UHASSELT
KNOWLEDGE IN ACTION

Maastricht University

Faculty of Sciences
School for Information Technology

Master of Statistics and Data Science

Master's thesis

Modelling Geomasked Spatial Data: Evaluation of Methods

Roel Jude Bagaforo

Thesis presented in fulfillment of the requirements for the degree of Master of Statistics and Data Science, specialization Quantitative Epidemiology

SUPERVISOR :

Prof. dr. Thomas NEYENS

2023
2024

Transnational University Limburg is a unique collaboration of two universities in two countries: the University of Hasselt and Maastricht University.

UHASSELT
KNOWLEDGE IN ACTION

www.uhasselt.be
Universiteit Hasselt
Campus Hasselt:
Martelarenlaan 42 | 3500 Hasselt
Campus Diepenbeek:
Agoralaan Gebouw D | 3590 Diepenbeek

Maastricht University

Faculty of Sciences
School for Information Technology
Master of Statistics and Data Science

Master's thesis

Modelling Geomasked Spatial Data: Evaluation of Methods

Roel Jude Bagaforo

Thesis presented in fulfillment of the requirements for the degree of Master of Statistics and Data Science, specialization Quantitative Epidemiology

SUPERVISOR :

Prof. dr. Thomas NEYENS

Contents

List of Tables

List of Figures

Abstract

In geostatistical data involving human subjects, privacy is a major concern. Various techniques, such as randomly perturbing the locations, known as geomasking, have been used to protect individual privacy while maintaining substantial spatial associations. However, geomasking has been shown to result in biased estimates and poor spatial predictions. In response, several methods have been proposed to account for positional uncertainty in a geostastitical model. In this thesis, we reviewed the methods that address geomasking and focused on evaluating the comprehensive and user-friendly `R` package, `GeoAdjust`. Through a case study and simulation studies, we demonstrated its performance in various scenarios.

We applied `GeoAdjust` to analyze the geomasked Philippines Demographic and Health Survey and found some interesting results. Subsequently, we subjected `GeoAdjust` to a simulation study. We simulated different data under multiple scenarios, such as varying sample sizes, different displacement radii, and combinations of different magnitudes of spatial and nugget variance. We used parameter estimate bias and predictive performance for comparison. The results indicated that `GeoAdjust` had the best performance in terms of covariance parameter bias when the sample size and displacement were large, and the spatial variance was relatively larger than the nugget variance. Similarly, predictions were found to be the best in these scenarios, particularly in terms of the uncertainties associated with the predictions.

Based on these results, we provided guidance and careful considerations for the usage of R package, `GeoAdjust`. Our evaluation of the method offers insights into the impacts of geomasking which are useful for future applications and development of new methods.

Chapter 1

Introduction

In recent decades, the integration of geographic information systems (GIS) and advanced spatial statistical techniques has facilitated more accurate mapping of disease distributions, identification of environmental risk factors, representation of economic activity, quantification of species abundance, and many other applications (Pfeiffer et al., 2008, Redding and Rossi-Hansberg, 2017, White et al., 2010). This integration has been pivotal in addressing complex spatial phenomena and supporting evidence-based decision-making across disciplines. Furthermore, there has been an exponential growth in spatial data, particularly at smaller resolutions, such as *geostatistical data* (Gelfand and Banerjee, 2017). For example, GPS geo-tagged data collected by government agencies, social media companies, and academic institutions are now widely available.

However, while it is crucial for geostatistical data to be available and accessible to researchers, analysts, policymakers, and the general public, it is equally important to maintain the privacy of the subjects in the data (Burgert et al., 2013). Several methods have been developed to protect the privacy of subjects in geostatistical data, with the most common being geographical masking through random perturbation or *geomasking*. Geomasking involves intentionally repositioning the original locations within a certain region around the original point (Armstrong et al., 1999). One popular method of geomasking is displacing the points within a circular region with equal probabilities, known as *uniform geomasking*. In this thesis, we focused on this type of geomasking. Uniform geomasking has been used in various applications such as in health surveys (Burgert et al., 2013), social media data like Twitter (Gao et al., 2019), and administrative data like 911 emergency phone calls (Allshouse et al., 2010).

Standard methods for geostatistical analysis, such as methods under model-based geostatistics (Diggle et al., 1998), often assume that the locations are collected accurately (Diggle et al., 2003, Diggle and Giorgi, 2019). With geomasking, location errors are introduced intentionally by displacing the true locations, which propagates through the analysis. Multiple studies have demonstrated that such location errors generally lead to biased model parameter estimates and poor spatial predictions (Gabrosek and Cressie, 2002, Jacquez, 2012, Goldberg and Cockburn, 2012, Kinnee et al., 2020). It has been theoretically illustrated that uniform geomasking significantly disrupts the spatial structure of the data, particularly affecting the covariance

parameters, which are inherently based on the distances between the locations (Fronterrè et al., 2018). This disruption can severely compromise the integrity of spatial analyses, making it challenging to draw accurate conclusions. The problem is further compounded as the degree of displacement increases, leading to more significant distortion of spatial relationships in the data. Additionally, factors such as sample sizes and the amount of variation may also contribute to this issue. While geomasking is crucial for protecting privacy, it poses a substantial challenge for maintaining the utility of spatial data.

A number of researchers have proposed methods to address general location errors in spatial data analysis (Gabrosek and Cressie, 2002, Cressie and Kornak, 2003). However, only a few have directly tackled the specific problem of geomasking. The first to propose an approach within a model-based geostatistics framework was Fanshawe and Diggle (2011). Their method, while pioneering, was significantly hampered by its enormous computational burden, which prevented further explorations and practical applications. A few years later, Fronterrè et al. (2018) addressed these limitations by refining the approach of Fanshawe and Diggle (2011). They developed methods that were more computationally feasible while still providing accurate spatial analyses. In a Bayesian paradigm, Wilson and Wakefield (2021) proposed a method for addressing geomasking. However, similar to the approach by Fanshawe and Diggle (2011), their method was also very slow. Recently, Altay, Paige, Riebler, and Fuglstad (2023) improved upon the method of Wilson and Wakefield (2021) and developed an `R` package, `GeoAdjust`, to enhance its computational efficiency and usability.

With this background, we reviewed the current methods available in the literature that address geomasking, focusing on the recent `R` package, `GeoAdjust`. We evaluated the performance of `GeoAdjust` using both real and simulated data. Specifically, it was used to analyze the geomasked Philippines Demographic and Health Survey (DHS). However, due to the absence of true data, we could not fully confirm the accuracy of the model estimates and the predictions. To address this, we conducted a simulation study, exploring various simulated data scenarios under different conditions. The developers of `GeoAdjust` previously conducted a limited simulation study that focused solely on the impact of the displacement radius, leaving many aspects of `GeoAdjust`'s performance yet to be investigated. For instance, it was still unclear how the number of sample locations or the magnitude of spatial variation in the data affects its performance. Understanding these factors is crucial because location errors propagated through geomasking can lead to a loss of information, potentially requiring specific scenarios for optimal usability. Our analyses aim to provide comprehensive guidance on the effective usage of `GeoAdjust`. The results offer insights into the strengths and weaknesses of this approach, highlighting areas for potential improvement and future research.

We organized this thesis into several chapters. We began with the standard inferences under model-based geostatistics which served as the foundation for the subsequent chapters (Chapter 2). In Chapter 3, geomasking was introduced and its effects were explored particularly on how it complicates the inferences along with a review and presentation of all available methods for making inferences with a geomasked spatial data. We then presented the case study and the

simulation study, covering motivations, methodologies employed, and key findings (Chapters 4, 5). Finally, we concluded this thesis with a discussion of the results and its implications, and potential extensions of our research (Chapter 6).

1.1 Relevance, Stakeholders, and Ethics

Many spatial datasets contain crucial information relevant to various fields. For instance, the dataset from the Philippines Demographic and Health Survey (DHS) includes information on household income, HIV status, violence against women and children, and many more. Due to the sensitive nature of these information, methods like geomasking have been developed and used multiple times to safeguard the spatial confidentiality of these data. However, information from sources like the Philippines DHS also play a crucial role in analyzing spatial trends for more effective programs and policies. Therefore, balancing privacy protection and information preservation is of prime importance.

In this thesis, our goal was to the review available methods that address geomasking, with a focus on `GeoAdjust`, aiming to provide insights to future researchers of geomasked data and, on the other hand, to data owners who perform geomasking in spatial data. Our results revealed certain scenarios that were optimal for the application of `GeoAdjust` on geomasked data. We found that `GeoAdjust` had smaller parameter bias and better spatial predictions when the geomasked data had larger displacement, more sample locations, and greater spatial variation. These findings provide valuable guidance for future users of `GeoAdjust` and researchers interested in improving and developing new methods. Alternatively, data owners, particularly privacy and ethical committees, can also use our results when applying geomasking to spatial datasets. For example, we have highlighted the necessity for larger sample locations and spatial variation. As such, when dealing with spatial data that needs to be geomasked, it might be beneficial to use smaller displacement whenever possible to limit the destruction of spatial associations, especially when working with smaller sample sizes and spatial variation.

In terms of the ethical standards, we analyzed a national household survey with a large number of respondents. All necessary documentary requirements were completed to access the datasets. The DHS program implemented essential data privacy measures, such as anonymization, before releasing the data to the public and provided guidelines on its use. We fully adhered to these guidelines and followed the intended purpose of the data. The data were used solely for academic purposes. Additionally, we ensured that ethical standards were maintained throughout the research process. This approach not only respected the privacy of the respondents but also upheld the integrity of the research. Our adherence to these protocols ensured the reliability and ethical soundness of our findings.

In general, this thesis contributes to the growing body of knowledge on spatial data privacy and aims to support researchers, analysts, and policymakers in making informed decisions when handling geomasked data.

Chapter 2

Overview of Model-based Geostatistics Methods

Suppose measurements, $y(x_1)$, $y(x_2)$, ..., $y(x_n)$, of a variable Y from a finite set of different locations, x_1, x_2,..., x_n, over a study region A are obtained. It is often assumed that these measurements are partial realizations of an underlying random process, $S(x)$, continuously spanning the region A. This set of measurements is commonly referred as geostatistical data. Based on this data, it is of interest to make a generalization of the characteristics of the process of the variable of study and to predict the measurements at unsampled locations.

