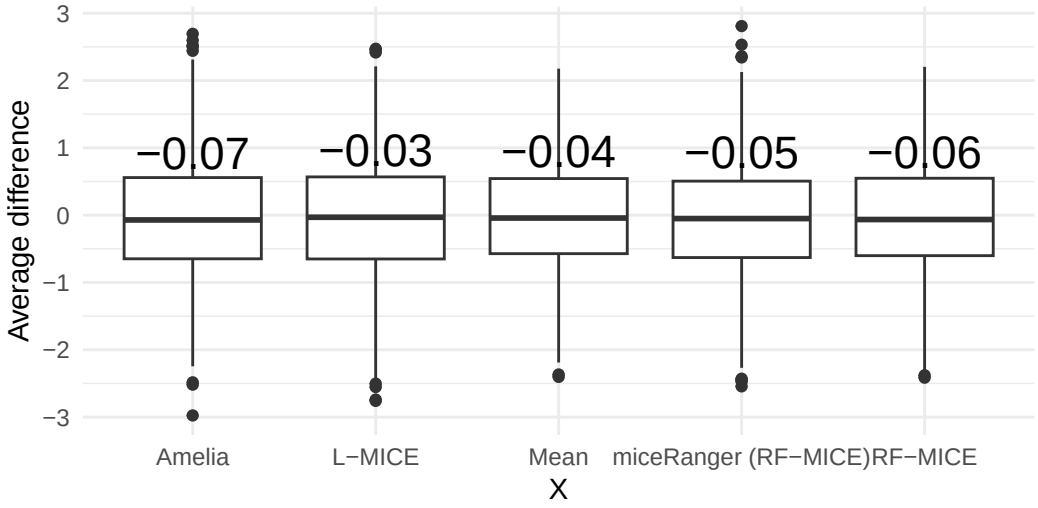
With these amputed datasets, I then apply some of the procedures I have discussed to impute these values. Interpolation is a quite simple procedure where I can fill in missing values by using the mean value of that particular variable for the non-missing observations. I per-

form 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 (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). As discussed before, the final procedure I use is a `RF` 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.

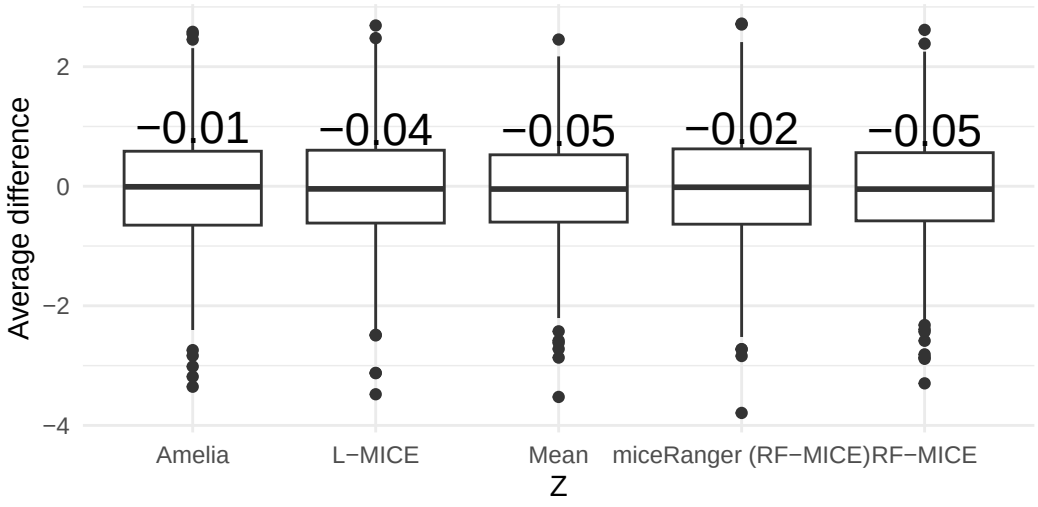
When producing the imputed datasets, I use the `tictoc` (Izrailev 2022) package to record the amount of time each procedure takes to complete the task on the 1000 samples as a measure of computational cost.[3] As a number of factors may affect the absolute computational costs for these procedures (e.g., 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. Figure 3 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.

