Implementation and Evaluation of Predictive Mean Matching Methods for Multiple Imputation in Python

The Anh Vu

Master Thesis
for the Department of Statistics
at Ludwig Maximilians University
Munich

written by
The Anh Vu

Munich, 23.06.2025

Dissertation supervisor: Dr. Anna-Carolina Haensch

Github repository:
https://github.com/Theanh2/Implementation_of_PMM_by_MICE/

Statement of authorship

I hereby declare that I am the sole author of this master thesis and that I have not used any sources other than those listed in the bibliography and identified as references. I have solely used LLMs to correct my phrasing and have not used it for anything else. I further declare that I have not submitted this thesis at any other institution in order to obtain a degree.

Munich, 23.06.2025 .....

Acknowledgements

I sincerely like to express my gratitude to everyone who supported me throughout the process of writing this thesis. It was supervised by Dr. Anna Carolina Haensch , to whom I would like to express my thanks for the excellent guidance and many helpful ideas during our meetings. Writing this thesis marks the conclusion of a rewarding Master's program in Statistics and Data Science at the LMU in Munich. My gratitude also extends to all the lecturers as well as my fellow students.

This Master's thesis could not have been completed without the invaluable support of my friends and parents, whose constant encouragement and presence provided the emotional strength and practical assistance I needed throughout this journey.

The Anh Vu

Abstract

This thesis develops a modular Python framework for multiple imputation inspired by the R package mice, with a focus on PMM and the more recent midastouch algorithm. The implementation offers full flexibility in defining distance metrics, donor selection rules, and imputation parameters.

To evaluate the performance of both imputation methods, a comprehensive simulation study was conducted across three types of variables: continuous, semicontinuous, and discrete. Each simulation scenario varied in terms of the missingness mechanism, including Missing Completely at Random, left tailed Missing at Random, and right tailed Missing at Random. The scenarios also differed in the proportion of missing data and the configuration of the imputation settings. The performance of the imputations was evaluated using bias, coverage, confidence interval width, and mean squared error.

Results show that PMM performs reliably under MCAR and mild MAR conditions, particularly when data are symmetrically distributed and sample sizes are moderate to large. However, its performance degrades under skewed distributions and structured missingness, leading to biased estimates and reduced coverage. midastouch, in contrast, consistently matches or outperforms PMM in terms of coverage and standard error estimation, especially in scenarios involving skewness or small sample sizes. Importantly, midastouch avoids the need to manually tune parameters such as donor size, and when paired with HowManyImputations also number of multiple imputations and offers an out of the box solution with strong empirical performance. The findings support the use of midastouch as a robust default imputation method.

Contents

List of Figures

List of Tables

List of Abbreviations and Symbols

I use the given Notation by Van Buuren's extensive book about Multiple Imputation: Flexible Imputation of Missing Data (van Buuren, 2018, P. 30), which he modified from Rubin's and Schafer's Notation (alphabetically):

α	Coefficient of analysis model
β	Coefficient of imputation model
$\dot{\beta}$	random draw from the posterior distribution of β
γ	fraction of missing information
$\hat{\beta}$	estimated coefficient of β
$\hat{SE}$	estimated standard deviation
κ	closeness parameter for MIDAS
ϕ	Parameters for imputation model
θ	Parameters for analysis model
$\tilde{\beta}$	Coefficient from posterior distribution of β
CV	Coefficient of Variation
HMI	HowManyImputations by von Hippel (2020)
k	Donor Size
m	Number of imputation iterations
MAR	Missing at random
$MCAR$	Missing completely at random
$MIDAS$	Multiple Imputation with distance aided selection of donors
$MNAR$	Missing not at random
n	Number of units in sample
PMM	Predictive Mean Matching
R	n x p boolean matrix, Response indicator
r_{ij}	Elements of R
T	Total variance of θ
U	variance-covariance matrix of θ
X	set of predictors in various types of models
Y	n x p matrix with p variables and n units in the sample
y_{ij}	Elements of Y

Introduction

Missing data is an omnipresent challenge in empirical research and applied statistics. Whether due to nonresponse in surveys or procedural issues during data collection, incomplete datasets can lead to biased estimates and reduced statistical power. To mitigate these risks, multiple imputation has become a widely endorsed approach. Among its various implementations, Predictive Mean Matching has emerged as a popular default method for imputing numerical variables across popular statistical software like R, Stata, SPSS, and SAS.

Despite its widespread use, the theoretical underpinnings and empirical behavior of PMM remain insufficiently understood. Existing studies, such as those by Yu et al. (2007), Vink et al. (2014), and Kleinke (2017), provide valuable insights but leave several questions open. As Kleinke (2018) notes, "currently, there are no evaluation studies that systematically tested if or under what conditions PMM can be used," highlighting the need for a deeper and more systematic investigation.

In this thesis, I aim to fill this gap by implementing a fully functional and customizable Python based multiple imputation framework inspired by the R package *mice*. My implementation includes both PMM and the more recently proposed midastouch algorithm. The framework emphasizes flexibility and transparency, enabling researchers to modify components such as distance metrics, donor selection strategies, and the number of donors features not currently available in any single Python package.

The empirical part of the thesis consists of a comprehensive simulation study that systematically evaluates the performance of PMM and midastouch across three data scenarios: continuous, semicontinuous, and discrete variables. Each simulation introduces varying degrees of missingness (MCAR, left-tailed MAR, right-tailed MAR) and considers multiple configurations of imputation parameters. The simulations follow a factorial design and are evaluated on multiple performance metrics, including bias, coverage rates, confidence interval widths,

and mean squared error.

The results support and extend previous findings by showing under which conditions PMM remains reliable, and where midastouch provides meaningful improvements. Moreover, the thesis introduces a working implementation of the "How Many Imputations?" procedure proposed by Gaffert et al. (2018), providing an easy way to determine the number of imputations required for replicability.

Overall, this thesis contributes both a Python implementation and a detailed empirical evaluation of PMM and midastouch, helping to inform best practices in handling missing data and highlighting the strengths and limitations of current default methods.

Response Model

3.1 Response Mechanism

Missing data is a frequent issue in real world applications and can arise from a variety of causes. Understanding the underlying mechanisms that lead to missing data is crucial for selecting appropriate imputation methods.

Little and Rubin (2002, P. 12) classifies these mechanisms into three categories. According to their theory, any data point has a likelihood of missing. If the probability is the same for all data points, the data is called *Missing completely at random (MCAR)*. Under MCAR, the missingness is entirely independent of the data itself. Although MCAR leads to unbiased estimates, it is often considered an unrealistic assumption in practice. A more plausible scenario is *Missing at Random (MAR)*, where the likelihood of missingness depends only on the observed data. MAR is a common assumption and serves as the starting point for most imputation techniques. *Missing not at Random* (MNAR) is when neither MCAR nor MAR holds. This means the missingness is also dependent on unobserved values.

$$\begin{aligned} Pr(R = 0|Y_{obs}, Y_{mis}, \psi) &= Pr(R = 0|\psi)(MCAR) \\ Pr(R = 0|Y_{obs}, Y_{mis}, \psi) &= Pr(R = 0|Y_{obs}, \psi)(MAR) \\ Pr(R = 0|Y_{obs}, Y_{mis}, \psi)&(MNAR) \end{aligned} \tag{3.1.1}$$

Rubin's classification is important for the concept of *ignorability*. The joint density $f(Y_{obs}, R|\theta, \psi)$ depends on ψ, which is unknown and not of scientific interest, and θ, our estimands of interest. The missing data mechanism (ψ) can be ignored for likelihood based inference if certain conditions are met (Little & Rubin, 2002, P. 119):

- *MAR: the missing data are missing at random; and*
- *Distinctness: the parameters θ and ψ are distinct, in the sense that the joint parameter space of (ψ, θ) is the product of the parameter space of θ and the parameter space of ψ.*

MAR is regarded as more important than distinctness because inference remains valid from a frequentist perspective, though it is not fully efficient (Little & Rubin, 2002, P. 120). If the nonresponse mechanism is ignorable, we can model the posterior distribution and draw imputations, as the distribution remains consistent between the response and nonresponse groups. (Little & Rubin, 2002, P. 120-122).

$$\begin{aligned} &P(Y_{mis}|Y_{obs}, R) = P(Y_{mis}|Y_{obs}) \\ &\Rightarrow P(Y|Y_{obs}, R = 1) = P(Y|Y_{obs}, R = 0) \end{aligned} \tag{3.1.2}$$

For this work we are simulating the missing data mechanisms, so non ignorability is not an issue and will not be discussed.

3.2 Why do we not use Complete Case?

Complete Case analysis is the default method to handle missing data in most statistical software. At first glance, this does not seem to be a big deal especially for students, but it does bring a loss of information. This results in loss of precision and bias when the missing data mechanisms is not MCAR (Little & Rubin, 2002, P. 41). For example, when estimating means, the bias introduced depends on both the proportion of incomplete cases and the differences between complete and incomplete observations. Regression coefficients may also be biased if the likelihood of an observation being complete depends on the outcome variable Y, even after controlling for covariates (Little & Rubin, 2002, P. 43). Therefore, complete case analysis should be used deliberately and cautiously, typically only when data are truly MCAR or when the proportion of missing data is negligible.

Other single imputation methods, such as mean, regression and deletion methods have been evaluated by van Buuren (2018, P. 8-15). While these methods can be easy to implement, their results are generally insufficient for reliable statistical inference. They tend to underestimate the variance, fail to preserve relationships

among variables, and often lead to biased parameter estimates when the data is not MCAR (van Buuren, 2018, Tbl. 1.1). These limitations highlight the need for more robust techniques that appropriately reflect uncertainty due to missingness. The current state of the art technique, Multiple Imputation, addresses these issues and is widely regarded as the most robust general approach for handling incomplete data.

3.3 What is the problem with Single Imputation?

Our goal for imputation is to get a unbiased and confidence valid estimate of θ. Single imputation does not account for the uncertainty inherent in the imputation process, which can lead to underestimation of variability and biased statistical inference. This means Given Y and U we want to have:

$$\begin{aligned} E(\hat{\theta}|Y) &= \theta \\ E(U|Y) &\geq Var(\hat{\theta}|Y) \end{aligned} \tag{3.3.1}$$

This is not achievable with single imputation, as the true value of θ is unknown and single imputation fails to account for the uncertainty associated with missing data. The possible values of θ from our observed values Y_{obs} are derived from the posterior distribution $P(\theta|Y_{obs})$ (Rubin, 1996, P. 475). Rubin (1987, P. 21) proposed connecting the posterior distribution based on the observed data with the posterior distribution we would have had if the missing data Y_{mis} were fully observed. Specifically, the observed data posterior distribution can be written as the integral over all possible values of the missing data:

$$\begin{aligned} P(\theta|Y_{obs}) &= \int p(\theta, Y_{mis}|Y_{obs})dY_{mis} \\ &= \int p(\theta|Y_{mis}, Y_{obs})p(Y_{mis}|Y_{obs})dY_{mis} \end{aligned} \tag{3.3.2}$$

When the posterior mean and variance are tolerable summaries of the posterior

distribution $p(\theta|Y_{mis}, Y_{obs})$ we get:

$$\begin{aligned} P(\theta|Y_{obs}) = E(\theta|Y_{obs}) &= E(E[\theta|Y_{mis}, Y_{obs})|Y_{obs}] \\ \Rightarrow V(\theta|Y_{obs}) &= E[V(\theta|Y_{obs}, Y_{mis})|Y_{obs}] + V(E(\theta|Y_{obs}, Y_{mis})|Y_{obs}] \\ &= withinVariance(U_\infty) + betweenVariance(B_\infty) \end{aligned} \tag{3.3.3}$$

Equation 3.3.2 shows that the posterior distribution can be simulated by first imputing missing values and then drawing θ from the imputed complete data (Little & Rubin, 2002, P. 210). In the context of Multiple Imputation, the actual posterior distribution of θ is approximated by averaging over repeated draws of θ.

The posterior variance of $P(\theta|Y_{obs})$ is composed of two variance components, as shown in Equation 3.3.3. The term $E(V(\theta|Y_{obs}, Y_{mis})|Y_{obs}]$ represents the average of the complete data posterior variances of θ across repeated imputations, known as the within imputation variance. The term $V(E(\theta|Y_{obs}, Y_{mis})|Y_{obs}]$ captures the variability of the complete data posterior means of θ, referred to as the between imputation variance. As our θ is estimated by a finite sample the total variance itself is shown to be:

$$\begin{aligned} T &= \bar{U} + B + B/m, \\ \bar{U} &= 1/m \sum_{l=1}^{m} \bar{U}_l, \\ &(Rubin's Rules) \end{aligned} \tag{3.3.4}$$

with l being the l_{th} imputation (Little & Rubin, 2002, P. 211). This highlights why Single Imputation is insufficient. It effectively replaces Y_{mis} with a single guess, as a result ignoring the variability between imputations. In contrast, Multiple Imputation accounts for this uncertainty by drawing m times and is therefore the preferred method for handling incomplete data.

Methods

4.1 Multiple Imputation with chained equations

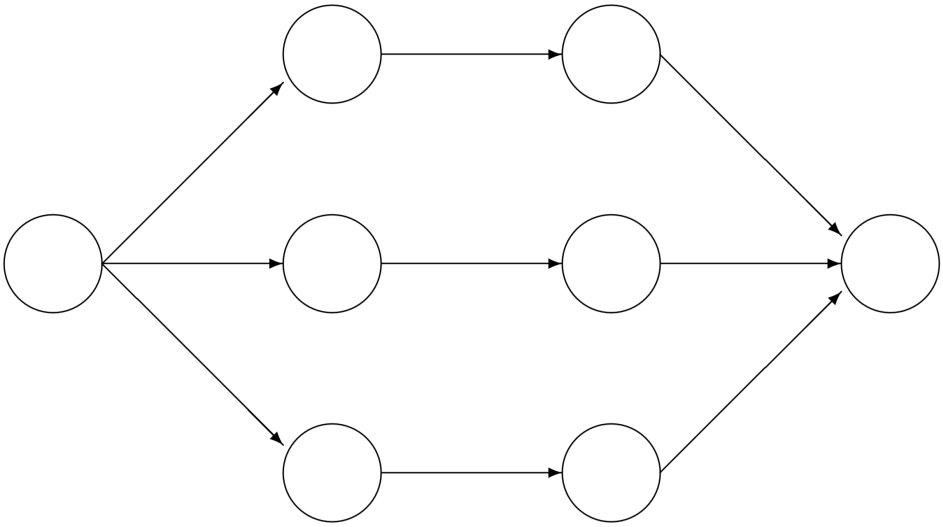


Figure 4.1: Multiple Imputation scheme (van Buuren, 2018, P. 17)

Multiple imputation is the natural extension of single imputation that produces valid statistical estimates. In the first step, an imputation model is used to generate m complete datasets by replacing missing values with plausible alternatives. Each iteration usually differ in the imputed values, because of the uncertainty of the imputation process. In the second step, θ is estimated from each of the m imputed datasets using an analysis model. Typically, the same analytical method is applied as would have been used if the data were fully observed (van Buuren, 2018, P. 16). In the final step, the parameter estimates from the imputed datasets are combined to produce an overall estimate along with its associated variance. This is accomplished using Rubin's Rules (Equation 3.3.4), which accounts for

both the within imputation variance and the between imputation variance.

For multivariate missing data, there are two strategies for specifying the imputation model. One is called Joint modeling, the other is Full Conditional Specification (FCS) or more commonly Multiple Imputation with Chained Equation (MICE). Joint modeling involves specifying a joint multivariate distribution for all variables with missing data. This approach treats the data as arising from a single probabilistic model, allowing imputation by directly drawing from the conditional distributions implied by the joint model. However, in practice, the exact form of the multivariate distribution is often unknown or difficult to specify. As a result, the more flexible MICE approach is used. MICE bypasses the need to define a joint distribution by directly specifying the conditional distribution for each variable, given the others (van Buuren, 2018, P. 108).

The core idea behind the MICE algorithm is to iteratively impute missing values for each variable by modeling its conditional distribution based on the current values of the other variables. When these other variables also contain missing data, their missing entries are temporarily filled using placeholders, such as the mean or by sampling from the observed values. This iterative process is sometimes referred to as the "Poor Man's Gibbs Sampler," as each conditional regression is estimated only using the observed values of the dependent variable.

The algorithm sequentially cycles through all variables with missing data. One complete pass through these variables is called a cycle, and multiple cycles are performed to ensure convergence. The order in which variables are updated is not important, as long as each variable is visited sufficiently often. In the case of monotone missing data patterns, convergence is typically achieved after just one cycle (van Buuren, 2018, P. 141).

A missing data pattern is considered monotone if the variable Y_j can be ordered such that if Y_j is missing then all variables Y_k with $k > j$ are also missing (van Buuren, 2018, P. 95). Usually the data is non monotone and multiple cycles have to be done.

Nevertheless, diagnostic plots, such as tracking the estimated values of parameters across iterations can provide insight. When convergence has been achieved, the values from different iterations should appear mixed and show no discernible trends. A practical indicator of convergence is when the between iteration variance does not exceed the within iteration variance (van Buuren, 2018, P.

142–143). Convergence issues may arise particularly when variables are highly interdependent and influence each other recursively.