In model-based geostatistics, a stochastic model is constructed to depict the underlying process $S(x)$. The most common and tractable way to model it is by assuming that $S(x)$ is a Gaussian process or specifically as a Gaussian random field (GRF). A Gaussian process is defined such that the joint probability distribution of the variable measurements sampled over the finite set of locations follows a multivariate normal distribution. In addition, it is often assumed that the covariance matrix of the multivariate normal distribution is stationary and isotropic. That is, the variance is assumed constant and the correlation between $S(x)$ and $S(x')$ only depends on the distance between x and x'. This correlation is defined by a symmetric correlation function $\rho(d)$, where d is the distance. Most common correlation functions are under the Matérn family.

The Matérn correlation function is defined as follows:

$$\rho(d;\phi,\kappa) = 2^{\kappa-1}\Gamma(\kappa)^{-1}(d/\phi)^{\kappa}K_{\kappa}(d/\phi) \tag{2.1}$$

where $\phi > 0$ and $\kappa > 0$ are the parameters and $K_{\kappa}(\cdot)$ is a modified Bessel function. The parameter ϕ controls the rate on how quickly the spatial correlation decays to zero as spatial distance increases while the parameter κ is a smoothness parameter that is related to the number of times the function is differentiable. A special case of the Matérn correlation function is the exponential correlation function, when $\kappa = 0.5$. In this thesis, the majority of the assumed correlation functions were of this type. The exponential correlation function is given by:

$$\rho(d;\phi) = exp(-d/\phi) : d \geq 0 \tag{2.2}$$

In the following sections, we outlined an overview of the methods for model-based statistics, starting with the likelihood-based inference and followed by Bayesian inference. These theories formed the foundation of the methods discussed in the subsequent chapter. The content was based on the books by Diggle and Giorgi (2019) and Moraga (2019). For a more detailed explanation of these methods, please refer to these references.

2.1 Likelihood-based Inference

To perform inference to draw conclusions from the data, the study variable is modelled with the following specifications:

$$Y(x_i) = \mu + S(x_i) + Z_i : i = 1, ..., n \tag{2.3}$$

where $Y(x_i)$ are the observations at locations x_i, μ is the intercept, and $S(x_i)$ and Z_i are the residual information, for n locations. $S(x_i)$ represents the spatially correlated residual variation. For instance, due to omitted important explanatory variables that have spatial trends. While, Z_i, also called the nugget effect, represents the spatially uncorrelated residual variation that is often interpreted as either the spatial variation that is at play at smaller distance than the minimum observed distance in the data or as measurement error. It is assumed that Z_i is normally distributed with zero mean and variance τ^2. On the other hand, $S(x_i)$ has variance σ^2 and correlation function as presented before. As follows, the correlation between two observations, $corr(Y(x_i), Y(x_j)) = \sigma^2\rho(d)/(\tau^2 + \sigma^2)$, which approaches $\sigma^2/(\tau^2 + \sigma^2)$ as distance d approaches zero. In essence, this formulation of a geostatistical model extends standard regression models to accommodate spatial correlation in the data.

The model presented above can also be specified as a joint probability distribution,

$$[Y, S; \theta] = [S; \theta][Y|S; \theta] \tag{2.4}$$

where $[S; \theta]$ refers to the process model and is the probabibility distribution of S given a set of unknown parameters θ, and $[Y|S; \theta]$ refers to the data model and is the probability distribution of the data, Y, conditional on S, given θ. This specification is a hierarchical model, in which the distribution of an observable set of variables, Y, is specified conditionally on an unobservable or latent process S. A class of linear geostatistical models, like 2.3, is obtained by assuming that conditional on S, $Y(x_i)$ are mutually independent variables with Gaussian conditional distributions,

$$[Y(x_i)|S; \theta] \sim N(\mu + S(x_i), \tau^2) \tag{2.5}$$

Now, to make inference, maximum likelihood estimation is often done to estimate the parameters θ. A likelihood function is defined, that is the joint probability distribution of the data considered as a function of the parameters θ. It is given by,

$$L(\theta) = [Y; \theta] = \int [Y, S; \theta][S; \theta] dS \tag{2.6}$$

This is the joint distribution specified in 2.4 evaluated over the distribution of the latent process S given parameters θ. Often, a log transformation of the likelihood function is done for mathematical convenience. Maximizing the likelihood or the log-likelihood function yields point estimates of the parameters θ, and correspondingly, computation of the information matrix results in the standard errors of the parameters.

In the case that the outcome of the study is not continuous, i.e. a count or a binomial outcome, the geostatistical model can be extended with the corresponding link function. However, this complicates the likelihood function making it intractable necessitating methods like Laplace approximation or Monte Carlo maximum likelihood.

2.2 Bayesian Inference

In likelihood-based inferences, as discussed in the previous section, the parameters θ are assumed to be unknown fixed constants. In a Bayesian approach, inference is based on the posterior distribution, which is the distribution of the unobserved quantities in the model, conditional on the data observed. The standard linear geostatistical model is expressed as a Bayesian hierarchical model.

$$Y|S,\theta \sim \pi(Y|S,\theta) \tag{2.7}$$
$$S|\theta \sim N(\mu(\theta), Q(\theta)^{-1}) \tag{2.8}$$
$$\theta \sim \pi(\theta) \tag{2.9}$$

The first line of the hierarchical model represents the data model or the observation layer, the second line represents the latent layer, and the third line is for the prior distribution of the hyperparameters which represents any *apriori* belief for the hyperparameters. For the latent process S, $\mu(\theta)$ and $Q(\theta)$ are the mean and precision matrix, respectively. With this, the posterior distribution of all the unknown quantities is defined as,

$$\pi(S,\theta|Y) = \frac{\pi(Y|S,\theta)\pi(S|\theta)\pi(\theta)}{\pi(Y)} \tag{2.10}$$

$$\pi(S,\theta|Y) \propto \pi(Y|S,\theta)\pi(S|\theta)\pi(\theta) \tag{2.11}$$

Often, derivation of this posterior distribution involves difficult and high-dimensional integrals that have no closed-form solutions. Markov chain Monte Carlo (MCMC) methods have been traditionally used for solving this problem implemented in software programs like `WinBUGS` or `JAGS`. MCMC methods have revolutionized statistical practice by enabling Bayesian inference for complex models. However, they are computationally intensive and often struggle with convergence issues.

Integrated nested Laplace approximation (INLA) is a computationally efficient alternative to MCMC for approximate Bayesian inference in latent Gaussian models like geostatistical models. INLA combines analytical approximations and numerical integration to approximate the marginal

posterior distributions of all the parameters, including both S and θ. For a more detailed explanation about INLA for spatial models, the paper by Blangiardo et al. (2013) provided more information about it and how it could be implemented using `R-INLA`.

2.3 Stochastic Partial Differential Equations (SPDE)

In geostatistical models, the precision matrix Q is often not sparse, leading to computational challenges. As the number of sample locations increases, the dimension of Q also grows. One of the ways developed to circumvent this issue is the use of stochastic partial differential equations (SPDE). As previously mentioned, one of the assumptions for modelling geostatistical data is that there is a GRF over the study region. Whittle (1963) demonstrated that a GRF with a Matérn covariance matrix can be represented as a solution to a continuous domain SPDE. An approximate solution to the SPDE can be obtained using the Finite Element method, which divides the spatial domain A into non-intersecting triangles, creating a triangulated mesh with n nodes and n basis functions. Basis functions $\psi_k(\cdot)$ are defined as piecewise linear functions on each triangle, equal to 1 at vertex k and 0 at the other vertices. Then, the continuously indexed Gaussian field $S(x)$ is represented as a discretely indexed Gaussian Markov random field (GMRF) using the finite basis functions defined on the triangulated mesh.

$$S(x) \approx \sum_{i}^{n} \psi_k(x) x_k \tag{2.12}$$

Since the continuous GRF is approximated using a discrete, sparse GMRF, there is a huge computational gain. Typically, the SPDE approach is employed together with INLA. Several considerations should be noted in using this approach such as the specification of the triangular mesh for the study region. The paper of Righetto et al. (2020) provided some points and guidance on how to select an optimal mesh.

Chapter 3

Geomasking and Review of Methods for Inference in the Presence of Geomasking

In the following texts, the primary problem of interest was discussed in detail. Additionally, its impact on the inferences presented in the previous chapter was examined, highlighting how it complicates these inferences. Methods to address this problem were also presented, along with a quick evaluation of their advantages and disadvantages.

3.1 Geomasking

In recent years, there has been a surge in data containing geospatial information, which benefits research by providing a better understanding through the incorporation of location data. However, these advancements have raised numerous concerns about data security, privacy, and protection. Both the public and federal, state, and local government organizations worry that geospatial information could be exploited by attackers, potentially compromising critical infrastructures and the security and privacy of individuals, properties, and systems (Bertino et al., 2008). For instance, health records or health surveys that include GIS information contain crucial details such as disease status, medications, and personal information about the patients or respondents. This has prompted researchers and data collectors to implement strategies to protect the subjects of their data collection processes.

One of most common ways employed to preserve confidentiality is *geographical masking* or *geomasking*. Armstrong et al. (1999) compiled and developed methods that fall under the general class of geomasking. Geomasking involves altering the true locations in the data such as by changing the scale of the coordinates or rotating the locations around a pivot point. However, the most common method is through point aggregation and random perturbation. Point aggregation involves aggregating all locations within a sufficiently large geographical area to a single location. While, random perturbation involves displacing the original location randomly over a specific region. It has been noted that the latter method has the best balance of location confidentiality and information preservation (Armstrong et al., 1999).

