MSc Dissertation

Spatially Correlated Missing Data: A Comparative Study

By

Miguel Torres

2305614

Supervisors:
David Rose
Yoko Chhana

A Dissertation submitted to the Faculty of Science, University of the Witwatersrand, Johannesburg, in fulfilment of the requirements for the degree of Master of Science

March 5, 2025

Dedication

I would like to dedicate this research report to my partner, family and friends for all their continued support.

Acknowledgements

Firstly, I acknowledge my supervisors Dr David Rose and Mrs Yoko Chhana, for their valuable discussions and feedback relating to this work. I certainly know the typesetting and layout would be much worse without their input.

I also acknowledge my partner, Jade, for proofreading and suggestions regarding visual clarity. Finally, I would like to thank my friend, Lazarus, for allowing me to run simulations on his server. The many hours of computation time would have taken far longer without this.

Abstract

Missing data is an important and well-researched topic in the literature, but there is a lack of analysis on the missing data problem in the context of spatially correlated data. Multiple imputation methods in the literature are not tailored for spatial data. This simulation study aimed to determine the potential gain of spatial imputation models over existing multiple imputation approaches. The bias, confidence interval width and coverage were used to compare the performance of imputation models on simulated data. The spatial lag model for imputation was applied to the Boston dataset to assess performance on real data. The spatial models were found to have potential improvement over existing methods, but are potentially sensitive to the missing not at random mechanism.

Declaration

I declare that this Dissertation is my own work. It is being submitted for the Degree of Master of Science to the University of the Witwatersrand, Johannesburg. It has not been submitted before for any degree or examination in any other University.

Signed on March 5, 2025.

The ethics of this study has been approved: waiver number HRECNMW24/06/01 with registration number REC-101114-044.

Contents

List of Figures

List of Tables

Code

List of Abbreviations and Acronyms

CAR	Conditional Autoregressive
EM	Expectation-maximisation
IID	Independent and Identically Distributed
MAR	Missing at Random
MCAR	Missing Completely at Random
MCMC	Markov Chain Monte Carlo
MI	Multiple Imputation
MICE	Multiple Imputation by Chained Equations
ML	Maximum Likelihood
MLE	Maximum Likelihood Estimate
MNAR	Missing Not at Random
OK	Ordinary Kriging
PMM	Predictive Mean Matching
SDEM	Spatial Durbin Error Model
SDM	Spatial Durbin Model
SEM	Spatial Error Model
SLM	Spatial Lag Model

Chapter 1

Introduction

1.1 Background

Analysis of spatial data is becoming increasingly prevalent, as Geographic Information Systems improve and epidemiology as a field expands (Haining, 2003; Puranik, 2021). Spatial statistical models are able to capture the structure in the data by incorporating spatial autocorrelation.

Spatial analysis considers three main types of data: geostatistical, areal and point patterns (Haining, 2003). The focus of this research lies in the field of areal data, where the spatial domain is partitioned into regions, which are often very different in size and shape in real-world applications. Continuous data is the main consideration, although methods could be extended to nominal and ordinal data by adjusting the imputation methods to estimate replacements for the missing values.

Despite the extensive development of spatial models, missing data in the covariates is a problem that is not well explored for areal data. Panzera et al. (2016) and Puranik (2021) are the main sources of literature regarding missing data in this context. A brief discussion of methods to handle missing data in spatial modelling is included in Haining (2003).

Rubin (1976) provides widely known guidelines for dealing with missing data. Although time series is mentioned in the analysis of missing data, Rubin's work does not examine spatially correlated data. An important aspect of missing data is the missingness mechanism. Data can be: missing completely at random (MCAR), missing at random (MAR) or missing not at random (MNAR). The implications for imputation methods depend on the missingness mechanism. Missingness is formally defined in Section 2.2.1.

1.2 Statement of the Problem

Existing literature includes limited analysis of the behaviour of imputation models across missingness mechanism, strength of autocorrelation and percentage of missingness. It is important to determine how these factors affect the performance of the imputation models. A comparison of the different methods under these conditions indicates any weaknesses the model may have (aside from misspecification which is a separate issue).

Incorrectly dealing with missing data can lead to biased parameter estimates and inaccurate predictions (Rubin, 1976). Furthermore, parameter standard errors can be underestimated resulting in narrow prediction intervals and misleading conclusions to hypothesis tests.

Imputation methods may also have computational costs or compatibility issues. Consider the commonly used R (R Development Core Team, 2006) *mice* package (van Buuren and Groothuis-Oudshoorn, 2011) as an example. The package handles missing data using an algorithm termed Multiple Imputation (MI) by Chained Equations (MICE - discussed in Section 2.2.2). The algorithm is a Gibbs sampler under certain conditions, which is known to be computationally expensive. Likelihood-based models can be faster to compute, but may suffer from convergence issues and software is specific to certain models (Allison, 2009). It is important to consider these factors when selecting an appropriate method.

1.3 Aims and Objectives

1.3.1 Aims

The focus of this study is on different methods for handling missing data that are spatially correlated. The autocorrelation in the data is an extra obstacle, but also an opportunity for improved imputation accuracy since the spatial autocorrelation (and correlation) of variables can be utilised. The aim is to examine competing imputation approaches in the context of spatially correlated data.

1.3.2 Objectives

- To simulate data with a variety of missingness mechanisms (MCAR, MAR and MNAR) and autocorrelation structures (Spatial Lag Model (SLM), Spatial Error Model (SEM) and Conditional Autoregressive (CAR) model).
- To compare common methods from literature using simulated data under different missingness mechanisms, percentages of missingness and strengths of autocorrelation in the case of data generated from a specific imputation model.
- To select a set of best-performing method(s) in each data generation scenario based on (average) bias, confidence level width and coverage.
- To apply the selected methods to a real dataset and compare the results.

1.4 Limitations and Assumptions

The generating processes used (refer to Sections 2.6.2 and 2.6.3) assume the data are normally distributed. The simulation approach imposes limitations neither on which missingness mechanisms may be considered nor on the autocorrelation structure in the data. Due to the issue of multiplicity, it may not be possible to explore all scenarios fully. For each model there are 27 combinations: three choices for each of the missing data mechanisms, strength of autocorrelation and missingness percentage. As a result, there are 81 different simulation scenarios to consider (since 3 models are considered).

Chapter 2

Literature Review

2.1 Introduction

Research on missing data was largely pioneered by Rubin (1976). Three main types of missingness, referred to as missingness mechanisms, were identified in the literature: MCAR, MAR and MNAR. Simple single imputation methods that replace a missing value with a plausible value such as the mean of the data (mean imputation) were commonly used for missing data. More robust statistical methods, which can be split into two main categories: MI and model-based imputation, have been developed. MI assigns multiple possible values to a missing value resulting in a 'multiple analysis'. Model-based imputation incorporates missing data into the model likelihood and parameter estimation. These methods are described in Rubin (1988), Schafer (1997) and Little and Rubin (2019).

Although the main focus lies on areal data, for completeness a brief discussion of missing data in geostatistics is included. In this field, different interpolation methods are used, but often the missing data problem is viewed as an inference task. Different kriging methods can be found in Bae et al. (2018) and Chung et al. (2019). Lattice data features many different spatial regression models: linear regression that incorporate spatial autocorrelation into the modelling process. The CAR model, SLM and SEM can be used for imputation and are discussed further in Sections 2.6.2 and 2.6.3

(Haining, 2003; Panzera et al., 2016; Puranik, 2021). In a similar way to the development of standard imputation techniques, single imputation methods are commonly used, but more robust methods have been developed (Haining, 2003).

2.2 Established Missingness Techniques

2.2.1 Missingness Mechanism

Missingness mechanisms were first identified by Rubin (1976), who proposed the theory that missingness can depend on observed or missing data in different ways. Following the notation used in Schafer (1997), let $\mathbf{R}$ denote the $n \times p$ matrix of indicator variables corresponding to whether an observation of the data matrix $\mathbf{Y}$ is missing or not. $\mathbf{Y}$ is split into the observed data $\mathbf{Y}^{\text{obs}}$ and the missing data $\mathbf{Y}^{\text{mis}}$

$$R_{ij} = \begin{cases} 1, & \text{if } y_{ij} \in \mathbf{Y}^{\text{mis}} \\ 0, & \text{if } y_{ij} \in \mathbf{Y}^{\text{obs}}. \end{cases}$$

It is assumed the missingness is related to some unknown parameters, denoted by $\boldsymbol{\xi}$. MCAR assumes that the probability of missingness does not depend on any of the observed or missing data, by satisfying the following property

$$\mathbb{P}(\mathbf{R} \mid \mathbf{Y}^{\text{obs}}, \mathbf{Y}^{\text{mis}}, \boldsymbol{\xi}) = \mathbb{P}(\mathbf{R} \mid \boldsymbol{\xi}).$$

MAR assumes the probability of missingness does not depend on the missing data, but only on the observed data (Schafer, 1997), which can be expressed as the following property

$$\mathbb{P}(\mathbf{R} \mid \mathbf{Y}^{\text{obs}}, \mathbf{Y}^{\text{mis}}, \boldsymbol{\xi}) = \mathbb{P}(\mathbf{R} \mid \mathbf{Y}_{obs}, \boldsymbol{\xi}).$$

Under MNAR the probability $\mathbb{P}(\mathbf{R} \mid \mathbf{Y}^{\text{obs}}, \mathbf{Y}^{\text{mis}}, \boldsymbol{\xi})$ still depends on the missing data $\mathbf{Y}^{\text{mis}}$, and can therefore not be simplified any further.
Further investigation of this concept was developed by Schafer (1997) and other authors in the literature. It was found that models can perform differently under the missingness mechanisms.

- Rubin (1976) proposed that under the MCAR mechanism complete case analysis does not produce biased estimates. Schafer (1997) added that MCAR is the strongest assumption. In further investigation of the literature, it was found that "The MCAR assumption is very strong, and is unlikely to be completely satisfied unless data are missing by design" (Allison, 2009, pg 635).

- Rubin (1976) explained that missingness is related to the observed data in some systematic way under the MAR mechanism. Allison (2009) comments that the MAR mechanism is subtly different from "ignorability" as defined by Rubin (1976), but in the literature if the missingness mechanism is assumed to be ignorable, then data are assumed to be MCAR or MAR. It is further explained by van Buuren (2018) that MAR is a more realistic representation of missing data and is often the default starting point for imputation models.

- Non-ignorable missingness was explained by Rubin (1976) and is linked to the MNAR mechanism in the literature. Both require the missingness mechanism to be modelled in order to produce unbiased estimates. Following this explanation, Allison (2009) stated that it is difficult to model the missingness mechanism since there is a multitude of possible models, so ignorability is often assumed when modelling with missing data in the literature. van Buuren and Groothuis-Oudshoorn (2011) asserted that in practice it cannot be proved that data are MNAR since the missing data are unobservable.

It was noted by van Buuren and Groothuis-Oudshoorn (2011, pg 50) that in the case of MNAR, "the model fitted to the complete cases is incorrect for the missing cases". The author recommended to perform a sensitivity analysis to determine the robustness of the model to the MNAR mechanism.

2.2.2 Multiple Imputation

MI was initially proposed by Rubin (1976) and explored for multivariate normal data by Schafer (1997). This technique was reviewed by Allison (2009) and later implemented in R with the *mice* package by van Buuren and Groothuis-Oudshoorn (2011).

Rubin (1988) explained the implementation of MI using rules to 'pool' parameter estimates. In this framework, m imputations are generated for each missing value resulting in m datasets. Furthermore, Rubin found that values as low as $m = 3$ are acceptable, provided 'proper' imputation methods are used. Schafer (1997, pg 146) supported this result and stated that "Monte Carlo error is a relatively small portion of the overall inferential uncertainty." Monte Carlo error refers to loss of accuracy in the Monte Carlo estimate due to the finite sample size, since the accuracy of Monte Carlo estimates are based on the law of large numbers i.e. an asymptotic result.

'Proper' in the context of imputation models refers to methods that incorporate an appropriate variability between imputations i.e. variability in the sampling assuming reasons for missingness are known, variability in the nonresponse and a third type that is ignorable for MCAR and MAR. As an example, the predictive mean matching (PMM) in *mice* is a proper method, while mean imputation is not (van Buuren and Groothuis-Oudshoorn, 2011). In fact, any purely deterministic imputation such as the median or a prediction from linear regression would be inappropriate. Predictive mean matching selects a number (five by default) of candidate 'donors' with predicted values close to the predicted value for the missing data. A random 'donor' is sampled and its value becomes the imputed value. Further details can be found in Section 3.4 of van Buuren (2018). Any deterministic imputation would display no variation across imputations and is therefore not 'proper'.

The algorithm implemented by van Buuren and Groothuis-Oudshoorn (2011) is called Multiple Imputation by Chained Equations (MICE) and it is a Markov Chain Monte Carlo (MCMC) technique. van Buuren (2018) explained that if the imputation model has valid (non-negative and normalisable) joint distributions, then MICE is a Gibbs sampler. In this study we consider missingness in one variable so the full conditional specification is equivalent to the joint modelling case (discussed in Sections 4.4 and 4.5) and is implemented in a Bayesian way, rather than through bootstrap techniques.

The MICE algorithm stated in Section 4.5.2 of van Buuren (2018) is given as Algorithm 1 below. Define $Y_{-j} = \{Y_1, Y_2, \ldots, Y_{j-1}, Y_{j+1}, \ldots, Y_p\}$ and denote the set of parameters used for imputation of Y_j by ϕ_j

Algorithm 1 MICE algorithm for imputation of multivariate missing data

1: Select an imputation model for each variable Y_j: $\mathbb{P}\left(Y_j^{\text{mis}} \mid Y_j^{\text{obs}}, Y_{-j}, \mathbf{R}\right)$ with $j = 1, \ldots, p$. Note $\mathbf{R}$ indicates if an observation is missing or not.
2: **for** each variable $j = 1, \ldots, p$ **do**
3: Draw $\dot{Y}_j^0$ from Y_j^{obs} (starting imputation).
4: **end for**
5: **for** each iteration $t = 1, \ldots, T$ **do**
6: **for** each variable $j = 1, \ldots, p$ **do**
7: Define $\dot{Y}_{-j}^t := \left(\dot{Y}_1^t, \ldots, \dot{Y}_{j-1}^t, \dot{Y}_{j+1}^t, \ldots, \dot{Y}_p^t\right)$
8: Draw $\dot{\boldsymbol{\phi}}_j^t \sim \mathbb{P}\left(\boldsymbol{\phi}_j^t \mid Y_j^{\text{obs}}, \dot{Y}_{-j}^t, \mathbf{R}\right)$
9: Draw imputations $\dot{Y}_j^t \sim \mathbb{P}\left(Y_j^{\text{mis}} \mid Y_j^{\text{obs}}, \dot{Y}_{-j}^t, \mathbf{R}, \dot{\boldsymbol{\phi}}_j^t\right)$
10: **end for**
11: **end for**

van Buuren (2018) noted some technical aspects of using the *mice* package. It is important to check the convergence of the MICE algorithm (as with any MCMC method). Typically, the same model is fitted to all of the m datasets in what is called a 'repeated analysis' (van Buuren and Groothuis-Oudshoorn, 2011).

The Gibbs samplers for the CAR model, SLM and SEM are discussed in Sections 2.6.2, 2.6.3. Note that line 3 in Algorithm 1 is inappropriate for autocorrelated data since any random observation could be a very poor starting imputation. An adjustment would be to randomly draw from the set of observed neighbours with the lowest order (note that all 1st-order neighbours may be missing so 2nd-order neighbours would need to be considered, similarly if all 2nd-order neighbours are missing, 3rd-order neighbours would be considered etc.).

To generate suitable imputations in line with Rubin (1976), variability across imputations must be appropriate. van Buuren (2018) explained that the variability is composed of two elements: 'noise' and 'parameter uncertainty'. Noise was included simply by adding a stochastic element to the imputed value (such as a random draw from

the posterior predictive distribution). Parameter uncertainty can be included in two ways: a Bayesian approach or using the bootstrap method. The bootstrap method assumes observations are independent and identically distributed (IID), which is not the case for spatially correlated data. There are extensions such as block bootstrapping, but there is limited literature regarding bootstrapping areal data. Consequently, the Bayesian approach was implemented.

Gibbs sampling is a technique to divide multivariate sampling into a set of alternating univariate conditional sampling steps. Assume that the multivariate distribution of $\mathbf{Y}$ is specified by a set of parameters $\boldsymbol{\theta}$. The posterior distribution of $\boldsymbol{\theta}$ is obtained by sampling from the conditional distributions (van Buuren and Groothuis-Oudshoorn, 2011):

$$\begin{aligned}
&\mathbb{P}(Y_1 \mid Y_2, \ldots, Y_p, \theta_1),\\
&\mathbb{P}(Y_2 \mid Y_1, \ldots, Y_p, \theta_2),\\
&\qquad\vdots\\
&\mathbb{P}(Y_p \mid Y_1, \ldots, Y_{p-1}, \theta_p).
\end{aligned}$$

Note that the parameters $\theta_1, \ldots, \theta_p$ are defined by the dependence of each Y_i on relevant parameter(s), and are not necessarily individual parameters of $\boldsymbol{\theta}$.

The iterative sampling was executed on lines 8 and 9 of Algorithm 1, which in this case becomes:

$$\begin{aligned}
\theta_1^{(t)} &\sim \mathbb{P}(\theta_1 \mid Y_1^{\text{obs}}, Y_2^{(t-1)}, \ldots, Y_p^{(t-1)}),\\
Y_1^{(t)} &\sim \mathbb{P}(Y_1 \mid Y_1^{\text{obs}}, Y_2^{(t-1)}, \ldots, Y_p^{(t-1)}, \theta_1^{(t)}),\\
\theta_2^{(t)} &\sim \mathbb{P}(\theta_2 \mid Y_1^{(t)}, Y_2^{\text{obs}}, \ldots, Y_p^{(t-1)}),\\
Y_2^{(t)} &\sim \mathbb{P}(Y_2 \mid Y_1^{(t)}, Y_2^{\text{obs}}, \ldots, Y_p^{(t-1)}, \theta_2^{(t)}),\\
&\qquad\vdots\\
\theta_p^{(t)} &\sim \mathbb{P}(\theta_p \mid Y_1^{(t)}, Y_2^{(t-1)}, \ldots, Y_p^{\text{obs}}),\\
Y_p^{(t)} &\sim \mathbb{P}(Y_p \mid Y_1^{(t)}, Y_2^{(t)}, \ldots, Y_p^{\text{obs}}, \theta_p^{(t)}),
\end{aligned}$$

where $\theta_i^{(t)}$ denotes the estimated value of the parameter θ_i in the t^{th} iteration and $Y_i^{(t)}$ is defined similarly. Y_i^{obs} is the observed data for Y_i.

Let $\beta_1, \ldots, \beta_m$ be the m parameters derived from the repeated analysis of the imputed datasets with estimates $\hat{\beta}_1, \ldots, \hat{\beta}$ and let $U_1, \ldots, U_m$ be their variances. The final estimate of β based on the imputed datasets is given by:

$$\bar{\beta} = \frac{1}{m} \sum_{j=1}^{m} \beta_j$$

The within and between imputation variance are defined as (Rubin, 1988):

$$\begin{aligned} \bar{U} =& \frac{1}{m} \sum_{i=1}^{m} U_i, \\ B =& \frac{1}{m-1} \sum_{j=1}^{m} (\hat{\beta}_j - \bar{\beta})^2. \end{aligned}$$

The ratio $r = \bar{U}/B$ estimates $(1-\gamma)/\gamma$, where "γ is the proportion of information about $\boldsymbol{\theta}$ missing due to non-response" (Rubin, 1988, pg 80). Furthermore, the variance of β is given by:

$$\text{Var}(\bar{\beta}) = \bar{U} + \left(1 + m^{-1}\right) B$$

The $1-\alpha$ confidence interval of $\bar{\beta}$ can be constructed using:

$$\begin{aligned} & \frac{\beta - \bar{\beta}}{\sqrt{\text{Var}(\bar{\beta})}} \sim t_\nu \\ & \nu = (m-1) \left(1 + \left[(1 + m^{-1}) \frac{B}{\bar{U}} \right]^{-1} \right)^2 \\ \implies & \text{CI}(\bar{\beta}) := \bar{\beta} \pm t_{\frac{1-\alpha}{2}, \nu} \sqrt{\text{Var}(\bar{\beta})} \end{aligned}$$

2.2.3 Model-based Imputation

Schafer (1997) explained that model-based methods assume a full model for all of the data (missing and observed), then a likelihood-based approach is used to estimate parameters. Since maximum likelihood estimates (MLEs) are not available in closed form, iterative methods are used. Generally, the expectation-maximisation (EM) algorithm is used to obtain parameter estimates.

The main idea behind the EM algorithm was explained by Schafer (1997). Missing data is imputed based on initial parameter estimates and then the parameters are recalculated. Thereafter, the missing data are estimated again based on the new parameter(s). This process is iterated until convergence. Suppose the complete data is written as (X, Y), where Y is observed and X is missing. The goal of EM is to maximise the log-likelihood (denoted f) with respect to parameter(s) θ: $\log(\theta) = \log\left(\int f(x, y)dx\right)$ which is not available directly. Instead we construct a function $Q(\cdot \mid \cdot)$ in the "E-step" and maximise it in the "M-step" iteratively:

$$Q(\theta \mid \theta_n) = \mathbb{E}[\log(f(X, Y \mid \theta)) \mid X, \theta_n]$$
$$\theta_{n+1} = \arg\max_{\theta} Q(\theta \mid \theta_n)$$

An important property of this procedure was emphasised by the author: the likelihood of the estimated parameters given the observed data are monotonically increasing, guaranteeing convergence (under the conditions that a MLE exists).

However, Schafer (1997) points out that as with other likelihood maximisation problems, convergence to a unique, global maximum is not guaranteed. Problems such as multiple modes or boundary solutions may occur, resulting in convergence to a local optimum. Another disadvantage of the EM likelihood approach compared to MI is that EM only provides point estimates, while MI makes standard errors available.

The review of missing data methods by Allison (2009) found similar performance between MI and ML methods. A significant disadvantage of ML is the requirement for specialised software that only caters to a small class of models. Allison (2009) found almost all implementations of EM assume multivariate normality.

A modern implementation of the ML approach using EM is provided by Feng et al. (2020) in the *imputeR* package. The authors were able to apply EM to offer regularised regression and tree-based models.

2.3 Missingness in Areal/Lattice Data

First areal data and some related properties must be defined. Let $\{Z(s) \mid s \in \mathcal{D}\}$ define a spatial process with domain $\mathcal{D}$. Areal data is defined in terms of discrete areas $s_1, s_2, \ldots, s_n$ within $\mathcal{D}$. These areas partition $\mathcal{D}$ into n non-overlapping, usually unequal regions. The concept of a 'neighbourhood' is of importance in modelling areal data. Essentially the spatial neighbourhood of a region is the set of regions that have a non-negligible influence on the given area.

The neighbourhood may be defined in terms of contiguity, a cut-off distance or a specified number of nearest neighbours. Points are considered adjacent if they share a common border (rook adjacency), a common vertex (bishop adjacency) or a common point or vertex (queen adjacency). The spatial neighbourhood matrix $\mathbf{N}$ is then defined by:

$$N_{ij} = \begin{cases} 1, & \text{if } s_i \text{ and } s_j \text{ are neighbours} \\ 0, & \text{otherwise.} \end{cases}$$

The spatial weight matrix $\mathbf{W}$ is constructed from $\mathbf{N}$ and used for modelling. Sometimes $\mathbf{W} = \mathbf{N}$ is chosen, but often $\mathbf{W}$ is constructed by 'normalising' $\mathbf{N}$ to be row stochastic.