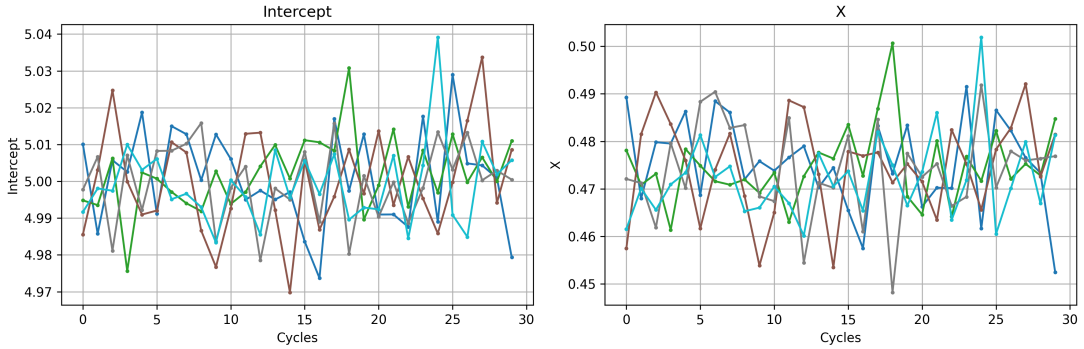


Figure 4.2: Example of convergence testing: Estimate vs Iteration plot with Simulation 1, 30 cycles, 5 iterations.

4.2 How many Imputations?

To obtain valid scientific estimates of the parameter of interest θ we need to iterate over our incomplete data m times. Each of these datasets is created by imputing the missing values based on the observed data. Each completed dataset is then analyzed separately, and the results are combined using Rubin's Rules (Equation 3.3.4).

Even with a relatively small number of imputations, the resulting parameter estimates for θ remain unbiased. Ideally, running $m = \infty$ imputations would eliminate all simulation error in the estimate of θ, but this is obviously infeasible in practice. That said, when the proportion of missing data is large or when accurate estimates of variability are important, using a higher number of imputations is recommended. Thanks to modern computational tools, increasing m incurs minimal cost, making larger numbers of imputations more common in applied settings to reduce Monte Carlo error and enhance precision. The total variance are related by (Little & Rubin, 2002; Rubin, 1987, P. 211, P. 114):

$$T = (1 + \frac{\gamma}{m})T_{\infty}, \tag{4.2.1}$$

where T_{∞} is the variance that would be obtained with an infinite number of imputations and γ represent the true population fraction of missing data. In the

univariate case γ corresponds directly to the fraction of missing data. However, in the multivariate case, γ depends on the extent to which missing data affects the estimation of the specific parameter of interest. It is generally less than the overall fraction of missing values and must be estimated from the data.

4.2.1 shows the impact of m on the total variance. For example with $\gamma = 0.2$ and $m = 40$ the variance inflation is only 5% bigger than $m = \infty$. Ultimately, the choice of m is a compromise between computational feasibility and the desired level of precision in parameter estimates (Graham et al., 2007).

White et al. (2011) suggests using the linear rule $m = 100\gamma$ to determine m to provide an adequate level of reproducibility. However, while easy to apply, this linear rule has notable limitations. As shown in von Hippel (2020, fig. 1), it tends to underestimate the required number of imputations when γ is high and overestimate it when γ is low.

von Hippel (2020) further explores the idea of reproducibility and also solves the problem of the unknown γ prior imputation. He proposes a quadratic Rule that approximates replicable point estimates as well as replicable standard errors:

$$m = 1 + \frac{1}{2} * \left(\frac{\gamma}{CV(\hat{SE}|Y_{obs})} \right)^2 \tag{4.2.2}$$

with the coefficient of Variation (CV):

$$CV(\hat{SE}|Y_{obs}) = \left(\frac{SD(\hat{SE}|Y_{obs})}{E(\hat{SE}|Y_{obs})} \right) \tag{4.2.3}$$

and additionally a two stage procedure to estimate the fraction of information that is missing about the parameter θ:

1. *In the first stage, you conduct a small m pilot analysis to obtain a pilot SE estimate* $\hat{SE}$ *and a conservative estimate of* γ *(*$logit(\hat{\gamma}) = +z\sqrt{2/m}$*). From the pilot analysis, you use the estimate* $\hat{SE}$ *and your goal* $SD(\hat{SE}|Y_{obs})$ *to estimate your target CV with 4.2.3.*
2. *In the second stage, you plug your estimate of* γ *and your target* $\hat{CV}$ *into the formula 4.2.2 to get the required number of imputations m. If your pilot m was at least as large as the required m, then your pilot analysis was good enough, and you can stop. Otherwise, you conduct a second and final*

analysis using the m that was recommended by the pilot analysis.

For the pilot stage we are not using the point estimate to get the estimated fraction of missing information $\hat{\gamma}$, but the upper bound of the confidence interval to guarantee m to achieve the desired degree of replicability with a $(1 - \frac{\alpha}{2})\%$ chance. For the pilot stage it does not really matter how we choose m as we ensure enough iterations in the second stage. The cost of low pilot m would be higher variability and therefore potentially higher computational costs. Simulation studies show that even with a low number of imputations in the pilot stage, the estimates of m tend to be very close to the target, as measured by the variability of the standard error $SD(\hat{SE}|Y_{obs})$ (von Hippel, 2020, tbl. 2). Ideally, the pilot stage value of m should be selected based on prior knowledge or expectations about γ.

4.3 Predictive Mean Matching

Little (1988, P. 291) first introduced Predictive Mean Matching (PMM) as a practical imputation technique aimed at incorporating information from observed data in a simple manner. The idea behind PMM is to preserve the distributional characteristics of the observed data by using actual observed values as imputations, rather than relying on model based predictions. The method begins by specifying an imputation model, typically a linear regression model, to predict the outcome variable Y based on covariates X. The model is fitted using the observed cases, and predicted values $\hat{y}$ are calculated for both observed and missing cases. For each predicted value $\hat{y}$ we match it to an observed value, known as a donor.

Algorithm 1 Predictive Mean Matching (Little, 1988, P. 291-292)

1. Calculate the predictive mean for the n_{obs} observed elements of $\hat{y}_i = X_i\hat{\beta}$
2. Calculate the predictive mean for the n_{obs} missing elements of $\hat{y}_i = X_i\hat{\beta}$
3. Match each element of $\hat{y}_j$ to its corresponding closest element of $\hat{y}_i$
4. Impute the observed y_i of the closest matches.

The core advantage of PMM is its robustness to model misspecification, as it

always imputes values from the observed data. This constraint ensures that imputed values remain within the plausible range of the original data, making PMM particularly well suited not only for continuous variables, but also for discrete and semicontinuous variables. PMM preserves data types and distributional features, leading to more realistic and interpretable imputations (Vink et al., 2014, P. 80). Furthermore, PMM exhibits robustness in scenarios where parametric methods may struggle. It performs well under non linear relationships, heteroscedastic residuals, and deviations from normality (Morris et al., 2014, fig. 1, 6–8) (Gaffert et al., 2018, P. 1026).

Algorithm 1 shows the most basic algorithm of PMM. For each step we have several degrees of freedom in the implementation that can affect the performance of the imputation. These include the method of matching, donor selection, and the number of donors used.

1. **Matching distance**

   - Type 0: $\hat{y} = X_{obs}\hat{\beta}$ matched with $\hat{y} = X_{mis}\hat{\beta}$
   - Type 1: $\hat{y} = X_{obs}\hat{\beta}$ matched with $\hat{y} = X_{mis}\dot{\beta}$
   - Type 2: $\hat{y} = X_{obs}\dot{\beta}$ matched with $\hat{y} = X_{mis}\dot{\beta}$

   Little (1988) initially suggested using Type 0 matching ($\hat{\beta}_{mis} = \hat{\beta}_{obs}$), where both observed and missing predicted values are based on the same point estimate $\hat{\beta}$. However, this approach ignores the sampling variability of $\hat{\beta}$ leading to improper imputations. To address this, Type 1 introduces stochasticity by drawing $\dot{\beta}$ from the posterior distribution of β, reflecting uncertainty in model parameters (van Buuren, 2018). Type 2 further generalizes this by applying posterior draws $\dot{\beta}$ to both observed and missing values.

   Morris et al. (2014, fig. 6-8) compared Type 1 and Type 2 matching and concluded that in general Type 1 matching is preferred, because it is better in terms of coverage and efficiency. However Gaffert et al. (2018, P. 15) advocates for Type 2 matching, arguing that it produces more proper multiple imputations. Most statistical software packages offer the option to choose among these types, as no single approach is universally superior.

2. **Donor selection**

Most of the time the donor is selected at random from among the k nearest neighbors. But, alternative approaches exist. For example, the related midastouch (Gaffert et al., 2018) assigns probability weights to donors based on their distance from the target case, giving closer matches a higher likelihood of being selected. Such approaches can enhance both robustness and efficiency.

3. **Number of donors**

   Using only a single donor ($k = 1$), as originally proposed (in algorithm 1), often results in the repeated selection of the same donor for Multiple Imputations. This reduces variability in the imputed values and can lead to overconfidence in parameter estimates due to an underestimation of uncertainty. On the other hand, setting k to a large value increases the likelihood of including poor matches whose observed values are not sufficiently similar to the predicted value of the missing case. This introduces bias into the estimates, as less relevant matches may distort the distribution of the imputed variable. A very small k may not fully reflect the uncertainty of the imputation process, again leading to underestimated standard errors. Simulation studies such as in Kleinke (2017, tbl. 2) and van Buuren (2018, tbl. 3.3) recommend using a relatively low k, between 3 and 10, as it improves point estimates in their simulations.

4.4 Multiple Imputation by Distance Aided Donor Selection

Using PMM for MI can lead to attenuation bias in the between imputation variance of parameter estimates θ (Gaffert et al., 2018, P. 13). Gaffert illustrates this bias by comparing PMM to both parametric imputation models and the Approximate Bayesian Bootstrap (ABB) approach. The key element is that the distribution from which the imputed values are drawn vary. In the parametric case, imputations are sampled from the underlying normal distribution defined by estimated parameters, while in the nonparametric case, they are sampled from the empirical distribution formed by the observed data.

PMM selects imputed values through random draws from a set of observed donors

whose predicted means are closest to those of the recipients. This results in the imputed values sharing the exact same predicted mean as the observed donors, effectively mimicking a simple hot deck imputation. As a consequence, PMM partly omits the posterior draw step, which leads to an underestimation of the between imputation variance. The extend of this underestimation is inversely proportional to the number of available donors (Gaffert et al., 2018, P. 14).

To address this issue, Gaffert proposes an improved method, *midastouch*, which builds on the theoretical foundation of MIDAS introduced by Siddique and Harel (2009). This approach performs ABB after conditioning on the regression parameters β, thereby making the imputation process more "proper" in Rubin's sense.

The flexible closeness parameter κ (see 4.4.1) determines the influence of the imputation model. κ reflects the model's goodness of fit, such that a better fitting model increases the likelihood of selecting donors close to the predicted value, while poorer fits allow for more dispersed donor selection.

4.4.2 generalizes the ABB imputation. However it is well known that in this case the total variance is underestimated if n_{obs} is finite (Kim, 2002; Parzen et al., 2005; Demirtas et al., 2007). Applying the Parzen correction in 4.4.3 mitigates this issue. Demirtas et al. (2007) investigated the correction and found that, while the correction increases precision, especially for small samples, it may reduce efficiency and increase root mean square error. In large samples, this tradeoff becomes negligible, as both corrected and uncorrected versions converge in performance. Thus, the decision to apply the correction should depend on the primary scientific interests.

Algorithm 2 midastouch (Gaffert et al., 2018, Alg. 4)

1. Obtain bootstrap frequencies ω for the donors to introduce the between variance

2. Draw β from a weighted least-squares regression with the weights ω_i and calculate the according coefficient of determination R_2

3. Calculate the elements of the $n_{mis}xn_{obs}$ distance matrix using the leave one out principle as follows: $\psi_{i,j} = |(x_i - x_j)\beta_{-i}|$. Here x_i denothes the row vector of X_i for the ith donor, x_j denotes the row vector of X_j for the jth recipient, and β_{-i} denotes the weighted least-squares parameter vector from the donor sample without the ith row

4. Calculate the closeness parameter as follows:

$$\kappa(R^2) = \left(\frac{50R^2}{1 + \epsilon - R^2}\right)^{3/8} \tag{4.4.1}$$

   where ϵ is a very small positive positive scalar number used to ensure real results for $R^2 = 1$

5. Insert $\omega, \varphi_{i,j}$ and κ and draw the donors:

$$\omega_{i,j} = f(\omega, y_i, y_j, \kappa) = \omega_i \varphi_{i,j}^{-} \kappa / \sum_{i=1}^{n_{obs}} (\omega_i \varphi_{i,j}^{-\kappa}) \tag{4.4.2}$$

6. Repeat the above steps m times, apply Rubin's rules and multiply the total variances of the means by the Parzen correction. Substitute n_{obs} with n_{eff} and $n = n_{eff} + n_{mis}$

$$\phi(n_{obs}, n_{mis}, m) = \frac{\frac{n^2}{n_{obs}} + \frac{n_{mis}}{m}\left(\frac{n-1}{n_{obs}} - \frac{n}{n_{obs}^2}\right)}{\frac{n^2}{n_{obs}} + \frac{n_{mis}}{m}\left(\frac{n-1}{n_{obs}} - \frac{n}{n_{obs}^2}\right) - \frac{n * n_{mis}}{n_{obs}}\left(\frac{3}{n} + \frac{1}{n_{obns}}\right)} \geq 1 \tag{4.4.3}$$

Implementation

5.1 MICE

There is currently no comprehensive Python package that fully implements PMM or the midastouch method. While a few existing packages offer basic PMM functionality, they typically support only limited configurations and lack the full set of researcher degrees of freedom available in established packages such as *mice* (van Buuren & Groothuis-Oudshoorn, 2011). To address this gap, I have developed a Python implementation that closely mirrors the behavior of the *mice* package in R, but within a more pythonic workflow. The core philosophy of this package is to combine ease of use with full flexibility giving users complete control over the imputation process, including the selection of methods and model parameters.

The foundation of the package is the *mice*() function, which initializes the basic framework of the MICE algorithm. It allows users to specify the number of imputations via m, the number of cycles with *maxit* and the method of initial imputation with *initial*. For guidance on the selection of m and *maxit* see 4.1 and 4.2.

Once initialized, the $.fit()$ method is used to execute the imputation procedure. This method runs both the imputation model and the analysis model, returning a summary of all relevant coefficients and diagnostic metrics. A predictor matrix must be specified to indicate which variables should be used to impute each target variable. This matrix can be manually defined or automatically generated using the *quickpred*() function. This function creates a predictor matrix using the variable selection procedure described in van Buuren et al. (1999, P. 687). It calculates two types of correlations for each variable pair. One based on observed values and the other based on the response indicator and includes a variable as a predictor if either correlation exceeds a specified threshold. Due to the reliance on correlation computations, this procedure is only applicable to

numerical variables.

By default PMM is used as the imputation method for both numerical and categorical variables. The imputation method can be specified using the *.set_methods*() function. Currently, only PMM and midastouch are supported, although additional methods are planned for future implementation. Adding custom imputation methods is straightforward, any function that returns an imputed vector can be integrated into the workflow.

To my knowledge, this package also provides the only working Python implementation of the HowManyImputations approach proposed by Gaffert et al. (2018). Although conceptually straightforward to program, this method has not yet been made available in Python. In my implementation, it overrides the m parameter defined in the *mice*() call and runs the two stage imputation procedure to estimate the appropriate number of imputations dynamically.

5.2 PMM

The PMM implementation in this package is designed to expose all degrees of freedom to the user. Namely, the choice of matching distance, donor selection strategy, and the number of donors (see 4.3). The implementation behavior is consistent with the R counterpart.

The algorithm begins by drawing values of β and σ from their posterior distributions using Bayesian linear regression. Matching distances are then computed according to the selected type (0, 1, or 2), and donor selection is carried out using either the nearest k neighbors or fixed radius neighbors with a k-dimensional tree. This speeds the computation up like the binary search implementation from mice (version $\geq$ 2.22). For categorical variables, the implementation either factorizes categories or applies canonical correlation analysis to quantify the relationships between the target and predictors. The imputed canonical scores are subsequently retransformed to preserve the original categorical structure.

5.3 midastouch

The midastouch implementation follows the original R version (see Algorithm 2), including the two minor deviations from the original algorithm. The first deviation involves the calculation of the closeness parameter κ. In the original implementation, a small constant ϵ is added to the denominator to prevent division by zero. However, this can introduce numerical inaccuracies, potentially causing the resulting imputation probabilities to deviate from a proper probability distribution (i.e., not summing to one). To address this issue, I instead define κ as follows:

$$\kappa = min(100, \frac{50R^2}{1 - R^2}^{3/8}) \tag{5.3.1}$$

In cases where κ can not be calculated, $\kappa = 3$ will be assumed as suggested from Gaffert et al. (2018) and Siddique and Harel (2009).