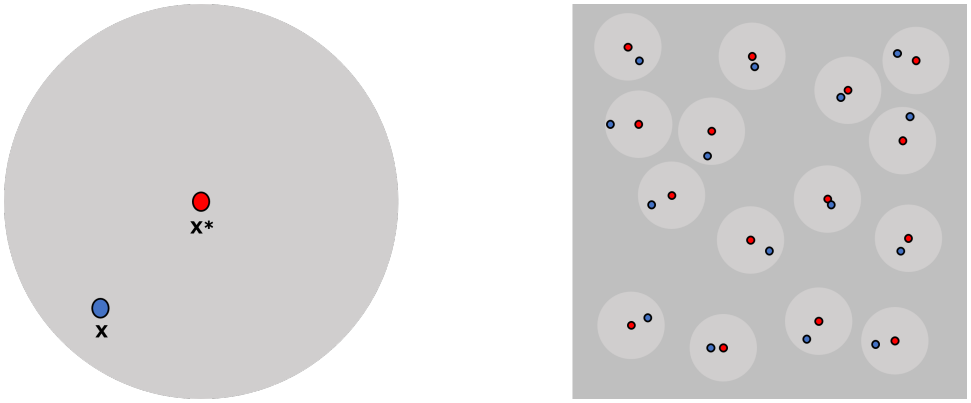

Figure 3.1: Mechanism of *uniform geomasking*

To apply random perturbation, often the region of displacement is a circle with the original location as the center and with a pre-specified radius, δ. Then, the original location is displaced using a random angle (0 to 2π) and a random distance (0 to δ). This technique is commonly called as *uniform geomasking* as the probabilities to be displaced over the region are equal. Shown in Figure 3.1 is the mechanism of uniform geomasking. The true locations x^* are denoted by the red dots while the geomasked locations x are denoted by the blue dots. On the left, the individual displacement of x^* to x is demonstrated, while the right shows the x^* and the x for the whole study region.

There are some other extensions of uniform geomasking and Zandbergen et al. (2014) had reviewed these techniques and its implementation. For example, *Gaussian geomasking* wherein the displacement probabilities are Gaussian distributed, or *donut geomasking* wherein the region is composed of two concentric circles like a donut. But for convenience, uniform geomasking and methods accounting for it are the subject of this thesis as it is the most studied and is the most common technique used in practice such as in the case study to be presented in the later chapter.

3.2 Inferences with Geomasked Spatial Data

In essence, (uniform) geomasked spatial data are spatial data with location errors. That means that locations x_i in the models specified in 2.3 are not the true locations. The good thing, however, contrary to pure location error, the procedure on how the location error was introduced is typically known and provided with geomasked spatial data. Several studies have illustrated that ignoring location errors, in general, lead to incorrect inferences and poor spatial predictions. It has been shown that it yields to biased mean such as disease rates (Goldberg and Cockburn, 2012), covariate effects such as exposure effects (Kinnee et al., 2020), and spatial covariance parameters (Gabrosek and Cressie, 2002).

Moreover, there have been studies as well that have documented similar effects of geomasking

(Fronterrè et al., 2018, Arbia et al., 2023). In particular, it has been demonstrated, both theoretically and through a simulation study, that (uniform) geomasking disrupts the spatial structure in the data (Fronterrè et al., 2018) resulting to biased covariance parameter estimates. Specifically, spatial variance σ^2 tends to be underestimated and spatial range ϕ to be overestimated. While nugget variance τ^2 tends to be overestimated due to an artificial nugget effect resulting from the reduction in the spatial structure. In contrast, the mean μ is not affected by geomasking as it does not depend on the distances. In addition, it has been noted that the effects of geomasking tend to be dependent on the ratio of displacement δ and true spatial range ϕ. Predictions are noted to be inaccurate and imprecise as a result of incorrect model estimates.

Location error, or geomasking in particular, in geostatistics has been mentioned occasionally, but it hasn't been fully integrated into spatial statistical analysis. This problem was first addressed by Gabrosek and Cressie (2002) through adjusting the standard kriging equations for spatial predictions, and later by Cressie and Kornak (2003) with a more general approach with adjustments for the mean component of the model.

The first to study and proposed a method in a model-based geostatistics framework were Fanshawe and Diggle (2011). They have extended the model in 2.4 incorporating that the true locations X^* were not the ones observed but the displaced locations X. Correspondingly, the model was written as,

$$[Y, S, X, X^*] = [Y|S, X, X^*] = [Y|S, X^*; \theta, \delta][S|X^*; \theta, \delta][X^*|X; \theta, \delta][X; \theta, \delta] \tag{3.1}$$

with the same model parameters θ and new parameter δ related to distribution of the location error. This new model proposed by Fanshawe and Diggle was just the same model as the standard geostatistical model but with a layered latent effect composed of the process model S and a location error model X^*. Then, the likelihood function was defined as follows,

$$\begin{aligned} L(\theta, \delta) &= \iint [Y, S, X, X^*; \theta, \delta] dS dX^* \\ &= \iint [Y|S, X^*; \theta, \delta][S|X^*; \theta, \delta][X^*|X; \theta, \delta][X; \theta, \delta] dS dX^* \\ &\propto \iint [Y|S, X^*; \theta, \delta][S|X^*; \theta, \delta][X^*|X; \theta, \delta] dS dX^* \end{aligned} \tag{3.2}$$

as X does not depend on any of the parameters. Fanshawe and Diggle evaluated the likelihood using Monte Carlo integration and maximized it using Nealder-Mead algorithm. Once the point estimates and standard errors were generated, inference and spatial predictions followed.

Extending the model for geomasked spatial data, since the distribution on how the location error is introduced and the parameter δ is known, δ can be removed in the model as a parameter to be estimated. Consequently, for geomasked spatial data, the model can be written as,

$$[Y, S, X, X^*] = [Y|S, X^*; \theta][S|X^*; \theta, \delta][X^*|X; \theta][X; \theta] \tag{3.3}$$

and the likelihood function as,

$$
\begin{aligned}
L(\theta) &= \iint [Y|S, X^*; \theta][S|X^*; \theta][X^*|X; \theta][X; \theta] dS dX^* \\
&\propto \iint [Y|S, X^*; \theta][S|X^*; \theta][X^*|X; \theta] dS dX^*
\end{aligned}
\tag{3.4}
$$

The main downside of this method was its computational burden. In the evaluation of the likelihood, Monte Carlo integration was done, where B samples were drawn from $[X^*|X; \theta, \delta]$ resulting to the approximation of the likelihood,

$$
L(\theta, \delta) \approx \frac{1}{B}[Y|X^*_{(b)}; \theta] \tag{3.5}
$$

entailing a computation time of $O(B \times n^3)$. Inverting matrices to compute the standard errors compounded the computational burden of the method.

Fronterrè et al. (2018) revisited and improved the work of Fanshawe and Diggle (2011) through the use of composite likelihood approach. To overcome the computational limits of the previous method, they approximated the likelihood in 3.4 by pairwise likelihood contributions. The resulting approximation was achieved by considering each pair of bivariate densities as independent, resulting in

$$
\begin{aligned}
logL(\theta) \approx logL(\theta)_{CL} &= \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} log[Y_i, Y_j; \theta] \\
&= \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} log \int_0^{\infty} [Y_i, Y_j|D^*_{ij}][D^*_{ij}|d_{ij}] dU^*_{ij}
\end{aligned}
\tag{3.6}
$$

where D^*_{ij} are the distances after geomasking is applied and d_{ij} are the true distances. They have approximated that the conditional distribution of D^*_{ij} given the true distances follows a Rice distribution with parameters d_{ij} and $\delta/\sqrt{6}$. More detailed explanation of the composite likelihood approach for geostatistical models are written in the works of Varin et al. (2011), Bevilacqua and Gaetan (2015), and Stein et al. (2004). This approximation necessitated the integration of only $n(n-1)/2$ univariate integrals, significantly speeding up the process. They have further used a quasi Monte Carlo method to enhance the computational speed of the method. From a simulation study, they concluded that their composite likelihood approach results in substantially smaller root mean square errors for parameter estimates compared to standard geostatistical modelling that ignores geomasking.

In a Bayesian framework, the first to evaluate this problem and proposed a method were done by Wilson and Wakefield (2021). They have extended the hierarchical Bayesian model, specifically by factorizing the posterior distribution into marginal posteriors of θ and the geomasked (observed) locations X. The distribution was specified as follows,

$$
\pi(\theta|Y, X, X^*) \propto \pi(Y|X, \theta)\pi(\theta) \tag{3.7}
$$

$$
\pi(X|Y, X^*, \theta) \propto \pi(Y|X, \theta)\pi(X|X^*) \tag{3.8}
$$

where X are the geomasked locations while X^* are the true locations, and $\pi(X|X^*)$ is the geomasking distribution. They have employed an INLA within MCMC procedure in an effort to make the computations faster. The procedure involved employing MCMC to sample the true locations, followed by the application of INLA for inference based on these true locations. More details on the approach are presented in their paper and the paper of Gómez-Rubio and Rue (2018). Still, this approach proved to be relatively slow, taking 52 hours to run a single scenario with 1000 iterations for 398 locations.

In summary, these were all the methods available, except for the next one to be presented below, that were developed to deal with geomasked spatial data. As mentioned, the method of Fanshawe and Diggle (2011), which was in a likelihood framework, and Wilson and Wakefield (2021), in a Bayesian framework, were noted with computational burden. However, the method proposed by Fronterrè et al. (2018) was incomplete in that it does not provide standard errors for the parameter estimates, thus rendering it unable to make predictions. In addition, the method of Fanshawe and Diggle (2011) and Fronterrè et al. (2018) only allowed Gaussian outcomes. Lastly, there were no packages available in `R` for all of these methods, which restricts their usability for the public.