A row stochastic matrix has the property that the elements of each row sum to one, hence $\mathbf{W}$ may be obtained from $\mathbf{N}$ by using

$$W_{ij} = \frac{N_{ij}}{\sum_{j=1}^{n} N_{ij}}.$$

In the context of lattice data, the first-order Markov property may be expressed as (Haining, 2003):

$$\mathbb{P}(Z(s) \mid \mathcal{Z}(s)) = \mathbb{P}(Z(s) \mid Z(s_j), s_j \in \mathcal{N}(s)),$$

where $\mathcal{Z}(s) := \{Z(s_j), s_j \in \mathcal{D}, s_j \neq s\}$ and $\mathcal{N}(s)$ is the set of neighbours of area s. The CAR model discussed in Section 2.6.2 assumes the first order Markov property.

It is uncommon to find MI or ML approaches in the literature on areal data. As discussed by the review of Haining (2003), missing data are most commonly handled by single imputation approaches. Of these, spatially weighted mean imputation with weights based on the area of neighbouring regions or on the length of the common border between neighbouring regions were popular methods. A more modern implementation is provided in the *areal* package in R (Prener et al., 2019).

Other approaches explained by Haining (2003) include spatial hot deck imputation (randomly sampling from the observed neighbours) and iterative imputation through the EM algorithm, which is explained in Section 2.2.3.

Baker et al. (2014) provided a case study of single imputation methods for spatial data relating to predicting risk of type II diabetes. A mean imputation approach was compared to a multivariate normal model and the CAR model.

Models were compared by Baker et al. (2014) using a cross-validation approach. Comparative measures included root mean squared error, bias and CI width. It was found that mean imputation was the best-performing model, which contradicts other literature (Rubin, 1988; van Buuren, 2018).

More recently, Panzera et al. (2016) implemented a MI approach in a Bayesian framework. The methods were applied in the context of normally distributed lattice data using the Besag model (Besag et al., 1991) to generate imputations. The authors examined the performance of the model on real data.

The labour productivity data used was from regions of France and Greece. Three different levels of autocorrelation were chosen to assess the effects on the imputed data: 0.06, 0.2 and 0.8. Panzera et al. (2016) investigated changing the neighbourhood structure of the model, but found no impact on imputations.

The performance of the imputation model was seen to improve at higher levels of autocorrelation. Panzera et al. (2016) also found poorer performance when neighbours were dissimilar as is the case for the Greek data.

Single imputation approaches remain prevalent in the literature, despite the issues pointed out by Rubin (1988) and van Buuren (2018). Puranik (2021) compared the performance of spatial regression models (discussed in Section 2.6.3) to stochastic regression for imputing public health survey data.

The spatial regression models considered by Puranik (2021) were the SLM, SEM, spatial Durbin model, spatial Durbin error model and the spatial lag of X model. The focus of the paper was to estimate the missing data, not to produce accurate parameters for an inferential model, so mean squared error was used to compare the imputations. It must be stated that this approach is incorrect since the goal of imputation is not to replicate the unobserved data (as this is impossible to verify in practice). Mean squared error is a misleading metric to characterise the performance of imputation models (van Buuren, 2018).

Puranik (2021) first compared the models using simulated data before testing on real data. Only the MAR mechanism was considered, while autocorrelation was split into 'high' and 'low' categories. The level of autocorrelation in the dependent variable was set to either 0.2 or 0.8 to represent 'low' and 'high'. For the independent variable 0.1 and 0.6 were the corresponding values.

A simulation of size 1000 was executed using India as the spatial region (to mirror the real data). Puranik (2021) found that the spatial regression models outperformed stochastic regression and similar results were obtained for the real data. Further details on spatial regression models can be found in Section 2.6.3.

Research by Stroud et al. (2017) applied both Bayesian and ML techniques to estimate parameters for a Gaussian process on an incomplete lattice. The approach was to assume an underlying Gaussian process and use ideas from geostatistics to build the model on a lattice. Simulated data was considered in addition to sea surface temperature data. For the Bayesian component Jeffreys' prior was used. The proposed ML method outperformed composite likelihood and spectral analysis. Spectral analysis refers to the approach of applying a Laplace transform to the random variable to work in the so called 'frequency domain'. This is often easier to work with than the covariance function.

2.4 Missingness in Geostatistical Data

In the context of geostatistical data, imputation is mainly carried out by using interpolation type methods such as kriging (Bae et al., 2018; Chung et al., 2019). Kriging is a linear interpolation technique that uses weights based on an estimated covariance function (or variogram). Simple kriging assumes a constant, known mean while ordinary kriging (OK) only assumes the mean is constant. Both these types of kriging can be extended to include covariates and this is referred to as cokriging.

This is sensible, since the imputation of missing observations can be viewed as an inference task to which kriging can be applied. If there are covariates to exploit, then cokriging can be used (Bae et al., 2018).

Chung et al. (2019) considered water level data collected from a national groundwater monitoring well in Korea. The data contained missing intervals of data, which were of interest. The authors compared OK to a technique called conditional simulation for estimating the missing data. Conditional simulation incorporates the random error of kriging so that simulated values have the same covariance function as that of the data.

Chung et al. (2019) noted that OK had a lower mean squared error, but conditional simulation more accurately represented fluctuations in the data. This is due to the 'smoothing' effect of the linear interpolation in kriging.

Bae et al. (2018) examined traffic flow speed data, which was collected by Remote Traffic Microwave Sensors in Knoxville, Tennessee. The approach of kriging had been found to perform better than historical averages or k-nearest neighbours. Four different types of kriging were considered: simple kriging, OK, simple cokriging and ordinary cokriging.

The work of Giorgi and Diggle (2015) discussed the niche problem of inferring the location of a given observation. It is uncommon to observe geostatistical data with a missing location, but the case should still be mentioned. The theoretical approach to solving this problem uses techniques from the area of point processes. It was not the focus of this study, therefore this topic was not explored.

Bae et al. (2018) considered MCAR, MAR and MNAR missingness mechanisms at 10%, 20%, 30% and 40% missingness. The four kriging methods were compared across all three mechanisms and the different percentages of missingness by using mean absolute error and mean absolute percentage error. It was found that cokriging does result in improved performance provided the secondary variable has reasonable correlation ($|\rho| > 0.5$ in the literature) with the variable of interest.

The literature on geostatistical data indicates a lack of MI approaches and few ML approaches for imputation. It would be desirable to explore MI methods in both areal data and geostatistical data, however, the scope would be too large. This study focused on methods for addressing missingness in areal data.

2.5 Simulation of Missing Data

Santos et al. (2019) provided a review of various methods for generating data with missingness, which covered univariate and multivariate generation of MCAR, MAR and MNAR data. The authors emphasised that when simulating missing data it is important to ensure that the assumptions of the missingness mechanism are not violated, otherwise the results of the investigation could be biased or invalid.

The most prevalent issues among methods are restrictions to the proportion of missing data that may be generated or that methods may generate a random amount of missing data i.e. the number of observations removed is not constant.

It was noted by Santos et al. (2019) that simulating missingness with nominal features remains a problem for further investigation in the literature. The generation of MAR data using multiple features was also discussed, since it is common for a single feature to be used in the literature.

The *ampute()* method in R developed by Schouten et al. (2018) is able to implement this for MAR and MNAR using multiple features. The authors proposed a multivariate "amputation" procedure for generating missing data. When compared to the "simple univariate amputation" approach the *ampute()* method was found to produce much more realistic missing data structures. In addition, the method is flexible, allowing the user to generate missing data scenarios as desired.

Two candidate procedures for generating missing data were considered. The first is from Santos et al. (2019) and the second from Schouten et al. (2018). The former research uses the convention "univa" to refer to missingness in one variable and "unifo" for missingness across multiple variables.

The simplest and most robust univariate MCAR implementation involves uniform random sampling in order to remove observations (referred to as 'MCAR2univa' in Santos et al. (2019)). This can also be extended easily to the multivariate case ('MCAR2unifo'). This method can be executed with uniform sampling (without replacement).

A possible MAR implementation is to select the highest values from one variable, and remove the corresponding entries from another variable (referred to as 'MAR2univa' in Santos et al. (2019)). A multivariate extension is possible by randomly selecting pairs of variables ('MAR1unifo'). An alternative is to select pairs of correlated variables.

For MNAR, the approach is to remove the entries from a variable with the highest or lowest values of that variable itself (referred to as 'MNAR1univa' Santos et al. (2019)). The multivariate extension is to take this approach for a number of variables ('MNAR1unifo'). In this case the missingness of a variable is dependent strictly on its own missing values (MNAR by definition may depend on both the observed and missing data). There are other implementations that allow observed variables to have an effect.

Another potential method was developed by Schouten et al. (2018). The process of inducing missingness in complete data was termed 'amputation' in their work. As a result, the R command developed is called *ampute()*. The methods are only applicable to numerical data. Any missingness mechanism may be introduced since the *pattern* (set of variables to be amputated), and *weights* (set of variables which affect amputation) may be specified.

A weighted sum of the variables in *weights* is utilised to extract a probability from a discrete or continuous distribution, chosen by the user. In the case of a continuous distribution, the logistic distribution is used (Schouten et al., 2018). The generated probabilities (that are based on the selected variables) are used to remove observations for a given covariate. It is possible to select no variables, resulting in data that are MCAR.

Additionally, the distribution can be adjusted to mimic different forms of missingness such as missingness concentrated in large values ('MARRIGHT' in Figure 2.5.1). Suppose Y_1 is the variable to be amputated and Y_2 is chosen to influence the amputation. Figure 2.5.1 (van Buuren, 2018, Section 3.2) below shows different possible ways to specify the missingness probabilities.

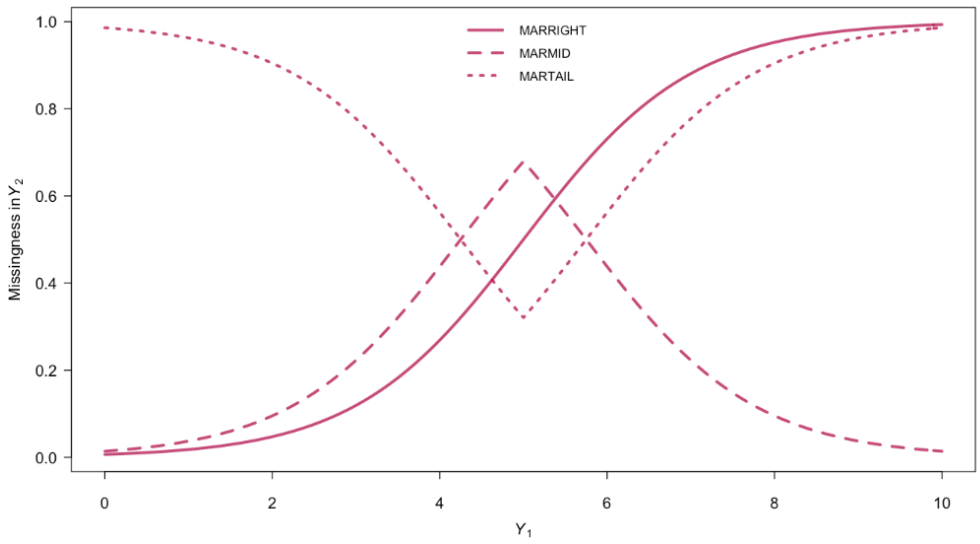


Figure 2.5.1: Generating missing values using a specified logistic distribution

As illustrated in Figure 2.5.1, the proposed method is adaptable and can "accurately mimic realistic missingness patterns" (Schouten et al., 2018, pg 8). Note that the use of the term "patterns" is misleading since this corresponds to the *type* argument in *ampute()* while *pattern* selects variables as discussed previously.

Schouten et al. (2018) is selected for satisfying the characteristics of Santos et al. (2019) and for having methods that are built into the *mice* package. Ultimately, the methods in Santos et al. (2019) could be implemented manually, but *ampute()* is flexible and could be used to extend this study.

Before explaining the methodology, the models used in this study must be defined and their properties explained according to the literature.

2.6 Review of Models in the Study

2.6.1 Introduction

The formal definitions and the mathematical formulations of models used in this study are provided in this section. The properties and requirements for the models are laid out. The distributions for the Gibbs sampling of the parameters are also defined.

2.6.2 CAR Model

The CAR model is specified as (Morris et al., 2019):

$$Z(s) \mid \mathcal{Z}(s) \sim \text{Normal}\left(\sum_{j=1}^{n} W_{ij} Z(s_j), \sigma_s^2\right),$$
$$\sigma_s^2 = \text{Var}(Z(s) \mid \mathcal{Z}(s)),$$

The intrinsic CAR model is often used due to the computational issues of estimating the CAR model (Morris et al., 2019):

$$Z(s) \mid \mathcal{Z}(s) \sim \text{Normal}\left(\frac{1}{|\mathcal{N}(s)|} \sum_{\{k:s_k \in \mathcal{N}(s)\}} Z(s_k), \sigma_s^2\right),$$

where $|\mathcal{N}(s)|$ is the cardinality of the set of neighbours. Note that the above models assume a zero-centred model. In the case of a non-zero mean, the model may be specified as (Panzera et al., 2016):

$$Z(s) \mid \mathcal{Z}(s) \sim \text{Normal}\left(\mathbb{E}[Z(s)] + \sum_{j=1}^{n} \rho W_{ij}(Z(s_j) - \mathbb{E}[Z(s_j)]), \sigma_s^2\right), \tag{2.6.1}$$

where ρ is the scalar spatial autocorrelation parameter and W_{ij} is the $(i,j)^{\text{th}}$ entry of the spatial neighbourhood matrix. According to the work of Besag et al. (1991), these conditional distributions have the joint distribution:

$$\mathbf{Z} \sim \text{Normal}_n(\boldsymbol{\mu}, \boldsymbol{\Sigma}), \tag{2.6.2}$$
$$\boldsymbol{\Sigma} = (\mathbf{I}_n - \rho\mathbf{W})^{-1}\mathbf{B},$$
$$\mathbf{B} = \text{diag}(\sigma_1^2, \sigma_2^2, \ldots, \sigma_n^2).$$

The model requires that $\boldsymbol{\Sigma}$ be symmetric and positive definite, properties which are satisfied when $W_{ij}\sigma_j = W_{ji}\sigma_i$ and $|\rho| < 1$ (Panzera et al., 2016). For $(\mathbf{I}_n - \rho\mathbf{W})$ to be invertible, it is required that $\lambda_{\text{min}} < \rho < \lambda_{\text{max}}$ (Ver Hoef et al., 2018) where λ_{min} and λ_{max} are the minimum and maximum eigenvalues of $\mathbf{W}$ respectively. In applications to lattice data, often a normalised $\mathbf{W}$ matrix based on contiguity is used. This leads to the simplification (Panzera et al., 2016):

$$\begin{aligned}\boldsymbol{\Sigma} &= (\mathbf{I}_n - \rho\mathbf{W})^{-1}\mathbf{B},\\ &= (\mathbf{I}_n - \rho\mathbf{W})^{-1}\sigma^2\mathbf{D}^{-1},\\ &= \sigma^2(\mathbf{D} - \rho\mathbf{N})^{-1},\end{aligned}$$

where $\mathbf{N}$ is the neighbourhood matrix, $\mathbf{W}$ is standardised to be row stochastic, and $\mathbf{D} = \text{diag}(\sum_{j=1}^{n} N_{1j}, \ldots, \sum_{j=1}^{n} N_{nj})$ is the matrix of row totals of $\mathbf{N}$. Note that $\mathbf{W} = \mathbf{D}^{-1}\mathbf{N}$ follows from standardising $\mathbf{W}$. When using the normalised $\mathbf{W}$, the restriction on ρ is: $\lambda_{\text{min}} < \rho < 1$, where $\lambda_{\text{min}} < -1$ usually. For practical modelling purposes, this is often restricted to $|\rho| < 1$ (Ver Hoef et al., 2018).

The posterior distributions from which to sample the parameters are provided in De Oliveira (2012), using uninformative prior distributions. Consider a CAR model on $\mathbf{y} = (y_1, y_2, \ldots y_n)$ with mean given by a linear combination of covariates $\mathbf{X}\boldsymbol{\beta}$. The parameters are sampled from Equations 2.6.3, 2.6.4 and 2.6.5.

$$\begin{aligned}\mathbf{y} \mid \mathbf{X}, \rho, \boldsymbol{\beta}, \sigma &\sim \text{Normal}_n(\mathbf{X}\boldsymbol{\beta}, \sigma^2(I_n - \rho\mathbf{W})^{-1})\\ \text{Let } \tilde{\mathbf{B}} &= (\mathbf{X}^T(\mathbf{I}_n - \rho\mathbf{W})\mathbf{X})^{-1}\mathbf{X}^T(\mathbf{I}_n - \rho\mathbf{W})\mathbf{y},\\ \boldsymbol{\beta} \mid \rho, \sigma^2, \mathbf{X}, \mathbf{y} &\sim \text{Normal}_p(\tilde{\mathbf{B}}, \sigma^2(\mathbf{X}^T(\mathbf{I}_n - \rho\mathbf{W})\mathbf{X})^{-1}), \qquad (2.6.3)\\ \text{Let } S_\rho^2 &= (\mathbf{y} - \mathbf{X}\tilde{\mathbf{B}})^T(\mathbf{I}_n - \rho\mathbf{W})(\mathbf{y} - \mathbf{X}\tilde{\mathbf{B}}),\\ \sigma^2 \mid \rho, \boldsymbol{\beta}, \mathbf{X}, \mathbf{y} &\sim \text{InvGam}\left(\frac{n-p}{2}, \frac{1}{2}S_\rho^2\right), \qquad (2.6.4)\end{aligned}$$

$$\pi(\rho \mid \boldsymbol{\beta}, \sigma^2, \mathbf{X}, \mathbf{y}) \propto \left(\frac{\left|\mathbf{X}^T(\mathbf{I}_n - \rho\mathbf{W})\mathbf{X}\right|}{|\mathbf{I}_n - \rho\mathbf{W}|}\right)^{-\frac{1}{2}} \left(S_\rho^2\right)^{\frac{n-p}{2}} \pi(\rho). \qquad (2.6.5)$$

Note that in Equation 2.6.4, 'InvGam' refers to the inverse gamma distribution (defined in Appendix A.1). De Oliveira (2012) shows the posterior can be normalised by

showing the integral of the likelihood is finite. When using a diffuse prior the posterior is proportional to the likelihood and the result was also shown to hold when using Jeffreys' Independence Prior. For simplicity and since there is little difference in using Jeffreys' Independence Prior (De Oliveira, 2012), the prior distribution of ρ is taken to be the uniform distribution over $(0, 1)$. Hence $\pi(\rho) = 1$. The R code for the CAR model is in Code Block D.2.1 in Appendix D.2.

2.6.3 Spatial Lag and Spatial Error Models

Spatial regression models extend linear regression by incorporating spatial dependence through the spatial weight matrix $\mathbf{W}$. Anselin (2009) distinguishes two main categories: spatial lag dependence (SLM and SDM) and spatial error dependence (SEM and SDEM). The SDM and SDEM are outside the scope of this study so focus is instead placed on the SLM and SEM. The SLM model may be defined by:

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \rho\mathbf{W}\mathbf{y} + \boldsymbol{\epsilon},$$

where $\mathbf{y}$ is a $n \times 1$ vector of the dependent variable, $\boldsymbol{\beta}$ is a $p \times 1$ vector of regression coefficients, $\mathbf{X}$ is an $n \times p$ matrix of independent variables, ρ is a scalar autocorrelation parameter, $\mathbf{W}$ is the $n \times n$ spatial weight matrix (that introduces spatial lag) and $\boldsymbol{\epsilon}$ is a random, IID $n \times 1$ vector of errors: $\boldsymbol{\epsilon} \sim \mathbf{N}_n(\mathbf{0}, \sigma^2\mathbf{I}_n)$. For this model, the spatial correlation is of interest.

Note that for row standardised $\mathbf{W}$, a weighted average of the neighbours of $\mathbf{y}$ is included in the regression. Writing the model in "reduced form" (Anselin, 2009, pg 7):

$$\mathbf{y} = (\mathbf{I}_n - \rho\mathbf{W})^{-1}(\mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}), \tag{2.6.6}$$

$$\mathbb{E}(\mathbf{y} \mid \mathbf{X}) = \mathbf{X}\boldsymbol{\beta} + \sum_{k=1}^{\infty} \rho^k \mathbf{W}^k \mathbf{X}\boldsymbol{\beta}. \tag{2.6.7}$$

The inverse $(\mathbf{I}_n - \rho\mathbf{W})^{-1}$ exists provided $\mathbf{W}$ is row stochastic and $|\rho| < 1$ (Anselin, 2009). Equation 2.6.7 follows from the expansion $(\mathbf{I}_n - \rho\mathbf{W})^{-1} = \mathbf{I}_n + \sum_{k=1}^{\infty} \rho^k \mathbf{W}^k$.

The distributions that were used for the Gibbs sampler as discussed in Section 2.2.2 are provided in Equations 2.6.8, 2.6.9 and 2.6.10. These equations are taken from LeSage (2000) by considering a 'diffuse prior' (uninformative prior) for the parameters $(\rho, \sigma, \boldsymbol{\beta})$. The posterior distributions are be denoted by $\pi(\cdot \mid \cdot)$.

$$\mathbf{y} \mid \mathbf{X}, \boldsymbol{\beta}, \rho, \sigma \sim \text{Normal}_n((\mathbf{I}_n - \rho\mathbf{W})^{-1}\mathbf{X}\boldsymbol{\beta}, \sigma^2(\mathbf{I}_n - \rho\mathbf{W})^{-1}((\mathbf{I}_n - \rho\mathbf{W})^{-1})^T)$$

$$\text{Let } \boldsymbol{\epsilon} = (\mathbf{I}_n - \rho\mathbf{W})\mathbf{y} - \mathbf{X}\boldsymbol{\beta}.$$

$$\pi(\sigma \mid \rho, \boldsymbol{\beta}, \mathbf{X}, \mathbf{y}) \propto \sigma^{-(n+1)} \exp\left(\frac{1}{2\sigma^2}\boldsymbol{\epsilon}^T\boldsymbol{\epsilon}\right). \tag{2.6.8}$$

$$\text{Let } \mathbf{B} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T(\mathbf{I}_n - \rho\mathbf{W})\mathbf{y}.$$

$$\boldsymbol{\beta} \mid \rho, \sigma, \mathbf{X}, \mathbf{y} \sim \text{Normal}_p(\mathbf{B}, \sigma^2(\mathbf{X}^T\mathbf{X})^{-1}), \tag{2.6.9}$$

$$\pi(\rho \mid \boldsymbol{\beta}, \sigma, \mathbf{X}, \mathbf{y}) \propto |\mathbf{I}_n - \rho\mathbf{W}| \exp\left(\frac{1}{2\sigma^2}\boldsymbol{\epsilon}^T\boldsymbol{\epsilon}\right). \tag{2.6.10}$$

The joint posterior distribution is not shown to be normalisable (have a finite integral with respect to the parameters) explicitly in the literature. As such the convergence of the Gibbs sampler must be monitored for divergent behaviour. The code for implementing the SLM model is in Code Block D.3.1 in Appendix D.3.