The second modification involves the Parzen correction (Parzen et al., 2005), which has been implemented as an optional feature. Although this correction can improve the performance of the algorithm in some cases, it is not universally beneficial (Demirtas et al., 2007) (see 4.4), and its effectiveness may depend on the structure of the data and the amount of missingness.

```
pm = quickpred(df, mincor= 0.1, minpuc = 0.1)
mice = mice(data = df, predictorMatrix = pm, initial = "sample",
    maxit = 5)
mice.set_methods(d = {"variable1": "pmm", "variable2": "midas"})
x = mice.fit(fml = y ~ variable1 + variable2, HMI = True, pilot =
    5)
```

simple code example (python)

Related Works

PMM is currently the default imputation method for numerical data in most widely used statistical software packages (R, Stata, SPSS, SAS). Its widespread adoption is largely due to its intuitive appeal and robust performance across a variety of scenarios. Despite its popularity, PMM has not been extensively studied in terms of its limitations, and a systematic understanding of where the method may fail remains somewhat underdeveloped (van Buuren, 2018; Kleinke, 2017). As discussed in 4.3, PMM offers advantages over fully parametric MI approaches, they preserve the underlying distribution as the imputed values are drawn from the observed values, thus reducing the potential for bias due to model misspecification.

The work by Yu et al. (2007) offers a comparative evaluation of common imputation methods for semicontinuous data across various software packages. Their findings conclude that PMM consistently outperforms methods based on normality assumptions, not only in terms of accuracy but also in maintaining the shape and distributional features of semicontinuous variables. Building on this, Vink et al. (2014) further examined PMM in the context of semicontinuous data and compared it to specialized imputation techniques designed for such data structures. Their study involved both simulated and real data under a MAR mechanism, evaluating performance at a point mass at zero with probability of 25% and 50% chance as well as 25% and 50% missingness. The assessment criteria included bias in the mean, median, and correlation, as well as coverage rates, size of the point mass, preservation of distributional shapes and plausibility of imputed values. Across all these metrics, PMM consistently demonstrated strong performance.

Gaffert et al. (2018) offered simulation based evidence comparing PMM with his proposed method, midastouch. His findings suggest that midastouch generally outperforms PMM, especially in situations with limited donor availability and small sample sizes.

Kleinke (2017) provided a thorough assessment of PMM in the context of skewed data, examining its performance under varying conditions of sample size, donor pool size, and missing data percentages. In his study, a complete continuous predictor X was used to impute an incomplete, positively skewed variable Y, simulated as a non negative integer drawn from a Poisson distribution. Each imputation scenario was repeated 1,000 times, and the evaluation metrics included coverage rates, bias, and confidence interval widths. The study concluded that PMM performs better when skewness is low, sample size is large, and somewhat counterintuitively the donor pool is limited to the nearest neighbor when the sample size is small. However, the authors note that this may reflect variance underestimation inherent in nearest neighbor matching.

In a subsequent paper, Kleinke (2018) revisited this question with a similar simulation design, this time focusing on the accuracy of PMM with small sample sizes and optimal donor pool configurations. He also incorporated midastouch into the analysis to determine whether it offers a more stable alternative to fixed size donor pools.

He reported three main findings:

- PMM and midastouch yield accurate results in most scenarios
- The magnitude of observed biases depends on the interaction between the size of the donor pool, the percentage of missing data, the sample size, and the size of regression coefficients in the data generating model.
- midastouch has more appropriate standard errors than PMM

All things considered, these findings indicate that while PMM is generally reliable and well suited for a wide range of applications, its performance can be sensitive to specific characteristics of the data, particularly in small samples. In contrast, midastouch offers a more robust alternative that does not require tuning of parameters and adapts well to data limitations. As Gaffert et al. (2018) and Kleinke (2018) both emphasize, midastouch should be preferred in smaller datasets. However, when feasible, increasing the sample size remains a simple and effective strategy to enhance the reliability of all imputation methods.

Simulations

As outlined in Chapter 6, the characteristics and performance boundaries of PMM remain insufficiently explored. While important contributions have been made by Yu et al. (2007), Vink et al. (2014), Kleinke (2017, 2018), and Gaffert et al. (2018), a comprehensive and systematic evaluation is still lacking. As noted by Kleinke (2018), "currently, there are no evaluation studies that systematically tested if or under what conditions PMM can be used,..."

7.1 Simulation setting

In this study, I evaluate PMM across three simulation settings, each tailored to a different target variable type: continuous, semicontinuous, and discrete. This simulation design is inspired by and expands upon prior work by Vink et al. (2014) and Kleinke (2017). To ensure full control over the data generating process, I generate artificial data sets with known statistical properties. This design based approach to simulation is often used in the case of performance assessment of imputation procedures in official statistics (Vink et al., 2014, P. 66) (Alfons et al., 2010). While the overall simulation structure remains consistent across all simulations, the construction of the underlying target variable Y varies to reflect the different data types with the covariate X being correlated to Y.

The objective is to compare PMM and midastouch with regard to their behavior under different target variable types. For each simulation scenario, a sample of $n = 3000$ observations is generated. This sample size was chosen to reflect realistic empirical data settings, while remaining small enough to observe the effects of noise and missingness.

For all simulations I generate the missingness with MCAR and MAR (left tailed and right tailed). For MCAR, missing values are generated by drawing from a binomial distribution with a fixed probability γ. For MAR we also draw from a binomial distribution, but the missingness probability is determined by the logit

	Simulation 1	Simulation 2	Simulation 3
Type	continuous	semicontinuous	discreet
distribution	$\sim N(\mu = 5, \sigma = 1)$	$Y \sim N(\mu = 5, \sigma = 1)$ $\begin{cases} Y^4/max(Y^3), & \text{1-p} \\ 0, & \text{p} \end{cases}$ with point mass p	$\sim Pois(\lambda = 4)$
m	5, 10, 25, 40, HMI		
k	1, 3, 5, 10, midas		
γ	10, 20, 40 (%)		
Missing Data Pattern	MCAR, MAR (left-tailed and right-tailed)		

Table 7.1: Overview of Simulations

model (Vink et al., 2014, P. 69) (Kleinke, 2017, eq. 2):

$$P(R = 0) = \frac{exp(a)}{1 + exp(a)}, \tag{7.1.1}$$

where a controls the probability based on the value of the observation x_i. In the left tailed MAR mechanism, values are more likely to be missing, when the value of x_i is low and vice versa for right tailed MAR. To shift the logistic curve and achieve a specified degree of missingness, the variable x_i is standardized and the constant c is calculated using the logit transformation of the target missingness proportion (7.1.2). The MAR mechanisms are visualized in 8.1:

$$\begin{aligned} a &= c + \frac{\bar{x} - x_i}{\sigma}, \, left\, tailed \\ a &= c - \frac{\bar{x} - x_i}{\sigma}, \, right\, tailed \\ \rightarrow c &= logit(target\_missingness) \end{aligned} \tag{7.1.2}$$

For all simulations, the MICE algorithm is initialized with a initial random sampling imputation and runs for five cycles. Convergence is monitored sporadically by plotting imputed values against iteration cycles, with no signs of non convergence observed during testing (see figure 4.1). The overall simulation is structured as a factorial experiment, incorporating combinations of different pa-

rameter settings. Specifically, the design includes three conditions for the number of imputations m, four settings for donor size k, and three levels of missingness proportions γ. Additionally these are crossed with three missingness mechanisms (MCAR, left tailed MAR, right tailed MAR) and three data types (continuous, semicontinuous, discrete). On top of that we include two advanced imputation strategies: the two stage procedure How Many Imputations, which dynamically adjusts m and midastouch, which allows donor selection based on distance rather than fixed k.

Each unique simulation setting was replicated 500 times. For each replication, mean together with the standard deviation of the estimate was recorded. In total, the simulation study comprises of 675 distinct configurations. The results of all simulations can be found in the Appendix (See 8) and are archived in .csv format accessible via the project's GitHub repository. Note that the bias values are multiplied by 100 for improved visibility, as their original values are extremely small. Vink et al. (2014) reports these bias values with two decimal places, yet frequently observes values as low as 0.00–0.01 in their simulations. This scaling allows for easier comparison and interpretation of the results.

The evaluation of the simulation results will be based on the bias of point estimates, coverage rates (CR), Confidence interval (CI) widths and Mean squared error (MSE). Bias is measured as the difference between the true parameter θ and its estimate $\hat{\theta}$. Coverage rate refers to the proportion of simulations in which the 95% confidence interval includes the true parameter value. According to Schafer and Graham (2002, P. 157), a coverage rate below 90% is undercoverage and may indicate poor standard error estimation.

The MSE is particularly relevant from a practical perspective, as complete datasets are returned after a single imputation iteration. Imputation methods that yield estimators with low bias, but excessively high variance are often less desirable than those that strike a better balance, even if slightly biased. Ideally, a robust imputation method should produce nearly unbiased estimates with narrow confidence intervals, high coverage, and low MSE.

While much of the existing literature has focused on bias, CI width and coverage, this study aims to extend the evaluation framework by incorporating MSE. The results should be comparable to the related work and serve to support and validate their conclusions. By replicating elements from prior work while intro-

ducing new variables and evaluation criteria, this simulation study contributes to a more comprehensive understanding.

Algorithm 3 Simulation Procedure

1. Generate a target variable Y and a covariate X such that X is correlated with Y at a specified level ρ.
2. Introduce missingness in Y based on a chosen mechanism (MCAR or MAR) with missingness probability γ.
3. Apply multiple imputation to handle missing values in the data.
4. Repeat steps 1–3 across all variable combinations, performing 500 replications for each setting, and record the performance metrics.
5. Aggregate the results by computing the mean and standard deviation for each evaluation metric across replications.

7.2 Simulation 1: Continuous variable

The first simulation is based on a modified version of the data generation process used by Vink et al. (2014), who evaluated the performance of PMM with semi-continuous variables. In our adaptation, we omit the point mass component to focus solely on a continuous target variable $Y \sim N(5, 1)$. This choice of distribution mirrors that in the original study, ensuring comparability of results across studies.

To introduce a correlation structure, the covariate X is constructed to achieve a target correlation of $\rho = 0.5$ with Y:

$$X = \rho \frac{y_i - \bar{y}}{\sigma} + \sqrt{1 - \rho^2} * \epsilon, \tag{7.2.1}$$

, with $\frac{y_i - \bar{y}}{\sigma}$ being the standardized random variate and $\epsilon \sim N(0, 1)$. This formula is simply a version of Cholesky Decomposition and often used in Monte Carlo Simulations to generate correlated data (Burgess, 2022, P. 10).

7.3 Simulation 2: Semicontinuous variable

The second simulation reproduces the semicontinuous setting used in Vink et al. (2014). The target variable $Y \sim N(5, 1)$, is transformed to induce right skewness using the function $Y = Y^4/max(Y^3)$. This transformation introduces right skewness into the distribution and shifts it toward the point mass at zero. Creating skewed variables like this introduces outliers, which can severely impact the performance of non robust imputation methods.

To reflect the semicontinuous nature, we then introduce a point mass at zero by drawing from a binomial distribution, assigning a value of zero to each observation with a probability of 25%

$$Y = \begin{cases} \sim Y^4/max(Y^3), & 1 \text{ - p} \\ 0, & \text{p} \end{cases} \quad with\, Y \sim N(5, 1)$$

The covariate X is again generated using the Cholesky decomposition from Simulation 1 (equation 7.2.1) with $\rho = 0.5$. , leveraging the fact that the continuous component of Y is still normally distributed. The resulting distribution structure is visualized in Figure 8.3.

7.4 Simulation 3: Poisson distributed variable

The third simulation is inspired by Kleinke (2017), who explored imputation performance with skewed data. However, our implementation diverges in several aspects, most notably by correlating the Covariate X with the Poisson distributed variable Y. Kleinke uses a standard normal distributed X and introduces dependency by simulating Y with $Y \sim Pois(exp(1 + b_1 x_i)$. In this setup Y increases exponentially with X, and and the skewness is influenced by varying the value of b_1. In contrast, for our simulation, we first draw values from the Poisson distribution and then generate corresponding X values to ensure a desired level of correlation.

Since Cholesky based correlation only ensures proper correlation for normally distributed variables, it is not suitable here. Instead, we employ a Gaussian copula approach (Schölzel & Friederichs, 2008, P. 763) (Implementation: copulas-in-

python) to simulate a correlated pair (X, Y) with a desired marginal distribution for $Y \sim Pois(\lambda = 4)$ and standard normal X.

To do so, we first sample from a bivariate normal distribution with a trial correlation and then transform the latent variables using the inverse CDF to achieve the intended Poisson margins. The empirical correlation between X and Y is then checked, and the latent correlation is iteratively adjusted until the observed correlation converges to $\rho = 0.5$ within a tolerance of 0.01. This method ensures a valid dependence structure even for non normal marginals and results in the poisson distribution shown in Figure 8.4.

7.5 Simulation Results

Previous studies have evaluated imputation methods under extreme conditions, such as very small sample sizes or highly skewed distributions (Kleinke, 2017), or have focused on varying correlation structures and point mass probabilities (Vink et al., 2014). In contrast, the present study is conducted under more realistic assumptions, a relatively large sample size ($n = 3000$) and a moderately skewed variable. The goal is to explore the performance of imputation methods in settings that are closer to typical applied research scenarios.

7.5.1 Underestimation of Bias

As discussed in Section 4.4, PMM has been criticized for underestimating the between imputation variance, because it partially omits the posterior draw step. This shortcoming is addressed by the midastouch method. Across all our simulations, we observe that MSE of PMM remains nearly constant within each level of missingness probability (γ), regardless of configuration. However, when comparing across increasing levels of γ, the MSE of PMM decreases. This pattern supports the concern that PMM underestimates the variance, as increasing the number of imputations results in a lower MSE, even though the amount of observed information decreases. In our simulation with high donor density, missing values are often imputed with plausible values close to observed data points. This leads the imputation to be skewed towards the mean, thereby reducing the overall variance and bias. The midastouch method shows a more stable MSE

across all levels of γ, indicating a more consistent estimation of variance. However, this comes at the cost of slightly higher bias, wider confidence intervals, and increased MSE in comparison to PMM, which is per se not worse in this context.

This is most evident in Simulation 2, where the MSE of PMM is noticeably lower than that of midastouch, despite PMM exhibiting significantly higher bias. This discrepancy highlights PMMs tendency to favor lower variance imputations, even when they introduce systematic bias. In the case of semicontinuous variables our result shows clear underestimation of bias when using PMM. While this pattern is not observed under MCAR, both MAR scenarios show substantially higher bias. Under MCAR, the bias remains comparable to that of the continuous and Poisson distributed variables.

The effect is driven by the distributional characteristics of Simulation 2 and the directional missingness under MAR (see Figure 8.3). In the case of left tailed MAR, this means more missing values towards the point mass as X is correlated to Y and conversely, for right tailed MAR, missingness is concentrated in the upper tail. The presence of both a point mass and extreme outliers introduced by the skewed distribution presents significant challenges for PMM, as these values are often too distant from potential donors to be reliably imputed. In contrast, midastouch generally performs better in these scenarios, exhibiting lower bias and higher coverage rates. This is expected as midastouch calculates the drawing probabilities depending on the distance to the recipient, compared to randomly selecting from k nearest neighbors.

Depending on the application, the trade off of accepting a higher bias in exchange for lower variance can be advantageous. However, in our case, additional metrics such as coverage must also be considered. Here, it becomes clear that the bias is excessively large, leading to PMM's low coverage and midastouch should be preferred.

7.5.2 CI widths

In general confidence interval widths increase with rising proportions of missing data. This is a well documented and desirable feature of multiple imputation as it incorporates the added uncertainty from missing data into the variance estimation, which results in wider CIs when more information is missing (Stavseth

	$\gamma = 0.1$	$\gamma = 0.2$	$\gamma = 0.4$
m = 5	20%	40%	80%
m = 10	10%	20%	40%
m = 25	4%	8%	16%
m = 40	2.5%	5%	10%

Table 7.2: Variance increase compared to m=∞ in percent

et al., 2019, e.g Tbl. 3). This can also be seen here. In our simulations, the CI width estimate remains relatively stable within each value of γ, largely due to the large sample size. In other simulations, we observe that smaller sample sizes tend to produce wider CI estimates (Vink et al., 2014, P.76) (Kleinke, 2017, Tbl. 1).

From Equation 4.2.1, we expect the total variance of the estimate to be decreasing depending on the number of imputations (see Table 7.2). This is reflected in the standard deviation of the CI width. Although the estimated width remains consistent across different values of m, its standard deviation decreases as m increases. This effect becomes more pronounced with larger values of γ, and is observed for both the PMM and midastouch methods. Notably, the HowManyImputations method consistently outperforms other configurations in terms of producing lower standard deviations. Even with lower γ and consequently less iterations than the fixed m the standard deviation remains within the range of the third decimal place.

7.5.3 Coverage and Missingness patterns

When evaluating the simulation results for PMM with a focus on coverage, it becomes clear that performance is influenced by several factors, including the missingness pattern, the probability of missingness, and the type of variable involved. In the case of MCAR, the coverage remains relatively stable across all variable types. Specifically, the coverage is approximately 1.00 for $\gamma = 0.1$, around 0.99 for $\gamma = 0.2$, and decreases slightly to about 0.94 for $\gamma = 0.4$. While some variability exists between different variable combinations, this can largely be attributed to inherent simulation variability rather than systematic deficiencies in PMM.

In Simulation 1, the coverage for MAR with both left and right tailed mechanisms closely mirrors the trends observed under MCAR. This pattern is also observed in Simulation 3, particularly for left tailed MAR scenarios. These results suggest that PMM performs robustly when the missingness mechanism does not heavily distort the distribution of the observed data.