3.2.1 GeoAdjust Package

Recently, Altay, Paige, Riebler, and Fuglstad (2023) developed a method improving the previous works on geomasking, focusing on the Demographic and Health Survey (DHS) data. More details were presented about DHS in the case study in the next chapter. They developed and published an R package called `GeoAdjust`, designed to implement empirical Bayesian geostatistical inference under geomasking (Altay et al., 2024). In their model, they started with defining the likelihood of the individual observations in geomasked spatial data. That is,

$$\begin{aligned} \pi(Y, X|S, \theta) &= \int \pi(Y, X, X^*|S, \theta) dX^* \\ &= \int \pi(Y|S, \theta)\pi(X|X^*)\pi(X^*) dX^* \end{aligned} \tag{3.9}$$

This two-dimensional integral was then approximated numerically using quadratures. Numerical integration was done by placing an integration point in the geomasked location X, and then placing more rings of points around X. Hence,

$$\begin{aligned} \int \pi(Y|S, \theta)\pi(X|X^*)\pi(X^*) dX^* &= \int \pi(Y|S, \theta) d[\pi(X|X^*)\pi(X^*)] \\ &\approx \sum_{j=1}^{J^1} \sum_{k=1}^{m_{ij}} \lambda_{ijk} \pi(y_i|S, \theta) \end{aligned} \tag{3.10}$$

where

$$\lambda_{ijk} \propto \int \pi(X|X^*)\pi(X^*) dX^* \tag{3.11}$$

with m_{ij} denoting the total number of integration points for observation i ring j and J^i the total number of rings for observation i. At each integration point, there was a corresponding

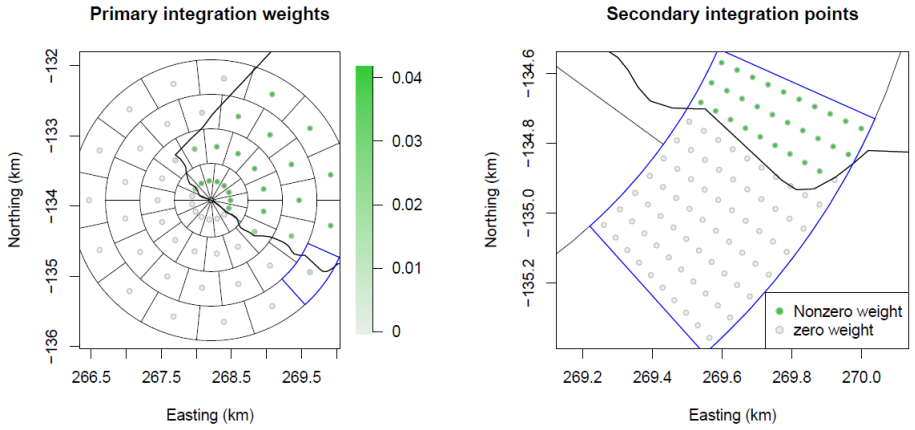

Figure 3.2: Illustration of the integration points by `GeoAdjust`. Figure obtained from Altay, Paige, Riebler, and Fuglstad (2023).

weight λ_{ijk}, where the $\sum_{ij}\sum_{ijk}\lambda_{ijk} = 1$ for each observation i. If the displacement distance was within the boundary of the region, the weights were adjusted accordingly by creating secondary integration regions for areas near the boundary, each with an associated 10x10 grid of integration points. The assignment of integration points is depicted in Figure 3.2.

They had m_{i1} set to 1 and m_{ij} set to 15 for the subsequent rings. In other words, after the single integration point at X, 15 integrations points were built for the next rings surrounding X. Since `GeoAdjust` was developed to analyze a DHS data, the maximum displacement radius was set to either 2km (urban clusters) or 10km (rural clusters). For 2km, 5 rings were built while for 10km, 10 rings were built. The technical derivation of the numerical integration procedure was presented in the appendix of the paper of Altay et al. (2022). Users can set the displacement radius on their own but it is currently limited to the multiples of the current displacement rule by DHS. Setting the displacement radius to 0km results in a standard geostatistical model with only a single integration point at X, without the succeeding rings of points around it.

In addition to the numerical integration of the likelihood, `GeoAdjust` utilized an SPDE approach for the estimation of S and used penalized complexity priors. PC priors are alternative for *ad hoc* reference priors which are better in terms of inference and avoidance of overfitting (Simpson et al., 2017). These PC priors are defined as probability statements reflecting prior information on the parameters. `GeoAdjust` implemented the PC priors developed by Fuglstad et al., 2019 for Gaussian random fields.

$$P(\sigma > \sigma_0) = \alpha_\sigma \quad P(\tau > \tau_0) = \alpha_\tau \quad P(\phi < \phi_0) = \alpha_\phi \tag{3.12}$$

The default priors were $\sigma_0 = \tau_0 = 1$, $\alpha_\sigma = \alpha_\tau = 0.05$, and $\alpha_\phi = 0.5$. It was recommended for ϕ_0 to be 1/4 of the maximum distance.

`GeoAdjust` also leveraged on the computational power of `Template Model Builder` (`TMB`). `TMB` (Kristensen et al., 2015) is an `R` package designed for fast implementation of models with

complex non-linear latent effects. It is typically used in complex spatio-temporal models such as state-space models in ecology (Albertsen et al., 2015, Auger-Méthé et al., 2017). In detail, for `GeoAdjust`, `TMB` was used to produce marginal maximum *a posteriori* (MMAP) estimates of fixed effects θ. The marginal posterior of θ was specified as,

$$\begin{aligned}\pi(\theta|Y) &= \int \pi(Y|S,\theta)\pi(S|\theta)\pi(\theta)dS \\ &= \int exp(log\pi(Y|S,\theta) + log\pi(S|\theta) + log\pi(\theta))dS \\ &= \int exp(-f(\theta,S))\end{aligned}$$

Then `TMB` was used to integrate out S, and autodifferentiate to maximize and to take a Laplace approximation of the posterior. Inference for random effects S then occured through empirical Bayes estimation by maximization of $f(\hat{\theta}, S)$ conditioned on parameter estimates $\hat{\theta}$. As such, posterior distributions of the covariance parameters were not generated. `TMB` is similar to `INLA` as both are approximation procedures. However, the model being used here can't be fitted using `INLA`. For more technical explanation, Osgood-Zimmerman and Wakefield (2023) reviewed and explained the procedure done by `TMB` and compared it to `INLA`.

By adopting the likelihood approximation, `TMB` implementation, and SPDE approach, `GeoAdjust` was noted to be very fast compared to its predecessors. It addressed the limitation of the previously presented methods, as it can perform inferences for continuous, binomial, and count outcomes. Predictions could also be generated. The authors of `GeoAdjust` have applied it on a real data and subjected it to a simulation study and found that it results to more accurate parameter estimates and enhanced predictive power especially when the displacement was large.

Chapter 4

Case Study: The Philippines Demographic and Health Survey

This thesis was primarily inspired by the Demographic and Health Survey (DHS) Program, which is predominantly conducted in developing countries. DHS are nationally representative household surveys that have been conducted in over 85 countries worldwide since 1984, allowing for comparability across different nations. The survey contains crucial data on child mortality, nutrition, antenatal care coverage, maternal mortality, family planning, domestic violence, and access to clean water and sanitation facilities, with some of these serving as key indicators for the Sustainable Development Goals (The DHS Program, 2024). A range of robust observational data analysis methods have been employed using the DHS data, including cross-sectional designs, repeated cross-sectional designs, spatial and multilevel analyses, intra-household designs, and cross-comparative analyses (Corsi et al., 2012).

In the past decade, alongside with the different information collected in DHS, GPS locations of the respondents were also recorded. This enabled researchers to analyze respondent locations spatially, allowing them to identify geographical patterns associated with specific demographic and health outcomes and programs (Burgert et al., 2013). But to protect the survey respondents, DHS datasets are disclosed to the public with a geomasked GPS data. As a case study to illustrate the use of `GeoAdjust`, we used the latest Philippines DHS as a subject for analysis.

4.1 Data Description

The Philippines National Demographic and Health Survey (NDHS) started in 1968 and is conducted every 5 years. The most recent survey round is the 2022 NDHS and is the seventh DHS conducted in the country in collaboration with the DHS Program (Philippine Statistics Authority and ICF, 2022). The collection of GPS data for NDHS started in the 2003 version. The 2022 NDHS survey was conducted following a two-stage stratified sampling scheme that was representative of the entire country, the 17 administrative regions, and both urban and rural areas. In the first stage, a systematic selection of primary sampling units (PSUs) were distributed by province and highly urbanized cities. In the second stage, a systematic random sampling method was used to select an equal number of either 22 or 29 housing units from each

sampled PSU.

For the GPS data included with the NDHS, Burgert et al. (2013) elaborated on the methodology used. Initially, all data originating from the same PSU were aggregated to a single point coordinate, which was the centroid of the PSU. Subsequently, depending on whether the PSU was classified as a rural or urban cluster, the coordinates underwent geomasking. Urban clusters were displaced by a distance of up to 2km (0-2 km), while rural clusters were displaced by up to 5km (0-5 km). Additionally, a further randomly selected 1% of rural clusters were displaced by a distance of up to 10km (0-10 km). The displaced locations were made sure to remain inside the boundaries of the region of interest. In the 2022 NDHS, there were 1247 PSUs or clusters, comprising of 505 urban clusters and 742 rural clusters (Philippine Statistics Authority and ICF, 2023).

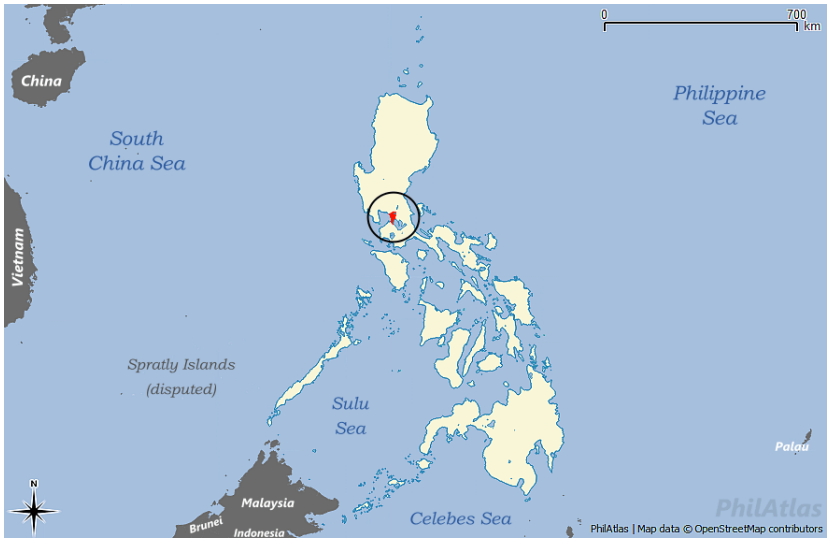

Figure 4.1: Location of Metro Manila in the Philippines (PhilAtlas, 2024)

Due to the geographical complexity of the Philippines as it is an archipelago, only the National Capital Region (NCR) or Metro Manila was considered as the study region. Metro Manila is the country's political and economic epicenter, and is composed of 16 cities and 1 municipality (Porio et al., 2019). For the 2022 NDHS, 126 clusters were sampled in Metro Manila, all of which were urban and displaced by up to 2km. The outcome analyzed in this case study was the distance, in minutes, to the nearest health facility. There have been several studies illustrating that longer distance times to nearest health facility are related to worse health outcomes (Kelly et al., 2016). Especially in Metro Manila, a highly dense region, this outcome is of importance. At the same time, this outcome was chosen as it was anticipated that the observations would exhibit spatial correlation. It is important that the variable of study has substantial spatial correlation to ensure the validity of geostatistical models (Diggle and Giorgi, 2019).