There are two ways to define the SEM model. One is based on specifying a function for the covariance between observations (direct representation), and the alternative is to specify a spatial process for the error term. Often for the former, the exponential distance decay function, is used (Anselin, 2009):

$$\mathbb{E}(\mathbf{u}'\mathbf{u}) = \sigma^2(I_n - \gamma\boldsymbol{\Phi}),$$
$$\Phi_{ij} = e^{\phi d_{ij}},$$

for a distance d_{ij} between locations s_i and s_j. The second formulation is the one that was considered in this study and can also be found in Puranik (2021).

If the error term (now defined by $\mathbf{u}$ to differentiate from the uncorrelated error $\boldsymbol{\epsilon}$) is considered to originate from the spatial autoregressive process:

$$\begin{aligned} &\mathbf{u} = \lambda\mathbf{W}\mathbf{u} + \boldsymbol{\epsilon}, \text{ then,} \\ \implies &\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \lambda\mathbf{W}\mathbf{u} + \boldsymbol{\epsilon}, \\ \implies &\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + (\mathbf{I}_n - \lambda\mathbf{W})^{-1}\boldsymbol{\epsilon}, \end{aligned} \tag{2.6.11}$$

where $\mathbf{u}$ is a $n \times 1$ vector of spatially correlated 'disturbances' and $\mathbf{y}$, $\mathbf{X}$, $\boldsymbol{\beta}$ and $\boldsymbol{\epsilon}$ are defined in the same way as the SLM (Puranik, 2021). In this model, spatial autocorrelation is treated as a 'nuisance' parameter.

The posterior distributions from which to sample are provided in Equations 2.6.12, 2.6.13 and 2.6.14. The equations are adjusted from LeSage (2000) where an uninformative prior was used.

$$\begin{aligned}
\mathbf{y} \mid \mathbf{X}, \lambda, \sigma &\sim \text{Normal}_n(\mathbf{X}\boldsymbol{\beta}, \sigma^2(I_n - \lambda\mathbf{W})^{-1}((I_n - \lambda\mathbf{W})^{-1})^T) \\
\text{Let } \boldsymbol{\epsilon} &= (\mathbf{I}_n - \lambda\mathbf{W})(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}). \\
\pi(\sigma \mid \lambda, \boldsymbol{\beta}, \mathbf{X}, \mathbf{y}) &\propto \sigma^{-(n+1)} \exp\left(\frac{1}{2\sigma^2}\boldsymbol{\epsilon}^T\boldsymbol{\epsilon}\right). \qquad (2.6.12) \\
\text{Let } \bar{\mathbf{B}} &= (\mathbf{X}^T(\mathbf{I}_n - \lambda\mathbf{W})^T(\mathbf{I}_n - \lambda\mathbf{W})\mathbf{X})^{-1}\mathbf{X}^T(\mathbf{I}_n - \lambda\mathbf{W})^T(\mathbf{I}_n - \lambda\mathbf{W})\mathbf{y}. \\
\boldsymbol{\beta} \mid \lambda, \sigma, \mathbf{X}, \mathbf{y} &\sim \text{Normal}_p(\bar{\mathbf{B}}, \sigma^2(\mathbf{X}^T(\mathbf{I}_n - \lambda\mathbf{W})^T(\mathbf{I}_n - \lambda\mathbf{W})\mathbf{X})^{-1}), \qquad (2.6.13) \\
\pi(\lambda \mid \boldsymbol{\beta}, \sigma, \mathbf{X}, \mathbf{y}) &\propto |\mathbf{I}_n - \lambda\mathbf{W}| \exp\left(\frac{1}{2\sigma^2}\boldsymbol{\epsilon}^T\boldsymbol{\epsilon}\right). \qquad (2.6.14)
\end{aligned}$$

Once again it is not explicitly shown that this model has a normalisable joint posterior. Note that there is a duality between the CAR model and the SEM implying the posterior should have a finite integral (Ver Hoef et al., 2018). The output of the Gibbs sampler should still be checked for divergent behaviour. The SEM is implemented in R code in Code Block D.4.1 (Appendix D.4).

Chapter 3

Methodology

To assess the performance of the imputation models, a simple 'top-level' regression model is considered with the covariates containing missing data. This is discussed in Section 3.4. The metrics used to compare imputation models are calculated using the pooled estimate, say $\acute{\beta}^{(m)}$, compared to the value used to generate the data $\acute{\beta}$.

The use of models in this simulation study can be broken down into eight steps:

1. Generate data for the chosen imputation model using a fixed set of parameter values and a chosen autocorrelation.
2. Use the generated data to simulate data from the regression model (top-level)
3. Introduce missingness in the chosen covariate through amputation (*ampute()* in *mice*).
4. Produce m imputed datasets based on the MICE algorithm (Figure 1).
5. Calculate parameter estimates for the linear regression model in each dataset and pool results.
6. Calculate bias, CI width and coverage (1 or 0 in this case) for these estimates.
7. Repeat steps 1-5 one hundred times to calculate average bias, average CI width and coverage (i.e. using Monte Carlo).

8. Repeat steps 1-6 for all 27 combinations of missingness mechanism, missingness percentage and autocorrelation.

Sections 3.6.1 and 3.6.2 describe how the first and third steps are implemented for the CAR model, SLM and SEM.

Note that there are two key parts of the process: the 'top-level' model of $\mathbf{z} = \acute{\beta}_0 + \acute{\beta}_1\mathbf{y} + \acute{\beta}_2\mathbf{x}$ (since $\mathbf{x}$ is univariate here) and the imputation model of $\mathbf{y}$ based on $\mathbf{x}$ which is either the CAR model, SLM or SEM. The imputations are drawn from a Gibbs sampler as discussed in Sections 2.6.2 and 2.6.3. The top level model is estimated using *lm*() in R (in the repeated analysis) and confidence intervals are calculated as defined at the end of Section 2.2.1 which is handled internally by the *mice*() method in R.

The objective of this process is to assess the performance of imputation methods in *mice* compared to data generated by spatial (imputation) models. These spatial models are fitted against data generated by themselves, which, while idealistic, determines if these models can be considered useful at all.

After analysis on simulated data, the models were tested on a real dataset to present a more realistic scenario. The chosen dataset was the widely used corrected Boston housing data.

3.1 Boston Data

The Boston dataset is contained in the *spData* package and is a corrected version of the original dataset (Harrison Jr and Rubinfeld, 1978). The spatial area contains 506 regions with 20 recorded variables for each region (described in Table 3.1.1). Figure 3.1.1 illustrates the spatial regions of the Boston dataset.

Figure 3.1.1: The spatial regions in the Boston dataset

Table 3.1.1: Description of variables in the Boston dataset

Variable	Description
TOWN	Town names
TOWNNO	A numeric representation of town names
TRACT	Tract ID number
LON	Tract point latitudes in decimal degrees
LAT	Tract point longitudes in decimal degrees
MEDV	Median values of owned homes in 1000's of USD
CMEDV	Corrected median house values in 1000's of USD
CRIM	Per capita crime
ZN	Proportions of residential land zoned for lots over 25000 square feet
INDUS	Proportions of non-retail business acres per town
CHAS	Indicates if the tract borders the Charles River
NOX	Nitric oxides concentration (parts per 10 million)
RM	Number of rooms per house
AGE	Proportion of properties built before 1940
DIS	Weighted distances to five employment centres
RAD	index of accessibility to radial highways per town
TAX	Full-value property-tax rate per 10,000 USD per town
PTRATIO	Pupil-teacher ratios per town
B	A function of the proportion of blacks, B_k: $B = 1000 * (B_k - 0.63)^2$
LSTAT	Percentage values of lower status population

Following the analysis of the corrected Boston dataset by Moraga (2023), a model for the (corrected) median house value of an area 'CMEDV' is modelled by the average number of rooms in an area 'RM' and the number of crimes per capita 'CRIM' under a spatial regression model. For these the SLM and SEM were considered and the best

one chosen in Section 4.6.

This was first fitted to the complete data to use as a benchmark and afterwards the data were amputated. Amputation was carried out in the same way as the simulated using the *ampute*() method set with MAR mechanism, 15% missingness and 'right' missingness. This is discussed in Section 3.2.

The imputations were calculated using the SEM and SLM, however both provided similar results so only the SLM imputations are reported. The imputations were used to compute pooled estimates (as discussed in Section 2.2.1) which were compared against parameters calculated from the complete data.

In the same way as the simulated data, imputations were also computed for the normal model and PMM and used to construct pooled estimates. The analysis can be found in Section 4.6.

3.2 Simulation Procedure - Amputation

Simulation of areal datasets was executed in the *spdep* package in R. Across these simulations missingness, of 5%, 15%, and 25% were compared, in addition to auto-correlations of 0.2, 0.5, and 0.8. These are values commonly used in literature for comparative purposes (Baker et al., 2014; Puranik, 2021). The spatial region used was that of Boston, so the real and simulated data had the same spatial weights matrix $\mathbf{W}$.

For MCAR data, the uniform sampling method discussed in Schouten et al. (2018) was used since it is simple, robust and should provide equivalent results to the *ampute()* method with zero weights.

In the case of MAR and MNAR data, the *ampute()* method was applied due to the flexibility and realistic simulation of missing data provided by this method.

The missingness percentage was defined in terms of a single feature as described in Santos et al. (2019) by considering an entire observation to be missing if one or more variables contain an unobserved value. The missingness percentage was taken to be equal to the proportion of rows with a missing value out of the total n. This measure was chosen since missingness was restricted to $\mathbf{y}$ (the spatial covariate). Since MCAR is a very strong assumption, that is often not completely satisfied, MCAR data was simulated for fewer scenarios compared to MAR and MNAR. It was believed to produce similar results to MAR, since models that can handle the MAR mechanism should have no issue with MCAR data (confirmed in Section 4).

In the case of MAR, the variables selected by the *pattern* argument are mutually exclusive to those in *weights* (refer to Section 2.5). This ensured that missingness depends on the observed data, thereby guaranteeing the MAR mechanism.

To simulate MNAR the variables selected by the *pattern* argument overlap with those in the *weights* to determine missingness. Specifying the parameters of the *ampute()* method in this way ensures that missingness depends on the missing values, which is characteristic of the MNAR mechanism.

There were no set guidelines in the literature with regards to the number of simulations that should be executed, nor the size of each dataset. Puranik (2021) makes use of 1000 simulations per scenario. The data contained 284 observations, corresponding to the 284 regions within India.

Due to the computational complexity of spatial data and the number of cases that must be simulated (27 scenarios for each imputation model), a size of 100 simulations (of

the imputation and analysis process) per model case was considered. Simulations were carried out using the spatial region in the Boston data (explained in Section 3.1).

3.3 Simulation Procedure - Imputation Models

A number of spatial models from the literature were considered for use as imputation models. The implementation and generation of data from these models are discussed in Section 3.6. According to Rüttenauer (2022), the SLM is the most common spatial regression model and other commonly used models are the SEM, SDM, the SDEM and the "spatial lag of X model."

The SDM and SDEM are extensions of the SLM and SEM respectively, so for this study methods were developed for the SLM and SEM. It would be possible to extend the methodology by using work on Bayesian versions of the SDM and SDEM, but this is beyond the scope of this study (refer to Section 5.5).

In the literature, there is a relationship between the SEM and the CAR model, as a result of the similar (but not identical) specifications (Ver Hoef et al., 2018). The CAR model was included as another imputation model for handling spatially correlated errors.

The main focus was on MI methods since likelihood-based methods become complex in the context of spatial data, which are not IID (Stroud et al., 2017). Furthermore ML methods require specialised software. Consequently, the imputation models are implemented using the MICE algorithm and the *mice* package in R.

The spatial models are implemented using a Bayesian approach (full-conditional specification) as discussed in Sections 2.6.2 and 2.6.3 and plugged into *mice* as a custom imputation method.

Imputations were also generated using PMM and the 'normal' (Gaussian) model of the *mice* package. The purpose of comparing these models is to gauge the robustness of *mice* methods and determine if spatial imputation models are worthwhile (considering the best case - data generated from the model itself). According to van Buuren (2018), appropriate comparative measures are bias, confidence interval (CI) width and coverage, with lower bias and confidence interval width and higher coverage indicating better performance. These metrics were used to compare the imputation models (only within their respective data generation scenario).

3.4 Simulation Procedure - Linear Regression

Recall a standard linear model was used to evaluate the performance of the imputation methods: $\mathbf{z} = \acute{\beta}_1\mathbf{y} + \acute{\beta}_2\mathbf{x} + \acute{\beta}_0$. For exposition, $\mathbf{z}$ was assumed to be complete and $\mathbf{x}$ univariate, but the extension to missing $\mathbf{z}$ and multivariate $\mathbf{x}$ can be made. The parameters were arbitrarily chosen to be $\acute{\beta}_1 = 1$, $\acute{\beta}_2 = -1$ and $\acute{\beta}_0 = 2$ and for the imputation models, $\beta_0 = 3$ and $\beta_1 = 2$. More parameter values could be tested, but that is outside the scope of this study.

Following the example of van Buuren (2018), bias and CI width were calculated as a mean of the corresponding values for the 100 simulations. Coverage was calculated as the proportion of cases in which the confidence interval contained the true parameter value.

The bias, CI width and coverage are estimated using Monte Carlo:

$$\text{Average Bias} = \frac{1}{N}\sum_{i=1}^{N}\bar{\acute{\beta}}_i - \acute{\beta}$$

$$\text{CI width} = \frac{1}{N}\sum_{i=1}^{N} 2t_{0.975,\nu}\sqrt{\text{Var}(\bar{\acute{\beta}}_i)}$$

$$\text{Coverage} = \frac{1}{N}\sum_{i=1}^{N}\mathbb{I}(\acute{\beta} \in \text{CI}(\bar{\acute{\beta}}_i))$$

where $\mathbb{I}(\cdot)$ is an indicator function. The $\bar{\hat{\beta}}$ and its variance is provided by the *mice* package as discussed at the end of Section 2.2.1.

Section 3.6 details how to sample from the imputation models and from the predictive distributions corresponding to these models (not discussed in the literature).

3.5 Simulation - Summary of Scenarios

The 27 possible simulation scenarios are given in Table 3.5.1:

Table 3.5.1: Simulation Scenarios for each model (CAR, SLM and SEM)

Autocorrelation	**Missingness (%)**	**Missingness Mechanism**
0.2	5	MCAR
0.2	10	MCAR
0.2	15	MCAR
0.5	5	MCAR
0.5	10	MCAR
0.5	15	MCAR
0.8	5	MCAR
0.8	10	MCAR
0.8	15	MCAR
0.2	5	MAR
0.2	10	MAR
0.2	15	MAR
0.5	5	MAR
0.5	10	MAR
0.5	15	MAR
0.8	5	MAR
0.8	10	MAR
0.8	15	MAR
0.2	5	MNAR
0.2	10	MNAR
0.2	15	MNAR
0.5	5	MNAR
0.5	10	MNAR
0.5	15	MNAR
0.8	5	MNAR
0.8	10	MNAR
0.8	15	MNAR

This is done for each of the three imputation models. In each of those cases imputations are also computed using the normal model and PMM within *mice*. Comparisons are made within each data generating model (same as the imputation models), since comparisons across these scenarios would not make sense.

3.6 Implementation of Models

3.6.1 CAR Model

Data was simulated for this model by randomly sampling from the distribution in Equation 2.6.2. First note that the mean $\boldsymbol{\mu} = \mathbf{X}\boldsymbol{\beta}$ is modelled by the covariates (De Oliveira, 2012). The last step before simulation was to select values for $\boldsymbol{\beta}$ and ρ and to generate $\mathbf{X}$. Sampling from a multivariate normal distribution can be executed in R using the *MASS* package (Venables and Ripley, 2002). It is possible to alter the autocorrelation structure by varying ρ. Note that $|\rho| < 1$ is required for a valid CAR model, but following the literature ρ is restricted to be positive as well (Ver Hoef et al., 2018).

Model imputations can be generated using Equation 2.6.1 for each missing value (note $\mathbb{E}[Z(s_i)] = \mathbf{X}_i\boldsymbol{\beta}$). First, the parameters are sampled using Equations 2.6.3, 2.6.4 and 2.6.5, then these sampled values are used in Equation 2.6.1. Simulation of the CAR model using MICE is implemented in Code Block D.2.2 (Appendix D.2).

3.6.2 Spatial Regression Models

Equation 2.6.6 was used to simulate data by selecting ρ and $\boldsymbol{\beta}$, generating $\mathbf{X}$ and $\boldsymbol{\epsilon}$. For the SLM, the posterior distributions of the parameters are given in Equations 2.6.8, 2.6.9 and 2.6.10. To draw imputations from the distribution of $\mathbf{y}^{\text{mis}} \mid \mathbf{y}^{\text{obs}}, \rho, \boldsymbol{\beta}, \sigma^2$, note that for complete data:

$$\begin{aligned} &\mathbf{y} \mid \rho, \sigma^2, \boldsymbol{\beta} \sim \text{Normal}_n\left(\mathbf{A}\mathbf{X}\boldsymbol{\beta}, \sigma^2\mathbf{A}\mathbf{A}^T\right), \\ &\mathbf{A} = (\mathbf{I}_n - \rho\mathbf{W})^{-1}. \end{aligned} \tag{3.6.1}$$

This follows from:

$$
\begin{aligned}
&\mathbf{y} = (\mathbf{I}_n - \rho\mathbf{W})^{-1}(\mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}),\\
&\text{and,}\\
&\mathbf{X} \sim \text{Normal}_n\left(\boldsymbol{\mu}, \boldsymbol{\Sigma}\right),\\
\implies &\mathbf{AX} \sim \text{Normal}_n\left(\mathbf{A}\boldsymbol{\mu}, \mathbf{A}\boldsymbol{\Sigma}\mathbf{A}^T\right).
\end{aligned}
$$

Then Equation A.2.1 (refer to Appendix A.2) was used on the partition of $\mathbf{y}$ into missing and observed data to obtain the distribution of $\mathbf{y}^{\text{mis}} \mid \mathbf{y}^{\text{obs}}, \rho, \boldsymbol{\beta}, \sigma^2$.

Similarly to the SLM, Equation 2.6.11 was used to simulate data and the parameters were drawn from the distributions given in Equations 2.6.12, 2.6.13 and 2.6.14. The conditional distribution of the missing data given the observed data, $\mathbf{y}^{\text{mis}} \mid \mathbf{y}^{\text{obs}}, \rho, \boldsymbol{\beta}, \sigma^2$ was derived similarly to the SLM:

$$
\mathbf{y} \mid \rho, \sigma^2, \boldsymbol{\beta} \sim \text{Normal}_n\left(\mathbf{X}\boldsymbol{\beta}, \sigma^2(\mathbf{I}_n - \lambda\mathbf{W})^{-1}((\mathbf{I}_n - \lambda\mathbf{W})^{-1})^T\right). \tag{3.6.2}
$$

Equation A.2.1 was applied to Equation 3.6.2 to obtain the distribution of $\mathbf{y}^{\text{mis}} \mid \mathbf{y}^{\text{obs}}, \rho, \boldsymbol{\beta}, \sigma^2$. The R code to simulate the SLM and SEM is contained in Code Blocks D.3.2 and D.4.2 (in Appendices D.3 and D.4 respectively).

3.7 A Note on Model Extensions

Models were simulated using univariate $\mathbf{y}$, since the multivariate case would add unnecessary complexity. Extension from univariate $\mathbf{y}$ to multivariate $\mathbf{Y}$ is not difficult, and would follow similarly to the extension of multiple regression to multivariate regression.

The more interesting case is missingness across non-spatial variable(s) in $\mathbf{X}$. The missing variable(s) could be imputed using conventional methods in *mice*, but it would be more efficient to leverage the relationship between $\mathbf{y}$ and $\mathbf{X}$.

Consider the SLM and assume $\mathbf{x}_i$ has missing values.

$$\begin{aligned}
&\mathbf{y} = (\mathbf{I}_n - \rho\mathbf{W})^{-1}(\mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}),\\
\Longrightarrow\ &\mathbf{X}\boldsymbol{\beta} = (\mathbf{I}_n - \rho\mathbf{W})\mathbf{y} - \boldsymbol{\epsilon},\\
\Longrightarrow\ &\mathbf{x}_i = \frac{1}{\beta_i}\tilde{\mathbf{y}} + \tilde{\mathbf{X}}\tilde{\boldsymbol{\beta}} + \tilde{\boldsymbol{\epsilon}},\\
&\tilde{\mathbf{X}} = (\mathbf{1} \quad \mathbf{x}_1 \quad \ldots \quad \mathbf{x}_{i-1} \quad \mathbf{x}_{i+1} \quad \ldots \quad \mathbf{x}_p),\\
&\tilde{\boldsymbol{\beta}} = \frac{-1}{\beta_i}(\beta_0 \quad \beta_1 \quad \ldots \quad \beta_{i-1} \quad \beta_{i+1} \quad \ldots \quad \beta_p),\\
&\tilde{\boldsymbol{\epsilon}} \sim \text{Normal}_n\left(\mathbf{0}, \frac{\sigma^2}{\beta_i^2}\mathbf{I}_n\right).
\end{aligned}$$

This implies, after transforming $\mathbf{y}$ using an estimate of ρ, that $\mathbf{x}_i$ can be imputed using normal linear regression with covariates of $\mathbf{y}$ and the remaining variables in $\mathbf{X}$. Before this should be considered, the performance of the spatial imputation models on $\mathbf{y}$ must be assessed. The comparative performance of these spatial methods with existing approaches in *mice* is the focus of Chapter 4.

Chapter 4

Analysis

4.1 Introduction

First, data were simulated as explained in Section 3.2, for each of the CAR model, SLM and SEM. The imputations were compared to those generated by the Gaussian regression (normal) model and predictive mean matching (PMM) in *mice* in Sections 4.3, 4.4 and 4.5.

After simulated results were analysed the method was further tested on the Boston dataset (discussed in Section 3.1). Values were amputated using the same methods as the simulated data. The best fitting spatial imputation model was compared to conventional methods in *mice* in Section 4.6.

The results for the CAR model, SLM and SEM on the simulated data are compared in this section. Due to the large number of combinations in the simulation conditions (missingness mechanism, autocorrelation and missingness percentage), only the most important results are discussed. Tables of other results can be found in Appendix B. In addition, note that the percentages in table captions refer to the missingness percentage.

Comparative measures for $\acute{\beta}_0$, $\acute{\beta}_1$ and $\acute{\beta}_2$ were similar in most cases so for ease of exposition only values for $\acute{\beta}_1$ are shown in the tables. Results in Appendix B contain

comparative measure values for $\acute{\beta}_0$, $\acute{\beta}_1$ and $\acute{\beta}_2$.

Note that data can be amputated using different distributions as specified in Section 3.2. Initial investigations indicated no substantial differences in results by changing the missingness distribution. The best-performing model remained the same and only marginal differences in the comparative metrics arose. Table 4.1.1 contains the results for the models across missingness distributions. Note to reflect the naming used in the *mice* package 'normal model' refers to a Gaussian model.

Table 4.1.1: SLM across missingness distributions under MAR 15% and $\rho = 0.8$

Model	**Mean Bias** ($\times 10^{-2}$)	**Mean CI Width** ($\times 10^{-3}$)	**Coverage**	**Distribution**
SLM	**0.13**	**0.78**	**1.00**	Right
Norm	-15.14	1.94	0.01	Right
PMM	-14.80	1.40	0.00	Right
SLM	**0.24**	**0.60**	**0.99**	Left
Norm	-14.83	1.65	0.03	Left
PMM	-15.03	1.90	0.00	Left
SLM	**0.16**	**0.89**	**1.00**	Mid
Norm	-14.74	1.24	0.02	Mid
PMM	-14.63	1.33	0.01	Mid
SLM	**0.06**	**1.14**	**1.00**	Tail
Norm	-14.67	2.41	0.03	Tail
PMM	-15.06	3.11	0.01	Tail

Table 4.1.1 indicates that the SLM is the best-performing model: lowest bias and CI width and highest coverage regardless of missingness distribution. It must be noted that the coverage being so dichotomous (either 0 or 1) points to an issue, likely to do with the CI width. It could also be the case that not enough noise was added to the sim-

ulation compared to the high autocorrelation resulting in very narrow CI widths. The normal model and PMM have very similar values in all cases, so it can be concluded that their performance is similar. The most important aspect of the table, the ranking of the models, was not affected by the missingness distribution. For this reason, only 'Right' missingness is used to amputate the data.