However, deviations in performance arise in the context of Poisson distributed variables under right tailed MAR mechanisms. In such cases especially when $\gamma = 0.4$ the coverage significantly deteriorates, resulting in substantial undercoverage. This decline is primarily due to the underestimation. Both Simulation 2 and Simulation 3 exhibit strong right skewness, meaning that higher value outliers are more likely to be missing under a right tailed MAR mechanism. As these outliers are systematically excluded, PMM struggles to accurately model the full range of the data, leading to biased imputations and reduced coverage.

This is particularly pronounced in Simulation 2. As discussed in Section 7.5.1, the estimates under the semicontinuous setting show significant bias. The skewed nature of the distribution, combined with the directional missingness, severely limits PMM's ability to recover the true distributional characteristics of the data. The more pronounced the skewness, the more distorted the imputations become (Kleinke, 2017, Tbl. 1).

7.5.4 Comparison of Results

The simulations do not reflect the effect of donor size, because donor availability is too high and the ratio of donor size to total sample size is too small. In contrast, Kleinke (2017) demonstrates a clear increase in bias when donor size is large and sample size is small. Specifically, when $n < 100$, the default setting of $k = 5$ donors leads to suboptimal results in terms of bias, particularly in situations with high skewness and substantial amounts of missing data. Even when adjusting k, the fundamental issue of limited suitable donor cases may remain unresolved. This problem becomes increasingly severe as sample size decreases and the proportion of missing data grows.

When comparing our results from Simulation 2 to those of Vink et al. (2014, Tbl. 2: Y3), I observe quiet big differences. Unlike in my simulation, their results show no issues with bias or coverage. Although the simulation settings are otherwise identical, they use a smaller sample size of $n = 500$ and $k = 5$.

		Vink et al. (2014)			Replicated		
MAR	pm	Bias	Coverage	Width	Bias	Coverage	Width
left	0.3	0.00	0.94	0.17	0.01	0.97	0.24
right	0.3	0.00	0.96	0.22	-0.04	0.83	0.32
left	0.5	0.00	0.96	0.15	0.00	0.97	0.21
right	0.5	0.00	0.86	0.19	-0.04	0.84	0.29

Table 7.3: Replication of Vink et al. (2014)
Table 2: $n = 500$, MAR with 50%, m = 5, k = 5, 100 replications

When I replicate their settings (see Table 7.3), my results align more closely with theirs. This suggests that the observed underperformance in my simulations is due to the smaller sample size, as neither m nor k appears to significantly influence coverage in my setting. The immediate proximity of donors from bigger sample size, small variance and skewness of the distribution makes imputation challenging.

Conclusion

This thesis has explored the performance of Predictive Mean Matching and midastouch in the presence of missing data, with a focus on how its effectiveness varies under different missingness mechanisms: MCAR, MAR (left and right tailed) and across variable types, including continuous, semicontinuous, and count data.

The results demonstrate that PMM performs reliably under MCAR conditions, with coverage remaining close to nominal levels even as the missingness probability increases. This consistency is maintained across most variable types, suggesting that PMM is well suited for scenarios where missingness is random and uninformative. Similarly, under MAR with moderate skew or when the tail affected by missingness is populated, PMM maintains reasonable performance.

However, the method's limitations become apparent under more complex or structured missingness patterns, especially in the presence of heavily skewed distributions. In these cases, the performance of PMM degrades, leading to systematic undercoverage and biased estimates. The simulations clearly indicate that the interaction between the severity of skewness and the direction of missingness can significantly affect PMM's accuracy. As a result, relying on PMM in such contexts without careful diagnostics and adjustments can be problematic.

While PMM can be effective, it is often quite finicky, requiring careful adjustment of variables based on the specific characteristics of the data structure. In practical applications, where data complexity and missingness patterns are not always known in advance, running full scale simulation studies to identify the optimal imputation configuration is impractical. From a pragmatic standpoint, researchers and practitioners are better served by a more robust and less parameter sensitive approach, that yields sensible results with minimal manual intervention.

midastouch consistently performs as well as, or better than, PMM in terms of coverage. Unlike PMM, midastouch tends not to underestimate bias, thereby

producing more reliable inference. This superior performance is particularly pronounced in settings involving small sample sizes or skewed distributions, where PMM struggles. The findings of this thesis are in line with previous simulation studies by Gaffert et al. (2018) and Kleinke (2018), which also report midastouch as a more stable and accurate method across a wide range of conditions.

An additional practical advantage of midastouch is its compatibility with HMI. When used in conjunction with HMI, midastouch eliminates the need to specify any parameters, reducing the risk of introducing error through poor parameter selection. This results in a ready to use imputation method without burdening the user with extensive tuning. Consequently, researchers can allocate more attention to specifying appropriate imputation models and analysis models.

The findings of this study carry several important implications. First, they highlight the importance of considering both the data structure and the missingness mechanism when choosing an imputation strategy. Methods that perform well under MCAR may fail under MAR, particularly in the presence of skewed data. Second, the results highlight that default settings in widely used software packages may not be optimal, especially when dealing with small samples or non normal distributions. The use of adaptive or model informed imputation techniques, such as midastouch, may provide a more reliable alternative.

Nevertheless, some limitations of the present study should be acknowledged. While this study considered a variety of variable types, the simulations were conducted in relatively controlled, low dimensional settings. The simulations focused on univariate missingness patterns, where only one variable at a time had missing values. Usually missingness is multivariate and interdependent. Additionally the study relied on predefined imputation model structures, assuming that the correct predictors were included and appropriately specified. In applied research, however, misspecification of the imputation model is a common source of error, particularly when variables are omitted or when non linear relationships are not adequately captured. The robustness of midastouch and PMM to such model misspecification was not systematically tested, and future research should explore how sensitive these methods are to modelling errors.

In conclusion, this thesis contributes to the growing body of evidence supporting midastouch as a strong default imputation method, particularly when paired with HMI. While PMM remains a viable option under well behaved data and simple

missingness structures, its sensitivity to data characteristics makes it less suitable for routine use without diagnostic checks. Midastouch, offers a practical and statistically sound solution for a wide range of missing data problems, aligning well with the needs of researchers seeking valid inference.

Appendix

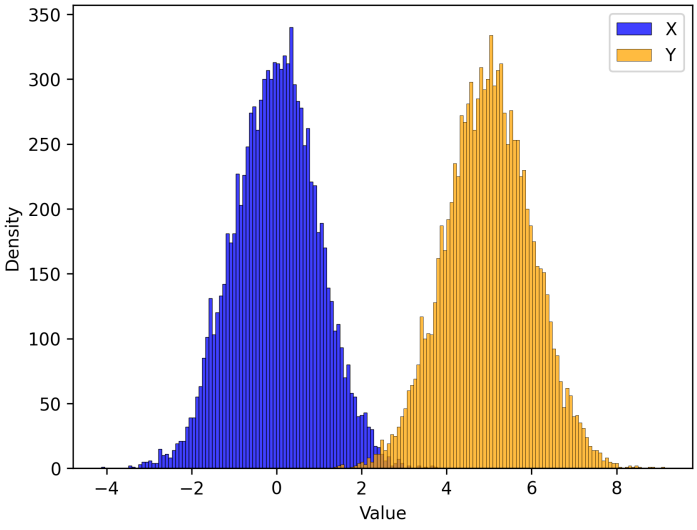


Figure 8.2: Distribution of the data in Simulation 1. The sample consists of $n = 10000$ observations where the target variable Y follows a normal distribution $Y \sim N(5, 1)$, and the covariate X is generated to have a correlation of $\rho = 0.5$ with Y using Cholesky decomposition.

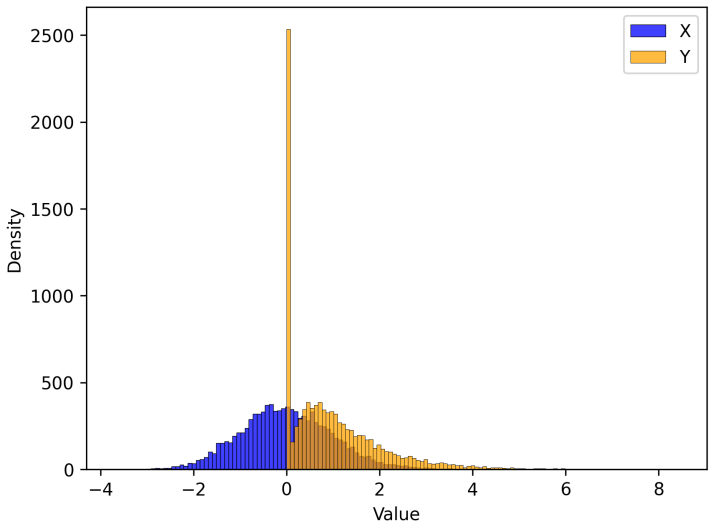


Figure 8.3: Distribution of the data in Simulation 2. The sample consists of $n = 10000$ observations, where the target variable Y is initially generated from a normal distribution $Y \sim N(5, 1)$ and then transformed using $Y^4/\max(Y^3)$ to create a semicontinuous distribution. A point mass at zero is introduced with a 25% probability. The covariate X is generated to have a correlation of $\rho = 0.5$ with Y using Cholesky decomposition.

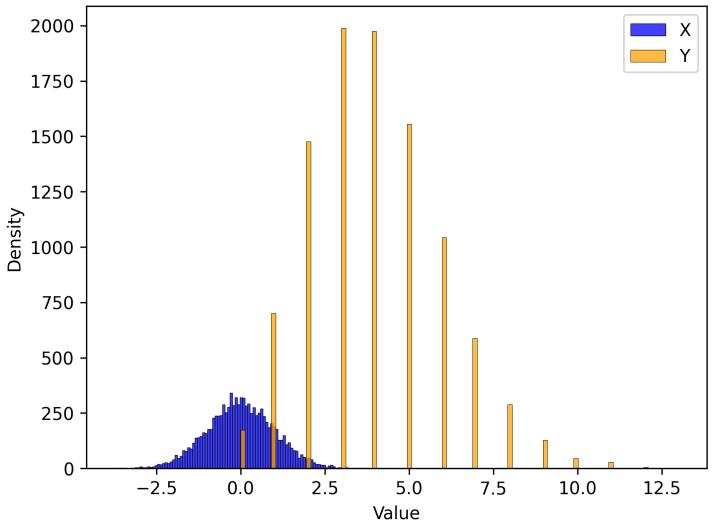


Figure 8.4: Distribution of the data in Simulation 3. The sample consists of $n = 10000$ observations, where the target variable Y follows a Poisson distribution $Y \sim Pois(4)$. The covariate X is generated to have a correlation of $\rho = 0.5$ with Y using a Gaussian copula.

Parameters				scaled by $1e^2$ Bias	Width	MSE	Coverage
MCAR	$\gamma = 10\%$	k=1	m=5	.005 (.938)	.059 (.001)	.675 (.022)	.998
			m=10	.001 (.967)	.059 (.001)	.674 (.022)	.996
			m=25	.044 (1.000)	.059 (.001)	.673 (.023)	1.000
			m=40	.097 (.925)	.059 (.001)	.676 (.021)	.998
			HMI 10 (0)	.002 (.944)	.059 (.001)	.674 (.022)	1.000
		k = 3	m=5	.032 (.878)	.059 (.001)	.675 (.020)	1.000
			m=10	-.091 (.914)	.059 (.001)	.675 (.022)	1.000
			m=25	.013 (.962)	.059 (.001)	.674 (.022)	.998
			m=40	-.039 (.942)	.059 (.001)	.675 (.023)	.994
			HMI 10 (0)	.019 (1.030)	.059 (.001)	.673 (.022)	1.000
		k = 5	m=5	-.063 (.972)	.059 (.001)	.676 (.021)	.998
			m=10	-.044 (.934)	.059 (.001)	.676 (.021)	.998
			m=25	.040 (.995)	.059 (.001)	.675 (.022)	1.000
			m=40	.009 (.973)	.059 (.001)	.674 (.021)	.996
			HMI 10 (0)	-.004 (.982)	.059 (.001)	.674 (.021)	.998
		k = 10	m=5	-.034 (.961)	.059 (.001)	.674 (.023)	.992
			m=10	-.026 (.950)	.059 (.001)	.675 (.022)	1.000
			m=25	-.010 (.980)	.059 (.001)	.674 (.023)	.998
			m=40	-.022 (.963)	.059 (.001)	.676 (.023)	1.000
			HMI 10 (0)	.024 (.884)	.059 (.001)	.674 (.021)	1.000
		midas	m=5	.056 (1.021)	.065 (.002)	.750 (.024)	.998
			m=10	.032 (1.062)	.064 (.002)	.748 (.024)	.994
			m=25	.030 (.973)	.064 (.001)	.749 (.024)	1.000
			m=40	.043 (.998)	.064 (.001)	.749 (.023)	.998
			HMI 12.00 (3.87)	-.029 (.930)	.064 (.001)	.749 (.025)	1.000
				estimate (std of estimate)			

Table 8.1: Simulation 1, MCAR, $\gamma = 10\%$

				scaled by $1e^2$			
Parameters				Bias	Width	MSE	Coverage
MCAR	$\gamma = 20\%$	k=1	m=5	-.030 (1.084)	.057 (.002)	.598 (.020)	.994
			m=10	.046 (1.079)	.057 (.001)	.598 (.020)	.994
			m=25	.024 (1.142)	.057 (.001)	.599 (.020)	.986
			m=40	.043 (1.068)	.057 (.001)	.600 (.021)	.996
			HMI 10.96 (2.01)	.090 (1.148)	.057 (.001)	.598 (.021)	.984
		k = 3	m=5	.091 (1.138)	.057 (.002)	.600 (.022)	.980
			m=10	-.024 (1.123)	.057 (.001)	.598 (.020)	.988
			m=25	-.046 (1.140)	.057 (.001)	.599 (.020)	.992
			m=40	.015 (1.070)	.057 (.001)	.601 (.021)	.994
			HMI 10.66 (1.87)	.062 (1.108)	.057 (.001)	.600 (.021)	.986
		k = 5	m=5	-.029 (1.133)	.057 (.002)	.601 (.020)	.986
			m=10	.029 (1.114)	.057 (.001)	.598 (.021)	.984
			m=25	.062 (1.148)	.057 (.001)	.599 (.019)	.998
			m=40	.018 (1.102)	.057 (.001)	.600 (.020)	.992
			HMI 10.75 (2.02)	.039 (1.145)	.057 (.001)	.600 (.021)	.990
		k = 10	m=5	-.004 (1.070)	.058 (.002)	.599 (.020)	.994
			m=10	-.061 (1.104)	.057 (.001)	.599 (.021)	.986
			m=25	-.023 (1.069)	.057 (.001)	.602 (.022)	.996
			m=40	-.015 (1.058)	.057 (.001)	.600 (.020)	.998
			HMI 10.62 (1.80)	-.078 (1.109)	.057 (.001)	.600 (.018)	.992
		midas	m=5	.070 (1.248)	.068 (.004)	.750 (.026)	.998
			m=10	-.058 (1.264)	.068 (.003)	.748 (.026)	.990
			m=25	-.066 (1.215)	.067 (.002)	.749 (.026)	.998
			m=40	-.003 (1.187)	.067 (.002)	.750 (.025)	.998
			HMI 29.54 (12.64)	.067 (1.233)	.067 (.002)	.750 (.025)	.990
				estimate (std of estimate)			

Table 8.2: Simulation 1, MCAR, $\gamma = 20\%$

				scaled by $1e^2$			
Parameters				Bias	Width	MSE	Coverage
MCAR	$\gamma = 40\%$	k=1	m=5	.047 (1.486)	.059 (.007)	.448 (.018)	.932
			m=10	-.019 (1.448)	.058 (.004)	.450 (.018)	.954
			m=25	-.006 (1.509)	.058 (.003)	.451 (.018)	.954
			m=40	.066 (1.534)	.058 (.002)	.450 (.018)	.936
			HMI 65.93 (20.81)	-.016 (1.547)	.058 (.002)	.451 (.019)	.940
		k = 3	m=5	.066 (1.809)	.067 (.012)	.439 (.017)	.918
			m=10	.296 (1.867)	.067 (.008)	.437 (.018)	.908
			m=25	.286 (1.761)	.065 (.005)	.438 (.018)	.918
			m=40	.170 (1.669)	.066 (.004)	.435 (.018)	.944
			HMI 67.77 (20.42)	-.123 (1.406)	.058 (.002)	.451 (.019)	.970
		k = 5	m=5	-.129 (1.611)	.059 (.007)	.450 (.018)	.920
			m=10	-.176 (1.430)	.059 (.005)	.450 (.019)	.952
			m=25	-.024 (1.485)	.058 (.003)	.450 (.019)	.940
			m=40	-.119 (1.539)	.058 (.002)	.450 (.019)	.934
			HMI 68.30 (20.30)	.041 (1.572)	.058 (.002)	.449 (.018)	.932
		k = 10	m=5	-.034 (1.538)	.059 (.007)	.450 (.019)	.940
			m=10	.079 (1.494)	.058 (.004)	.450 (.019)	.954
			m=25	-.088 (1.420)	.058 (.003)	.450 (.019)	.962
			m=40	.035 (1.603)	.058 (.002)	.449 (.018)	.932
			HMI 68.98 (20.06)	-.083 (1.472)	.058 (.002)	.450 (.018)	.938
		midas	m=5	.016 (1.801)	.078 (.010)	.749 (.031)	.970
			m=10	.143 (1.744)	.077 (.007)	.750 (.030)	.964
			m=25	.035 (1.692)	.076 (.004)	.749 (.029)	.980
			m=40	.000 (1.668)	.076 (.003)	.749 (.029)	.978
			HMI 77.85 (21.05)	.065 (1.633)	.076 (.003)	.748 (.029)	.982
			estimate (std of estimate)				