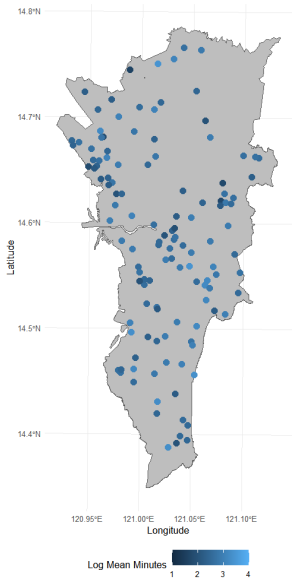

Figure 4.2: Locations of the samples

4.2 Methods

The data and the shapefiles needed for the analysis were provided by the DHS Program. The GPS data and the study area boundaries were prepared using `QGIS`. In the data, the distance time to the nearest facility were given per each respondent in the survey. There were an average of 22 respondents for each of the 126 cluster locations. However, due to the limitations of the model to be used in the later part, only allowing single observation per location, the distance times were aggregated for each of the cluster locations. It should be noted that aggregation may result in a loss of information, such as the variation between the respondents. There are available techniques that handle inferences for aggregated spatial data but this was beyond the scope of this analysis. The outcome to be modeled was the natural log transformed mean distance time.

After the data and shapefiles were cleaned and prepared, statistical analysis proceeded as follows. Firstly, exploratory data analysis was done. The samples were plotted and tested for spatial correlation. Test for residual spatial correlation was done using empirical variogram (Diggle and Giorgi, 2019). In brief, the test involved a Monte Carlo strategy to simulate empirical variograms under spatial independence and assessed if the variogram from the data was inside the envelope of the simulated empirical variograms. The number of simulations were set to 1000 and distance bins were set to different values to check consistency. Afterwards, model fittings were performed. Two hierarchical Bayesian models were fitted: one assuming the locations were correct, and another incorporating the geomasked locations. In the ensuing texts, the first model was called as the 'naive' model while the second model was called as '`GeoAdjust`' model. The naive model was defined similar to a standard geostatistical model.

$$Y(x_i)|S;\theta \sim N(\mu + S(x_i), \tau^2) \tag{4.1}$$

where Y is the log mean distance time (in minutes) to the nearest facility at locations x for $i = 1, ..., n$ and μ is the overall mean. While S represents the residual information capturing the spatial variation assumed to follow a multivariate normal distribution with an exponential correlation function and variance, σ^2. The remaining unstructured variation are described by τ^2. Similarly, the `GeoAdjust` model had similar form with an addition of another latent effect introduced by geomasking.

$$Y(x_i)|S, X, X^*; \theta, \delta \sim N(\mu + S(x_i), \tau^2) \qquad X|X^*; \delta \sim \pi(X|X^*) \tag{4.2}$$

where X are the geomasked locations and X^* are the true (unobserved) locations, with $\pi(X|X^*)$ as the geomasking distribution where δ set to 2km. The priors for both model covariance parameters were PC priors and a flat prior was used for μ. A triangulated mesh was then constructed for the the SPDE approach. Different PC prior specifications and meshes were done for sensitivity analysis.

Both models were then fitted in `TMB` using `GeoAdjust`. As noted in the previous chapter, a 0km displacement radius results to the standard geostatistical model. Predictions and uncertainty around this predictions for both model were produced and visualized graphically.

4.3 Results

The locations of the cluster samples are presented in Figure 4.2. Figure 4.3 indicates that according to the residual spatial correlation test there was substantial spatial correlation in the data. However, this result should be interpreted taking into consideration that the locations were geomasked. It is highly probable that the true spatial correlation was much higher.

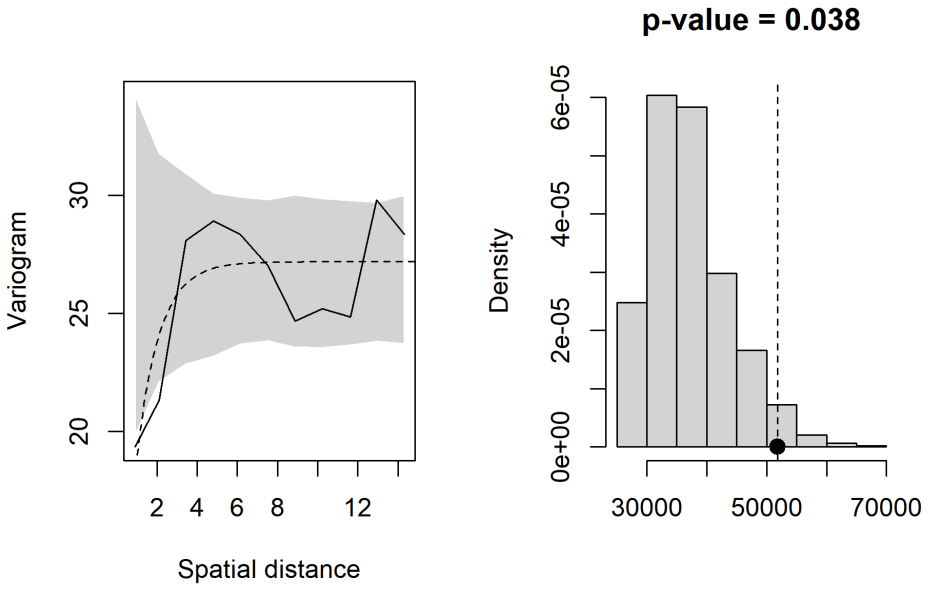

Figure 4.3: Results for test of residual spatial correlation

Results for the two fitted models are shown in Table 4.1. For this data with 126 locations, all of the models ran less than a minute. As mentioned in the previous chapter, since the overall mean μ was not dependent on the distance, both the naive and `GeoAdjust` model estimated it similarly. The differences from the two models arose in the estimates for the covariance parameters. In particular, the `GeoAdjust` model had a smaller estimate for the spatial variance σ^2 and a bigger estimate for the nugget variance τ^2 than the naive model. For spatial range ϕ, the `GeoAdjust` model had bigger estimates than the naive model.

Parameter	**Naive**	**GeoAdjust**
μ	2.6022 (2.49, 2.71)	2.5966 (2.48, 2.71)
σ^2	0.0351	0.0289
ϕ	4.7818	6.2253
τ^2	0.1109	0.1196
Time elapsed (in sec)	16.69	42.14

Table 4.1: Parameter estimates for the Naive and `GeoAdjust` model

From the parameter estimates of the two models, predictions for the whole study region were made with the corresponding standard deviations. Figure 4.4 presents the comparison of the values for the two models. The prediction maps of the models seemed similar but the map from the `GeoAdjust` model was smoother. This can be explained by the difference in the estimates for spatial range. The primary distinction in the model predictions seemed to be the level of uncertainty. The `GeoAdjust` model generated more precise estimates, with higher precision evident in the sample location points. Notably, areas in the southeast of the region appeared to have the longest travel times to the nearest facility compared to other areas.

The results presented above cannot be confirmed as accurate since only the geomasked dataset was available. Assuming that the results of `GeoAdjust` were indeed the true values of the parameters, it seemed counterintuitive to observe smaller spatial variance and larger nugget variance when geomasking was considered. However, it appeared that geomasking in this particular dataset might have affected the spatial range only. Nonetheless, these results warranted further scrutiny.