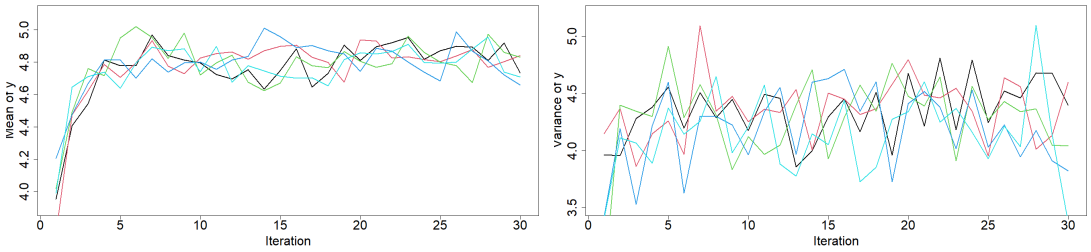
It must also be stated that due to the relatively high correlation between $\mathbf{z}$, $\mathbf{y}$ and $\mathbf{X}$ (by construction of the simulated data) the models displayed more robustness to MNAR than is typically expected. The use of the Boston data aims to provide a more realistic example. Refer to Section 4.6 for an analysis of the Boston data.

4.2 Convergence and Model Assessment

The *mice* package includes plots for assessing the convergence of the chosen imputation model. The plots display the mean and variance of the imputed variable across iterations of the MCMC simulation. Good convergence is indicated by the 'mixing' of the chains and the absence of a continuous trend (van Buuren, 2018).

Convergence was for most models, besides the SEM at high levels of missingness under MAR. Examples of convergence are provided in Appendix C, however, note that these convergence plots only illustrate one overall simulation of MICE. It is not possible to comment on convergence for all 100 simulations of each model, but poor convergence would be reflected in poor estimates. A model that has not adequately converged would be unlikely to produce representative imputations (resulting in poor pooled estimates).

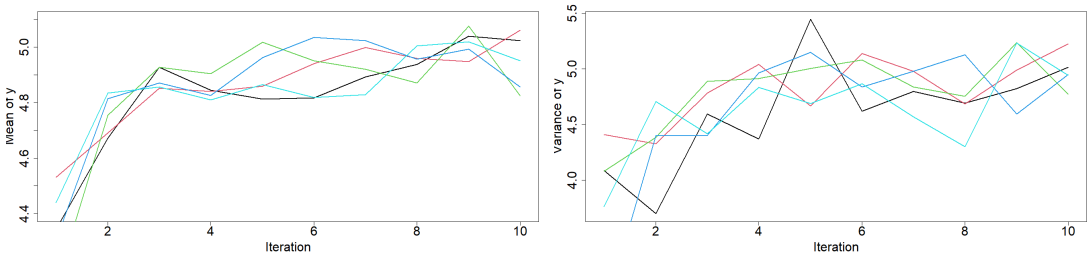
Figures 4.2.1 and C.1.1 provide examples of good convergence for the CAR model and the SLM. The figures illustrate the mean (left) and standard deviation (right) of $\mathbf{y}$ across iterations of MICE (horizontal axis).



(a) Mean across MICE iterations (b) Variance across MICE iterations

Figure 4.2.1: Convergence of the CAR imputation model

The initial trend dies away quickly for both the mean and variance and there is prevalent mixing of the chains indicating good convergence of MICE. Note that the CAR model is run for more iterations than the SLM, since initially convergence of the CAR model was inadequate. Figure 4.2.2 is an example of poor convergence for the SEM model, but this was not seen in all SEM simulations, only under MAR with 15% missingness.



(a) Mean across MICE iterations (b) Variance across MICE iterations

Figure 4.2.2: Example of poor convergence of the SEM

Trend is present throughout the iterations in the mean and variance which indicates poor convergence of the MICE algorithm.

Importantly, the CAR model had an issue with the intercept. Across all imputation models (CAR, PMM and normal), including the intercept resulted in unusually high bias for all the models' estimates. Table 4.2.1 provides these results.

Table 4.2.1: Bias of the CAR model with intercept

Model	**Mean Bias** ($\times 10^{-2}$)	**Mean CI Width** ($\times 10^{-3}$)	**Coverage**	**Parameter**
CAR	**37.53**	4.49	0.15	Intercept
Norm	**45.13**	3.78	0.05	Intercept
PMM	**53.83**	11.23	0.05	Intercept
CAR	-12.55	1.41	0.13	Slope of y
Norm	-15.05	1.25	0.03	Slope of y
PMM	-17.79	3.57	0.04	Slope of y

The unexpected bias may be due to an implementation issue with the CAR model. As a result of the unexpected bias, the intercept was removed for the CAR model in further simulations by setting $\acute{\beta}_0 = 0$. This approach succeeded in reducing the bias of the model estimates.

Unfortunately, the graphical tools to assess the imputation models only provide information for a single simulation of MICE. Examples of resulting plots are given in Figures 4.2.3, 4.2.4 and 4.2.5. The Gaussian kernel is used to estimate the densities in these figures for the variable $\mathbf{y}$ in the case of the MNAR mechanism with 15% missingness and ρ=0.5. The models compared are the SLM, the normal model and PMM.

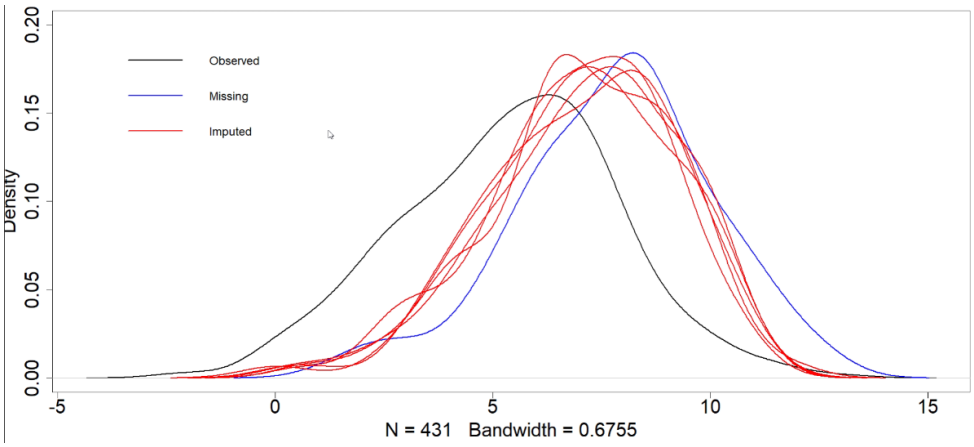


Figure 4.2.3: An example fit of the SLM imputation model

The densities of the imputed data match the density of the missing data well which indicates good performance of the imputation model.

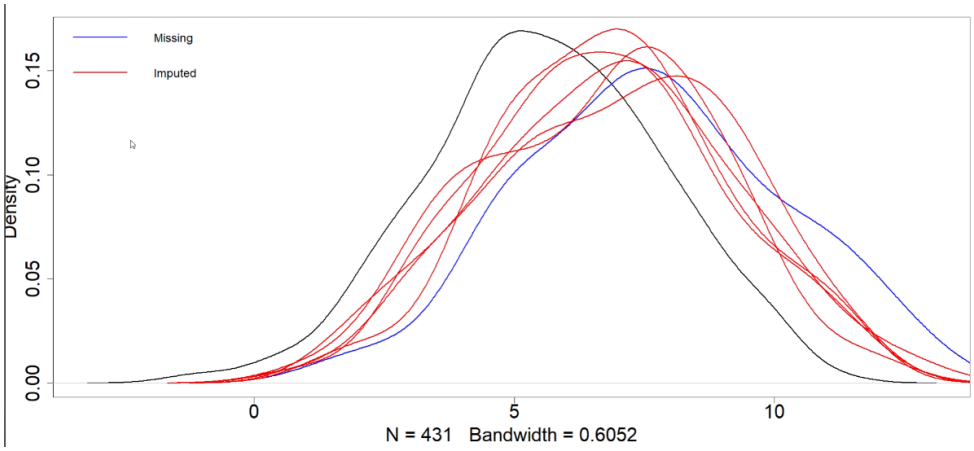


Figure 4.2.4: An example of the normal model with SLM data

For this imputation the normal model appears to perform well as the density of the imputed data is similar to that of the missing data, with slight bias to the left.

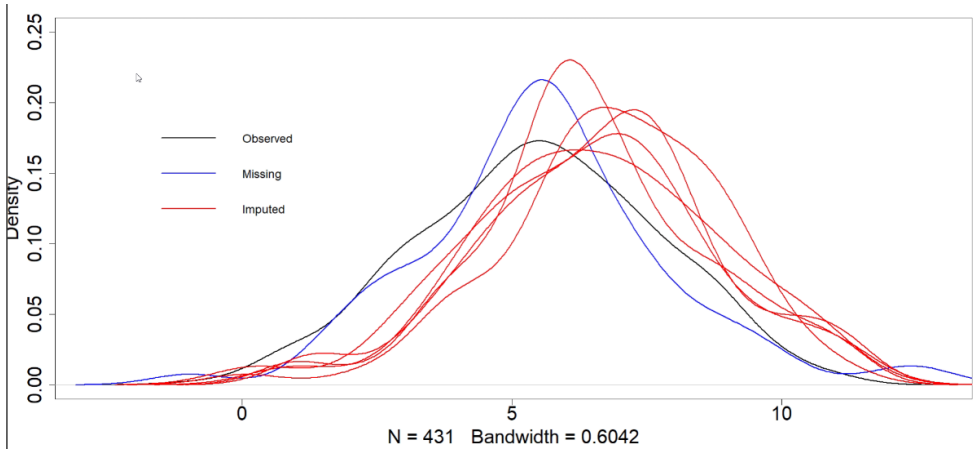


Figure 4.2.5: An example of PMM under SLM data

PMM performs poorly in this imputation, since the density of the imputed data lies to the right of the missing data which implies bias.

The plots cannot be used to infer much about the simulation as they omit the 99 other simulated results. The plots do show a closer fit to the density of the missing data for the SLM. Instead of using these plots, tables with the mean bias, mean CI width and coverage are used. It is important to note that mean squared error is a misleading metric for comparing imputation models, so it is omitted (van Buuren, 2018).

4.3 CAR Model Results

The autocorrelation did not seem to affect the comparative measures of the models. Consider Table 4.3.1 that provides results for MAR missingness of 15% with $\rho = 0.8$ and $\rho = 0.2$.

Table 4.3.1: Effect of autocorrelation on comparative measures for the CAR model

Model	**Mean Bias** ($\times 10^{-2}$)	**Mean CI Width** ($\times 10^{-3}$)	**Coverage**	ρ
CAR	-4.48	2.50	0.91	0.80
Norm	-0.59	1.44	0.92	0.80
PMM	-0.67	1.54	0.94	0.80
CAR	-3.30	2.72	0.98	0.20
Norm	-3.30	2.38	0.98	0.20
PMM	-1.51	2.91	0.97	0.20

Table 4.3.1 indicates that higher autocorrelation results in reduced coverage, but lower bias, except for the CAR model, which had a higher bias. The table also indicates that the CAR model performs worse ($\rho = 0.8$ higher bias, CI width and lower coverage) or just as well as PMM and the normal model ($rho = 0.2$ same bias and coverage as the normal model). This behaviour is unexpected since the CAR model is similar to the SEM, which outperformed PMM and the normal model by having lower bias, CI width and coverage in most scenarios. Both models should produce similar results (see Section 4.5). The models both contain spatial autocorrelation incorporated into the covariance matrix. These findings point to a possible implementation issue with the CAR model.

However, the CAR model performs favourably in other simulations, such as the case of MAR with 10% missingness and $\rho = 0.8$ in Table 4.3.2. The CAR model is (barely) the best-performing model: highest coverage, similar CI width and second best bias, but the comparative measures are all very similar.

Table 4.3.2: Comparison under MAR 10% with $\rho = 0.8$

Model	**Mean Bias** ($\times 10^{-2}$)	**Mean CI Width** ($\times 10^{-3}$)	**Coverage**
CAR	-0.23	0.22	0.98
Norm	-0.26	0.22	0.97
PMM	-0.19	0.25	0.97

This was also the case for the same conditions under the MNAR mechanism. Table 4.3.3 shows that the CAR model performs the best across all metrics, but once again the difference between models is minimal.

Table 4.3.3: Comparison under MNAR 10% with $\rho = 0.8$

Model	**Mean Bias** ($\times 10^{-2}$)	**Mean CI Width** ($\times 10^{-3}$)	**Coverage**
CAR	0.23	0.20	0.99
Norm	0.27	0.23	0.97
PMM	0.37	0.24	0.96

In the case of MCAR, the CAR model was no longer the best-performing model, but the comparative measures were once again similar. These results are provided in Table 4.3.4.

Table 4.3.4: Comparison under MCAR 10% with $\rho = 0.8$

Model	**Mean Bias** ($\times 10^{-2}$)	**Mean CI Width** ($\times 10^{-3}$)	**Coverage**
CAR	-0.32	0.25	0.89
Norm	-0.27	0.21	1.00
PMM	-0.25	0.23	0.99

In general, results were not very different in terms of comparative model performance when changing from MAR to MNAR or MCAR under the same simulation conditions

(same missingness percentage and parameter choices). Biases and CI widths remained similar while coverage fluctuated. There were no clear patterns that point to the missingness mechanism influencing the comparative measures.

Considering the difference in performance between the CAR model and the other methods, there appears to be no substantial benefit in using the CAR model in the case of data that follows the distribution of the CAR model. PMM and the normal model provide comparable performance with a much less effort.

4.4 SLM Results

The results to follow indicate poor performance in applying PMM and the normal model to data generated using the SLM. Even under MAR at 5% missingness and low autocorrelation ($\rho = 0.2$), these models had comparatively high bias and low coverage. These results are displayed in Table 4.4.1.

Table 4.4.1: Comparison of the SLM under MAR 5% with $\rho = 0.2$

Model	**Mean Bias** ($\times 10^{-2}$)	**Mean CI Width** ($\times 10^{-3}$)	**Coverage**
SLM	-1.96	0.81	0.93
Norm	-4.89	1.61	0.65
PMM	-5.35	1.03	0.48

The performance of the SLM appeared to improve with higher autocorrelation. Table 4.4.2 indicates the performance of the SLM at different missingness percentages and missingness mechanisms grouped by the strength of autocorrelation.

Table 4.4.2: Effect of autocorrelation on SLM under MAR 10%

Model	**Mean Bias** ($\times 10^{-2}$)	**Mean CI Width** ($\times 10^{-3}$)	**Coverage**	ρ
SLM	-3.92	1.20	0.76	0.20
SLM	-1.72	1.20	0.98	0.50
SLM	0.04	0.66	1.00	0.80

In Table 4.4.2 a clear trend is present: bias and CI width decrease as autocorrelation increases, while coverage increases with autocorrelation. At $\rho = 0.8$, it was found in some cases that the estimated coverage of the model was 100% or very close to 100%. This is either caused by the model overestimating variance, or more likely, the IID noise used across simulations was not enough to offset the high autocorrelation, resulting in higher than normal coverage. This trend is also present under the MNAR mechanism, as illustrated in Table 4.4.3.

Table 4.4.3: Effect of autocorrelation on the SLM under MNAR 10%

Model	**Mean Bias** ($\times 10^{-2}$)	**Mean CI Width** ($\times 10^{-3}$)	**Coverage**	ρ
SLM	-4.26	1.43	0.80	0.20
SLM	-1.59	0.83	0.98	0.50
SLM	0.93	0.80	0.94	0.80

The higher coverage for $\rho = 0.5$ compared to $\rho = 0.8$ is likely due to the simulation size of 100. Other simulations do not show a decrease in coverage from $\rho = 0.5$ to $\rho = 0.8$. These findings agree with previous literature that increased autocorrelation improves the performance of a spatial imputation model (Panzera et al., 2016).

MCAR and MAR mechanism did not have a significant influence on the simulation outcome. Consider the results in Table 4.4.4, which indicate very similar results for the SLM regardless of MAR or MCAR. This may indicate that the MAR generated data is MCAR like. PMM and the normal model display lower coverage under MAR

compared to MCAR, but this trend was not evident in other simulation scenarios: different missingness percentages and values of ρ.

Table 4.4.4: Comparison of MAR and MCAR for the SLM with 10% missingness and $\rho = 0.8$

Model	**Mean Bias** ($\times 10^{-2}$)	**Mean CI Width** ($\times 10^{-3}$)	**Coverage**	**Mechanism**
SLM	**0.11**	**0.47**	**1.00**	MCAR
Norm	-9.81	1.09	0.13	MCAR
PMM	-9.75	1.84	0.08	MCAR
SLM	**0.04**	**0.66**	**1.00**	MAR
Norm	-9.76	1.62	0.06	MAR
PMM	-10.23	1.19	0.06	MAR

Simulated results point to increased bias, CI width and decreased coverage as the missingness percentage increases. Intuitively this makes sense as more missing data should result in comparatively worse estimates and more uncertainty in estimated parameters. These claims are supported by comparing Table 4.4.1 to Table 4.4.5.

Table 4.4.5: Comparison of models under the SLM MAR 15% $\rho = 0.2$

Model	**Mean Bias** ($\times 10^{-2}$)	**Mean CI Width** ($\times 10^{-3}$)	**Coverage**
SLM	-6.04	1.31	0.53
Norm	-14.76	3.15	0.01
PMM	-15.60	2.06	0.01

Compared to Table 4.4.1, bias and CI width increased across all models, while coverage decreased. In Table 4.4.5, the models all perform poorly, but the SLM performs the best. The low autocorrelation is likely to blame for this, since the SLM cannot identify strong relations in the data. A possible solution would be to increase the number

of imputed datasets m. The SLM should be used with caution when autocorrelation is low. The trend of higher bias and CI width and lower coverage as missingness is increased occurs at higher autocorrelations and under MNAR.

There were no major differences between results under MCAR, MAR or MNAR. When keeping all simulation conditions the same and comparing results, coverage would sometimes increase and at other times would decrease. Bias and CI width values also fluctuated between missingness mechanisms, showing no clear pattern.

Overall, the SLM displays a clear improvement over PMM and the normal model that is worth the computational cost, considering the poor performance of the other two methods.

4.5 SEM Results

Similarly to the results of SLM generated data, PMM and the normal model showed very poor performance compared to the SEM for SEM data, even at low levels of missingness and low autocorrelation. In Table 4.5.1 the SEM model performs the best and comparatively, PMM and the normal model show high bias and low coverage.

Table 4.5.1: Comparison under MAR 5% with $\lambda = 0.2$ of the SEM

Model	**Mean Bias** ($\times 10^{-2}$)	**Mean CI Width** ($\times 10^{-3}$)	**Coverage**
SEM	-1.64	0.63	0.92
Norm	-4.88	0.89	0.57
PMM	-5.36	0.75	0.47

Although the trend is not as strong for the SEM as the SLM, autocorrelation seemed to affect bias, CI width and coverage. Higher autocorrelation resulted in lower bias and CI width and greater coverage for the SEM. PMM and the normal model did not benefit from increased autocorrelation. These findings are supported by the comparative measures in Table 4.5.2

Table 4.5.2: Effect of autocorrelation on the SEM under MAR 10%

Model	**Mean Bias** ($\times 10^{-2}$)	**Mean CI Width** ($\times 10^{-3}$)	**Coverage**	λ
SEM	-3.54	0.97	0.83	0.20
Norm	-10.18	1.64	0.08	0.20
PMM	-10.11	0.95	0.10	0.20
SEM	-1.48	0.63	0.87	0.80
Norm	-9.73	1.85	0.11	0.80
PMM	-10.09	1.34	0.05	0.80

The coverage of the normal model increased, but this is likely just due to variation in the simulations since this trend was not evident in other simulations. Interestingly, in the case of comparing MAR to MNAR, results tend to have a lower bias and a higher CI width in the case of MNAR, resulting in better coverage. Table 4.5.3 contains results for the SEM under 10% missingness and $\lambda = 0.5$.

Table 4.5.3: Comparison of the SEM under MAR and MNAR with 10% missingness and $\lambda = 0.5$

Model	**Mean Bias** ($\times 10^{-2}$)	**Mean CI Width** ($\times 10^{-3}$)	**Coverage**	**Mechanism**
SEM	**-3.06**	**0.76**	**0.80**	MAR
Norm	-10.02	1.58	0.12	MAR
PMM	-10.07	1.18	0.11	MAR
SEM	**-2.68**	**0.81**	**0.92**	MNAR
Norm	-10.02	1.60	0.14	MNAR
PMM	-9.87	1.16	0.14	MNAR

Similar to the patterns of the SLM, MAR and MCAR displayed similar results. Table 4.5.4 indicates very small changes in the comparative measures between the SEM un-

der MCAR and MAR. Other simulations also implied no notable difference between results for the SEM under MCAR or MAR. This is further indication that the results of the *ampute()* method are MCAR-like data.

Table 4.5.4: Comparison of MAR and MCAR for the SEM with 10% missingness and $\lambda = 0.8$

Model	**Mean Bias** ($\times 10^{-2}$)	**Mean CI Width** ($\times 10^{-3}$)	**Mechanism**	λ
SEM	**-1.45**	**0.23**	**0.91**	MCAR
Norm	-9.95	1.33	0.10	MCAR
PMM	-9.96	2.13	0.03	MCAR
SEM	**-1.48**	**0.63**	**0.87**	MAR
Norm	-9.73	1.86	0.11	MAR
PMM	-10.09	1.34	0.05	MAR

Just like the SLM, the SEM has higher bias, CI width and lower coverage for higher missingness percentages. Due to the performance improvements over PMM and the normal model, the computational cost is worth the more accurate parameter estimates.

4.6 Results of Imputation on the Boston Dataset

First possible models for the complete data were considered. The logarithm of 'CMEDV' was fitted in a linear regression with 'CRIM' and 'RM' as covariates. Testing the residuals with Moran's I indicated residual spatial autocorrelation (Schabenberger and Gotway, 2017). Spatial models were compared and the SEM model proved to be the best fit for the data, with the SLM performing comparably. Models were compared using mean squared error and the Akaike Information Criterion in Figure C.2.1 (shown in Appendix C).

Data amputation was carried out on variable 'CRIM' under MAR and MNAR with

15% missingness to see the effect of missingness (10% and 5% would be similar but with smaller differences). The SLM was found to fit better than the SEM for imputing 'CRIM' based on 'RM' by comparing the same metrics for the complete model fit (refer to Figure C.2.2). Finally, a simulation of size 100 was run using the SLM, PMM and the normal model. The model convergence was good and can be seen in Figure C.1.2 in Appendix C. The comparative measures for the pooled estimates of the coefficient for 'CRIM' are provided in Tables 4.6.1, 4.6.2 and 4.6.3.