Table 8.3: Simulation 1, MCAR, $\gamma = 40\%$

				scaled by $1e^2$			
Parameters				Bias	Width	MSE	Coverage
MAR (right tailed)	$\gamma = 10\%$	k=1	m=5	.035 (.975)	.059 (.001)	.649 (.020)	1.000
			m=10	.040 (1.033)	.059 (.001)	.649 (.021)	.996
			m=25	.081 (1.023)	.059 (.001)	.649 (.021)	1.000
			m=40	.089 (1.063)	.059 (.001)	.651 (.021)	1.000
			HMI 10.42 (1.65)	-.087 (1.027)	.059 (.001)	.649 (.022)	.998
		k = 3	m=5	-.040 (1.024)	.059 (.001)	.651 (.022)	1.000
			m=10	.010 (1.014)	.059 (.001)	.650 (.021)	.994
			m=25	-.014 (.965)	.059 (.001)	.649 (.021)	.996
			m=40	.026 (.983)	.059 (.001)	.650 (.020)	.998
			HMI 10.58 (1.87)	.042 (1.010)	.059 (.001)	.649 (.022)	.994
		k = 5	m=5	.076 (.980)	.059 (.001)	.651 (.020)	1.000
			m=10	.090 (1.007)	.059 (.001)	.649 (.021)	.998
			m=25	.046 (.994)	.059 (.001)	.648 (.022)	.998
			m=40	.086 (1.066)	.059 (.001)	.649 (.021)	.996
			HMI 10.57 (2.03)	.003 (1.010)	.059 (.001)	.648 (.021)	.996
		k = 10	m=5	.076 (1.003)	.059 (.001)	.650 (.022)	.996
			m=10	.066 (1.028)	.059 (.001)	.648 (.022)	.994
			m=25	.065 (1.061)	.059 (.001)	.649 (.021)	.992
			m=40	-.006 (1.070)	.059 (.001)	.650 (.022)	.994
			HMI 10.45 (1.76)	.041 (1.011)	.059 (.001)	.650 (.023)	.996
		midas	m=5	.092 (1.110)	.067 (.004)	.750 (.025)	.998
			m=10	.023 (1.073)	.067 (.002)	.752 (.024)	.998
			m=25	.191 (1.097)	.066 (.002)	.752 (.024)	.996
			m=40	.161 (1.118)	.066 (.002)	.751 (.025)	.998
			HMI 35.22 (18.99)	.096 (1.097)	.066 (.002)	.748 (.025)	.994
				estimate (std of estimate)			

Table 8.4: Simulation 1, MAR (right tailed), $\gamma = 10\%$

				scaled by $1e^2$			
Parameters				Bias	Width	MSE	Coverage
MAR (left tailed)	$\gamma = 20\%$	k=1	m=5	.022 (1.290)	.059 (.004)	.571 (.019)	.976
			m=10	.128 (1.358)	.059 (.002)	.572 (.021)	.962
			m=25	.116 (1.229)	.059 (.002)	.571 (.020)	.978
			m=40	.166 (1.217)	.059 (.002)	.570 (.021)	.984
			HMI 27.55 (14.49)	.061 (1.263)	.058 (.002)	.572 (.020)	.984
		k = 3	m=5	.024 (1.308)	.059 (.004)	.570 (.021)	.978
			m=10	.153 (1.244)	.059 (.002)	.570 (.020)	.976
			m=25	.024 (1.197)	.059 (.002)	.572 (.020)	.988
			m=40	.038 (1.262)	.059 (.001)	.571 (.021)	.986
			HMI 28.14 (14.10)	.103 (1.206)	.058 (.002)	.572 (.021)	.986
		k = 5	m=5	.053 (1.316)	.059 (.004)	.571 (.019)	.972
			m=10	.036 (1.262)	.059 (.002)	.571 (.021)	.986
			m=25	.105 (1.202)	.059 (.002)	.572 (.020)	.982
			m=40	.007 (1.229)	.059 (.001)	.570 (.021)	.984
			HMI 28.17 (14.87)	.017 (1.247)	.058 (.002)	.570 (.021)	.982
		k = 10	m=5	.201 (1.222)	.059 (.004)	.572 (.020)	.984
			m=10	.030 (1.158)	.059 (.003)	.570 (.020)	.986
			m=25	.044 (1.242)	.059 (.002)	.570 (.020)	.982
			m=40	.080 (1.275)	.059 (.001)	.570 (.020)	.986
			HMI 27.79 (14.37)	.139 (1.230)	.058 (.002)	.570 (.021)	.980
		midas	m=5	.121 (1.492)	.072 (.007)	.751 (.029)	.988
			m=10	.197 (1.456)	.072 (.005)	.752 (.026)	.978
			m=25	.242 (1.344)	.071 (.003)	.751 (.026)	.986
			m=40	.291 (1.333)	.071 (.002)	.752 (.029)	.996
			HMI 70.33 (26.51)	.178 (1.402)	.071 (.002)	.755 (.027)	.992
			estimate (std of estimate)				

Table 8.5: Simulation 1, MAR (left tailed), $\gamma = 20\%$

				scaled by $1e^2$			
Parameters				Bias	Width	MSE	Coverage
MAR (left tailed)	$\gamma = 40\%$	k=1	m=5	.157 (1.770)	.068 (.012)	.438 (.017)	.944
			m=10	.241 (1.713)	.067 (.008)	.437 (.018)	.944
			m=25	.066 (1.792)	.066 (.005)	.438 (.019)	.934
			m=40	.241 (1.759)	.065 (.004)	.436 (.017)	.922
			HMI 97.49 (24.69)	.153 (1.732)	.065 (.003)	.439 (.018)	.956
		k = 3	m=5	.066 (1.809)	.067 (.012)	.439 (.017)	.918
			m=10	.296 (1.867)	.067 (.008)	.437 (.018)	.908
			m=25	.286 (1.761)	.065 (.005)	.438 (.018)	.918
			m=40	.170 (1.669)	.066 (.004)	.435 (.018)	.944
			HMI 99.34 (24.83)	.209 (1.683)	.065 (.003)	.439 (.019)	.946
		k = 5	m=5	.150 (1.706)	.067 (.012)	.438 (.017)	.956
			m=10	.220 (1.793)	.067 (.008)	.438 (.017)	.922
			m=25	.138 (1.703)	.066 (.005)	.437 (.017)	.946
			m=40	.045 (1.691)	.066 (.004)	.438 (.017)	.940
			HMI 97.97 (24.75)	.124 (1.727)	.065 (.003)	.439 (.018)	.932
		k = 10	m=5	.126 (1.819)	.067 (.011)	.438 (.017)	.920
			m=10	.235 (1.768)	.066 (.007)	.439 (.018)	.916
			m=25	.286 (1.739)	.066 (.005)	.437 (.018)	.936
			m=40	.140 (1.711)	.065 (.004)	.438 (.018)	.946
			HMI 96.21 (26.49)	.180 (1.743)	.065 (.003)	.438 (.018)	.926
		midas	m=5	.446 (2.084)	.088 (.016)	.753 (.034)	.948
			m=10	.558 (1.989)	.086 (.010)	.754 (.033)	.956
			m=25	.504 (1.983)	.085 (.007)	.756 (.033)	.972
			m=40	.348 (1.961)	.085 (.005)	.754 (.031)	.974
			HMI 123.34 (24.5)	.372 (2.081)	.085 (.004)	.755 (.032)	.960
			estimate (std of estimate)				

Table 8.6: Simulation 1, MAR (left tailed), $\gamma = 40\%$

				scaled by $1e^2$			
Parameters				Bias	Width	MSE	Coverage
MAR (right tailed)	$\gamma = 10\%$	k=1	m=5	-.003 (1.054)	.059 (.001)	.647 (.022)	.994
			m=10	-.031 (1.036)	.059 (.001)	.649 (.021)	.996
			m=25	-.051 (1.061)	.059 (.001)	.648 (.022)	.998
			m=40	-.000 (1.040)	.059 (.001)	.650 (.021)	.996
			HMI 10.65 (2.32)	-.104 (1.021)	.059 (.001)	.650 (.022)	.990
		k = 3	m=5	-.055 (1.097)	.059 (.001)	.649 (.022)	.996
			m=10	-.056 (1.009)	.059 (.001)	.649 (.022)	.992
			m=25	-.072 (1.038)	.059 (.001)	.652 (.021)	.992
			m=40	-.077 (.997)	.059 (.001)	.651 (.023)	.998
			HMI 10.41 (1.49)	-.076 (1.011)	.059 (.001)	.650 (.022)	.996
		k = 5	m=5	-.031 (1.089)	.059 (.001)	.648 (.021)	.994
			m=10	-.099 (1.033)	.059 (.001)	.647 (.022)	.990
			m=25	-.030 (1.051)	.059 (.001)	.650 (.022)	.988
			m=40	.015 (1.030)	.059 (.001)	.649 (.022)	.996
			HMI 10.47 (1.36)	-.094 (1.053)	.059 (.001)	.649 (.022)	.994
		k = 10	m=5	-.037 (1.013)	.059 (.001)	.650 (.022)	.996
			m=10	-.090 (1.038)	.059 (.001)	.651 (.022)	.998
			m=25	.013 (1.020)	.059 (.001)	.648 (.021)	.996
			m=40	-.028 (1.007)	.059 (.001)	.649 (.021)	.992
			HMI 10.52 (1.82)	-.019 (.979)	.059 (.001)	.650 (.022)	1.000
		midas	m=5	-.131 (1.131)	.067 (.003)	.751 (.026)	.992
			m=10	-.071 (1.125)	.066 (.002)	.750 (.025)	.998
			m=25	-.047 (1.119)	.066 (.002)	.751 (.024)	.998
			m=40	-.030 (1.068)	.066 (.002)	.751 (.025)	.998
			HMI 35.19 (18.94)	-.125 (1.072)	.066 (.002)	.751 (.026)	.998
				estimate (std of estimate)			

Table 8.7: Simulation 1, MAR (right tailed), $\gamma = 10\%$

Parameters				scaled by $1e^2$ Bias	Width	MSE	Coverage
MAR (right tailed)	$\gamma = 20\%$	k=1	m=5	-.138 (1.316)	.059 (.004)	.571 (.020)	.974
			m=10	-.041 (1.247)	.059 (.002)	.572 (.021)	.980
			m=25	.020 (1.270)	.059 (.002)	.570 (.020)	.976
			m=40	-.060 (1.272)	.059 (.001)	.571 (.021)	.974
			HMI 27.40 (14.57)	-.050 (1.146)	.058 (.002)	.571 (.020)	.982
		k = 3	m=5	-.170 (1.255)	.059 (.004)	.571 (.021)	.986
			m=10	-.070 (1.356)	.059 (.002)	.571 (.020)	.966
			m=25	-.163 (1.279)	.059 (.002)	.569 (.021)	.978
			m=40	-.161 (1.274)	.059 (.001)	.571 (.019)	.982
			HMI 27.06 (14.62)	-.075 (1.258)	.058 (.002)	.570 (.020)	.974
		k = 5	m=5	.009 (1.312)	.060 (.004)	.569 (.019)	.974
			m=10	-.046 (1.236)	.059 (.003)	.570 (.021)	.982
			m=25	-.043 (1.219)	.059 (.002)	.570 (.020)	.976
			m=40	-.079 (1.284)	.059 (.001)	.569 (.021)	.972
			HMI 27.29 (13.75)	-.151 (1.239)	.058 (.002)	.570 (.019)	.984
		k = 10	m=5	-.176 (1.274)	.059 (.004)	.570 (.022)	.976
			m=10	-.179 (1.262)	.059 (.002)	.572 (.021)	.974
			m=25	-.130 (1.291)	.059 (.002)	.571 (.020)	.974
			m=40	-.106 (1.233)	.059 (.001)	.570 (.021)	.984
			HMI 27.87 (13.95)	-.068 (1.233)	.058 (.002)	.571 (.021)	.982
		midas	m=5	-.236 (1.407)	.072 (.007)	.750 (.027)	.984
			m=10	-.169 (1.418)	.072 (.005)	.752 (.029)	.986
			m=25	-.198 (1.328)	.071 (.003)	.752 (.027)	.990
			m=40	-.153 (1.376)	.071 (.003)	.751 (.027)	.980
			HMI 70.50 (26.34)	-.272 (1.327)	.071 (.002)	.754 (.026)	.984
				estimate (std of estimate)			

Table 8.8: Simulation 1, MAR (right tailed), $\gamma = 20\%$

				scaled by $1e^2$			
Parameters				Bias	Width	MSE	Coverage
MAR (right tailed)	$\gamma = 40\%$	k=1	m=5	-.189 (1.742)	.067 (.011)	.438 (.017)	.942
			m=10	-.152 (1.798)	.066 (.008)	.438 (.017)	.926
			m=25	-.137 (1.678)	.066 (.005)	.439 (.017)	.936
			m=40	-.162 (1.701)	.066 (.004)	.438 (.017)	.934
			HMI 97.67 (24.61)	-.175 (1.719)	.065 (.003)	.437 (.019)	.940
		k = 3	m=5	-.129 (1.827)	.067 (.012)	.437 (.018)	.930
			m=10	-.222 (1.750)	.067 (.008)	.435 (.018)	.932
			m=25	-.105 (1.744)	.066 (.004)	.437 (.017)	.942
			m=40	-.205 (1.688)	.066 (.004)	.437 (.018)	.950
			HMI 97.79 (25.09)	-.243 (1.796)	.065 (.003)	.437 (.018)	.924
		k = 5	m=5	-.095 (1.717)	.067 (.011)	.437 (.017)	.936
			m=10	-.285 (1.696)	.066 (.008)	.437 (.017)	.938
			m=25	-.347 (1.703)	.066 (.004)	.438 (.017)	.950
			m=40	-.248 (1.685)	.066 (.004)	.438 (.018)	.956
			HMI 97.04 (25.06)	-.265 (1.760)	.065 (.003)	.439 (.018)	.924
		k = 10	m=5	-.115 (1.903)	.068 (.012)	.436 (.017)	.914
			m=10	-.099 (1.744)	.066 (.008)	.438 (.018)	.938
			m=25	-.206 (1.670)	.066 (.005)	.438 (.018)	.952
			m=40	-.266 (1.774)	.066 (.004)	.438 (.018)	.932
			HMI 96.47 (26.07)	-.197 (1.696)	.065 (.003)	.438 (.016)	.936
		midas	m=5	-.510 (2.110)	.087 (.016)	.753 (.034)	.942
			m=10	-.430 (2.116)	.085 (.010)	.754 (.034)	.950
			m=25	-.443 (1.886)	.085 (.007)	.756 (.033)	.970
			m=40	-.563 (1.969)	.085 (.006)	.754 (.031)	.962
			HMI 120.25 (24.6)	-.268 (1.903)	.085 (.004)	.757 (.031)	.976
				estimate (std of estimate)			

Table 8.9: Simulation 1, MAR (right tailed), $\gamma = 40\%$

				scaled by $1e^2$			
Parameters				Bias	Width	MSE	Coverage
MCAR	$\gamma = 10\%$	k=1	m=5	.007 (.950)	.059 (.006)	.687 (.137)	.998
			m=10	-.026 (.959)	.060 (.006)	.700 (.139)	.998
			m=25	.102 (.984)	.060 (.006)	.692 (.132)	1.000
			m=40	.059 (.960)	.060 (.006)	.693 (.144)	.996
			HMI 10 (0)	.015 (1.008)	.060 (.006)	.690 (.134)	.998
		k = 3	m=5	.018 (.948)	.060 (.006)	.697 (.142)	1.000
			m=10	.027 (.931)	.060 (.006)	.693 (.137)	1.000
			m=25	.018 (.954)	.060 (.006)	.693 (.138)	1.000
			m=40	.034 (.926)	.059 (.006)	.688 (.145)	.998
			HMI 10 (0)	-.026 (.982)	.060 (.006)	.696 (.142)	1.000
		k = 5	m=5	.076 (.918)	.059 (.006)	.688 (.144)	1.000
			m=10	.048 (.967)	.059 (.006)	.687 (.131)	.998
			m=25	-.108 (.995)	.060 (.006)	.691 (.144)	1.000
			m=40	-.007 (.933)	.060 (.006)	.695 (.134)	.998
			HMI 10 (0)	-.006 (.917)	.060 (.006)	.700 (.142)	.998
		k = 10	m=5	.045 (.957)	.059 (.006)	.677 (.138)	1.000
			m=10	-.036 (.982)	.059 (.006)	.686 (.140)	.992
			m=25	-.066 (.961)	.060 (.006)	.701 (.146)	.998
			m=40	.012 (.942)	.059 (.006)	.689 (.138)	1.000
			HMI 10 (0)	-.016 (.933)	.060 (.006)	.689 (.135)	1.000
		midas	m=5	.016 (1.032)	.065 (.007)	.760 (.148)	.996
			m=10	-.038 (.944)	.065 (.007)	.772 (.154)	1.000
			m=25	.015 (1.017)	.065 (.006)	.766 (.145)	1.000
			m=40	-.044 (1.016)	.064 (.007)	.755 (.157)	1.000
			HMI 14.55 (8.22)	.049 (1.009)	.064 (.007)	.762 (.164)	.998
				estimate (std of estimate)			