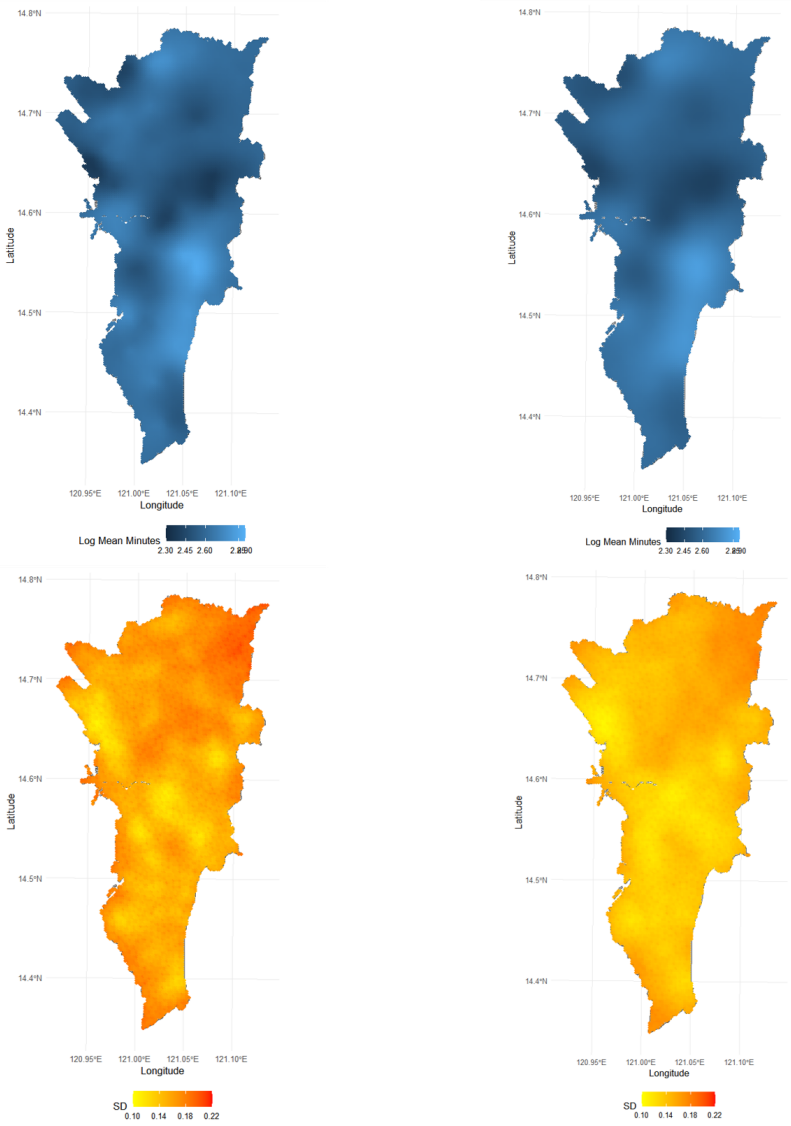

Figure 4.4: Predictions of the naive (*right*) and `GeoAdjust` (*left*) model

Chapter 5

Simulation Study

Previously, `GeoAdjust` was demonstrated using a case study, specifically the Philippines DHS. It was observed that there were some differences when `GeoAdjust` was implemented compared to a standard geostatistical analysis ignoring location error. In this chapter, a simulation study was conducted to determine the performance of `GeoAdjust` and the scenarios where it is applied best. Altay et al. (2022) performed a simulation study to demonstrate the performance of `GeoAdjust` under a standard DHS displacement procedure, where data points were displaced by 2km, 5km, or 10km. Their general findings indicated that `GeoAdjust` performed better when the displacement was less than the true spatial range. If the displacement exceeded the true spatial range, the spatial correlation was entirely destroyed. They also found that it performed better in larger displacement settings. In this simulation study, these results were extended in more detail. Instead of considering multiple displacement radii present in the data, only a single displacement was focused on. Several combinations of spatial variance (sill) and nugget variance (nugget) were also considered, and the impact of sample sizes was examined. Lastly, the performance was measured in terms of both parameter estimation and predictions.

5.1 Simulation Set up

Using the parameter estimates from the case study results, several simulation settings were established. The mean parameter μ was set to 2.5 and the spatial range ϕ was set to 8 km to accommodate larger displacements. The spatial variance (sill) σ^2 and nugget variance (nugget) τ^2 were varied, with values set to either 0.2 or 0.02. This created four combinations of sill and nugget values. Sample sizes were chosen based on those in the case study, with values of 75, 125, and 250 locations. For the displacement radius, both the original 2km and a larger 6km were selected. Combining all these parameter settings resulted in a total of 24 simulation scenarios.

The following steps were the summary of the simulation study:

1. Data generation: Generate 3,000 + n locations (X^*) over an area randomly. The n locations will be used as sample locations (X^*) and the 3,000 locations will be used as prediction locations ($\tilde{X}$). Then, simulate the outcome Y given the locations (X^* or $\tilde{X}$) and the parameter settings.

2. Geomasking: Apply uniform geomasking to X^*, by randomly displacing over a circular region given displacement parameter, to have the geomasked locations (X).

3. Analysis: Perform an analysis using the naive model and by GeoAdjust. Generate predictions for the $\tilde{X}$ using the models.

4. Parameter estimate measure: Compute parameter bias and relative bias.

5. Prediction measure: Compute prediction root mean squared error (RMSE) and continuous ranked probability score (CRPS).

For data generation, n referred to the different sample size settings. These steps were done 100 times for the 24 settings for a total of 2400 runs. To save computational time, parallel runs were implemented using the Flemish Supercomputer Center (VSC, *Vlaams Supercomputer Centrum*). It took a total of approximately 4 days to complete all of the runs.

Bias and relative bias were used as measurement for the the parameter estimates of the models. Bias for any parameter θ is

$$= \hat{\theta} - \theta_0 \tag{5.1}$$

where $\hat{\theta}$ is the model's estimate and θ_0 is the true value. While, relative bias on the other hand is,

$$= \frac{\hat{\theta} - \theta_0}{\theta_0} \tag{5.2}$$

Predictive performance was measured using root mean square error (RMSE),

$$\text{RMSE} = \sqrt{\frac{1}{3000}\sum_{i=1}^{3000}(y_i - \hat{y}_i)^2} \tag{5.3}$$

However, RMSE does not consider the uncertainty around the predictions and may lead to incorrect comparison results. Hence, continuous ranked probability score (CRPS) was also used as a predictive measure. CRPS can be used to compare observations and predictions accounting for the uncertainty (Matheson and Winkler, 1976). The CRPS for observation y is a score function that compares the Cumulative Distribution Function (CDF) of the prediction distribution (F) with the degenerate CDF of the observation ($1[u \geq y]$) (Moraga, 2023). It is given by,

$$\text{CRPS}(F, y) = \int_{-\infty}^{\infty}(F(u) - 1[u \geq y])^2 du \tag{5.4}$$

CRPS simplifies to mean absolute error (MAE) when the predicted distribution is a single point estimate rather than a full distribution. The mean CRPS for all 3000 prediction locations was computed and was used to compare the results. For all of the measures, namely bias, relative bias, RMSE, and CRPS, 0 was the perfect score. Thus, a smaller or value close to 0 signified better performance. Since in each 24 settings, 100 simulated data were analyzed using two models, pairwise comparisons were made. Specifically, pairwise bias ratio for the parameters and pairwise differences for the RMSE and CRPS were computed for each of the 100 runs.

$$\text{Bias ratio} = \frac{\text{bias}_{GeoAdjust}}{\text{bias}_{naive}} \tag{5.5}$$

$$\text{RMSE difference} = \text{RMSE}_{GeoAdjust} - \text{RMSE}_{naive} \tag{5.6}$$

$$\text{CRPS difference} = \text{CRPS}_{GeoAdjust} - \text{CRPS}_{naive} \tag{5.7}$$

5.2 Results

As shown in the additional figures for the simulation study in Appendix B, the parameter estimates from the naive analysis generally aligned with the theory described by Fronterrè et al. (2018) regarding the effects of geomasking. Specifically, the relative bias for the mean parameter μ across different simulation settings was close to zero. However, the estimates for the covariance parameters showed some discrepancies: the spatial variance σ^2 tended to be underestimated, the nugget variance τ^2 was overestimated, and the spatial range ϕ was also overestimated, on average, all in line with the theoretical expectations.

Figures 5.1, 5.2, 5.3, and 5.4 present the pairwise bias ratio for the model parameters across all settings. As anticipated, the mean parameter μ was consistently estimated correctly by both methods, with bias ratios in all settings very close to 1, indicating similar bias. However, the results for the covariance parameters differed notably between the methods. For the spatial variance σ^2, the bias was generally smaller when using `GeoAdjust` compared to the naive approach. This improvement was more pronounced as the sample size increased, and especially when the displacement radius was set to 6km. The enhanced performance of `GeoAdjust` over the naive method was particularly evident in scenarios where the true spatial variance was greater than or equal to the true nugget variance. Similarly, for the nugget variance parameter τ^2, `GeoAdjust` showed better performance as the sample size increased and the displacement radius was set to 6km. This improvement was also more noticeable when the true spatial variance was greater than or equal to the nugget variance. Although a similar trend was observed for the spatial range parameter ϕ, there were exceptions. Specifically, in instances where the displacement radius was set to 2km, the naive method sometimes outperformed `GeoAdjust`.

Additionally, when comparing the bias ratios for the three covariance parameters, `GeoAdjust` demonstrated better performance than the naive method for both the nugget variance τ^2 and the spatial range ϕ. For instance, with a true spatial variance of 0.2, a nugget variance of 0.02, 250 sample locations, and a 6km displacement, the median bias for τ^2 and ϕ using `GeoAdjust` was nearly half that of the naive method. Under the same conditions, `GeoAdjust` had a median bias for the spatial variance σ^2 of about 75% of the naive method's bias. These results indicated that using `GeoAdjust` led to clearer improvements in the estimation of nugget variance and spatial range compared to spatial variance.

In terms of predictive performance, pairwise differences in RMSE and CRPS revealed similar distinctions between the naive method and `GeoAdjust`. For RMSE, a noticeable difference between the two methods emerged only when the true spatial variance was significantly greater than the true nugget variance. Conversely, when CRPS was used as the measure of predictive performance, `GeoAdjust` consistently outperformed the naive method. Larger sample sizes and a displacement radius of 6km yielded better predictive scores, especially when the spatial variance

was greater than or equal to the nugget variance.

Computational times were recorded for all the runs, with longer times noted for `GeoAdjust`, especially when the sample size increased and the displacement radius was 6km. Spatial variance and nugget variance had no noticeable effects on computational time. This was expected since the method constructs integration points for each sample location and its surroundings, depending on the displacement size and the number of locations. The difference in run times between `GeoAdjust` and the naive method was not recorded to be greater than 500 seconds, supporting that the method was relatively fast compared to previous methods developed for geomasking. It was also observed that among the 2400 runs, 3 runs failed. These failures occurred in scenarios where the sample size was 75 locations.

These results indicated that while `GeoAdjust` generally provided better estimates for the covariance parameters and better predictions, its performance varied depending on the number of sample locations, size of the displacement, and the relative magnitudes of the spatial and nugget variances.