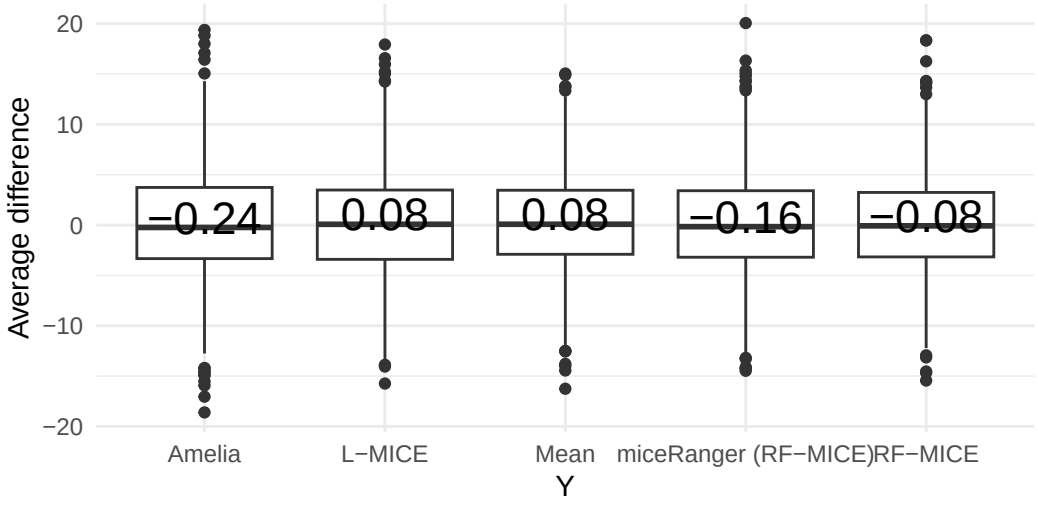
[3]It is important to note that these benchmarks are based on a computer 16 GB of RAM, with a Apple Silicon M2 Pro Processor and a 10-core graphics card.



(a) X variable



(b) Z variable



(c) Y variable

Figure 3: Discrepancies between original and imputed data

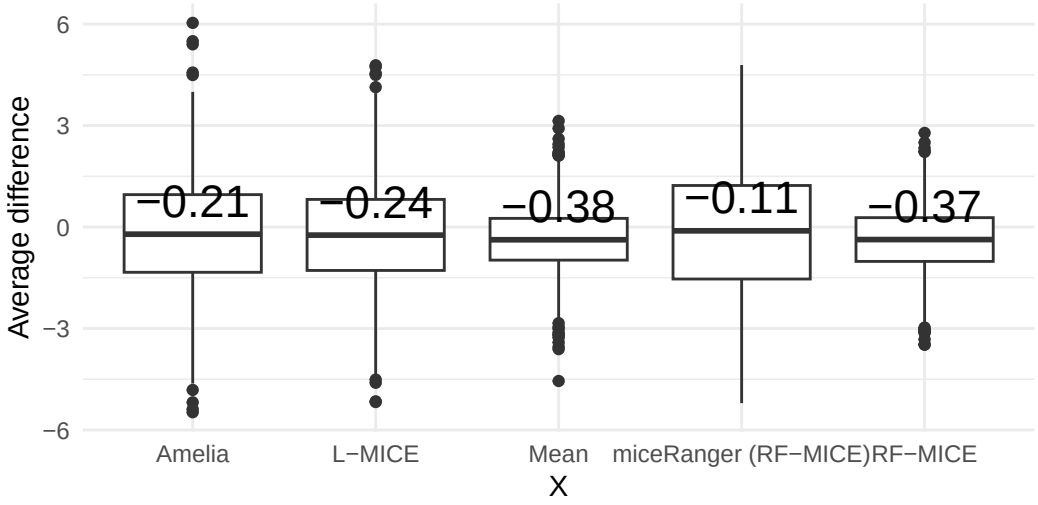
Overall, the procedures do quite well in that the average difference between the actual data and the imputed data are quite small across the datasets. We see that `RF-MICE` when implemented with `miceRanger` (Wilson 2021) consistently does a good job at coming closer to the correct value than the other 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 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.