Table 8.10: Simulation 2, MCAR, $\gamma = 10\%$

				scaled by $1e^2$			
Parameters				Bias	Width	MSE	Coverage
MCAR	$\gamma = 20\%$	k=1	m=5	-.033 (1.152)	.058 (.006)	.607 (.122)	.982
			m=10	-.032 (1.160)	.058 (.006)	.611 (.127)	.990
			m=25	-.032 (1.175)	.057 (.006)	.608 (.123)	.982
			m=40	.010 (1.143)	.058 (.006)	.624 (.127)	.986
			HMI 10.56 (1.72)	-.089 (1.079)	.058 (.006)	.613 (.119)	.982
		k = 3	m=5	-.111 (1.109)	.058 (.006)	.614 (.124)	.992
			m=10	.011 (1.168)	.058 (.006)	.615 (.124)	.990
			m=25	.003 (1.142)	.057 (.006)	.606 (.130)	.990
			m=40	-.077 (1.099)	.057 (.006)	.602 (.123)	.990
			HMI 10.75 (2.06)	.011 (1.270)	.058 (.006)	.615 (.125)	.974
		k = 5	m=5	.059 (1.159)	.058 (.006)	.610 (.124)	.992
			m=10	-.069 (1.163)	.057 (.006)	.610 (.119)	.988
			m=25	-.069 (1.076)	.057 (.006)	.610 (.123)	.998
			m=40	.004 (1.149)	.057 (.006)	.599 (.123)	.986
			HMI 10.71 (1.79)	-.003 (1.164)	.058 (.006)	.619 (.130)	.982
		k = 10	m=5	-.025 (1.148)	.058 (.006)	.609 (.120)	.992
			m=10	-.078 (1.124)	.058 (.006)	.611 (.129)	.990
			m=25	.070 (1.137)	.058 (.006)	.612 (.125)	.980
			m=40	.024 (1.096)	.057 (.006)	.608 (.122)	.990
			HMI 10.63 (1.74)	-.009 (1.142)	.057 (.006)	.612 (.125)	.994
		midas	m=5	-.078 (1.301)	.068 (.008)	.753 (.153)	.990
			m=10	-.057 (1.172)	.067 (.007)	.761 (.157)	.994
			m=25	-.138 (1.222)	.067 (.007)	.762 (.160)	.994
			m=40	.008 (1.142)	.067 (.007)	.760 (.155)	.996
			HMI 35.24 (18.21)	-.076 (1.164)	.067 (.008)	.763 (.162)	.998
				estimate (std of estimate)			

Table 8.11: Simulation 2, MCAR, $\gamma = 20\%$

				scaled by $1e^2$			
Parameters				Bias	Width	MSE	Coverage
MCAR	$\gamma = 40\%$	k=1	m=5	.023 (1.542)	.059 (.010)	.452 (.094)	.934
			m=10	-.093 (1.533)	.058 (.008)	.452 (.096)	.932
			m=25	.095 (1.570)	.058 (.007)	.458 (.095)	.930
			m=40	.155 (1.457)	.058 (.006)	.463 (.090)	.950
			HMI 68.10 (19.97)	.032 (1.588)	.058 (.007)	.454 (.098)	.934
		k = 3	m=5	-.050 (1.650)	.060 (.009)	.457 (.089)	.916
			m=10	.024 (1.476)	.059 (.008)	.456 (.093)	.950
			m=25	-.047 (1.548)	.058 (.007)	.460 (.092)	.944
			m=40	-.010 (1.488)	.058 (.006)	.458 (.095)	.946
			HMI 67.20 (20.11)	-.031 (1.500)	.058 (.006)	.463 (.091)	.942
		k = 5	m=5	.059 (1.542)	.059 (.009)	.462 (.094)	.946
			m=10	-.062 (1.463)	.059 (.008)	.462 (.098)	.952
			m=25	-.033 (1.507)	.058 (.007)	.453 (.097)	.942
			m=40	-.106 (1.623)	.058 (.006)	.455 (.094)	.926
			HMI 67.31 (20.31)	.008 (1.570)	.058 (.006)	.461 (.090)	.938
		k = 10	m=5	.073 (1.546)	.059 (.009)	.452 (.092)	.948
			m=10	.119 (1.445)	.059 (.008)	.455 (.091)	.940
			m=25	-.060 (1.500)	.058 (.007)	.458 (.095)	.956
			m=40	-.192 (1.564)	.058 (.006)	.459 (.094)	.942
			HMI 66.76 (21.45)	.010 (1.425)	.058 (.006)	.463 (.092)	.956
		midas	m=5	-.185 (1.668)	.077 (.013)	.764 (.157)	.980
			m=10	-.338 (1.646)	.076 (.011)	.756 (.159)	.978
			m=25	-.058 (1.609)	.076 (.009)	.767 (.158)	.982
			m=40	-.085 (1.610)	.076 (.009)	.762 (.162)	.988
			HMI 85.92 (25.47)	-.135 (1.618)	.076 (.008)	.772 (.160)	.976
				estimate (std of estimate)			

Table 8.12: Simulation 2, MCAR, $\gamma = 40\%$

				scaled by $1e^2$			
Parameters				Bias	Width	MSE	Coverage
MAR (left tailed)	$\gamma = 10\%$	k=1	m=5	-1.297 (1.007)	.061 (.006)	.698 (.139)	.968
			m=10	-1.363 (1.031)	.060 (.006)	.690 (.136)	.946
			m=25	-1.271 (.988)	.060 (.006)	.691 (.139)	.958
			m=40	-1.392 (1.031)	.061 (.007)	.696 (.145)	.944
			HMI 10.24 (1.15)	-1.419 (1.050)	.060 (.006)	.684 (.144)	.932
		k = 3	m=5	-1.387 (1.045)	.061 (.006)	.704 (.134)	.966
			m=10	-1.323 (.980)	.061 (.006)	.692 (.139)	.948
			m=25	-1.380 (.978)	.061 (.006)	.693 (.138)	.956
			m=40	-1.334 (1.027)	.060 (.006)	.688 (.140)	.946
			HMI 10.14 (0.87)	-1.402 (1.012)	.061 (.006)	.693 (.132)	.950
		k = 5	m=5	-1.340 (1.026)	.060 (.007)	.691 (.149)	.944
			m=10	-1.348 (.978)	.061 (.007)	.695 (.146)	.964
			m=25	-1.413 (.984)	.061 (.006)	.705 (.143)	.958
			m=40	-1.348 (.995)	.060 (.007)	.683 (.146)	.950
			HMI 10.23 (1.13)	-1.336 (1.051)	.060 (.006)	.690 (.136)	.948
		k = 10	m=5	-1.320 (1.061)	.061 (.006)	.695 (.137)	.942
			m=10	-1.320 (1.037)	.061 (.006)	.707 (.142)	.948
			m=25	-1.326 (1.030)	.061 (.006)	.700 (.136)	.950
			m=40	-1.376 (1.045)	.061 (.006)	.698 (.143)	.942
			HMI 10.17 (1.18)	-1.369 (1.012)	.061 (.006)	.701 (.145)	.950
		midas	m=5	-.017 (.989)	.065 (.007)	.767 (.155)	.998
			m=10	-.018 (.998)	.065 (.007)	.763 (.153)	.994
			m=25	.063 (.986)	.065 (.007)	.777 (.152)	.998
			m=40	.028 (1.002)	.064 (.007)	.763 (.156)	1.000
			HMI 12.78 (5.28)	.076 (1.021)	.064 (.007)	.762 (.152)	.998
				estimate (std of estimate)			

Table 8.13: Simulation 2, MAR (left tailed), $\gamma = 10\%$

				scaled by $1e^2$			
Parameters				Bias	Width	MSE	Coverage
MAR (left tailed)	$\gamma = 20\%$	k=1	m=5	-2.414 (1.261)	.062 (.007)	.626 (.127)	.694
			m=10	-2.316 (1.175)	.062 (.007)	.639 (.128)	.752
			m=25	-2.358 (1.245)	.061 (.006)	.637 (.127)	.720
			m=40	-2.331 (1.203)	.060 (.007)	.620 (.141)	.730
			HMI 24.72 (12.41)	-2.338 (1.260)	.061 (.007)	.636 (.132)	.740
		k = 3	m=5	-2.438 (1.213)	.062 (.007)	.639 (.122)	.714
			m=10	-2.402 (1.239)	.062 (.007)	.638 (.131)	.698
			m=25	-2.353 (1.244)	.062 (.007)	.643 (.135)	.734
			m=40	-2.398 (1.308)	.061 (.007)	.638 (.130)	.700
			HMI 24.00 (12.95)	-2.293 (1.229)	.061 (.007)	.638 (.136)	.736
		k = 5	m=5	-2.428 (1.297)	.062 (.007)	.632 (.121)	.708
			m=10	-2.400 (1.207)	.062 (.007)	.640 (.124)	.724
			m=25	-2.369 (1.243)	.061 (.006)	.625 (.128)	.714
			m=40	-2.358 (1.295)	.062 (.006)	.649 (.126)	.714
			HMI 24.13 (12.53)	-2.461 (1.212)	.061 (.006)	.646 (.128)	.702
		k = 10	m=5	-2.319 (1.270)	.062 (.007)	.635 (.130)	.714
			m=10	-2.411 (1.194)	.062 (.007)	.638 (.131)	.740
			m=25	-2.416 (1.273)	.062 (.007)	.638 (.133)	.674
			m=40	-2.344 (1.223)	.061 (.007)	.637 (.132)	.724
			HMI 24.33 (12.27)	-2.396 (1.230)	.061 (.006)	.636 (.127)	.722
		midas	m=5	.059 (1.148)	.068 (.008)	.766 (.147)	.998
			m=10	.117 (1.164)	.067 (.007)	.758 (.151)	.998
			m=25	.126 (1.129)	.068 (.007)	.771 (.156)	1.000
			m=40	.060 (1.183)	.068 (.007)	.775 (.159)	.998
			HMI 28.12 (13.70)	.114 (1.110)	.067 (.007)	.759 (.159)	.994
				estimate (std of estimate)			

Table 8.14: Simulation 2, MAR (left tailed), $\gamma = 20\%$

				scaled by $1e^2$			
Parameters				Bias	Width	MSE	Coverage
MAR (left tailed)	$\gamma = 40\%$	k=1	m=5	-4.359 (1.738)	.073 (.014)	.527 (.105)	.344
			m=10	-4.342 (1.683)	.072 (.012)	.520 (.112)	.336
			m=25	-4.277 (1.713)	.071 (.009)	.520 (.107)	.330
			m=40	-4.275 (1.686)	.071 (.009)	.527 (.116)	.328
			HMI 97.02 (25.60)	-4.373 (1.719)	.071 (.007)	.534 (.098)	.330
		k = 3	m=5	-4.338 (1.696)	.072 (.014)	.524 (.101)	.342
			m=10	-4.237 (1.685)	.071 (.011)	.527 (.107)	.332
			m=25	-4.512 (1.688)	.071 (.009)	.522 (.099)	.290
			m=40	-4.425 (1.737)	.071 (.008)	.527 (.107)	.304
			HMI 94.36 (26.15)	-4.373 (1.714)	.070 (.008)	.523 (.104)	.302
		k = 5	m=5	-4.430 (1.739)	.073 (.016)	.526 (.108)	.346
			m=10	-4.291 (1.602)	.072 (.011)	.535 (.108)	.348
			m=25	-4.461 (1.787)	.071 (.009)	.516 (.106)	.310
			m=40	-4.220 (1.747)	.071 (.008)	.526 (.100)	.366
			HMI 95.27 (25.80)	-4.277 (1.694)	.071 (.008)	.533 (.107)	.316
		k = 10	m=5	-4.285 (1.847)	.073 (.015)	.522 (.107)	.374
			m=10	-4.467 (1.719)	.073 (.012)	.525 (.107)	.302
			m=25	-4.382 (1.682)	.071 (.009)	.519 (.107)	.296
			m=40	-4.472 (1.742)	.071 (.008)	.534 (.104)	.292
			HMI 94.92 (26.65)	-4.334 (1.643)	.070 (.008)	.521 (.104)	.302
		midas	m=5	.153 (1.650)	.078 (.013)	.774 (.164)	.982
			m=10	.136 (1.722)	.077 (.010)	.775 (.157)	.976
			m=25	.064 (1.524)	.076 (.009)	.771 (.156)	.980
			m=40	.063 (1.583)	.075 (.008)	.758 (.155)	.992
			HMI 73.87 (22.70)	.111 (1.620)	.075 (.008)	.764 (.151)	.986
				estimate (std of estimate)			

Table 8.15: Simulation 2, MAR (left tailed), $\gamma = 40\%$

				scaled by $1e^2$			
Parameters				Bias	Width	MSE	Coverage
MAR (right tailed)	$\gamma = 10\%$	k=1	m=5	-1.869 (1.133)	.056 (.006)	.592 (.123)	.790
			m=10	-1.917 (1.134)	.056 (.006)	.585 (.123)	.772
			m=25	-1.889 (1.119)	.056 (.006)	.587 (.122)	.784
			m=40	-1.934 (1.092)	.056 (.006)	.596 (.125)	.794
			HMI 11.34 (3.53)	-1.831 (1.177)	.056 (.006)	.587 (.126)	.804
		k = 3	m=5	-1.901 (1.201)	.056 (.006)	.595 (.121)	.774
			m=10	-1.927 (1.118)	.056 (.006)	.591 (.125)	.782
			m=25	-1.986 (1.096)	.055 (.006)	.581 (.126)	.744
			m=40	-1.999 (1.109)	.056 (.006)	.587 (.125)	.760
			HMI 3.53 (2.55)	-1.899 (1.115)	.056 (.006)	.587 (.126)	.778
		k = 5	m=5	-1.954 (1.099)	.056 (.006)	.598 (.127)	.774
			m=10	-1.890 (1.143)	.057 (.006)	.603 (.126)	.796
			m=25	-1.907 (1.086)	.056 (.006)	.583 (.122)	.772
			m=40	-1.967 (1.113)	.056 (.006)	.602 (.124)	.766
			HMI 11.13 (2.96)	-1.986 (1.125)	.056 (.006)	.593 (.131)	.760
		k = 10	m=5	-1.896 (1.127)	.056 (.006)	.590 (.119)	.798
			m=10	-1.992 (1.102)	.056 (.006)	.596 (.123)	.746
			m=25	-1.975 (1.165)	.056 (.006)	.592 (.123)	.752
			m=40	-1.885 (1.120)	.056 (.006)	.590 (.125)	.798
			HMI 10.99 (2.80)	-1.951 (1.111)	.056 (.006)	.585 (.129)	.768
		midas	m=5	-.704 (1.337)	.069 (.009)	.744 (.162)	.978
			m=10	-.614 (1.275)	.069 (.008)	.749 (.156)	.982
			m=25	-.745 (1.251)	.069 (.008)	.762 (.161)	.986
			m=40	-.651 (1.322)	.069 (.007)	.763 (.151)	.984
			HMI 95.88 (31.21)	-.698 (1.280)	.069 (.007)	.748 (.157)	.980
				estimate (std of estimate)			

Table 8.16: Simulation 2, MAR (right tailed), $\gamma = 10\%$

				scaled by $1e^2$			
Parameters				Bias	Width	MSE	Coverage
MAR (right tailed)	$\gamma = 20\%$	k=1	m=5	-2.871 (1.422)	.054 (.007)	.486 (.108)	.452
			m=10	-2.922 (1.425)	.054 (.006)	.490 (.102)	.432
			m=25	-3.010 (1.361)	.054 (.006)	.489 (.103)	.396
			m=40	-2.995 (1.309)	.054 (.006)	.491 (.104)	.388
			HMI 30.02 (14.56)	-2.928 (1.441)	.054 (.006)	.485 (.103)	.400
		k = 3	m=5	-2.848 (1.479)	.055 (.007)	.498 (.111)	.470
			m=10	-2.917 (1.353)	.054 (.006)	.491 (.102)	.420
			m=25	-2.944 (1.431)	.054 (.006)	.493 (.107)	.410
			m=40	-2.862 (1.429)	.054 (.006)	.494 (.103)	.438
			HMI 30.59 (15.41)	-3.013 (1.334)	.054 (.006)	.494 (.105)	.398
		k = 5	m=5	-2.899 (1.370)	.055 (.007)	.495 (.110)	.446
			m=10	-2.874 (1.372)	.054 (.006)	.492 (.103)	.480
			m=25	-2.940 (1.402)	.054 (.006)	.489 (.107)	.450
			m=40	-2.902 (1.427)	.054 (.006)	.489 (.105)	.424
			HMI 29.35 (15.26)	-2.927 (1.401)	.054 (.006)	.494 (.108)	.416
		k = 10	m=5	-2.979 (1.389)	.055 (.007)	.494 (.101)	.408
			m=10	-2.795 (1.425)	.054 (.006)	.489 (.101)	.464
			m=25	-2.981 (1.421)	.054 (.006)	.491 (.106)	.404
			m=40	-2.911 (1.337)	.054 (.006)	.497 (.107)	.424
			HMI 30.28 (15.82)	-2.867 (1.343)	.054 (.006)	.485 (.105)	.440
		midas	m=5	-1.186 (1.717)	.079 (.015)	.748 (.164)	.930
			m=10	-1.206 (1.677)	.077 (.011)	.739 (.162)	.924
			m=25	-1.224 (1.725)	.076 (.010)	.741 (.164)	.914
			m=40	-1.218 (1.613)	.076 (.009)	.738 (.156)	.932
			HMI 129.83 (27.4)	-1.054 (1.746)	.076 (.009)	.744 (.156)	.934
				estimate (std of estimate)			