Table 4.6.1: Imputation Models under MCAR on the Boston data

Model	**Mean Bias** ($\times 10^{-2}$)	**Mean CI Width** ($\times 10^{-3}$)	**Coverage**
SLM	-0.01	0.08	0.99
Norm	0.24	0.05	0.49
PMM	0.27	0.06	0.61

Table 4.6.2: Imputation Models under MAR on the Boston data

Model	**Mean Bias** ($\times 10^{-2}$)	**Mean CI Width** ($\times 10^{-3}$)	**Coverage**
SLM	-0.01	0.06	0.99
Norm	0.27	0.06	0.52
PMM	0.21	0.05	0.89

Table 4.6.3: Imputation Models under MNAR on the Boston data

Model	**Mean Bias** ($\times 10^{-2}$)	**Mean CI Width** ($\times 10^{-3}$)	**Coverage**
SLM	-0.46	0.10	0.62
Norm	0.15	0.09	0.91
PMM	0.21	0.08	0.91

Tables 4.6.1 and 4.6.2 indicate the SLM performs much better than the other models, since the SLM has the lowest bias and the highest coverage. However, the opposite case occurs under MNAR in Table 4.6.3. This is unexpected for the case of MNAR, but could be attributed to the missing values being impactful on the model's estimate for spatial autocorrelation ρ.

Bias in ρ would affect the other parameter estimates in the imputation model. R code for analysis of the Boston data is in Code Block D.5.1 in Appendix D.5. Plots of the locations of the missing data under MAR and MCAR do not seem to indicate any patterns that would explain the poor performance of the SLM (refer to Figures C.2.1 and C.2.2 in Appendix C).

4.7 Conclusion

The simulation results imply that the SLM and SEM can provide improved pooled estimates compared to PMM and the normal model, which displayed high bias. The performance of PMM and the normal model indicated that they can be inappropriate imputation models for certain spatially correlated data. The CAR model was found to have no significant difference between PMM and the normal model. The results from the Boston dataset indicate that the SLM imputation model performed well under the MAR mechanism, but may be more sensitive to the case of MNAR. the similarity between results under MCAR and MAR warrant further investigation of the amputated data.

Chapter 5

Conclusion and Recommendations

5.1 Summary

Missing data are a common occurrence across many data types and spatial data are no exception. Current approaches to dealing with missingness in spatial data in the literature are mainly tailored to a specific problem. Research in non-statistical fields commonly utilise single imputation, which can lead to biased estimates.

In this study, the effectiveness of spatial imputation models in comparison to existing MI methods in *mice* in R was assessed. Three different spatial models were considered: the CAR model, the SLM and the SEM. For each of these models, autocorrelations of 0.2, 0.5 and 0.8 were used to simulate data.

The simulated data were amputated using the *ampute()* method in *mice* under the MCAR, MAR and MNAR mechanisms, at 5%, 10% and 15% missingness. Imputations were generated for each spatial model and compared with imputations generated by PMM and the normal (Gaussian) model.

Simulation results indicated no substantial difference between the CAR model, PMM and the normal model. In contrast, the results for the SLM and SEM were superior to those of PMM and the normal model. These results were investigated further through the use of real data.

The results of the imputation models on the Boston dataset indicated that the SLM imputation model performs well under MAR, but may be sensitive to MNAR. The computational cost of the spatial models is likely worthwhile, since PMM and the normal model can produce poor imputations in strongly correlated spatial data.

5.2 Conclusion

The objectives of this study were completed as MCAR, MAR and MNAR were investigated for the SLM, SEM and the CAR model. The spatial imputation models were compared to PMM and the normal model from *mice* at 5%, 10% and 15% missingness.

The CAR model did not perform significantly better or worse than PMM or the normal model. The SLM and SEM, however, showed significantly better performance in the simulated data. The real data supported these findings for MAR, but suggests further analysis is needed in the case of MNAR.

5.3 Recommendations

The implementation of the imputation methods could be executed in a more computationally efficient manner regarding matrix calculations (mainly calculation of inverses). Adjusting the code for the independent MICE simulations to run in parallel would result in great performance benefits.

5.4 Limitations

The issue related to the intercept of the CAR model is a clear limitation that could be addressed. Considering the similarity between the CAR and SEM models, it is unclear why the intercept of the CAR model should cause any issue.

The simulation size of 100 for the MICE algorithm resulted in reduced accuracy of the comparative measures for the models. The results indicated that bias and CI width remain similar across simulations so coverage had the most loss in accuracy.

It is not possible to verify the amputated data are MAR and not MCAR for only one type of missingness pattern (Jamshidian and Jalal, 2010). Some small investigations are included in Listing D.7.1 and show that complete case analysis gives unbiased estimates, which should not happen for MAR data.

5.5 Scope for Future Work

An extension of this methodology to the SDM and SDEM can be explored as these are also common spatial regression models. Semi-parametric and non-parametric spatial models could also be considered in future work.

It would also be possible to extend application of the imputation model to missingness in multiple variables by utilising Section 3.7. Separately, an extension of block bootstrapping could be compared to the Bayesian results provided for the imputation models.

Finally, MI methods could be investigated for geostatistical data and compared to conventional approaches in the literature. The methodology applied in this study building on the *mice* package could be used in the context of geostatistical data.

References

Allison, P. D. (2009). Missing data. *The SAGE Handbook of Quantitative Methods in Psychology*, pages 631–657. 2, 6, 11

Anselin, L. (2009). Spatial regression. *The SAGE Handbook of Spatial Analysis*, 1:255–276. 22, 23

Bae, B., Kim, H., Lim, H., Liu, Y., Han, L. D., and Freeze, P. B. (2018). Missing data imputation for traffic flow speed using spatio-temporal cokriging. *Transportation Research Part C: Emerging Technologies*, 88:124–139. 4, 15, 16

Baker, J., White, N., and Mengersen, K. (2014). Missing in space: an evaluation of imputation methods for missing data in spatial analysis of risk factors for type II diabetes. *International Journal of Health Geographics*, 13:1–13. 13, 29

Besag, J., York, J., and Mollié, A. (1991). Bayesian image restoration, with two applications in spatial statistics. *Annals of the Institute of Statistical Mathematics*, 43:1–20. 13, 20

Chung, S. Y., Venkatramanan, S., Elzain, H. E., Selvam, S., and Prasanna, M. (2019). Supplement of missing data in groundwater-level variations of peak type using geostatistical methods. *GIS and Geostatistical Techniques for Groundwater Science*, pages 33–41. 4, 15, 16

De Oliveira, V. (2012). Bayesian analysis of conditional autoregressive models. *Annals of the Institute of Statistical Mathematics*, 64:107–133. 21, 22, 35

Feng, L., Moritz, S., Nowak, G., Welsh, A. H., and O'Neill, T. J. (2020). *imputeR: A General Multivariate Imputation Framework*. R package version 2.2, https://CRAN.R-project.org/package=imputeR. 12

Giorgi, E. and Diggle, P. J. (2015). On the inverse geostatistical problem of inference on missing locations. *Spatial Statistics*, 11:35–44. 16

Haining, R. (2003). *Spatial Data Analysis*. Cambridge University Press, Cambridge, United Kingdom. 1, 5, 13

Harrison Jr, D. and Rubinfeld, D. L. (1978). Hedonic housing prices and the demand for clean air. *Journal of Environmental Economics and Management*, 5(1):81–102. 27

Holt, W. and Nguyen, D. (2023). Essential aspects to Bayesian data imputation. *SSRN*. doi: http://dx.doi.org/10.2139/ssrn.4494311. 62

Jamshidian, M. and Jalal, S. (2010). Tests of Homoscedasticity, Normality and Missing Completely at Random for Incomplete Multivariate Normal Data. *Psychometrika*, 4(75):649–674. doi:10.1007/s11336-010-9175-3. 57

LeSage, J. P. (2000). Bayesian estimation of limited dependent variable spatial autoregressive models. *Geographical Analysis*, 32(1):19–35. 23, 24

Little, R. J. and Rubin, D. B. (2019). *Statistical Analysis with Missing Data*, volume 793. John Wiley & Sons, Hoboken, USA. 4

Moraga, P. (2023). *Spatial Statistics for Data Science: Theory and Practice with R*. Chapman & Hall/CRC Data Science Series, Boca Raton, Florida. 28

Morris, M., Wheeler-Martin, K., Simpson, D., Mooney, S. J., Gelman, A., and DiMaggio, C. (2019). Bayesian hierarchical spatial models: Implementing the Besag York Mollié model in Stan. *Spatial and Spatio-temporal Epidemiology*, 31:100301. 20

Panzera, D., Benedetti, R., and Postiglione, P. (2016). A Bayesian approach to parameter estimation in the presence of spatial missing data. *Spatial Economic Analysis*, 11(2):201–218. 1, 5, 13, 14, 20, 21, 48

Prener, Christopher, Revord, and Charles (2019). areal: An R package for areal weighted interpolation. *Journal of Open Source Software*, 4(37). https://doi.org/10.21105/joss.01221. 13

Puranik, A. (2021). Estimation of missing values in aggregate level spatial data. *Clinical Epidemiology and Global Health*, 9:1–6. 1, 5, 14, 23, 24, 29, 30

R Development Core Team (2006). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0, http://www.R-project.org. 2

Rubin, D. B. (1976). Inference and missing data. *Biometrika*, 63(3):581–592. 2, 4, 5, 6, 8

Rubin, D. B. (1988). An Overview of Multiple Imputation. In *Proceedings of the Survey Research Methods Section of the American Statistical Association*, volume 79, page 84. Citeseer. 4, 7, 10, 13, 14

Rüttenauer, T. (2022). Spatial regression models: A systematic comparison of different model specifications using Monte Carlo experiments. *Sociological Methods & Research*, 51(2):728–759. 31

Santos, M. S., Pereira, R. C., Costa, A. F., Soares, J. P., Santos, J., and Abreu, P. H. (2019). Generating synthetic missing data: A review by missing mechanism. *IEEE Access*, 7:11651–11667. 17, 18, 19, 30

Schabenberger, O. and Gotway, C. A. (2017). *Statistical Methods for Spatial Data Analysis*. CRC Press, Boca Raton, Florida. 52

Schafer, J. L. (1997). *Analysis of incomplete multivariate data*. CRC Press, Boca Raton, Florida. 4, 5, 6, 7, 11

Schouten, R. M., Lugtig, P., and Vink, G. (2018). Generating missing values for simulation purposes: a multivariate amputation procedure. *Journal of Statistical Computation and Simulation*, 88(15):2909–2930. 17, 18, 19, 30

Stroud, J. R., Stein, M. L., and Lysen, S. (2017). Bayesian and maximum likelihood estimation for Gaussian processes on an incomplete lattice. *Journal of Computational and Graphical Statistics*, 26(1). 15, 31

van Buuren, S. (2018). *Flexible Imputation of Missing Data. Second Edition*. CRC Press, Boca Raton, Florida. 6, 7, 8, 13, 14, 19, 32, 40, 44

van Buuren, S. and Groothuis-Oudshoorn, K. (2011). mice: Multivariate imputation by chained equations in R. *Journal of Statistical Software*, 45(3):1–67. doi: http://dx.doi.org/10.18637/jss.v045.i03. 2, 6, 7, 8, 9

Venables, W. N. and Ripley, B. D. (2002). *Modern Applied Statistics with S, Fourth Edition*. Springer, New York. ISBN 0-387-95457-0, https://www.stats.ox.ac.uk/pub/MASS4/. 35

Ver Hoef, J. M., Hanks, E. M., and Hooten, M. B. (2018). On the relationship between conditional (CAR) and simultaneous (SAR) autoregressive models. *Spatial Statistics*, 25:68–85. 21, 24, 31, 35

Appendix A

Distributional Results Used

A.1 Inverse Gamma Distribution

If $X \sim \text{InvGamma}(\alpha, \beta)$, then the probability density function of X is given by:

$$f(x) = \frac{\beta^{\alpha}}{\Gamma(\alpha)} \left(\frac{1}{x}\right)^{(\alpha+1)} \exp\left(-\frac{\beta}{x}\right)$$

A.2 Partitioning of Normal Distribution

Let $\mathbf{y}$ be an n-dimensional multivariate normal distribution, $\mathbf{y}_1$ be q chosen variables from $\mathbf{y}$ and $\mathbf{y}_2$ be the remaining $n - q$ variables. The conditional distribution $\mathbf{y}_1 \mid \mathbf{y}_2$ can be expressed by Equation A.2.1 (Holt and Nguyen, 2023):

$$\begin{aligned} \begin{pmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \end{pmatrix} &\sim \text{Normal}_n \left(\begin{pmatrix} \boldsymbol{\mu}_1 \\ \boldsymbol{\mu}_2 \end{pmatrix}, \begin{pmatrix} \boldsymbol{\Sigma}_{11} & \boldsymbol{\Sigma}_{12} \\ \boldsymbol{\Sigma}_{21} & \boldsymbol{\Sigma}_{22} \end{pmatrix} \right) \\ \therefore \mathbf{y}_1 \mid \mathbf{y}_2 &\sim \text{Normal}_q(\tilde{\boldsymbol{\mu}}, \tilde{\boldsymbol{\Sigma}}) \qquad \text{(A.2.1)} \\ \tilde{\boldsymbol{\mu}} &= \boldsymbol{\mu}_1 + \boldsymbol{\Sigma}_{12}\boldsymbol{\Sigma}_{22}^{-1}(\mathbf{y}_2 - \boldsymbol{\mu}_2) \\ \tilde{\boldsymbol{\Sigma}} &= \boldsymbol{\Sigma}_{11} - \boldsymbol{\Sigma}_{12}\boldsymbol{\Sigma}_{22}^{-1}\boldsymbol{\Sigma}_{21} \end{aligned}$$

Appendix B

Results

The following raw results are taken from R and contain the comparative measures for each simulation scenario and all the respective model parameters. The caption provides the missingness percentage for each result.

B.1 CAR Model Results

Table B.1.1: CAR under MCAR 10% $\rho = 0.8$

```
> resTablepmm
          Bias     CI Width  Coverage
y -0.002468345 0.0002251063      0.99
x  0.007020188 0.0007705082      0.97
> resTableCAR
          Bias     CI Width  Coverage
y -0.003195102 0.0002451270      0.89
x  0.004961599 0.0008680277      0.95
> resTableNorm
          Bias     CI Width  Coverage
y -0.002679013 0.0002135312      1.00
x  0.002925467 0.0008687173      0.98
```

Table B.1.2: CAR under MCAR 10% $\rho = 0.2$

```
> resTablepmm
          Bias     CI Width  Coverage
y -0.001346745 0.0002228288      0.99
x  0.017436821 0.0007390700      0.87
> resTableCAR
          Bias     CI Width  Coverage
y -0.002745247 0.0002337114      0.93
x  0.005201283 0.0008756500      0.99
> resTableNorm
          Bias     CI Width  Coverage
y -0.001768061 0.0001857655      0.97
x  0.002395228 0.0007001102      0.97
```

Table B.1.3: CAR under MAR 5% $\rho = 0.8$

```
> resTablepmm
          Bias     CI Width  Coverage
y 0.0002867929 0.0001393859      0.99
x 0.0005181669 0.0004758983      0.95
> resTableCAR
           Bias     CI Width  Coverage
y -0.0005650591 0.0001660378      0.97
x  0.0028524288 0.0006800134      0.95
> resTableNorm
          Bias     CI Width  Coverage
y -0.000214030 0.0001474652      0.96
x  0.001177937 0.0005348184      0.93
```

Table B.1.4: CAR under MAR 10% $\rho = 0.8$

```
> resTablepmm
          Bias     CI Width  Coverage
y -0.001881326 0.0002547455      0.97
x  0.018234947 0.0008080874      0.86
> resTableCAR
          Bias     CI Width  Coverage
y -0.002254548 0.0002191958      0.98
x  0.005385838 0.0007989347      1.00
> resTableNorm
          Bias     CI Width  Coverage
y -0.002568428 0.0002157793      0.97
x  0.004580926 0.0008233110      0.98
```

Table B.1.5: CAR under MAR 15% $\rho = 0.8$

```
> resTablepmm
           Bias     CI Width  Coverage
y -0.0006749765 0.0001538981      0.94
x  0.0016821145 0.0005435716      0.95
> resTableCAR
           Bias     CI Width  Coverage
y -0.004475324 0.0002504602      0.91
x  0.009295742 0.0010466567      0.97
> resTableNorm
           Bias     CI Width  Coverage
y -0.0005855294 0.0001438330      0.92
x  0.0026400472 0.0005147938      0.96
```

Table B.1.6: CAR under MAR 5% $\rho = 0.5$

```
> resTablepmm
           Bias     CI Width  Coverage
y -0.0008248649 0.0001892089      0.99
x  0.0082369584 0.0006719943      0.94
> resTableCAR
           Bias     CI Width  Coverage
y -0.001118987 0.0001977527      0.98
x  0.003224857 0.0006976834      0.96
> resTableNorm
           Bias     CI Width  Coverage
y -0.001059208 0.0001925533      0.98
x  0.003691880 0.0006117986      0.97
```

Table B.1.7: CAR under MAR 10% $\rho = 0.5$

```
> resTablepmm
           Bias     CI Width  Coverage
y -0.002267917 0.0002141077      0.97
x  0.014481998 0.0007940167      0.90
> resTableCAR
           Bias     CI Width  Coverage
y -0.002260719 0.0002165158      0.98
x  0.002717808 0.0008307178      0.98
> resTableNorm
           Bias     CI Width  Coverage
y -0.001936071 0.0002891535      0.96
x  0.003741966 0.0007084088      0.99
```

Table B.1.8: CAR under MAR 15% $\rho = 0.5$

```
> resTablepmm
          Bias     CI Width  Coverage
y -0.004088424 0.0002446731      0.94
x -0.002191233 0.0009836571      0.97
> resTableCAR
          Bias     CI Width  Coverage
y -0.002898199 0.0002258000      0.95
x  0.006538751 0.0008187672      0.99
> resTableNorm
          Bias     CI Width  Coverage
y -0.003302084 0.0002938971      0.94
x -0.005270344 0.0007602285      0.99
```

Table B.1.9: CAR under MAR 5% $\rho = 0.2$

```
> resTablepmm
          Bias     CI Width  Coverage
y -0.000977268 0.0002005323      0.98
x  0.010787014 0.0006320119      0.92
> resTableCAR
          Bias     CI Width  Coverage
y -0.000501847 0.0001998597      0.99
x  0.001403253 0.0006932511      0.99
> resTableNorm
          Bias     CI Width  Coverage
y -0.001591068 0.0001676349      0.95
x  0.003370108 0.0006264472      0.98
```

Table B.1.10: CAR under MAR 10% $\rho = 0.2$

```
> resTablepmm
          Bias     CI Width  Coverage
y -0.001718437 0.0002329351      0.99
x  0.014435222 0.0006951500      0.90
> resTableCAR
          Bias     CI Width  Coverage
y -0.002081026 0.0002095745      0.96
x  0.006070912 0.0007348578      0.98
> resTableNorm
          Bias     CI Width  Coverage
y -0.001501330 0.0002026566      0.99
x  0.003504205 0.0006725829      0.98
```

Table B.1.11: CAR under MAR 15% $\rho = 0.2$

```
> resTablepmm
          Bias     CI Width  Coverage
y -0.001511881 0.0002906275      0.97
x  0.027667054 0.0008343820      0.80
> resTableCAR
          Bias     CI Width  Coverage
y -0.003304429 0.0002721525      0.98
x  0.005958302 0.0008676673      1.00
> resTableNorm
          Bias     CI Width  Coverage
y -0.003304989 0.0002379314      0.98
x  0.008137850 0.0008749220      0.98
```

Table B.1.12: CAR under MNAR 5% $\rho = 0.8$

```
> resTablepmm
           Bias     CI Width  Coverage
y -0.0009864431 0.0002221502      0.97
x  0.0098690645 0.0007825098      0.95
> ##
> resTableCAR
          Bias     CI Width  Coverage
y 0.0008923839 0.0001707049      0.97
x 0.0049190505 0.0005699609      0.97
> resTableNorm
          Bias     CI Width  Coverage
y -0.001765085 0.0001858876      0.97
x  0.005310095 0.0006134330      0.97
```

Table B.1.13: CAR under MNAR 10% $\rho = 0.8$

```
> resTablepmm
         Bias     CI Width  Coverage
y 0.003726129 0.0002347652      0.96
x 0.017046771 0.0008970481      0.87
> resTableCAR
         Bias     CI Width  Coverage
y 0.002339640 0.0002035984      0.99
x 0.003112211 0.0007383191      0.98
> resTableNorm
         Bias     CI Width  Coverage
y 0.002666929 0.0002321912      0.97
x 0.006602459 0.0008498369      0.97
```

Table B.1.14: CAR under MNAR 15% $\rho = 0.8$

```
> resTablepmm
        Bias     CI Width  Coverage
y 0.00566466 0.0002572793      0.92
x 0.02867370 0.0009847798      0.81
> resTableCAR
         Bias     CI Width  Coverage
y 0.003149752 0.0002508720      0.97
x 0.006149267 0.0008210745      1.00
> resTableNorm
         Bias     CI Width  Coverage
y 0.004034854 0.0002484979      0.98
x 0.007898685 0.0007693527      0.99
```

Table B.1.15: CAR under MNAR 5% $\rho = 0.5$

```
> resTablepmm
          Bias     CI Width  Coverage
y -0.001108673 0.0001829309      0.98
x -0.001291464 0.0006526256      0.92
> resTableCAR
           Bias     CI Width  Coverage
y -0.0007582023 0.0001766683      1.00
x  0.0021027996 0.0006449320      0.99
> resTableNorm
          Bias     CI Width  Coverage
y -0.001168433 0.0001667561      0.96
x -0.003155405 0.0006466759      0.99
```

Table B.1.16: CAR under MNAR 10% $\rho = 0.5$

```
> resTablepmm
          Bias     CI Width  Coverage
y -0.008768988 0.0002453489      0.71
x  0.041899801 0.0009953273      0.63
> resTableCAR
          Bias     CI Width  Coverage
y -0.006409738 0.0002081189      0.83
x  0.022021701 0.0008162609      0.91
> resTableNorm
          Bias     CI Width  Coverage
y -0.008292369 0.000277243      0.74
x  0.030254848 0.001265351      0.83
```

Table B.1.17: CAR under MNAR 15% $\rho = 0.5$

```
> resTablepmm
         Bias     CI Width  Coverage
y 0.004602311 0.0002749512      0.94
x 0.024323342 0.0018106987      0.81
> resTableCAR
         Bias     CI Width  Coverage
y 0.002865937 0.0002680258      0.94
x 0.008624048 0.0008703491      0.99
> resTableNorm
         Bias     CI Width  Coverage
y 0.003478986 0.0002445092      0.98
x 0.009509034 0.0008361528      0.95
```

Table B.1.18: CAR under MNAR 5% $\rho = 0.2$

```
> resTablepmm
          Bias     CI Width  Coverage
y -0.0004141142 0.0001747865      0.97
x  0.0080225891 0.0005890918      0.95
> resTableCAR
         Bias     CI Width  Coverage
y 0.001003436 0.0001743494      0.99
x 0.005478309 0.0006370440      0.95
> resTableNorm
          Bias     CI Width  Coverage
y -0.001596556 0.0001607404      0.97
x  0.002982386 0.0005486082      0.94
```

Table B.1.19: CAR under MNAR 10% $\rho = 0.2$

```
> resTablepmm
         Bias     CI Width  Coverage
y 0.00264344 0.0002152744      0.97
x 0.01913735 0.0007941098      0.89
> resTableCAR
         Bias     CI Width  Coverage
y 0.002373817 0.0002002336      0.99
x 0.004023931 0.0007845728      0.99
> resTableNorm
         Bias     CI Width  Coverage
y 0.002736537 0.0001999036      0.97
x 0.008335860 0.0006641473      0.96
```