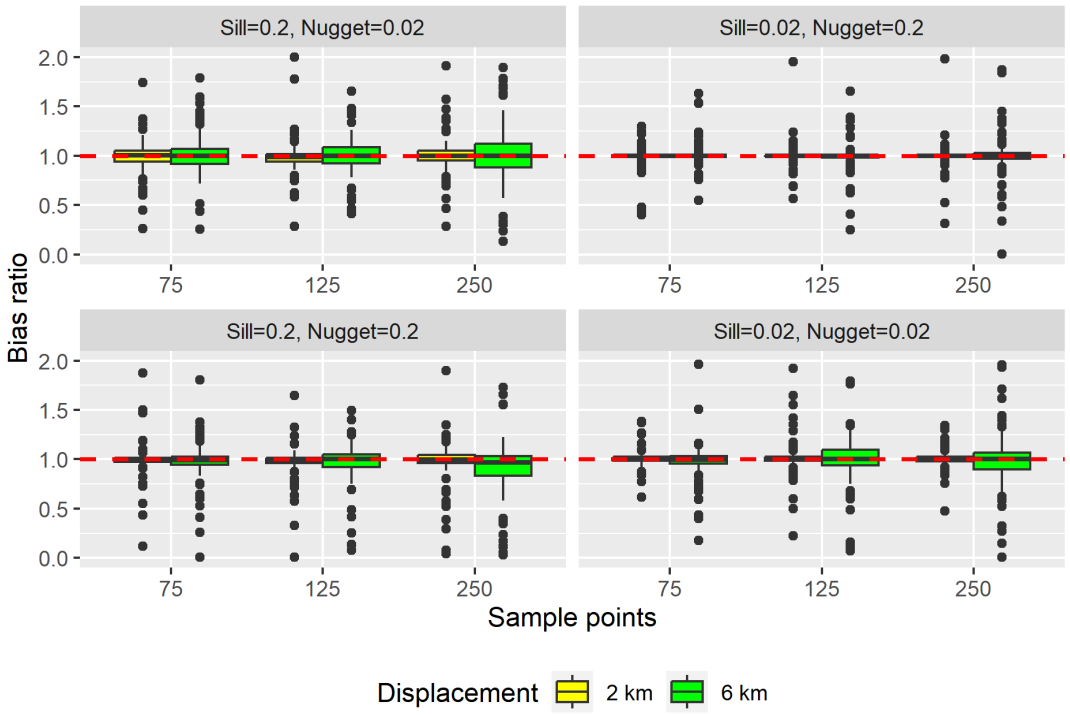

Figure 5.1: Pairwise bias ratio for the μ parameter

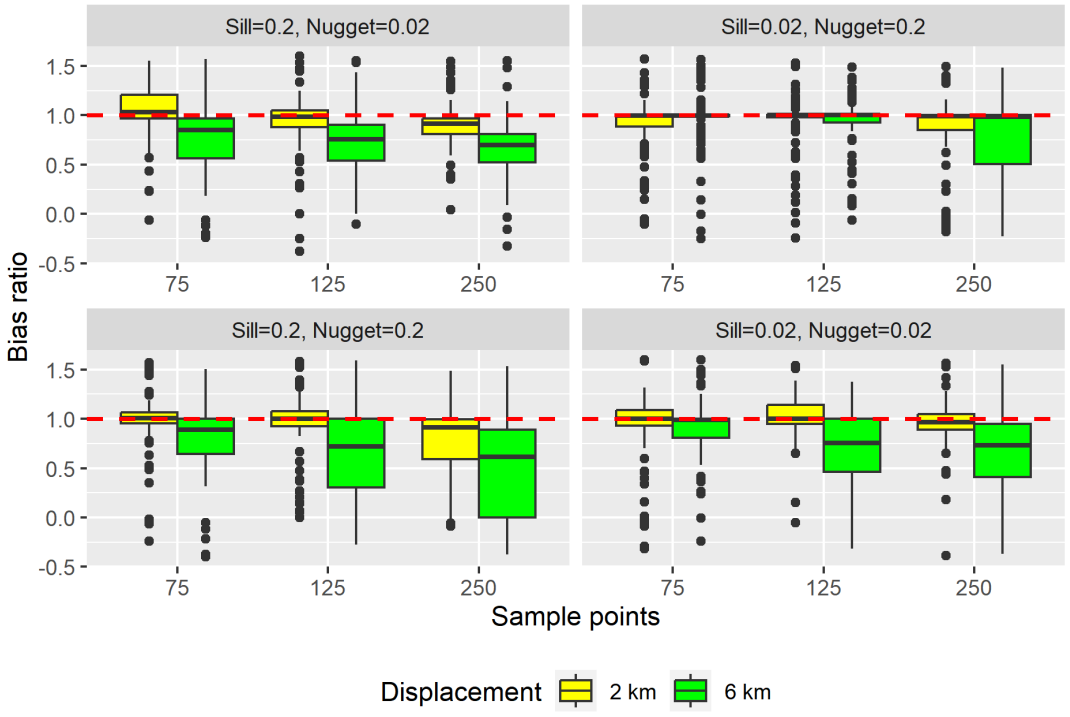

Figure 5.2: Pairwise bias ratio for the σ^2 parameter

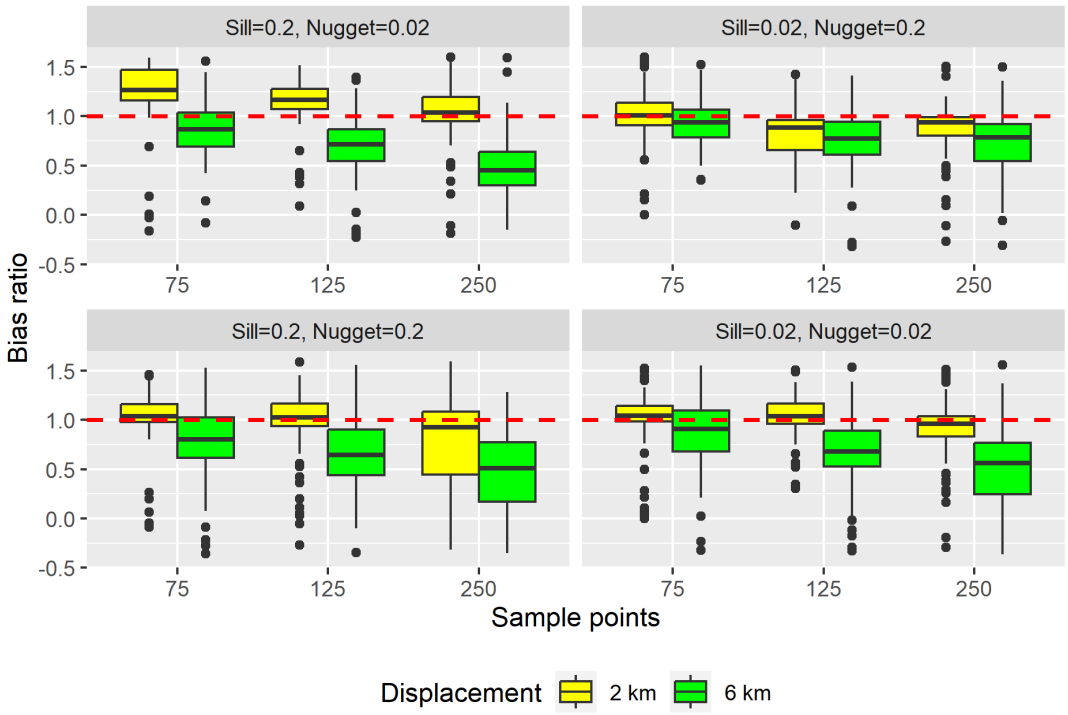

Figure 5.3: Pairwise bias ratio for the ϕ parameter

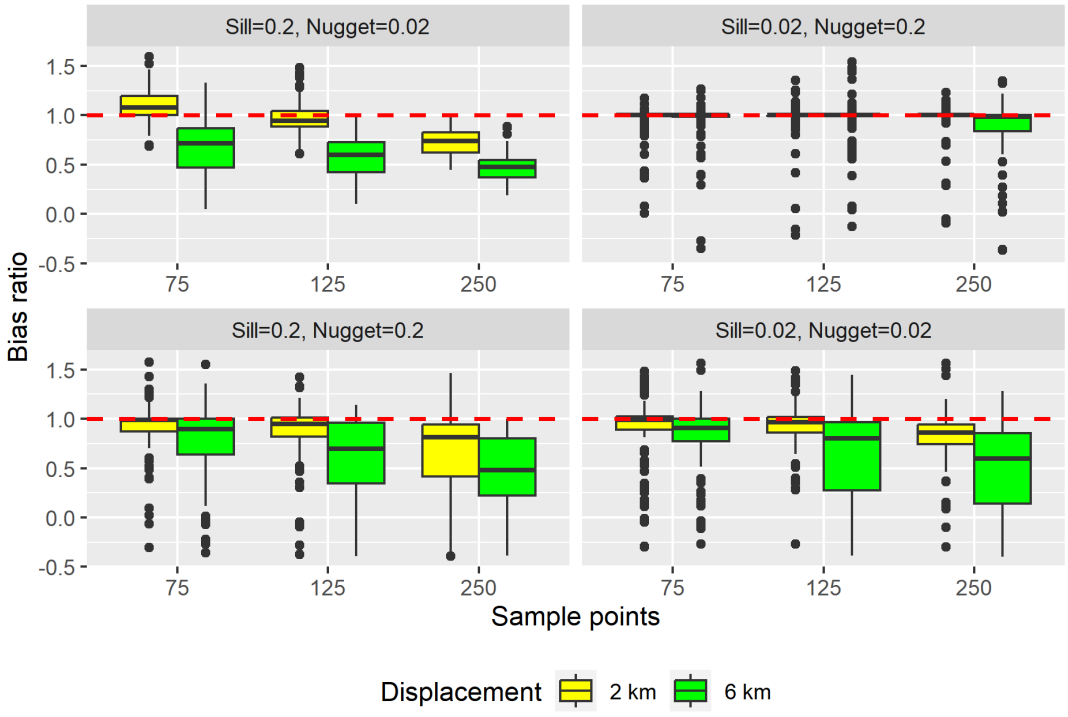

Figure 5.4: Pairwise bias ratio for the τ^2 parameter

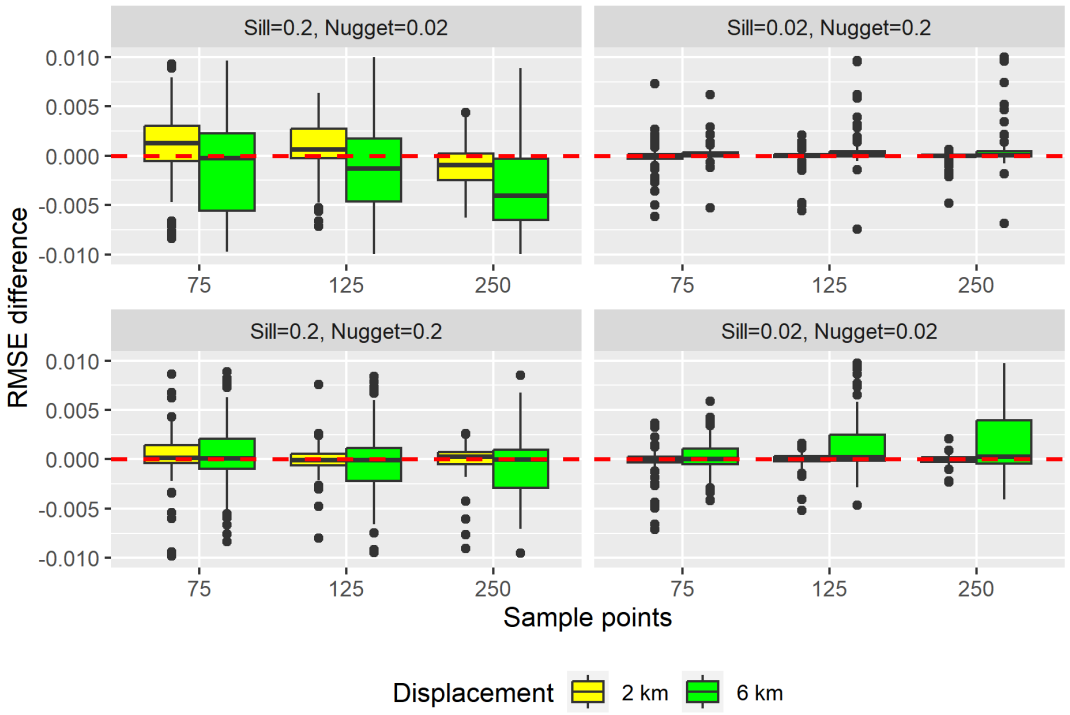

Figure 5.5: Pairwise RMSE difference

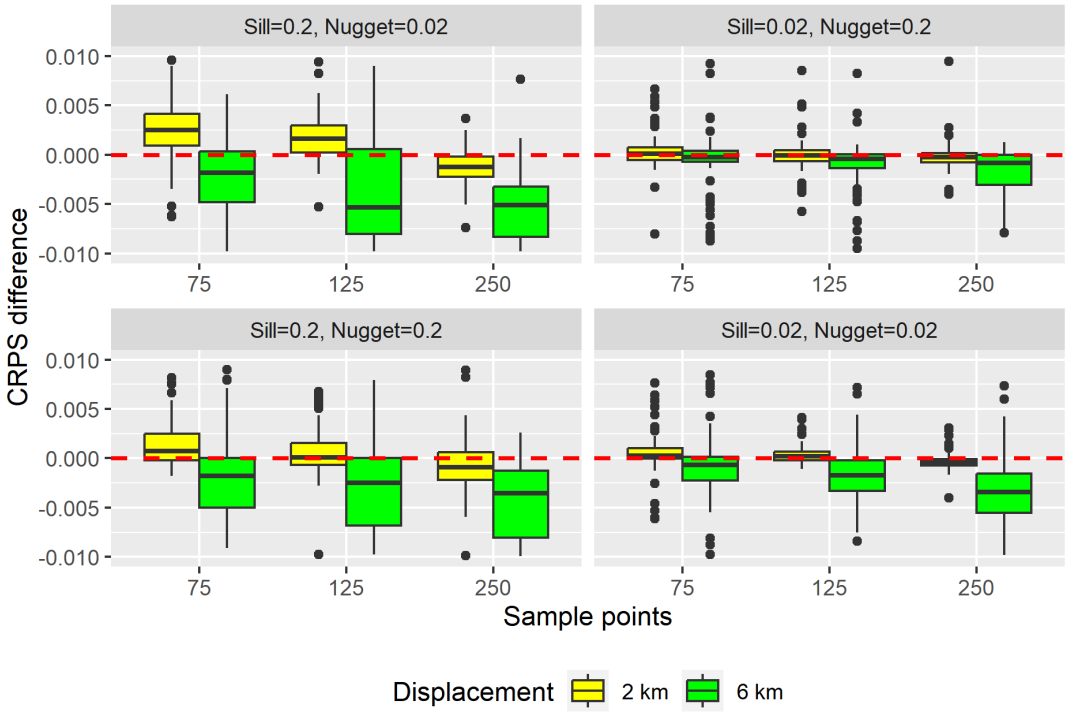

Figure 5.6: Pairwise CRPS difference

Chapter 6

Discussion and Conclusion

Geographical masking through random perturbation over a circular region has been shown to lead to biased parameter estimates and poor spatial predictions. In this thesis, we reviewed methods designed to address the geomasking problem, providing a brief evaluation of each, with a particular focus on `GeoAdjust` developed by Altay, Paige, Riebler, and Fuglstad (2023). We implemented `GeoAdjust` on the Philippines DHS dataset and compared the results with those from a naive model. Although differences were observed, the true data were unknown, prompting a simulation study to evaluate performance. Various scenarios were created to assess the flexibility and effectiveness of `GeoAdjust`. Simulation results indicated that `GeoAdjust` performed better with a sample size of approximately 250 locations, a displacement of 6km, and a larger spatial variance. In this chapter, we presented possible justifications and limitations of these results, together with guidance and suggestions for future work.

To begin with, we found in both the case study and simulation study that the overall mean μ was estimated similarly by `GeoAdjust` and by the naive model that ignores location error. This indicated that μ was unaffected by geomasking, aligning with the theory (Fronterrè et al., 2018). Since μ did not depend on the distances, introducing error in the locations did not impact μ. In practice, the focus is often not only on μ but also on the effects of various covariates. Although this study did not explicitly model covariate effects, the results for μ suggest that geomasking primarily affects distance-based covariates. Indeed, a study on raster-based covariates found that the strength of the association of spatially-structured covariates was attenuated by geomasking, thereby affecting prediction performance as well (Altay et al., 2024). However, whether `GeoAdjust` can estimate covariate effects correctly at different scenarios remained to be seen.

In terms of covariance parameters, we obtained a range of results and observations from our analyses. Consistent with the findings of Altay et al. (2024), we found that larger displacements led to better covariance parameter estimates with `GeoAdjust`. Specifically, a 6km displacement compared to a 2km displacement yielded better bias ratios against the naive model. This can be attributed to the fact that larger displacements cause more significant disruption in spatial associations, allowing `GeoAdjust` to recover more effectively. On the other hand, with smaller displacements like 2km, the disruption is less pronounced, which resulted in estimates that were closer to those of the naive model. We also observed that larger sample sizes were optimal,