Table 8.17: Simulation 2, MAR (right tailed), $\gamma = 20\%$

				scaled by $1e^2$			
Parameters				Bias	Width	MSE	Coverage
MAR (right tailed)	$\gamma = 40\%$	k=1	m=5	-4.118 (1.920)	.059 (.012)	.349 (.074)	.288
			m=10	-4.249 (1.979)	.059 (.009)	.346 (.073)	.242
			m=25	-4.221 (1.895)	.058 (.007)	.350 (.072)	.250
			m=40	-4.412 (1.972)	.058 (.007)	.346 (.072)	.224
			HMI 98.40 (25.59)	-4.250 (2.049)	.057 (.007)	.345 (.073)	.250
		k = 3	m=5	-4.269 (1.902)	.060 (.011)	.350 (.073)	.244
			m=10	-4.313 (2.013)	.059 (.010)	.351 (.076)	.252
			m=25	-4.374 (1.925)	.058 (.008)	.345 (.078)	.206
			m=40	-4.307 (1.920)	.058 (.007)	.348 (.076)	.218
			HMI 98.48 (23.99)	-4.181 (1.914)	.058 (.007)	.348 (.076)	.224
		k = 5	m=5	-4.287 (2.083)	.060 (.013)	.342 (.077)	.264
			m=10	-4.282 (1.896)	.059 (.009)	.345 (.073)	.226
			m=25	-4.458 (2.016)	.058 (.008)	.348 (.076)	.224
			m=40	-4.418 (1.880)	.058 (.007)	.348 (.071)	.204
			HMI 98.41 (24.65)	-4.190 (1.881)	.058 (.006)	.347 (.072)	.254
		k = 10	m=5	-4.193 (1.969)	.059 (.013)	.341 (.077)	.274
			m=10	-4.176 (1.865)	.058 (.009)	.344 (.072)	.256
			m=25	-4.256 (2.019)	.058 (.008)	.352 (.081)	.250
			m=40	-4.305 (1.892)	.058 (.007)	.347 (.074)	.230
			HMI 97.96 (25.77)	-4.357 (2.013)	.058 (.007)	.348 (.072)	.222
		midas	m=5	-2.124 (2.279)	.100 (.025)	.738 (.154)	.854
			m=10	-2.204 (2.371)	.097 (.019)	.739 (.160)	.840
			m=25	-2.079 (2.410)	.095 (.015)	.732 (.167)	.838
			m=40	-1.843 (2.442)	.096 (.015)	.730 (.164)	.862
			HMI 160.15 (19.4)	-1.837 (2.304)	.096 (.013)	.751 (.166)	.886
				estimate (std of estimate)			

Table 8.18: Simulation 2, MAR (right tailed), $\gamma = 40\%$

				scaled by $1e^2$			
Parameters				Bias	Width	MSE	Coverage
MCAR	$\gamma = 10\%$	k=1	m=5	.043 (2.215)	.119 (.002)	2.701 (.078)	.994
			m=10	.046 (2.065)	.118 (.002)	2.699 (.077)	.994
			m=25	-.085 (2.082)	.119 (.002)	2.705 (.085)	.990
			m=40	.044 (2.148)	.118 (.002)	2.703 (.080)	.996
			HMI 10 (0)	-.103 (2.165)	.119 (.002)	2.710 (.082)	.990
		k = 3	m=5	-.108 (2.149)	.119 (.002)	2.709 (.082)	.998
			m=10	.022 (2.143)	.118 (.002)	2.697 (.080)	.990
			m=25	.007 (1.966)	.118 (.002)	2.704 (.075)	1.000
			m=40	.027 (2.005)	.118 (.002)	2.700 (.078)	.998
			HMI 10 (0)	.018 (2.085)	.119 (.002)	2.709 (.089)	1.000
		k = 5	m=5	-.030 (2.096)	.119 (.002)	2.707 (.077)	.994
			m=10	-.055 (2.122)	.119 (.002)	2.708 (.077)	.996
			m=25	.002 (2.034)	.118 (.002)	2.703 (.081)	.996
			m=40	-.061 (2.150)	.119 (.002)	2.709 (.079)	.996
			HMI 10 (0)	.165 (2.092)	.119 (.002)	2.710 (.080)	.992
		k = 10	m=5	.041 (2.172)	.119 (.002)	2.703 (.077)	.998
			m=10	.093 (2.061)	.119 (.002)	2.704 (.075)	.998
			m=25	-.038 (2.019)	.118 (.002)	2.704 (.077)	.998
			m=40	-.003 (2.217)	.118 (.002)	2.700 (.080)	.992
			HMI 10 (0)	.129 (2.080)	.119 (.002)	2.706 (.086)	1.000
		midas	m=5	.030 (2.274)	.130 (.004)	3.004 (.084)	.994
			m=10	.079 (2.283)	.129 (.003)	3.004 (.082)	.990
			m=25	-.031 (2.307)	.129 (.002)	3.004 (.088)	.998
			m=40	.004 (2.283)	.129 (.002)	3.005 (.090)	.992
			HMI 12.18 (3.86)	-.092 (2.230)	.129 (.003)	3.008 (.092)	.996
				estimate (std of estimate)			

Table 8.19: Simulation 3, MCAR, $\gamma = 10\%$

				scaled by $1e^2$			
Parameters				Bias	Width	MSE	Coverage
MCAR	$\gamma = 20\%$	k=1	m=5	.175 (2.533)	.115 (.003)	2.406 (.070)	.978
			m=10	.062 (2.521)	.115 (.003)	2.404 (.076)	.966
			m=25	.193 (2.425)	.114 (.002)	2.400 (.078)	.984
			m=40	.116 (2.332)	.115 (.002)	2.403 (.078)	.980
			HMI 10.81 (2.30)	-.098 (2.524)	.115 (.002)	2.402 (.074)	.976
		k = 3	m=5	.171 (2.405)	.115 (.003)	2.398 (.075)	.986
			m=10	-.081 (2.390)	.115 (.002)	2.398 (.075)	.980
			m=25	.138 (2.310)	.115 (.002)	2.406 (.077)	.980
			m=40	.032 (2.404)	.115 (.002)	2.406 (.075)	.986
			HMI 10.75 (2.13)	-.090 (2.472)	.115 (.002)	2.406 (.080)	.978
		k = 5	m=5	-.057 (2.518)	.115 (.003)	2.410 (.078)	.986
			m=10	.026 (2.289)	.115 (.002)	2.404 (.079)	.988
			m=25	-.027 (2.314)	.114 (.002)	2.403 (.074)	.984
			m=40	-.153 (2.512)	.115 (.002)	2.410 (.076)	.984
			HMI 10.80 (2.14)	-.044 (2.401)	.115 (.002)	2.401 (.076)	.982
		k = 10	m=5	-.105 (2.289)	.115 (.003)	2.406 (.080)	.992
			m=10	.042 (2.411)	.115 (.002)	2.407 (.073)	.972
			m=25	.086 (2.410)	.115 (.002)	2.402 (.078)	.978
			m=40	.117 (2.422)	.115 (.002)	2.405 (.077)	.992
			HMI 10.74 (2.06)	.032 (2.390)	.115 (.002)	2.403 (.077)	.980
		midas	m=5	-.302 (2.620)	.136 (.009)	3.005 (.093)	.988
			m=10	-.026 (2.575)	.136 (.006)	3.009 (.099)	.994
			m=25	.071 (2.567)	.135 (.004)	3.005 (.096)	.990
			m=40	-.037 (2.569)	.134 (.003)	3.001 (.093)	.994
			HMI 29.83 (12.60)	.054 (2.650)	.135 (.004)	3.004 (.097)	.990
				estimate (std of estimate)			

Table 8.20: Simulation 3, MCAR, $\gamma = 20\%$

				scaled by $1e^2$			
Parameters				Bias	Width	MSE	Coverage
MCAR	$\gamma = 40\%$	k=1	m=5	.106 (3.406)	.119 (.015)	1.802 (.070)	.924
			m=10	.303 (3.154)	.117 (.009)	1.807 (.067)	.932
			m=25	-.049 (3.006)	.116 (.006)	1.808 (.069)	.940
			m=40	.061 (3.140)	.116 (.004)	1.796 (.067)	.926
			HMI 67.99 (20.14)	.035 (3.063)	.115 (.004)	1.807 (.069)	.946
		k = 3	m=5	-.260 (3.048)	.119 (.015)	1.805 (.072)	.946
			m=10	.078 (3.036)	.117 (.009)	1.806 (.071)	.944
			m=25	.241 (3.296)	.116 (.006)	1.806 (.070)	.914
			m=40	.084 (3.230)	.116 (.005)	1.803 (.069)	.924
			HMI 67.12 (20.44)	.086 (3.133)	.115 (.004)	1.801 (.071)	.932
		k = 5	m=5	.004 (3.042)	.119 (.014)	1.805 (.069)	.932
			m=10	.105 (3.215)	.117 (.009)	1.804 (.070)	.924
			m=25	.343 (3.471)	.116 (.006)	1.803 (.069)	.904
			m=40	-.073 (3.017)	.116 (.005)	1.803 (.070)	.950
			HMI 68.12 (20.46)	.005 (3.116)	.116 (.004)	1.804 (.068)	.920
		k = 10	m=5	.009 (3.057)	.120 (.014)	1.805 (.066)	.946
			m=10	-.071 (3.283)	.117 (.009)	1.802 (.069)	.914
			m=25	-.049 (3.132)	.116 (.006)	1.802 (.071)	.928
			m=40	.109 (3.217)	.116 (.005)	1.804 (.071)	.942
			HMI 67.93 (20.88)	.128 (3.006)	.115 (.004)	1.799 (.070)	.946
		midas	m=5	.067 (3.397)	.157 (.020)	3.004 (.119)	.968
			m=10	.280 (3.686)	.155 (.014)	3.003 (.115)	.960
			m=25	.154 (3.663)	.153 (.009)	3.006 (.114)	.964
			m=40	-.218 (3.649)	.153 (.007)	3.005 (.108)	.950
			HMI 76.43 (21.34)	-.016 (3.242)	.152 (.005)	3.011 (.110)	.980
				estimate (std of estimate)			

Table 8.21: Simulation 3, MCAR, $\gamma = 40\%$

				scaled by $1e^2$			
Parameters				Bias	Width	MSE	Coverage
MAR (left tailed)	$\gamma = 10\%$	k=1	m=5	-.337 (2.124)	.120 (.003)	2.658 (.075)	.998
			m=10	-.230 (2.152)	.119 (.002)	2.655 (.082)	.992
			m=25	-.323 (2.219)	.119 (.002)	2.651 (.082)	.990
			m=40	-.355 (2.182)	.119 (.002)	2.655 (.080)	.996
			HMI 10.58 (2.12)	-.282 (2.169)	.119 (.002)	2.652 (.077)	.994
		k = 3	m=5	-.145 (2.284)	.120 (.003)	2.663 (.083)	.992
			m=10	-.396 (2.152)	.119 (.002)	2.650 (.080)	.990
			m=25	-.386 (2.195)	.119 (.002)	2.657 (.076)	.994
			m=40	-.334 (2.283)	.119 (.002)	2.658 (.080)	.988
			HMI 10.52 (1.97)	-.545 (2.155)	.119 (.002)	2.660 (.078)	.988
		k = 5	m=5	-.444 (2.211)	.119 (.003)	2.654 (.075)	.990
			m=10	-.493 (2.132)	.119 (.002)	2.656 (.079)	.990
			m=25	-.318 (2.151)	.119 (.002)	2.654 (.081)	1.000
			m=40	-.283 (2.236)	.119 (.002)	2.662 (.079)	.990
			HMI 10.48 (2.10)	-.194 (2.232)	.119 (.002)	2.658 (.077)	1.000
		k = 10	m=5	-.573 (2.305)	.119 (.003)	2.650 (.076)	.988
			m=10	-.448 (2.179)	.119 (.002)	2.654 (.077)	.996
			m=25	-.379 (2.242)	.119 (.002)	2.655 (.081)	.992
			m=40	-.606 (2.255)	.119 (.002)	2.649 (.080)	.996
			HMI 10.51 (1.67)	-.421 (2.243)	.119 (.002)	2.659 (.078)	.992
		midas	m=5	.296 (2.226)	.132 (.006)	3.007 (.087)	.998
			m=10	.158 (2.440)	.131 (.004)	3.007 (.084)	.994
			m=25	.202 (2.246)	.131 (.003)	3.011 (.087)	.994
			m=40	.150 (2.192)	.131 (.003)	3.015 (.088)	.996
			HMI 24.24 (12.30)	.216 (2.443)	.131 (.003)	3.011 (.095)	.994
				estimate (std of estimate)			

Table 8.22: Simulation 3, MAR (left tailed), $\gamma = 10\%$

				scaled by $1e^2$			
Parameters				Bias	Width	MSE	Coverage
MAR (left tailed)	$\gamma = 20\%$	k=1	m=5	-.797 (2.637)	.121 (.007)	2.366 (.074)	.970
			m=10	-.863 (2.695)	.120 (.005)	2.369 (.077)	.948
			m=25	-.526 (2.543)	.119 (.003)	2.366 (.077)	.984
			m=40	-.673 (2.699)	.120 (.003)	2.373 (.078)	.956
			HMI 27.33 (13.48)	-.631 (2.732)	.119 (.003)	2.365 (.081)	.962
		k = 3	m=5	-.635 (2.685)	.121 (.007)	2.368 (.072)	.972
			m=10	-.603 (2.474)	.120 (.005)	2.372 (.078)	.976
			m=25	-.532 (2.712)	.119 (.003)	2.366 (.076)	.970
			m=40	-.569 (2.627)	.119 (.003)	2.373 (.074)	.972
			HMI 27.35 (14.45)	-.719 (2.558)	.119 (.003)	2.370 (.081)	.976
		k = 5	m=5	-.505 (2.577)	.120 (.007)	2.373 (.075)	.978
			m=10	-.490 (2.584)	.120 (.005)	2.372 (.075)	.966
			m=25	-.678 (2.516)	.119 (.003)	2.365 (.074)	.980
			m=40	-.854 (2.650)	.119 (.003)	2.366 (.079)	.966
			HMI 27.71 (14.19)	-.660 (2.656)	.119 (.003)	2.369 (.077)	.982
		k = 10	m=5	-.839 (2.623)	.121 (.008)	2.372 (.072)	.968
			m=10	-.723 (2.782)	.120 (.005)	2.360 (.082)	.960
			m=25	-.520 (2.563)	.120 (.003)	2.364 (.073)	.972
			m=40	-.644 (2.735)	.119 (.003)	2.364 (.074)	.972
			HMI 26.87 (13.10)	-.614 (2.701)	.119 (.003)	2.369 (.076)	.968
		midas	m=5	.247 (2.883)	.141 (.011)	3.020 (.094)	.970
			m=10	.400 (2.783)	.141 (.008)	3.012 (.095)	.986
			m=25	.449 (2.665)	.140 (.005)	3.013 (.097)	.990
			m=40	.398 (2.844)	.140 (.005)	3.016 (.098)	.986
			HMI 55.43 (22.37)	.233 (2.752)	.139 (.004)	3.014 (.096)	.990
				estimate (std of estimate)			

Table 8.23: Simulation 3, MAR (left tailed), $\gamma = 20\%$

				scaled by $1e^2$			
Parameters				Bias	Width	MSE	Coverage
MAR (left tailed)	$\gamma = 40\%$	k=1	m=5	-.907 (3.528)	.139 (.026)	1.855 (.069)	.932
			m=10	-1.003 (3.530)	.137 (.016)	1.856 (.071)	.930
			m=25	-.899 (3.606)	.136 (.010)	1.860 (.076)	.926
			m=40	-.810 (3.475)	.135 (.008)	1.862 (.069)	.938
			HMI 96.92 (25.45)	-1.131 (3.627)	.134 (.006)	1.855 (.072)	.928
		k = 3	m=5	-.883 (3.693)	.138 (.024)	1.859 (.070)	.914
			m=10	-1.023 (3.481)	.136 (.016)	1.860 (.067)	.944
			m=25	-.938 (3.325)	.135 (.010)	1.858 (.074)	.954
			m=40	-.766 (3.393)	.135 (.009)	1.861 (.070)	.954
			HMI 95.79 (27.00)	-.845 (3.625)	.134 (.006)	1.856 (.072)	.934
		k = 5	m=5	-.853 (3.572)	.138 (.024)	1.857 (.072)	.934
			m=10	-1.276 (3.511)	.135 (.016)	1.855 (.071)	.918
			m=25	-.723 (3.539)	.136 (.010)	1.860 (.072)	.936
			m=40	-.701 (3.374)	.135 (.008)	1.855 (.069)	.944
			HMI 96.35 (25.40)	-.955 (3.297)	.134 (.006)	1.856 (.067)	.950
		k = 10	m=5	-1.003 (3.506)	.139 (.026)	1.860 (.073)	.926
			m=10	-.949 (3.515)	.138 (.016)	1.860 (.071)	.934
			m=25	-.950 (3.563)	.135 (.009)	1.867 (.073)	.936
			m=40	-.861 (3.444)	.135 (.008)	1.855 (.074)	.928
			HMI 95.92 (26.22)	-.637 (3.334)	.134 (.006)	1.855 (.069)	.954
		midas	m=5	.914 (3.854)	.166 (.026)	3.035 (.122)	.944
			m=10	1.019 (4.192)	.165 (.017)	3.038 (.120)	.932
			m=25	.782 (3.995)	.165 (.012)	3.032 (.121)	.964
			m=40	.716 (3.585)	.164 (.009)	3.025 (.112)	.972
			HMI 106.56 (25.8)	.619 (3.860)	.163 (.008)	3.033 (.115)	.960
				estimate (std of estimate)			