Table B.1.20: CAR under MNAR 15% $\rho = 0.2$

```
> resTablepmm
         Bias     CI Width  Coverage
y 0.004399227 0.0003180998      0.96
x 0.024258147 0.0008434220      0.85
> resTableCAR
         Bias     CI Width  Coverage
y 0.003261515 0.0002754568      0.95
x 0.007598705 0.0010608343      0.98
> resTableNorm
         Bias     CI Width  Coverage
y 0.004306508 0.0002260739      0.97
x 0.006800060 0.0007860537      0.99
```

B.2 SLM Results

Table B.2.1: SLM under MCAR 10% $\rho = 0.8$

```
> resTablepmm
                Bias    CI Width  Coverage
Intercept  1.46403388 0.028306485      0.08
y         -0.09751127 0.001836250      0.08
x          0.23965499 0.003767866      0.26
> resTableSLM
                Bias    CI Width  Coverage
Intercept -0.02139039 0.007032502         1
y          0.00110672 0.000470895         1
x         -0.01230829 0.001549100         1
> resTableNorm
                Bias    CI Width  Coverage
Intercept  1.47357038 0.016241143      0.14
y         -0.09813132 0.001094642      0.13
x          0.23792548 0.003081584      0.29
```

Table B.2.2: SLM under MCAR 10% $\rho = 0.2$

```
> resTablepmm
                Bias    CI Width  Coverage
Intercept  0.3642930 0.004788377      0.14
y         -0.0964653 0.001305564      0.10
x          0.1986101 0.002598574      0.14
> resTableSLM
                 Bias    CI Width  Coverage
Intercept  0.15249852 0.003907734      0.79
y         -0.04032619 0.000915983      0.78
x          0.08204736 0.002079649      0.82
> resTableNorm
                 Bias    CI Width  Coverage
Intercept  0.36402639 0.005469296      0.14
y         -0.09765775 0.001373688      0.11
x          0.19719627 0.002564023      0.17
```

Table B.2.3: SLM under MAR 5% $\rho = 0.8$

```
Intercept  0.0117334619 0.0033620977      0.94
y         -0.0007036526 0.0002202687      0.93
x          0.0010013545 0.0006711641      0.95
> resTableSLM
                  Bias     CI Width  Coverage
Intercept -0.007976134 0.0048349929      1.00
y          0.000154388 0.0003221301      1.00
x         -0.011920931 0.0012953187      0.98
> resTableNorm
                   Bias     CI Width  Coverage
Intercept  0.0028235925 0.0028935174      0.96
y         -0.0003025166 0.0002028154      0.96
x          0.0017128514 0.0006497265      0.98
```

Table B.2.4: SLM under MAR 10% $\rho = 0.8$

```
Intercept  1.5435487 0.017635245      0.10
y         -0.1022503 0.001188710      0.06
x          0.2705484 0.002770735      0.22
> resTableSLM
                  Bias     CI Width  Coverage
Intercept -0.0185151435 0.0095538823         1
y          0.0004468626 0.0006624737         1
x         -0.0166165701 0.0023636467         1
> resTableNorm
                 Bias    CI Width  Coverage
Intercept  1.46432552 0.022234364      0.04
y         -0.09763562 0.001621621      0.06
x          0.24225857 0.004176333      0.30
```

Table B.2.5: SLM under MAR 15% $\rho = 0.8$

```
> resTablepmm
               Bias    CI Width  Coverage
Intercept  2.2300789 0.021585054      0.00
y         -0.1479988 0.001396256      0.00
x          0.3914647 0.004423698      0.05
> resTableSLM
                Bias     CI Width  Coverage
Intercept -0.037945633 0.0121845956      1.00
y          0.001258649 0.0007791316      1.00
x         -0.028960387 0.0021803433      0.98
> resTableNorm
               Bias    CI Width  Coverage
Intercept  2.2681003 0.028772699      0.01
y         -0.1514473 0.001937016      0.01
x          0.3808155 0.005078566      0.05
```

Table B.2.6: SLM under MAR 5% $\rho = 0.5$

```
                Bias    CI Width  Coverage
Intercept  0.34323469 0.009012347      0.51
y         -0.05048225 0.001476821      0.58
x          0.13875773 0.003260333      0.47
> resTableSLM
                Bias     CI Width  Coverage
Intercept  0.054877676 0.0035643464      0.95
y         -0.009504447 0.0005820284      0.95
x          0.018207760 0.0014100879      0.96
> resTableNorm
               Bias    CI Width  Coverage
Intercept  0.3333199 0.006748253      0.59
y         -0.0499388 0.001147601      0.65
x          0.1295662 0.002178621      0.56
```

Table B.2.7: SLM under MAR 10% $\rho = 0.5$

```
> resTablepmm
               Bias     CI Width  Coverage
Intercept  0.6837035 0.0054176260      0.06
y         -0.1031086 0.0008403282      0.16
x          0.2628092 0.0023578141      0.06
> resTableSLM
                Bias    CI Width  Coverage
Intercept  0.14838182 0.007312206      0.94
y         -0.01716615 0.001196788      0.98
x          0.06549686 0.003083031      0.88
> resTableNorm
               Bias    CI Width  Coverage
Intercept  0.6740468 0.007694546      0.09
y         -0.1006076 0.001283426      0.13
x          0.2568513 0.003132414      0.06
```

Table B.2.8: SLM under MAR 15% $\rho = 0.5$

```
                Bias    CI Width  Coverage
Intercept  0.9130307 0.009829086      0.02
y         -0.1518005 0.001720475      0.01
x          0.3350201 0.003678110      0.01
> resTableSLM
                 Bias    CI Width  Coverage
Intercept  0.18222341 0.010425536      0.93
y         -0.03086996 0.001790440      0.92
x          0.06748711 0.004085891      0.93
> resTableNorm
                Bias    CI Width  Coverage
Intercept  0.9013515 0.012673874      0.01
y         -0.1500954 0.002032792      0.01
x          0.3205465 0.004862483      0.02
```

Table B.2.9: SLM under MAR 5% $\rho = 0.2$

```
> resTablepmm
                Bias    CI Width  Coverage
Intercept  0.2045786 0.004056829      0.48
y         -0.0535356 0.001033542      0.48
x          0.1138544 0.002737788      0.59
> resTableSLM
                 Bias     CI Width  Coverage
Intercept  0.07684881 0.0032514235      0.90
y         -0.01957406 0.0008076176      0.93
x          0.04032570 0.0018870055      0.94
> resTableNorm
                Bias    CI Width  Coverage
Intercept  0.18096893 0.005447872      0.65
y         -0.04891446 0.001612294      0.65
x          0.09617146 0.002640503      0.69
```

Table B.2.10: SLM under MAR 10% $\rho = 0.2$

```
                Bias     CI Width  Coverage
Intercept  0.3995950 0.0034265779      0.09
y         -0.1048773 0.0008507835      0.06
x          0.2242205 0.0021818081      0.10
> resTableSLM
                 Bias    CI Width  Coverage
Intercept  0.15260916 0.004753381      0.77
y         -0.03923797 0.001201822      0.76
x          0.08855663 0.002919029      0.78
> resTableNorm
                Bias    CI Width  Coverage
Intercept  0.36902025 0.004820008      0.13
y         -0.09788841 0.001311115      0.09
x          0.19790443 0.002567626      0.20
```

Table B.2.11: SLM under MAR 15% $\rho = 0.2$

```
> resTablepmm
                Bias    CI Width  Coverage
Intercept  0.5893587 0.008171259      0.01
y         -0.1560301 0.002063783      0.01
x          0.3305203 0.005109224      0.01
> resTableSLM
                 Bias    CI Width  Coverage
Intercept  0.23115759 0.005064725      0.57
y         -0.06038192 0.001311725      0.53
x          0.13091996 0.002614845      0.61
> resTableNorm
                Bias    CI Width  Coverage
Intercept  0.5558539 0.011690643      0.02
y         -0.1475867 0.003150390      0.01
x          0.2967175 0.006420465      0.03
```

Table B.2.12: SLM under MNAR 5% $\rho = 0.8$

```
> resTablepmm
                Bias    CI Width  Coverage
Intercept  0.7963753 0.019380485      0.60
y         -0.0480416 0.001216099      0.68
x          0.1571041 0.003088502      0.61
> resTableSLM
                 Bias     CI Width  Coverage
Intercept -0.05292885 0.0051036814      0.99
y          0.00510957 0.0003359057      0.99
x         -0.00326301 0.0012494756      0.99
> resTableNorm
                Bias     CI Width  Coverage
Intercept  0.84131123 0.0126361300      0.48
y         -0.05117634 0.0008104738      0.59
x          0.16251411 0.0030609906      0.54
```

Table B.2.13: SLM under MNAR 10% $\rho = 0.8$

```
                Bias    CI Width  Coverage
Intercept  1.58942853 0.025981778      0.11
y         -0.09661444 0.001728347      0.19
x          0.31992629 0.004476846      0.06
> resTableSLM
                 Bias     CI Width  Coverage
Intercept -0.085756711 0.0110722885      0.96
y          0.009273848 0.0007975781      0.94
x         -0.004774063 0.0020549394      0.99
> resTableNorm
                Bias    CI Width  Coverage
Intercept  1.62569993 0.025377109      0.13
y         -0.09909065 0.001722930      0.16
x          0.30975853 0.005376014      0.12
```

Table B.2.14: SLM under MNAR 15% $\rho = 0.8$

```
                Bias    CI Width  Coverage
Intercept  2.4010793 0.026805295      0.03
y         -0.1468941 0.001792628      0.05
x          0.4614621 0.004898589      0.01
> resTableSLM
                 Bias     CI Width  Coverage
Intercept -0.11651929 0.0134298221      0.98
y          0.01273082 0.0009868095      0.97
x         -0.00965093 0.0042994432      1.00
> resTablepmm
                Bias    CI Width  Coverage
Intercept  2.4010793 0.026805295      0.03
y         -0.1468941 0.001792628      0.05
x          0.4614621 0.004898589      0.01
```

Table B.2.15: SLM under MNAR 5% $\rho = 0.5$

```
> resTablepmm
                 Bias     CI Width  Coverage
Intercept  0.32878630 0.0049209574      0.49
y         -0.04788851 0.0007673897      0.58
x          0.13008564 0.0019826381      0.52
> resTableSLM
                 Bias     CI Width  Coverage
Intercept  0.066212362 0.0040333273      0.97
y         -0.007264702 0.0006836533      1.00
x          0.029801245 0.0017535706      0.95
> resTableNorm
                Bias    CI Width  Coverage
Intercept  0.33878239 0.008965549      0.44
y         -0.04999078 0.001440617      0.59
x          0.13166107 0.003059310      0.48
```

Table B.2.16: SLM under MNAR 10% $\rho = 0.5$

```
> resTablepmm
                Bias    CI Width  Coverage
Intercept  0.6738201 0.006748282      0.07
y         -0.1006944 0.001147654      0.12
x          0.2601892 0.002573277      0.12
> resTableSLM
                 Bias     CI Width  Coverage
Intercept  0.14095410 0.0049674484      0.93
y         -0.01589197 0.0008287759      0.98
x          0.06067350 0.0020020187      0.89
> resTableNorm
                Bias    CI Width  Coverage
Intercept  0.66355617 0.008451948      0.08
y         -0.09990429 0.001336215      0.14
x          0.24918287 0.003275684      0.09
```

Table B.2.17: SLM under MNAR 15% $\rho = 0.5$

```
> resTablepmm
                Bias    CI Width  Coverage
Intercept  1.0132774 0.018360256      0.00
y         -0.1523284 0.003071770      0.02
x          0.3866126 0.006528477      0.01
> resTableSLM
                Bias    CI Width  Coverage
Intercept  0.21625502 0.007873584      0.91
y         -0.02458141 0.001248776      0.97
x          0.09351643 0.003690684      0.85
> resTableNorm
                Bias    CI Width  Coverage
Intercept  0.9837769 0.012916853      0.02
y         -0.1482417 0.002062644      0.04
x          0.3660076 0.004743601      0.02
```

Table B.2.18: SLM under MNAR 5% $\rho = 0.2$

```
                Bias    CI Width  Coverage
Intercept  0.22306859 0.004243943      0.49
y         -0.05183727 0.001030535      0.64
x          0.12905304 0.002401405      0.50
> resTableSLM
                Bias     CI Width  Coverage
Intercept  0.10271696 0.0028145738      0.86
y         -0.02019029 0.0007384575      0.94
x          0.05992127 0.0016447141      0.89
> resTableNorm
                Bias    CI Width  Coverage
Intercept  0.22303513 0.002912559      0.47
y         -0.05169533 0.000747222      0.62
x          0.12449702 0.001783033      0.57
```

Table B.2.19: SLM under MNAR 10% $\rho = 0.2$

```
                Bias    CI Width  Coverage
Intercept  0.4304448 0.004974357      0.04
y         -0.1009032 0.001206016      0.11
x          0.2448951 0.002898510      0.07
> resTableSLM
                Bias    CI Width  Coverage
Intercept  0.21268077 0.005553026      0.47
y         -0.04264651 0.001430539      0.80
x          0.12486181 0.003088997      0.49
> resTableNorm
                Bias    CI Width  Coverage
Intercept  0.4375193 0.005689635      0.06
y         -0.1033568 0.001471389      0.10
x          0.2397130 0.003370526      0.08
```

Table B.2.20: SLM under MNAR 15% $\rho = 0.2$

```
                Bias    CI Width  Coverage
Intercept  0.6531667 0.006346387      0.00
y         -0.1532175 0.001655712      0.00
x          0.3612792 0.003645737      0.01
> resTableSLM
                Bias    CI Width  Coverage
Intercept  0.29856484 0.006168686      0.31
y         -0.05971379 0.001520955      0.64
x          0.17107761 0.003744844      0.35
> resTableNorm
                Bias    CI Width  Coverage
Intercept  0.6400544 0.006918848      0.02
y         -0.1502950 0.001854752      0.05
x          0.3476185 0.003694352      0.03
```

B.3 SEM Results

Table B.3.1: SEM under MCAR 10% $\lambda = 0.8$

```
> resTablepmm
                Bias    CI Width  Coverage
Intercept  0.30602530 0.006617497      0.10
y         -0.09958282 0.002129618      0.03
x          0.19833208 0.004473257      0.18
> resTableSEM
                Bias     CI Width  Coverage
Intercept  0.04063622 0.0008210608     0.935
y         -0.01450496 0.0002323643     0.905
x          0.02829652 0.0006588904     0.945
> resTableNorm
                Bias    CI Width  Coverage
Intercept  0.2977961 0.003516606      0.08
y         -0.0994525 0.001326991      0.10
x          0.1973714 0.002494946      0.22
```

Table B.3.2: SEM under MCAR 10% $\lambda = 0.2$

```
> resTablepmm
                Bias    CI Width  Coverage
Intercept  0.29279717 0.003603624      0.17
y         -0.09681652 0.001191044      0.11
x          0.19771922 0.002364436      0.14
> resTableSEM
                Bias     CI Width  Coverage
Intercept  0.10733103 0.0028460228      0.84
y         -0.03590444 0.0008987705      0.80
x          0.07027543 0.0019914710      0.83
> resTableNorm
                Bias    CI Width  Coverage
Intercept  0.29221267 0.004521655      0.16
y         -0.09812402 0.001416653      0.09
x          0.19693192 0.002547251      0.15
```

Table B.3.3: SEM under MAR 5% $\lambda = 0.8$

```
> resTablepmm
                Bias    CI Width  Coverage
Intercept  0.15374990 0.003816587      0.53
y         -0.04956288 0.001282175      0.50
x          0.10605311 0.002764296      0.64
> resTableSEM
                 Bias    CI Width  Coverage
Intercept  0.020707052 0.001591837      0.98
y         -0.007365255 0.000487108      0.93
x          0.014571382 0.001290723      0.94
> resTableNorm
                Bias    CI Width  Coverage
Intercept  0.14172007 0.003853177      0.69
y         -0.04839352 0.001179075      0.60
x          0.09628073 0.003054735      0.77
```

Table B.3.4: SEM under MAR 10% $\lambda = 0.8$

```
> resTablepmm
                Bias    CI Width  Coverage
Intercept  0.3040684 0.004448324      0.10
y         -0.1009185 0.001341770      0.05
x          0.2143432 0.003423571      0.20
> resTableSEM
                 Bias     CI Width  Coverage
Intercept  0.04346071 0.0020877613      0.90
y         -0.01480431 0.0006297354      0.87
x          0.03001355 0.0014613017      0.92
> resTableNorm
                Bias    CI Width  Coverage
Intercept  0.29574737 0.006010854      0.18
y         -0.09729218 0.001851902      0.11
x          0.19657345 0.004244823      0.34
```

Table B.3.5: SEM under MAR 15% $\lambda = 0.8$

```
> resTablepmm
                Bias    CI Width  Coverage
Intercept  0.4492201 0.004998805      0.00
y         -0.1479283 0.001608883      0.00
x          0.3162421 0.004420944      0.02
> resTableSEM
                 Bias     CI Width  Coverage
Intercept  0.07276586 0.0024040836      0.81
y         -0.02366710 0.0007242536      0.78
x          0.04762113 0.0017802771      0.83
> resTableNorm
                Bias    CI Width  Coverage
Intercept  0.4483192 0.005201096      0.04
y         -0.1484997 0.001671885      0.02
x          0.2925713 0.004037722      0.09
```

Table B.3.6: SEM under MAR 5% $\lambda = 0.5$

```
> resTablepmm
                 Bias    CI.Width Coverage
Intercept  0.15532187 0.003220859     0.57
y         -0.05113644 0.001071689     0.49
x          0.11220578 0.002040946     0.53
> resTableSEM
                 Bias     CI.Width Coverage
Intercept  0.03528655 0.0021861148     0.94
y         -0.01282209 0.0006983817     0.93
x          0.02528582 0.0014995319     0.96
> resTableNorm
                 Bias    CI.Width Coverage
Intercept  0.15113480 0.003203974     0.58
y         -0.04942092 0.001047263     0.60
x          0.10209515 0.001862194     0.67
```

Table B.3.7: SEM under MAR 10% $\lambda = 0.5$

```
> resTablepmm
                Bias    CI Width  Coverage
Intercept  0.3043936 0.003502672      0.16
y         -0.1007240 0.001177560      0.11
x          0.2124709 0.002408052      0.20
> resTableSEM
                 Bias     CI Width  Coverage
Intercept  0.09361349 0.0025575142      0.79
y         -0.03059431 0.0007605424      0.80
x          0.06339909 0.0017791431      0.81
> resTableNorm
                Bias    CI Width  Coverage
Intercept  0.3036668 0.004528684      0.14
y         -0.1001965 0.001584866      0.12
x          0.2000629 0.002895424      0.20
```

Table B.3.8: SEM under MAR 15% $\lambda = 0.5$

```
> resTablepmm
                Bias    CI Width  Coverage
Intercept  0.5401290 0.003906358      0.00
y         -0.1522199 0.001187931      0.01
x          0.3609468 0.002889983      0.01
> resTableSEM
                Bias    CI Width  Coverage
Intercept  0.15497567 0.003437476      0.54
y         -0.05089677 0.001064958      0.44
x          0.10544658 0.002715249      0.56
> resTableNorm
                Bias    CI Width  Coverage
Intercept  0.5296923 0.003765078      0.01
y         -0.1508800 0.001309521      0.04
x          0.3444332 0.002942656      0.03
```

Table B.3.9: SEM under MAR 5% $\lambda = 0.2$

```
                Bias     CI Width  Coverage
Intercept  0.16163240 0.0024290287      0.50
y         -0.05357393 0.0007450641      0.47
x          0.11254231 0.0017413130      0.56
> resTableSEM
                Bias     CI Width  Coverage
Intercept  0.04847508 0.0020003948      0.94
y         -0.01644611 0.0006331119      0.92
x          0.03169693 0.0014102930      0.95
> resTableNorm
                Bias     CI Width  Coverage
Intercept  0.14599736 0.0027652993      0.63
y         -0.04876713 0.0008882772      0.57
x          0.09653175 0.0019894034      0.72
```

Table B.3.10: SEM under MAR 10% $\lambda = 0.2$

```
> resTablepmm
                Bias    CI Width  Coverage
Intercept  0.3101137 0.003087165      0.12
y         -0.1010994 0.000946315      0.10
x          0.2102922 0.002365487      0.15
> resTableSEM
                Bias     CI Width  Coverage
Intercept  0.10872289 0.0031325295      0.80
y         -0.03539491 0.0009724899      0.83
x          0.07221155 0.0025404388      0.80
> resTableNorm
                Bias    CI Width  Coverage
Intercept  0.3049598 0.004665672      0.10
y         -0.1017616 0.001639821      0.08
x          0.2019061 0.002957179      0.13
```

Table B.3.11: SEM under MAR 15% $\lambda = 0.2$

```
> resTableEpmm
                Bias    CI Width  Coverage
Intercept  0.4691202 0.003770403         0
y         -0.1541829 0.001199612         0
x          0.3239120 0.002612622         0
> resTableSEM
                 Bias    CI Width  Coverage
Intercept  0.16197190 0.004587462      0.58
y         -0.05343763 0.001463200      0.53
x          0.10166785 0.003093601      0.65
> resTableNorm
                Bias    CI Width  Coverage
Intercept  0.4551668 0.005207753      0.01
y         -0.1508029 0.001772071      0.02
x          0.3002964 0.003899731      0.03
```

Table B.3.12: SEM under MNAR 5% $\lambda = 0.8$

```
                Bias    CI Width  Coverage
Intercept  0.15446414 0.00388588      0.56
y         -0.04968391 0.00135353      0.49
x          0.10604683 0.00299561      0.65
> resTableSEM
                 Bias     CI Width  Coverage
Intercept  0.025553477 0.0016055139      0.92
y         -0.002658695 0.0004876181      0.93
x          0.016834959 0.0011080052      0.95
> resTableNorm
                Bias     CI Width  Coverage
Intercept  0.19981646 0.0027446403      0.39
y         -0.05120614 0.0007874082      0.62
x          0.12958612 0.0021509878      0.60
```

Table B.3.13: SEM under MNAR 10% $\lambda = 0.8$

```
                Bias    CI Width  Coverage
Intercept  0.3151060 0.003996111      0.11
y         -0.1014894 0.001338198      0.10
x          0.2149603 0.002458244      0.22
> resTableSEM
                 Bias     CI Width  Coverage
Intercept  0.060120236 0.0019287294      0.83
y         -0.006212423 0.0006035783      0.98
x          0.032428328 0.0014846467      0.89
> resTableNorm
                Bias    CI Width  Coverage
Intercept  0.3955410 0.005464838      0.06
y         -0.1012272 0.001361457      0.17
x          0.2523350 0.003467447      0.09
>
```

Table B.3.14: SLM under MNAR 15% $\lambda = 0.8$

```
                Bias    CI Width  Coverage
Intercept  0.4720252 0.003844895      0.01
y         -0.1557353 0.001299157      0.01
x          0.3364159 0.002589760      0.02
> resTableSEM
                 Bias    CI Width  Coverage
Intercept  0.08915940 0.002334402      0.73
y         -0.01101433 0.000648025      0.96
x          0.05213240 0.001436149      0.85
> resTableNorm
                Bias    CI Width  Coverage
Intercept  0.4498412 0.005230600      0.02
y         -0.1493365 0.001548621      0.00
x          0.2997842 0.003632098      0.08
```