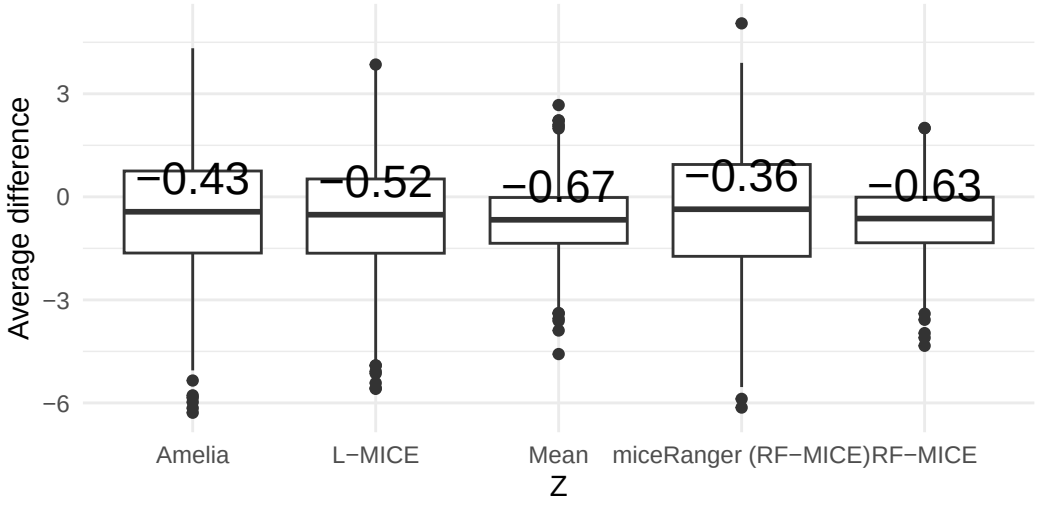
To examine this claim, I take the amputed and imputed datasets (10) and use Rubin's rule (Rubin 1996) to pool across the regression models performed on each amputed and imputed sample. I then calculate the discrepancy by subtracting the parameter value from the point estimate.

Figure 4 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`.



(a) X variable



(b) Z variable

Figure 4: Discrepancies of estimates between original and imputed data

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 RF 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, RF-MICE implemented 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?" *In-*

ternational Journal of Methods in Psychiatric Research 20 (1): 40–49. https://doi.org/10.1002/mpr.329.

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.

Izrailev, Sergei. 2022. "Tictoc: Functions for Timing r Scripts, as Well as Implementations of "Stack" and "List" Structures." https://CRAN.R-project.org/package=tictoc.

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.

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

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

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.

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.

Izrailev, Sergei. 2022. "Tictoc: Functions for Timing r Scripts, as Well as Implementations of "Stack" and "List" Structures." https://CRAN.R-project.org/package=tictoc.

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.

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

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

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.

Supplementary Materials

Summaries of types of missing data and solutions

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

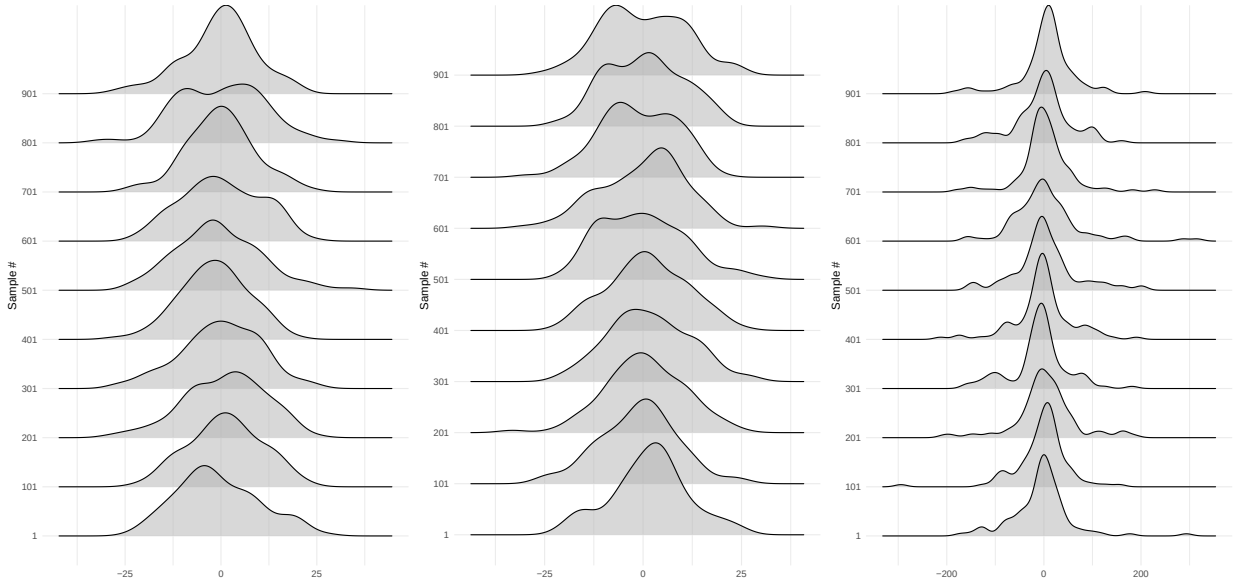
Procedure	Assumptions	What it does	Fixes
LWD	Sources of missingness are random.	Removes rows that have a missing value for *any* variable in the statistical model	MCAR
Simple	Sources of missingness may be explained by some type of autocorrelation (spatial, temporal, non-independence of observations).	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	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.	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, `RF` 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.	MAR