Table 8.24: Simulation 3, MAR (left tailed), $\gamma = 40\%$

Parameters				scaled by $1e^2$ Bias	Width	MSE	Coverage
MAR (right tailed)	$\gamma = 10\%$	k=1	m=5	-.588 (2.277)	.117 (.003)	2.552 (.077)	.978
			m=10	-.594 (2.288)	.117 (.002)	2.552 (.082)	.988
			m=25	-.639 (2.250)	.117 (.002)	2.547 (.072)	.984
			m=40	-.516 (2.189)	.117 (.002)	2.553 (.075)	.996
			HMI 10.42 (1.48)	-.625 (2.340)	.117 (.002)	2.548 (.075)	.990
		k = 3	m=5	-.537 (2.250)	.117 (.003)	2.554 (.078)	.984
			m=10	-.440 (2.279)	.117 (.002)	2.546 (.072)	.990
			m=25	-.744 (2.338)	.117 (.002)	2.550 (.076)	.982
			m=40	-.584 (2.121)	.117 (.002)	2.558 (.077)	.992
			HMI 10.42 (1.69)	-.593 (2.301)	.117 (.002)	2.545 (.078)	.984
		k = 5	m=5	-.398 (2.490)	.117 (.003)	2.549 (.074)	.984
			m=10	-.624 (2.174)	.117 (.002)	2.549 (.078)	.990
			m=25	-.662 (2.370)	.117 (.002)	2.548 (.074)	.978
			m=40	-.578 (2.182)	.117 (.002)	2.547 (.073)	.990
			HMI 10.56 (2.27)	-.495 (2.131)	.117 (.002)	2.552 (.077)	.992
		k = 10	m=5	-.527 (2.438)	.117 (.002)	2.547 (.077)	.984
			m=10	-.628 (2.233)	.117 (.002)	2.548 (.074)	.984
			m=25	-.632 (2.304)	.117 (.002)	2.551 (.077)	.986
			m=40	-.490 (2.252)	.117 (.002)	2.552 (.072)	.990
			HMI 10.47 (1.59)	-.528 (2.273)	.117 (.002)	2.546 (.074)	.986
		midas	m=5	-.263 (2.512)	.135 (.008)	3.016 (.093)	.988
			m=10	-.238 (2.417)	.134 (.005)	3.015 (.092)	.992
			m=25	-.090 (2.517)	.134 (.004)	3.012 (.098)	.990
			m=40	-.225 (2.582)	.134 (.003)	3.003 (.096)	.986
			HMI 43.31 (21.51)	-.188 (2.509)	.133 (.003)	3.014 (.093)	.996
				estimate (std of estimate)			

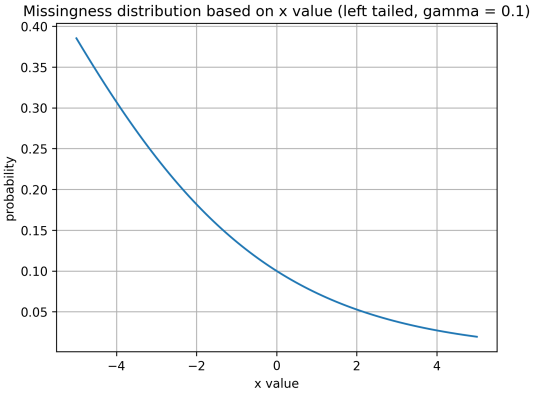
Table 8.25: Simulation 3, MAR (right tailed), $\gamma = 10\%$

				scaled by $1e^2$			
Parameters				Bias	Width	MSE	Coverage
MAR (right tailed)	$\gamma = 20\%$	k=1	m=5	-1.039 (2.683)	.117 (.007)	2.210 (.073)	.946
			m=10	-.794 (2.772)	.116 (.005)	2.209 (.070)	.960
			m=25	-1.059 (2.683)	.115 (.003)	2.205 (.074)	.964
			m=40	-1.003 (2.685)	.115 (.003)	2.209 (.075)	.960
			HMI 27.36 (12.98)	-1.021 (2.684)	.115 (.003)	2.214 (.069)	.948
		k = 3	m=5	-.899 (2.713)	.117 (.007)	2.205 (.073)	.960
			m=10	-.976 (2.686)	.116 (.005)	2.209 (.075)	.964
			m=25	-1.061 (2.816)	.116 (.003)	2.213 (.070)	.938
			m=40	-.839 (2.737)	.115 (.003)	2.210 (.074)	.954
			HMI 27.34 (13.00)	-1.069 (2.718)	.114 (.003)	2.205 (.077)	.946
		k = 5	m=5	-.891 (2.781)	.117 (.007)	2.205 (.071)	.958
			m=10	-1.123 (2.804)	.116 (.004)	2.205 (.070)	.944
			m=25	-1.137 (2.871)	.116 (.003)	2.206 (.073)	.944
			m=40	-1.069 (2.700)	.115 (.003)	2.205 (.075)	.940
			HMI 28.25 (13.73)	-.951 (2.783)	.115 (.003)	2.211 (.070)	.956
		k = 10	m=5	-.951 (2.834)	.117 (.007)	2.212 (.072)	.940
			m=10	-.937 (2.817)	.116 (.004)	2.211 (.073)	.942
			m=25	-.921 (2.733)	.116 (.003)	2.206 (.071)	.958
			m=40	-1.091 (2.741)	.115 (.003)	2.210 (.072)	.956
			HMI 27.47 (13.63)	-1.171 (2.590)	.115 (.003)	2.206 (.072)	.968
		midas	m=5	-.619 (3.210)	.147 (.014)	3.007 (.109)	.962
			m=10	-.440 (2.971)	.146 (.009)	3.019 (.100)	.982
			m=25	-.319 (2.916)	.145 (.007)	3.005 (.108)	.984
			m=40	-.547 (3.064)	.145 (.006)	3.020 (.108)	.978
			HMI 83.13 (27.73)	-.555 (2.982)	.145 (.005)	3.011 (.110)	.988
				estimate (std of estimate)			

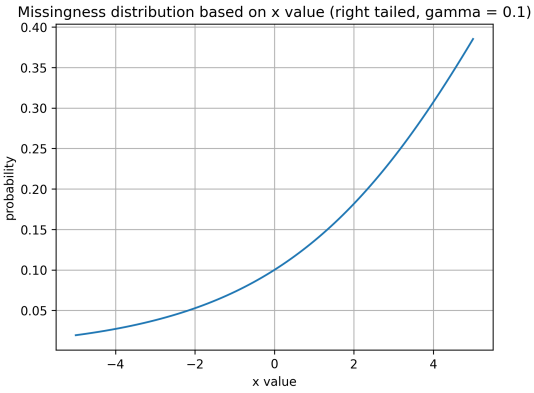
Table 8.26: Simulation 3, MAR (right tailed), $\gamma = 20\%$

				scaled by $1e^2$			
Parameters				Bias	Width	MSE	Coverage
MAR (right tailed)	$\gamma = 40\%$	k=1	m=5	-1.878 (3.736)	.131 (.023)	1.652 (.065)	.878
			m=10	-1.590 (3.713)	.129 (.015)	1.653 (.065)	.876
			m=25	-1.767 (3.792)	.128 (.010)	1.649 (.063)	.868
			m=40	-1.682 (3.643)	.127 (.008)	1.644 (.065)	.890
			HMI 95.67 (24.82)	-1.844 (3.756)	.126 (.005)	1.653 (.062)	.856
		k = 3	m=5	-2.155 (3.961)	.132 (.024)	1.652 (.062)	.828
			m=10	-1.533 (3.713)	.128 (.015)	1.643 (.061)	.890
			m=25	-1.747 (3.954)	.127 (.009)	1.650 (.063)	.850
			m=40	-1.660 (3.521)	.128 (.007)	1.648 (.062)	.888
			HMI 97.19 (25.70)	-1.633 (3.476)	.127 (.005)	1.654 (.062)	.896
		k = 5	m=5	-1.759 (3.926)	.131 (.024)	1.655 (.062)	.848
			m=10	-1.708 (3.871)	.129 (.016)	1.654 (.064)	.874
			m=25	-1.968 (3.686)	.127 (.009)	1.654 (.063)	.866
			m=40	-1.854 (3.690)	.127 (.008)	1.651 (.062)	.878
			HMI 95.68 (25.39)	-1.560 (3.666)	.127 (.006)	1.653 (.062)	.874
		k = 10	m=5	-2.089 (3.710)	.130 (.023)	1.647 (.061)	.850
			m=10	-1.767 (3.641)	.129 (.015)	1.652 (.063)	.880
			m=25	-1.542 (3.763)	.127 (.009)	1.655 (.066)	.880
			m=40	-1.872 (3.727)	.127 (.007)	1.648 (.066)	.868
			HMI 96.60 (25.19)	-1.562 (3.834)	.126 (.006)	1.649 (.063)	.880
		midas	m=5	-.956 (4.624)	.179 (.036)	3.025 (.144)	.926
			m=10	-1.210 (4.821)	.179 (.023)	3.011 (.130)	.906
			m=25	-.894 (4.639)	.176 (.015)	3.022 (.132)	.936
			m=40	-1.127 (4.105)	.176 (.013)	3.025 (.137)	.954
			HMI 131.93 (25.5)	-.832 (4.152)	.176 (.010)	3.021 (.133)	.964
				estimate (std of estimate)			

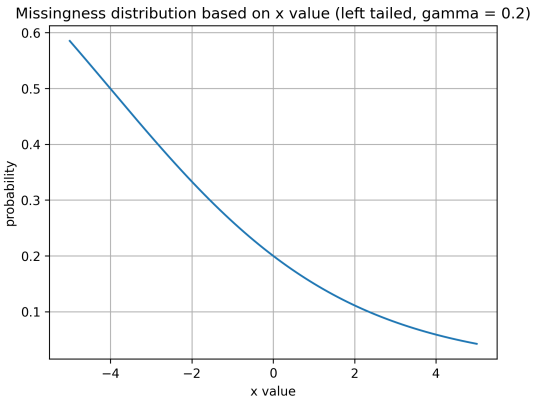
Table 8.27: Simulation 3, MAR (right tailed), $\gamma = 40\%$



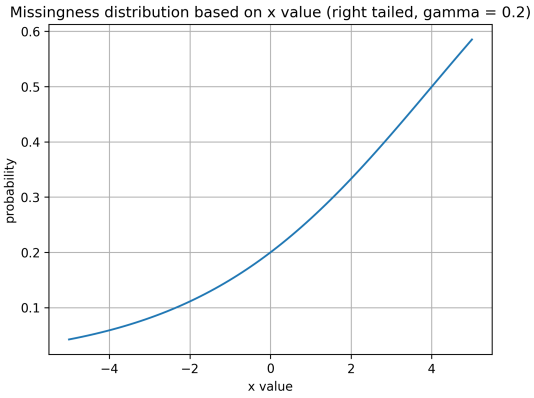
(a) left tailed MAR with $\gamma = 0.1$



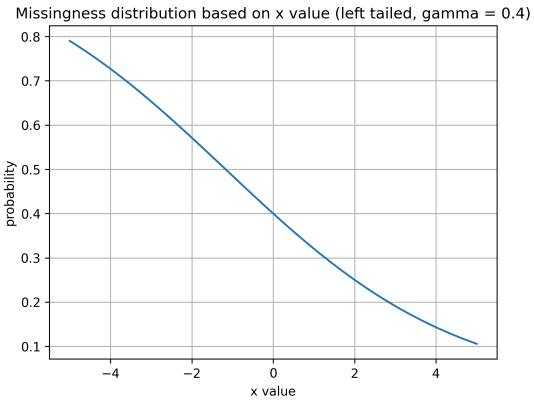
(b) right tailed MAR with $\gamma = 0.1$



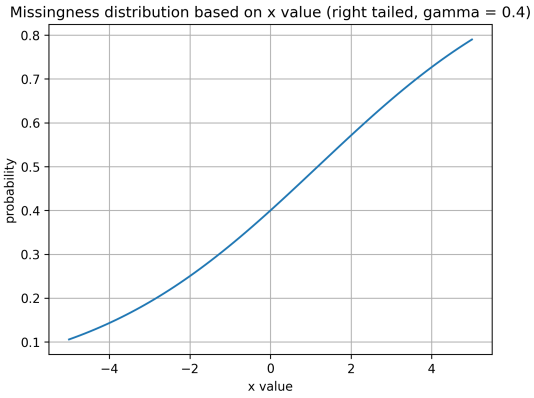
(c) left tailed MAR with $\gamma = 0.2$



(d) right tailed MAR with $\gamma = 0.2$



(e) left tailed MAR with $\gamma = 0.4$



(f) right tailed MAR with $\gamma = 0.4$

Figure 8.1: MAR (left and right tailed), Probability based on x

Bibliography

van Buuren, S. (2018). Flexible imputation of missing data, second edition (2nd ed.)

von Hippel, P. T. (2020). How many imputations do you need? a two-stage calculation using a quadratic rule.

Yu, L.-M., Burton, A., & Rivero-Arias, O. (2007). Evaluation of software for multiple imputation of semi-continuous data [PMID: 17621470]. *Statistical Methods in Medical Research*, *16*(3), 243–258.

Vink, G., Frank, L., Pannekoek, J., & Buuren, S. (2014). Predictive mean matching imputation of semicontinuous variables. *Statistica Neerlandica*, *68*.

Kleinke. (2017). Multiple imputation under violated distributional assumptions: A systematic evaluation of the assumed robustness of predictive mean matching. *Journal of Educational and Behavioral Statistics*, *42*(4), 371–404.

Kleinke. (2018). Multiple imputation by predictive mean matching when sample size is small. *Psychologische Methodenberichte*, *25*(2), 141–149.

Gaffert, P., Meinfelder, F., & Bosch, V. (2018). Towards multiple-imputation-proper predictive mean matching. *JSM Proceedings*, 14.

Little & Rubin. (2002). *Statistical analysis with missing data*. Wiley.

Rubin. (1996). Multiple imputation after 18+ years. *Journal of the American Statistical Association*, *91*(434), 473–489.

Rubin. (1987). *Multiple imputation for nonresponse in surveys*. Wiley.

Graham, J. W., Olchowski, A. E., & Gilreath, T. D. (2007). How many imputations are really needed? some practical clarifications of multiple imputation theory. *Prevention science*, *8*(3), 206–213.

White, I., Royston, P., & Wood, A. (2011). White ir, royston p, wood ammultiple imputation using chained equations: Issues and guidance for practice. stat med 30(4): 377-399. *Statistics in medicine*, *30*, 377–99.

Little. (1988). Missing-data adjustments in large surveys. *Journal of Business Economic Statistics*, *6*(3), 287–296.

Morris, T., White, I., & Royston, P. (2014). Tuning multiple imputation by predictive mean matching and local residual draws. *BMC medical research methodology*, *14*, 75.

Siddique, J., & Harel, O. (2009). Midas: A sas macro for multiple imputation using distance-aided selection of donors. *Journal of Statistical Software*, *29*.

Kim, J. K. (2002). A note on approximate bayesian bootstrap imputation. *Biometrika*, *89*(2), 470–477.

Parzen, M., Lipsitz, S. R., & Fitzmaurice, G. M. (2005). A note on reducing the bias of the approximate bayesian bootstrap imputation variance estimator. *Biometrika*, *92*(4), 971–974.

Demirtas, H., Arguelles, L. M., Chung, H., & Hedeker, D. (2007). On the performance of bias–reduction techniques for variance estimation in approximate bayesian bootstrap imputation. *Computational Statistics Data Analysis*, *51*(8), 4064–4068.

van Buuren, S., & Groothuis-Oudshoorn, K. (2011). mice: Multivariate imputation by chained equations in r. *Journal of Statistical Software*, *45*(3), 1–67.

van Buuren, S., Boshuizen, H. C., & Knook, D. L. (1999). Multiple imputation of missing blood pressure covariates in survival analysis. *Statistics in Medicine*, *18*(6), 681–694.

Alfons, A., Templ, M., & Filzmoser, P. (2010). An object-oriented framework for statistical simulation: The r package simframe. *Journal of Statistical Software*, *37*(3), 1–36.

Schafer, J. L., & Graham, J. W. (2002). Missing data: Our view of the state of the art. *Psychological Methods*, *7*(2), 147–177.

Burgess, N. (2022). Correlated monte carlo simulation using cholesky decomposition [visited: 02.06.2025].

Schölzel, C., & Friederichs, P. (2008). Multivariate non-normally distributed random variables in climate research ndash; introduction to the copula approach. *Nonlinear Processes in Geophysics*, *15*(5), 761–772.

Stavseth, M. R., Clausen, T., & Røislien, J. (2019). How handling missing data may impact conclusions: A comparison of six different imputation methods for categorical questionnaire data [PMID: 30671242]. *SAGE Open Medicine*, *7*, 2050312118822912.