Table B.3.15: SEM under MNAR 5% $\lambda = 0.5$

```
> resTablepmm
                Bias    CI Width  Coverage
Intercept  0.1804272 0.003833439      0.42
y         -0.0493914 0.001091254      0.63
x          0.1232557 0.002612746      0.53
> resTableSEM
                 Bias    CI Width  Coverage
Intercept  0.06216538 0.002056452      0.84
y         -0.01305005 0.000656071      0.97
x          0.04436344 0.001529524      0.92
> resTableNorm
                Bias    CI Width  Coverage
Intercept  0.18140077 0.002942070      0.49
y         -0.04992574 0.000815655      0.57
x          0.11895919 0.001986408      0.57
```

Table B.3.16: SEM under MNAR 10% $\lambda = 0.5$

```
> resTablepmm
                Bias    CI Width  Coverage
Intercept  0.3613594 0.003751814      0.05
y         -0.1009679 0.001161776      0.14
x          0.2453714 0.002677552      0.09
> resTableSEM
                 Bias     CI Width  Coverage
Intercept  0.12876473 0.0026401812      0.60
y         -0.02683534 0.0008102991      0.92
x          0.08056328 0.0017811507      0.75
> resTableNorm
                Bias    CI Width  Coverage
Intercept  0.35544879 0.004986128      0.09
y         -0.09868764 0.001601493      0.14
x          0.23488767 0.003761454      0.12
```

Table B.3.17: SEM under MNAR 15% $\lambda = 0.5$

```
> resTablepmm
                Bias    CI Width  Coverage
Intercept  0.5440004 0.004027374         0
y         -0.1534073 0.001353896         0
x          0.3700962 0.003330124         0
> resTableSEM
                Bias    CI Width  Coverage
Intercept  0.1916733 0.003820309      0.29
y         -0.0410631 0.001266535      0.79
x          0.1190210 0.003189094      0.54
> resTableNorm
                Bias    CI Width  Coverage
Intercept  0.5256166 0.006255885      0.00
y         -0.1485658 0.001648865      0.01
x          0.3370596 0.004081973      0.01
```

Table B.3.18: SEM under MNAR 5% $\lambda = 0.2$

```
                Bias     CI Width  Coverage
Intercept  0.18576560 0.0024544627      0.43
y         -0.05197567 0.0007398064      0.53
x          0.12664113 0.0018355166      0.50
> resTableSEM
                 Bias     CI Width  Coverage
Intercept  0.08730868 0.0023791851      0.84
y         -0.01991110 0.0007521524      0.97
x          0.05768572 0.0016083995      0.86
> resTableNorm
                Bias    CI Width  Coverage
Intercept  0.1713012 0.003526662      0.52
y         -0.0484532 0.001032550      0.60
x          0.1137523 0.002044333      0.52
```

Table B.3.19: SEM under MNAR 10% $\lambda = 0.2$

```
                Bias    CI Width  Coverage
Intercept  0.3565167 0.003045867      0.07
y         -0.1013527 0.001000512      0.14
x          0.2423128 0.002169804      0.08
> resTableSEM
                 Bias    CI Width  Coverage
Intercept  0.15799358 0.003118251      0.47
y         -0.03546519 0.001020211      0.84
x          0.09866590 0.002258600      0.65
> resTableNorm
                Bias    CI Width  Coverage
Intercept  0.3537378 0.004483657      0.08
y         -0.1015982 0.001326903      0.17
x          0.2316475 0.003489446      0.12
```

Table B.3.20: SEM under MNAR 15% $\lambda = 0.2$

```
                Bias    CI Width  Coverage
Intercept  0.5473011 0.005949555      0.01
y         -0.1576212 0.001752025      0.01
x          0.3680343 0.004030129      0.01
> ##
> resTableSEM
                Bias    CI Width  Coverage
Intercept  0.23274036 0.003591736      0.19
y         -0.05272422 0.001149619      0.71
x          0.14781353 0.002544067      0.41
> resTableNorm
                Bias    CI Width  Coverage
Intercept  0.5207353 0.004631439      0.00
y         -0.1500826 0.001543149      0.02
x          0.3426635 0.003250294      0.01
>
```

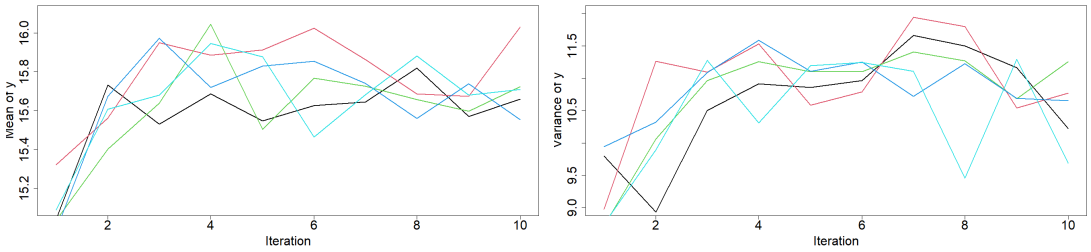
Appendix C

Graphs

C.1 Convergence

Note that the CAR model was run for 30 iterations (in *mice*: 'maxit'), while the SLM and SEM were run for 10. This is because the SLM and SEM converged in fewer iterations than the CAR model which also happened to run faster so running 30 iterations was not an issue.

Figures 4.2.1, C.1.1 and 4.2.2 provide the mean (left) and standard deviation (right) of $\mathbf{y}$ across iterations of MICE (horizontal axis).

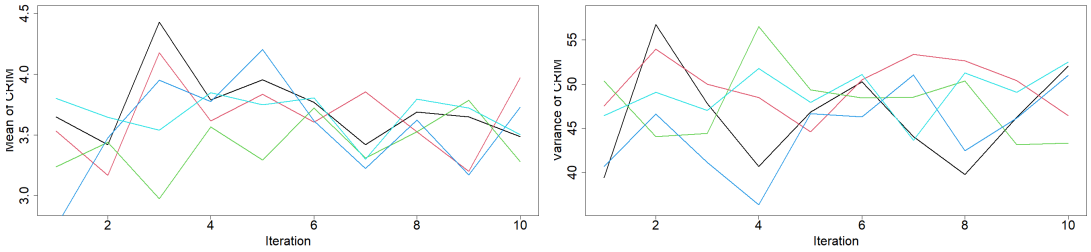


(a) Mean across MICE iterations

(b) Variance across MICE iterations

Figure C.1.1: Convergence of the SLM

There is no trend (aside from some initial trend), and the chains mix indicating adequate convergence of the MICE algorithm.



(a) Mean across MICE iterations

(b) Variance across MICE iterations

Figure C.1.2: Convergence of the SLM on the Boston dataset

There is no trend evident in the mean or variance across the iterations and mixing is sufficient to imply satisfactory convergence.

C.2 Boston Modelling

The 'top level' regression model was considered first.

Table C.2.1: Comparison of the SLM and SEM on the complete Boston data

Model	AIC	Residual MSE
SLM	-280.24	14.87
SEM	-284.41	13.48

The SEM is slightly better than the SLM since it has lower AIC and MSE. Next, the imputation models were compared.

Table C.2.2: Comparison of the SLM and SEM for Imputing variable 'CRIM'

Model	AIC	Residual MSE
SLM	3346.34	19077.86
SEM	3350.72	19152.76

For the imputation model, the SLM appears slightly better but the models are basically equivalent. Results using the SEM were very similar so only results for the SLM are included.

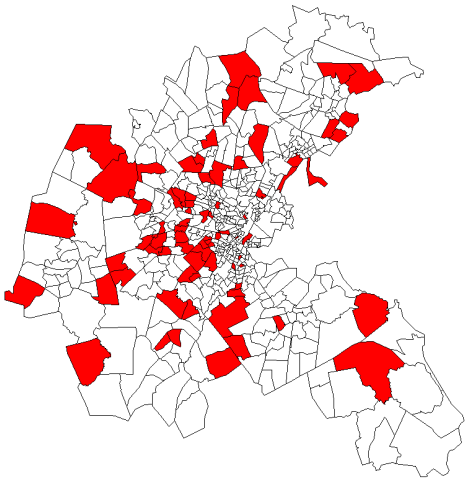


Figure C.2.1: Amputated areas under MAR for the Boston data

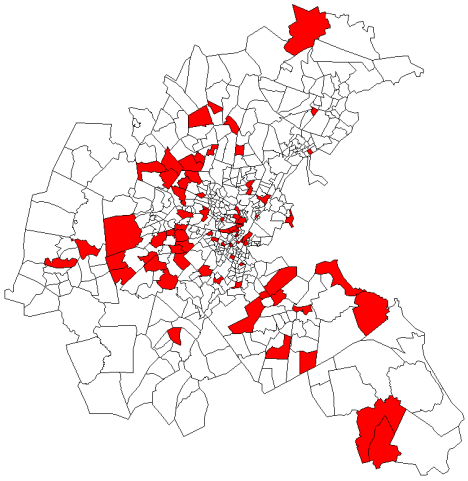


Figure C.2.2: Amputated areas under MNAR for the Boston data

There does not appear to be any pattern that would indicate a major difference in results under MAR or MNAR based on the amputated regions.

Appendix D

R Code

The code has been commented as much as possible for legibility. Large sections may be commented and these are often used to chop and change (specifically for the kind of data amputation to use).

D.1 Code Setup

Code Block D.1.1: General Setup Code

```
#Libraries
library(sp)
library(spdep)
library(mice)

library(spdep)
library(armspp)

library(MASS)
library(spatialreg)

#Data - Boston Housing (spatial)
bostonData <- read_sf(system.file("shapes/boston_tracts.shp",
package="spData")[1])

#starting imputations
```

```
multifill <- function(data,neighlist) {
  data <- as.data.frame(data)
  matchingvar <- which.max(colSums(is.na(data)))
  nmis <- sum(is.na(data[,matchingvar]))
  mis <- which(is.na(data[,matchingvar]))
  for(i in 1:nmis) {
    ###code from fill.nn
    flag <- TRUE
    nn <- neighlist[[mis[i]]]
    while(flag) {
      #check for any non-missing values
      if(length(which(!is.na(data[nn,matchingvar])))>0){
        flag <- FALSE
        ind <- nn[which(!is.na(data[nn,matchingvar]))] #neighbours with real
            values
        ind <- sample(ind,size = 1) #change to average of 2
        #ind <- sample(ind,size = 2) #change to average of all
        #set value to (average of) random neighbour(s)
        data[mis[i],matchingvar] <- mean(data[ind,matchingvar])
      }
      #if not expand order of neighbours
      else {
        for(j in 1:length(nn)) {
          nn <- c(nn,neigh[[nn[j]]])
        }
        nn <- unique(nn) #no repeats
      }
    }
    ###
    #for each variable check if missing and fill in using same neighbour(s)
    for(j in 1:ncol(data)) {
      if(is.na(data[mis[i],j])) {
        data[mis[i],j] <- mean(data[ind,j])
      }

    }
  }

  return(data)
}
```

```
#global variables
neigh <- poly2nb(bostonData)
n <- nrow(bostonData)
W <- nb2mat(neigh,style="W")
C <- nb2mat(neigh,style="B")
Beta <- c(2,1,-1)
nsims <- 100
mis.prop <- 0.1
rho.set <- c(0.2,0.5,0.8)
rho <- rho.set[2]
```

D.2 CAR Model Simulations

Code Block D.2.1: CAR Setup Code

```
#CAR Model
Beta[1] <- 0 #intercept removed due to bias
genCAR <- function(rho) {
  card <- as.numeric(rowSums(C))
  sigma <- 1
  Sig <- sigma^2*solve(diag(card)-rho*C)
  #set.seed(2305614)
  x <- rnorm(n,0,1)

  mu <- 2*x+3 +rnorm(n,0,0.125)

  #set.seed(2305614)
  y <- mvrnorm(n=1,mu=mu,Sigma=Sig)
  x.copy <- x
  y.copy <- y

  z <- Beta[2]*y+Beta[3]*x+Beta[1]+rnorm(n,0,0.25)
  return(data.frame(z=z,y=y,x=x))
}

##Amputation - copy and paste into loop
#MCAR
#ind <- sample(n,size=floor(mis.prop*n))
#CARdata$y[ind] <- NA
```

```
#MAR
#mis.prop set to 10% here, 15% other session
#temp <- ampute(CARdata,prop=mis.prop,patterns = c(1,0,1),weights=c(0,0,1)
#                ,type = "RIGHT",mech = "MAR")

#while(sum(is.na(temp$amp$y))!=floor(mis.prop*n)) {
  #  temp <- ampute(CARdata,prop=mis.prop,patterns = c(1,0,1),weights=c(0,0,1)
  #                  ,type = "RIGHT",mech = "MAR")
  #}
#CARdata$y <- temp$amp$y
#MNAR
#temp <- ampute(CARdata,prop=mis.prop,patterns = c(1,0,1),weights=c(1,1,0)
#                ,type = "RIGHT",mech="MNAR")
#while(sum(is.na(temp$amp$y))!=floor(mis.prop*n)) {
  #  temp <- ampute(CARdata,prop=mis.prop,patterns = c(1,0,1),weights=c(1,1,0)
  #                  ,type = "RIGHT",mech="MNAR")
  #}
#CARdata$y <- temp$amp$y
##

##Imputation
mice.impute.CARfill <- function(y,ry,x,wy=NULL,...) {
  if(is.null(wy)) {
    wy <- !ry
  }

  #neighbour fill with matching regions
  if(any(is.na(y)) || any(is.na(x))) {
    temp <- multifill(data.frame(y=y,x=x),neigh)
    x <- temp[,-1]
    y <- temp[,1]

  }



  x <- as.data.frame(x)
  x <- matrix(unlist(x),ncol=ncol(x))
  x <- cbind(1,x)

  card <- as.numeric(rowSums(C))
```

```
sig.phi.inv <- function(rho) {
  diag(1,n)-rho*W
}
B.func <- function(rho,x,y) {
  solve(t(x)%*%sig.phi.inv(rho)%*%x)%*%t(x)%*%sig.phi.inv(rho)%*%y
}
S.func <- function(b.phi,sig.phi.inv,x,y) {
  t(y-x%*%b.phi)%*%sig.phi.inv%*%(y-x%*%b.phi)
}
#Equation 2.14
rho.log <- function(rho) {
  sig.phi.inv <- sig.phi.inv(rho)
  beta.phi <- B.func(rho,x,y)
  value <- (-1/2)*log(1/det(sig.phi.inv)*det(t(x)%*%sig.phi.inv%*%x))
  #prior is uniform
  value <- value - ((n-ncol(x))/2)*log(S.func(beta.phi,sig.phi.inv,x,y))
  return(value)
}
rho.post <-  arms(1,log_pdf = rho.log,lower=0,upper=1,metropolis = FALSE)
#Equation 2.13
a <- (n-ncol(x))/2
b <- 1/2*S.func(B.func(rho.post,x,y),sig.phi.inv(rho.post),x,y)

sigma.post <- 1/rgamma(n=1,shape=a,rate=b)*5
#Equation 2.12
B.post <- mvrnorm(n=1,mu=B.func(rho.post,x,y),
Sigma = sigma.post*solve(t(x)%*%sig.phi.inv(rho.post)%*%x))

#Equation 2.10
imputes <- vector(length=sum(wy))
ind <- which(wy)
for(i in 1:sum(wy)) {
  sum <- 0
  for(j in 1:n) {
    sum <- sum + rho.post*W[ind[i],j]*(y[j]-x[j,]%*%B.post)
  }
  imputes[i] <- rnorm(n=1,mean=x[ind[i],]%*%B.post+sum,
  sd=sqrt(sigma.post/card[ind[i]]))
}

#print(paste("rho: ",rho.post," sigma^2: ",sigma.post," Beta: ",B.post[1],"
```

```
      "
  #            ,B.post[2])) #debugging
  if(length(wy)>0) {
    return(imputes)
  }
  else {
    print("No missing values")
    return(0)
  }
}
##
```

Code Block D.2.2: CAR Simulation Code

```
#repeated MICE runs
results <- list()
#imputation matrix and method
impMat <- diag(0,nrow=3)
impMat[2,3] <- 1
meth <- c("","CARfill","")
#
for(i in 1:nsims) {
  print(paste("Iteration: ",i))
  CARdata <- genCAR(rho)
  #paste here
  temp <- ampute(CARdata,prop=mis.prop,patterns = c(1,0,1),weights=c(0,0,1)
  ,type = "RIGHT",mech = "MAR")

  while(sum(is.na(temp$amp$y))!=floor(mis.prop*n)) {
    temp <- ampute(CARdata,prop=mis.prop,patterns = c(1,0,1),weights=c(0,0,1)
    ,type = "RIGHT",mech = "MAR")
  }
  #
  y.copy <- CARdata$y
  CARdata$y <- temp$amp$y
  yCAR <- mice(CARdata,method=meth,predictorMatrix=impMat,maxit=30,m=5)
  milm <- with(yCAR,lm(z~y+x+0))
  results[[i]] <- summary(pool(milm))
}

##Analysis Table
#Bias, CI width, Coverage
```

```
Beta <- c(1,-1)
makeTable <- function(var) {
  bias <- 0
  CI.w <- 0
  coverage <- 0

  for(i in 1:nsims) {
    bias <- bias+(results[[i]]$estimate[var]-Beta[var])
    tval <- qt(0.975,results[[i]]$df[var])
    ll <- results[[i]]$estimate[var] - tval*results[[i]]$std.error[var]
    ul <- results[[i]]$estimate[var] + tval*results[[i]]$std.error[var]
    CI.w <- ul-ll
    if(ll<Beta[var] && ul>Beta[var]) {
      coverage <- coverage+1
    }
  }
  table <- data.frame("Bias"=bias,"CI Width" = CI.w, "Coverage"=coverage)
  table <- 1/nsims * table
  return(table)
}

resTableCAR <- makeTable(1)
resTableCAR <- rbind(resTableCAR,makeTable(2))

row.names(resTableCAR) <- c("y","x")
colnames(resTableCAR) <- c("Bias","CI Width"," Coverage")
resTableCAR
##

## Comparison with normal and pmm
Beta <- c(0,1,-1)
for(i in 1:nsims) {
  CARdata <- genCAR(rho)
  #paste here
  temp <- ampute(CARdata,prop=mis.prop,patterns = c(1,0,1),weights=c(0,0,1)
  ,type = "RIGHT",mech = "MAR")

  while(sum(is.na(temp$amp$y))!=floor(mis.prop*n)) {
    temp <- ampute(CARdata,prop=mis.prop,patterns = c(1,0,1),weights=c(0,0,1)
    ,type = "RIGHT",mech = "MAR")
```

```
  }
  #
  y.copy <- CARdata$y
  CARdata$y <- temp$amp$y
  ynorm <- mice(CARdata,method=c("","norm",""),
  predictorMatrix=impMat,maxit=30,m=5,printFlag = FALSE)
  milm <- with(ynorm,lm(z~y+x+0))
  results[[i]] <- summary(pool(milm))
}
Beta <- c(1,-1)
resTableNorm <- makeTable(1)
resTableNorm <- rbind(resTableNorm,makeTable(2))

row.names(resTableNorm) <- c("y","x")
colnames(resTableNorm) <- c("Bias","CI Width"," Coverage")
resTableNorm
Beta <- c(0,1,-1)
for(i in 1:nsims) {
  CARdata <- genCAR(rho)
  #paste here
  temp <- ampute(CARdata,prop=mis.prop,patterns = c(1,0,1),weights=c(0,0,1)
  ,type = "RIGHT",mech = "MAR")

  while(sum(is.na(temp$amp$y))!=floor(mis.prop*n)) {
    temp <- ampute(CARdata,prop=mis.prop,patterns = c(1,0,1),weights=c(0,0,1)
    ,type = "RIGHT",mech = "MAR")
  }
  #
  y.copy <- CARdata$y
  CARdata$y <- temp$amp$y
  ypmm <- mice(CARdata,method=c("","pmm",""),
  predictorMatrix=impMat,maxit=30,m=5,printFlag = FALSE)
  milm <- with(ypmm,lm(z~y+x+0))
  results[[i]] <- summary(pool(milm))
}
Beta <- c(1,-1)
resTablepmm <- makeTable(1)
resTablepmm <- rbind(resTablepmm,makeTable(2))

row.names(resTablepmm) <- c("y","x")
colnames(resTablepmm) <- c("Bias","CI Width"," Coverage")
```

```
resTablepmm
##
```

D.3 SLM Simulations

Code Block D.3.1: SLM Setup Code

```
Beta <- c(2,1,-1)
#SLM

genSLM <- function(rho) {
  x <- rnorm(n,0,1)
  err <- rnorm(n,0,1)

  y <- 2*x + err +3
  y <- solve(diag(rep(1,n)) - rho * W) %*% matrix(y,ncol=1)

  #add top level z
  z <- Beta[2]*y+Beta[3]*x+Beta[1]+rnorm(n,0,0.25)
  return(data.frame(z=z,y=y,x=x))
}

SLMdata <- genSLM(rho)

#Amputation- once again ,copy and paste into loop
#MCAR
#ind <- sample(n,size=floor(mis.prop*n))
#SLMdata$y[ind] <- NA
#MAR
#temp <- ampute(SLMdata,prop=mis.prop,patterns = c(1,0,1),weights=c(0,0,1)
#                ,type = "RIGHT",mech = "MAR")
#while(sum(is.na(temp$amp$y))!=floor(mis.prop*n)) {
  #  temp <- ampute(SLMdata,prop=mis.prop,patterns = c(1,0,1),weights=c(0,0,1)
  #                  ,type = "RIGHT",mech = "MAR")
  #}
#SLMdata$y <- temp$amp$y
#MNAR
#temp <- ampute(SLMdata,prop=mis.prop,patterns = c(1,0,1),weights=c(1,1,0)
#                ,type = "RIGHT",mech="MNAR")
```

```
#while(sum(is.na(temp$amp$y))!=floor(mis.prop*n)) {
  #  temp <- ampute(SLMdata,prop=mis.prop,patterns = c(1,0,1),weights=c(1,1,0)
  #                  ,type = "RIGHT",mech="MNAR")
  #}
#SLMdata$y <- temp$amp$y
#

#Imputation
mice.impute.SLMfill <- function(y,ry,x,wy=NULL,...) {

  if(is.null(wy)) {
    wy <- !ry
  }

  if(any(is.na(y)) || any(is.na(x))) {
    temp <- multifill(data.frame(y=y,x=x),neigh)
    x <- temp[,-1]
    y <- temp[,1]

  }



  x <- as.data.frame(x)
  x <- matrix(unlist(x),ncol=ncol(x))
  x <- cbind(1,x)

  B.func <- function(rho,x,y) {
    solve(t(x)%*%x)%*%t(x)%*%(diag(1,n)-rho*W)%*%y
  }

  error <- function(rho,beta,x,y) {
    (diag(1,n)-rho*W)%*%y - x%*%beta
  }

  rho.post <- runif(1)
  B.post <- B.func(rho.post,x,y)
  e <- error(rho.post,B.post,x,y)
  #Equation 2.4
  B.post <- mvrnorm(n=1,mu=B.func(rho.post,x,y),
  Sigma=as.numeric(mean(e^2))*solve(t(x)%*%x))
  #Equation 2.3
```

```
sigma.log <- function(sig) {
  e <- error(rho.post,B.post,x,y)
  value <- -(n+1)*log(sig)
  value <- value -1/(2*sig^2)*t(e)%*%e
  return(value)
}

sigma.post <- arms(10,sigma.log,lower=0,upper=10*var(y))
sigma.post <- sigma.post[10]
#Equation 2.5
rho.log <- function(rho) {
  e <- error(rho,B.post,x,y)
  value <- log(det(diag(1,n)-rho*W))-(n+1)*log(sigma.post)
  value <- value -1/(2*sigma.post^2)*t(e)%*%e
  return(value)
}

rho.post <- arms(1,rho.log,0,1)

#gibbs sampler needs to converge to target dist. of parameters
n_sims <- 10
for(i in 1:n_sims) {
  sigma.post <- arms(1,sigma.log,lower=0,upper=10*var(y))
  e <- error(rho.post,B.post,x,y)
  B.post <- mvrnorm(n=1,mu=B.func(rho.post,x,y),
  Sigma=as.numeric(var(e))*solve(t(x)%*%x))
  rho.post <- arms(1,rho.log,0,1)
}
#print(paste("Final estimates: rho: ",rho.post,", sigma: ",
#            sigma.post," Beta: ",B.post[1]," ",B.post[2])) #debugging
#Equation 3.1 and A.1
Sig <- solve((diag(1,n)-rho.post*W))
mu <- Sig%*%x%*%B.post
Sigma <- sigma.post^2*Sig%*%t(Sig)
obs <- which(ry)
mis <- which(wy)
Sig.oo <- Sigma[obs,obs]
Sig.om <- Sigma[obs,mis]
Sig.mm <- Sigma[mis,mis]
Sig.mo <- Sigma[mis,obs]
```

```
  mu.con <- mu[mis] + Sig.mo %*% solve(Sig.oo) %*% matrix(y[obs]-mu[obs],ncol
      =1)
  Sig.con <- Sig.mm - Sig.mo %*% solve(Sig.oo) %*% Sig.om
  imputes <- mvrnorm(n=1,mu=mu.con,Sigma=Sig.con)
  return(imputes)
}
#
```