Distributions of simulated data

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 5 and the density distributions for the amputed data are presented in Figure 6.



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

Figure 5: Distributions of complete sample data

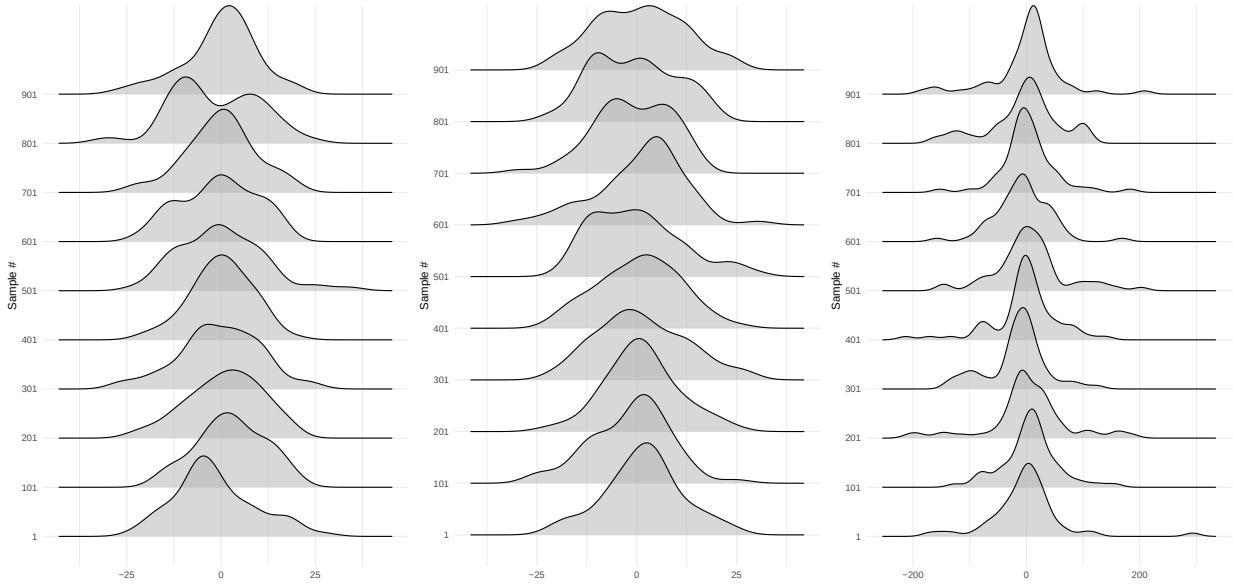
Code

```
# Install libraries

install.packages(

  c(

      "AMELIA"
```



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

Figure 6: Distributions of amputed sample data

```
      , "mice"
      , "miceRanger"
  )
)
# Load libraries
library(Amelia) # for MI
library(mice) # for many MICE and interpolation procedures
library(miceRanger) # for RF-Mice procedure
# Dataset


df
```

```r
# Listwise Deletion
dfImputed <- df[complete.cases(df), ] # exclude rows that have missing values in any col


# Interpolation
dfImputed <- mice(
  df # dataframe
  , m = 10 # number of imputations
  , method = "mean" # mean interpolation
)


# Amelia
dfImputed <- amelia(
  df # dataframe
  , m = 10 # number of imputations
)


# Linear Bayesian MICE
dfImputed <- mice(
  df # dataframe
  , m = 10 # number of imputations
  , method = "linear" # Bayesian linear MICE
)
```

```r
# RF-MICE with mice package
dfImputed <- mice(
  df # dataframe
  , m = 10 # number of imputations
  , method = "rf" # RF-MICE
)


# RF-MICE with miceRanger package
dfImputed <- miceRanger(
  df # dataframe
  , m = 10 # number of imputations
)
```