Code Block D.3.2: SLM Simulation Code

```
#repeated MICE runs
#nsims <- 200
results <- list()
#imputation matrix and method
impMat <- diag(0,nrow=3)
impMat[2,3] <- 1
meth <- c("","SLMfill","")
#
for(i in 1:nsims) {
  print(paste("Iteration: ",i))
  SLMdata <- genSLM(rho)
  #paste here
  temp <- ampute(SLMdata,prop=mis.prop,patterns = c(1,0,1),weights=c(0,0,1)
  ,type = "RIGHT",mech="MAR")
  while(sum(is.na(temp$amp$y))!=floor(mis.prop*n)) {
    temp <- ampute(SLMdata,prop=mis.prop,patterns = c(1,0,1),weights=c(0,0,1)
    ,type = "RIGHT",mech="MAR")
  }
  #
  y.copy <- SLMdata$y
  SLMdata$y <- temp$amp$y
  ySLM <- mice(SLMdata,method=meth,predictorMatrix=impMat,maxit=10,m=5)
  milm <- with(ySLM,lm(z~y+x))
  results[[i]] <- summary(pool(milm))
}

##Analysis Table
#Bias, CI width, Coverage

makeTable <- function(var) {
  bias <- 0
  CI.w <- 0
```

```
  coverage <- 0

  for(i in 1:nsims) {
    bias <- bias+(results[[i]]$estimate[var]-Beta[var])
    tval <- qt(0.975,results[[i]]$df[var])
    ll <- results[[i]]$estimate[var] - tval*results[[i]]$std.error[var]
    ul <- results[[i]]$estimate[var] + tval*results[[i]]$std.error[var]
    CI.w <- ul-ll
    if(ll<Beta[var] && ul>Beta[var]) {
      coverage <- coverage+1
    }
  }
  table <- data.frame("Bias"=bias,"CI Width" = CI.w, "Coverage"=coverage)
  table <- 1/nsims * table
  return(table)
}

resTableSLM <- makeTable(1)
resTableSLM <- rbind(resTableSLM,makeTable(2))
resTableSLM <- rbind(resTableSLM,makeTable(3))
row.names(resTableSLM) <- c("Intercept","y","x")
colnames(resTableSLM) <- c("Bias","CI Width"," Coverage")
resTableSLM
##

## Comparison
for(i in 1:nsims) {
  SLMdata <- genSLM(rho)
  #paste here
  temp <- ampute(SLMdata,prop=mis.prop,patterns = c(1,0,1),weights=c(1,1,0)
  ,type = "RIGHT",mech="MNAR")
  while(sum(is.na(temp$amp$y))!=floor(mis.prop*n)) {
    temp <- ampute(SLMdata,prop=mis.prop,patterns = c(1,0,1),weights=c(1,1,0)
    ,type = "RIGHT",mech="MNAR")
  }
  #
  y.copy <- SLMdata$y
  SLMdata$y <- temp$amp$y
  ynorm <- mice(SLMdata,method=c("","norm",""),
  predictorMatrix=impMat,maxit=10,m=5,printFlag = FALSE)
  milm <- with(ynorm,lm(z~y+x))
```

```
  results[[i]] <- summary(pool(milm))
}

resTableNorm <- makeTable(1)
resTableNorm <- rbind(resTableNorm,makeTable(2))
resTableNorm <- rbind(resTableNorm,makeTable(3))

row.names(resTableNorm) <- c("Intercept","y","x")
colnames(resTableNorm) <- c("Bias","CI Width"," Coverage")
resTableNorm

for(i in 1:nsims) {
  SLMdata <- genSLM(rho)
  #paste here
  temp <- ampute(SLMdata,prop=mis.prop,patterns = c(1,0,1),weights=c(0,0,1)
  ,type = "RIGHT",mech="MAR")
  while(sum(is.na(temp$amp$y))!=floor(mis.prop*n)) {
    temp <- ampute(SLMdata,prop=mis.prop,patterns = c(1,0,1),weights=c(0,0,1)
    ,type = "RIGHT",mech="MAR")
  }
  #
  SLMdata$y <- temp$amp$y
  ypmm <- mice(SLMdata,method=c("","pmm",""),
  predictorMatrix=impMat,maxit=10,m=5,printFlag = FALSE)
  milm <- with(ypmm,lm(z~y+x))
  results[[i]] <- summary(pool(milm))
}

resTablepmm <- makeTable(1)
resTablepmm <- rbind(resTablepmm,makeTable(2))
resTablepmm <- rbind(resTablepmm,makeTable(3))

row.names(resTablepmm) <- c("Intercept","y","x")
colnames(resTablepmm) <- c("Bias","CI Width"," Coverage")
resTablepmm
##
```

D.4 SEM Simulations

Code Block D.4.1: SEM Setup Code

```
#SEM
lambda <- rho.set[3] #choose lambda
genSEM <- function(lambda) {
  x <- rnorm(n,0,1)
  err <- rnorm(n,0,1)
  err <- matrix(err,ncol=1)
  err <- solve(diag(1,n)-lambda*W)%*%err
  y <- 2*x + err +3

  z <- Beta[2]*y+Beta[3]*x+Beta[1]+rnorm(n,0,0.25)
  return(data.frame(z=z,y=y,x=x))
}

#Amputation - once again copy and paste
#MCAR
#ind <- sample(n,size=floor(mis.prop*n))
#SEMdata$y[ind] <- NA
#MAR
#temp <- ampute(SEMdata,prop=mis.prop,patterns = c(1,0,1),weights=c(1,0,1)
#               ,type = "RIGHT",mech = "MAR")
#while(sum(is.na(temp$amp$y))!=floor(mis.prop*n)) {
  #  temp <- ampute(SEMdata,prop=mis.prop,patterns = c(1,0,1),weights=c(1,0,1)
  #                 ,type = "RIGHT",mech = "MAR")
  #}
#SEMdata$y <- temp$amp$y
#MNAR
#temp <- ampute(SEMdata,prop=mis.prop,patterns = c(1,0,1),weights=c(1,1,0)
#               ,type = "RIGHT",mech="MNAR")
#while(sum(is.na(temp$amp$y))!=floor(mis.prop*n)) {
  #  temp <- ampute(SEMdata,prop=mis.prop,patterns = c(1,0,1),weights=c(1,1,0)
  #                 ,type = "RIGHT",mech="MNAR")
  #}
#SEMdata$y <- temp$amp$y
#

#Imputation
mice.impute.SEMfill <- function(y,ry,x,wy=NULL,...) {

  if(is.null(wy)) {
    wy <- !ry
```

```
}

if(any(is.na(y)) || any(is.na(x))) {
  temp <- multifill(data.frame(y=y,x=x),neigh)
  x <- temp[,-1]
  y <- temp[,1]

}

x <- as.data.frame(x)
x <- matrix(unlist(x),ncol=ncol(x))
x <- cbind(1,x)

B.func <- function(lam,x,y) {
  B <- diag(1,n)-lam*W
  solve(t(x)%*%t(B)%*%B%*%x)%*%(t(x)%*%t(B)%*%B%*%y)
}

error <- function(lam,beta,x,y) {
  (diag(1,n)-lam*W)%*%(y-x%*%beta)
}

lambda.post <- runif(1)
B.post <- B.func(lambda.post,x,y)
e <- error(lambda.post,B.post,x,y)
B <- diag(1,n)-lambda.post*W
varSEM <- as.numeric(var(e))*solve(t(x)%*%t(B)%*%B%*%x)
#Equation 2.7
B.post <- mvrnorm(n=1,mu=B.func(lambda.post,x,y),Sigma=varSEM)
#Equation 2.8
sigma.log <- function(sig) {
  e <- error(lambda.post,B.post,x,y)
  return(-(n+1)*log(sig)-1/(2*sig^2)*t(e)%*%e)
}
sigma.post <- arms(1,sigma.log,0,10*var(y))
#Equation 2.9
lambda.log <- function(lambda) {
  e <- error(lambda,B.post,x,y)
  value <- log(det(diag(1,n)-lambda*W))-(n+1)*log(sigma.post)
  value <- value - 1/(2*sigma.post^2)*t(e)%*%e
  return(value)
```

```
  }
  lambda.post <- arms(1,lambda.log,0,1)

  n_sims <- 10 #within gibbs sampling must converge
  for(i in 1:n_sims) {
    sigma.post <- arms(1,sigma.log,lower=0,upper=10*var(y))#3 params
    e <- error(lambda.post,B.post,x,y)
    B <- diag(1,n)-lambda*W
    varSEM <- as.numeric(var(e))*solve(t(x)%*%t(B)%*%B%*%x)
    B.post <- mvrnorm(n=1,mu=B.func(lambda.post,x,y),Sigma=varSEM)
    lambda.post <- arms(1,lambda.log,0,1)
  }
  #print(paste("Optimal Estimates: lambda: ",lambda.post," sigma: ",sigma.post
      ,
  #             " Beta: ",B.post[1]," ",B.post[2])) #debugging

  Sig <- solve(diag(1,n)-lambda.post*W)
  #Equation 3.2 and A.1
  mu <- x%*%B.post
  Sigma <- sigma.post^2*Sig%*%t(Sig)
  obs <- which(ry)
  mis <- which(wy)
  Sig.oo <- Sigma[obs,obs]
  Sig.om <- Sigma[obs,mis]
  Sig.mm <- Sigma[mis,mis]
  Sig.mo <- Sigma[mis,obs]

  mu.con <- mu[mis] + Sig.mo %*% solve(Sig.oo) %*% matrix(y[obs]-mu[obs],ncol
      =1)
  Sig.con <- Sig.mm - Sig.mo %*% solve(Sig.oo) %*% Sig.om
  imputes <- mvrnorm(n=1,mu=mu.con,Sigma=Sig.con)
  return(imputes)
}
#
```

Code Block D.4.2: SEM Simulation Code

```
#repeated MICE runs
results <- list()
#imputation matrix and method
impMat <- diag(0,nrow=3)
impMat[2,3] <- 1
```

```
meth <- c("","SEMfill","")
#
for(i in 1:nsims) {
  print(paste("Iteration: ",i))
  SEMdata <- genSEM(lambda)
  #paste here
  temp <- ampute(SEMdata,prop=mis.prop,patterns = c(1,0,1),weights=c(0,0,1)
  ,type = "RIGHT",mech = "MAR")
  while(sum(is.na(temp$amp$y))!=floor(mis.prop*n)) {
    temp <- ampute(SEMdata,prop=mis.prop,patterns = c(1,0,1),weights=c(0,0,1)
    ,type = "RIGHT",mech = "MAR")
  }
  #
  SEMdata$y <- temp$amp$y
  ySEM <- mice(SEMdata,method=meth,predictorMatrix=impMat,maxit=10,m=5)
  milm <- with(ySEM,lm(z~y+x))
  results[[i]] <- summary(pool(milm))
}

##Analysis Table
#Bias, CI width, Coverage

makeTable <- function(var) {
  bias <- 0
  CI.w <- 0
  coverage <- 0

  for(i in 1:nsims) {
    bias <- bias+(results[[i]]$estimate[var]-Beta[var])
    tval <- qt(0.975,results[[i]]$df[var])
    ll <- results[[i]]$estimate[var] - tval*results[[i]]$std.error[var]
    ul <- results[[i]]$estimate[var] + tval*results[[i]]$std.error[var]
    CI.w <- ul-ll
    if(ll<Beta[var] && ul>Beta[var]) {
      coverage <- coverage+1
    }
  }
  table <- data.frame("Bias"=bias,"CI Width" = CI.w, "Coverage"=coverage)
  table <- 1/nsims * table
  return(table)
}
```

```
resTableSEM <- makeTable(1)
resTableSEM <- rbind(resTableSEM,makeTable(2))
resTableSEM <- rbind(resTableSEM,makeTable(3))
row.names(resTableSEM) <- c("Intercept","y","x")
colnames(resTableSEM) <- c("Bias","CI Width"," Coverage")
resTableSEM
##

## Comparison
for(i in 1:nsims) {
  SEMdata <- genSEM(lambda)
  #paste here
  temp <- ampute(SEMdata,prop=mis.prop,patterns = c(1,0,1),weights=c(1,1,0)
  ,type = "RIGHT",mech = "MNAR")
  while(sum(is.na(temp$amp$y))!=floor(mis.prop*n)) {
    temp <- ampute(SEMdata,prop=mis.prop,patterns = c(1,0,1),weights=c(1,1,0)
    ,type = "RIGHT",mech = "MNAR")
  }
  #
  SEMdata$y <- temp$amp$y
  ynorm <- mice(SEMdata,method=c("","norm",""),
  predictorMatrix=impMat,maxit=10,m=5,printFlag = FALSE)
  milm <- with(ynorm,lm(z~y+x))
  results[[i]] <- summary(pool(milm))
}

resTableNorm <- makeTable(1)
resTableNorm <- rbind(resTableNorm,makeTable(2))
resTableNorm <- rbind(resTableNorm,makeTable(3))

row.names(resTableNorm) <- c("Intercept","y","x")
colnames(resTableNorm) <- c("Bias","CI Width"," Coverage")
resTableNorm

for(i in 1:nsims) {
  SEMdata <- genSEM(lambda)
  #paste here
  temp <- ampute(SEMdata,prop=mis.prop,patterns = c(1,0,1),weights=c(1,1,0)
  ,type = "RIGHT",mech = "MNAR")
  while(sum(is.na(temp$amp$y))!=floor(mis.prop*n)) {
```

```
    temp <- ampute(SEMdata,prop=mis.prop,patterns = c(1,0,1),weights=c(1,1,0)
    ,type = "RIGHT",mech = "MNAR")
  }
  #
  SEMdata$y <- temp$amp$y
  ypmm <- mice(SEMdata,method=c("","pmm",""),
  predictorMatrix=impMat,maxit=10,m=5,printFlag = FALSE)
  milm <- with(ypmm,lm(z~y+x))
  results[[i]] <- summary(pool(milm))
}

resTablepmm <- makeTable(1)
resTablepmm <- rbind(resTablepmm,makeTable(2))
resTablepmm <- rbind(resTablepmm,makeTable(3))

row.names(resTablepmm) <- c("Intercept","y","x")
colnames(resTablepmm) <- c("Bias","CI Width"," Coverage")
resTablepmm
##
```

D.5 Boston Data

Code Block D.5.1: Boston Modelling Code

```
###Boston Data
# https://www.paulamoraga.com/book-spatial/bayesian-spatial-models.html -
# description and other analysis
# Corrected median house value in 1000's of USD - CMEDV
# Rooms per dwelling - RM
# Crime per capita in area - CRIM

bostonFit <- as.data.frame(bostonData[,c("CMEDV","CRIM","RM")])
bostonFit <- bostonFit[,-4]
#plot(bostonFit)
bostonFit <- cbind(bostonFit,logMEDV = log(bostonFit$CMEDV))
temp <- lm(logMEDV~CRIM+RM,data = bostonFit)
summary(temp)

moran.test(temp$residuals,nb2listw(neigh))
```

```
temp <- lm(CRIM~RM,data=bostonFit)
moran.test(temp$residuals,nb2listw(neigh))

temp.err <- errorsarlm(CRIM~RM,data = bostonFit,listw = nb2listw(neigh))
#summary(temp.err)

temp.lag <- lagsarlm(CRIM~RM,data = bostonFit,listw = nb2listw(neigh))
#summary(temp.lag)

mean(sum((temp.lag$residuals)^2))
mean(sum((temp.err$residuals)^2))

moran.test(temp.lag$residuals,listw=nb2listw(neigh))

mod.lag <- lagsarlm(logMEDV~CRIM+RM,data = bostonFit,listw = nb2listw(neigh))
#summary(mod.lag)

mod.err <- errorsarlm(logMEDV~CRIM+RM,data = bostonFit,listw = nb2listw(neigh)
    )
#summary(mod.err)

mean(sum((mod.lag$residuals)^2))
mean(sum((mod.err$residuals)^2))
AIC(mod.lag)
AIC(mod.err)

#SEM is better top-level model

Beta.mod <- mod.err$coefficients
B.var.mod <- mod.err$rest.se
lambda.mod <- mod.err$lambda
lambda.se.mod <- mod.err$lambda.se

##100 run sim

makeBostonTable <- function() {
  bias <- rep(0,4)
  CI.w <- rep(0,4)
  coverage <- rep(0,4)
  parms <- c(Beta.mod[1:3],lambda.mod)
  for(i in 1:nsims) {
```

```
    bias <- bias + (results[[i]]$estimate-parms)
    for(j in 1:4) {
      tval <- qt(0.975,results[[i]]$df[j])
      ll <- results[[i]]$estimate[j] - tval*results[[i]]$std.error[j]
      ul <- results[[i]]$estimate[j] + tval*results[[i]]$std.error[j]
      CI.w[j] <- ul-ll
      if(ll<parms[j] && ul>parms[j]) {
        coverage[j] <- coverage[j]+1
      }
    }
  }
  table <- data.frame("Bias"=bias,"CI Width" = CI.w, "Coverage"=coverage)
  table <- 1/nsims * table
  return(table)
}
#repeat for methods norm and pmm, then for MNAR with weights = c(1,0,1)
impMat <- diag(0,nrow=3)
impMat[1,2] <- 1
meth <- c("SLMfill","","")
results <- list()
for(i in 1:nsims) {
  bostonFit <- as.data.frame(bostonData[,c("CMEDV","CRIM","RM")])
  bostonFit <- bostonFit[,-4]
  bostonFit <- cbind(bostonFit,logMEDV = log(bostonFit$CMEDV))
  bostonFit <- bostonFit[,-1]
  temp <- ampute(bostonFit,prop=mis.prop,patterns = c(0,1,1),weights=c(0,1,0)
  ,type = "RIGHT",mech = "MAR")
  while(sum(is.na(temp$amp$CRIM))!=floor(mis.prop*nrow(bostonFit))) {
    temp <- ampute(bostonFit,prop=mis.prop,patterns = c(0,1,1),weights=c
        (0,1,0)
    ,type = "RIGHT",mech = "MAR")
  }
  bostonFit$CRIM <- temp$amp$CRIM
  ySLM <- mice(bostonFit,method=meth,predictorMatrix=impMat,maxit=10,m=5)

  mierrorsar <- with(ySLM,errorsarlm(logMEDV~CRIM+RM,listw = nb2listw(neigh)))
  results[[i]] <- summary(pool(mierrorsar))

}
```

D.6 Miscellaneous Code

Code Block D.6.1: Miscellaneous Plot Code

```
##plot code
ind <- which(!is.na(CARdata$y))
plot(density(CARdata$y[ind]),lwd=2,main="",cex.lab=2,cex.axis=2,col="black")
ind <- which(is.na(CARdata$y))
lines(density(y.copy[ind]),col="blue",lwd=2)
lines(density(yCAR$imp$y$`1`),col="red",lwd=2)
lines(density(yCAR$imp$y$`2`),col="red",lwd=2)
lines(density(yCAR$imp$y$`3`),col="red",lwd=2)
lines(density(yCAR$imp$y$`4`),col="red",lwd=2)
lines(density(yCAR$imp$y$`5`),col="red",lwd=2)
legend(-4,0.2,legend=c("Observed","Missing","Imputed"),
col=c("black","blue","red"),lwd=2,lty=1,bty = "n",cex=1.2)

ind <- which(!is.na(SLMdata$y))
plot(density(SLMdata$y[ind]),lwd=2,main="",cex.lab=2,cex.axis=2,col="black")
ind <- which(is.na(SLMdata$y))
lines(density(y.copy[ind]),col="blue",lwd=2)
lines(density(ySLM$imp$y$`1`),col="red",lwd=2)
lines(density(ySLM$imp$y$`2`),col="red",lwd=2)
lines(density(ySLM$imp$y$`3`),col="red",lwd=2)
lines(density(ySLM$imp$y$`4`),col="red",lwd=2)
lines(density(ySLM$imp$y$`5`),col="red",lwd=2)
legend(-4,0.2,legend=c("Observed","Missing","Imputed"),
col=c("black","blue","red"),lwd=2,lty=1,bty = "n",cex=1.2)

ind <- which(!is.na(SEMdata$y))
plot(density(SEMdata$y[ind]),lwd=2,main="",cex.lab=2,cex.axis=2,col="black")
ind <- which(is.na(SEMdata$y))
lines(density(y.copy[ind]),col="blue",lwd=2)
lines(density(ySEM$imp$y$`1`),col="red",lwd=2)
lines(density(ySEM$imp$y$`2`),col="red",lwd=2)
lines(density(ySEM$imp$y$`3`),col="red",lwd=2)
lines(density(ySEM$imp$y$`4`),col="red",lwd=2)
lines(density(ySEM$imp$y$`5`),col="red",lwd=2)
legend(-4,0.2,legend=c("Observed","Missing","Imputed"),
col=c("black","blue","red"),lwd=2,lty=1,bty = "n",cex=1.2)
##
```

D.7 Investigation of MCAR and MAR

Code Block D.7.1: Miscellaneous Plot Code

```
set.seed(2305614)
mis.prop <- 0.3
SLMdata <- genSLM(rho)
temp <- ampute(SLMdata,prop=mis.prop,patterns = c(1,0,1),weights=c(0,0,1)
,type = "RIGHT",mech = "MAR")
#actually cant test with only one missing pattern (one var)
#mcar(temp$amp,method = "auto")
#mcar(temp$amp,method= "hawkins")
#mcar(temp$amp,method="nonparametric")

lm.cc <- lm(z~y+x,data=temp$amp)
SLMdata <- temp$amp
impMat <- diag(0,nrow=3)
impMat[2,3] <- 1
yslm <- mice(SLMdata,method=c("","SLMfill",""),
predictorMatrix=impMat,maxit=10,m=5,printFlag = FALSE)
milm <- with(yslm,lm(z~y+x))
summary(lm.cc)
summary(pool(milm))
```