with 250 sample locations producing the smallest biases for the parameter estimates. This made sense because the model included a layered latent effect, which introduced more noise and required more information for a statistical signal to be detected. This layered latent effect was similar to the mechanism of preferential sampling in spatial models. In spatial data with preferential samples, Diggle et al. (2010) noted that a larger sample size is needed compared to random samples due to these latent effects. We considered extending the maximum sample size to 500, but most of the runs failed. The crashes may have been caused by computational burden or other factors, possibly including the high density of points in a small area leading to overlapping geomasking regions, which may had further distorted the spatial structure. It prevented us from evaluating the performances for sample locations above 250. This could be a potential extension of the simulation study to determine whether increasing the sample size continuously improves the results or if there is an optimal sample size. Our current analyses showed that improvements in some parameter estimates continued to increase with sample size, while others plateaued at a certain point. Another interesting area to explore is the impact of sampling design, such as the consideration of adaptive sampling (Chipeta et al., 2016). In the current case study and simulation study, the sample locations were randomly generated. Applying adaptive sampling could potentially improve the estimates and predictions, especially given the application of geomasking in the data.

The relative magnitude of spatial variance compared to nugget variance was found to impact the performance of `GeoAdjust`. We observed that `GeoAdjust` performed better in parameter estimation when the true spatial variance was greater than or equal to the true nugget variance. In contrast, only small improvements were seen when the nugget variance exceeded the spatial variance. This outcome was expected since the effects of geomasking are heavily related to the spatial component of the data. With more spatial variation, geomasking causes more distortion, requiring `GeoAdjust` to recover more information. However, the actual magnitudes of these variances are typically unknown beforehand, as was the case in the Philippines DHS where only the geomasked data were available. This raises the question of whether using `GeoAdjust` will be beneficial. To address this, it is advisable to first check for substantial spatial correlation in the data, as suggested by Diggle and Giorgi (2019). Even if the spatial correlation is weak, as in our case study where the p-value was not highly significant, we still observed some improvement, albeit small, in estimates when the spatial variance was much smaller than the nugget variance in the simulation study. Therefore, our recommendation is to apply `GeoAdjust` whenever there is substantial spatial structure in the data.

In spatial analysis, it is common that the interest is more on the effects of certain covariates and prediction maps, and less on covariance parameters itself (Diggle et al., 2003). Hence, our simulation study highlighted results on the predictions as well. Similar scenarios were found to be optimal for better spatial predictions by `GeoAdjust`, which were larger sample size, displacement, and spatial variance. Interestingly, however, we found that when RMSE was used to compare the predictive performances, the separations between the naive and `GeoAdjust` were not that clear. In other words, the point estimates of the predictions by both models were almost similar. But when uncertainty around the point estimates was taken into consideration,

through the use of CRPS as comparison, it was more vivid that `GeoAdjust` had better predictive performance.

One missing scenario in the simulation study was varying the spatial range. We kept it fixed at 8km to ensure it was larger than the 2km and 6km displacements, preventing the complete loss of spatial structure. Recent papers had suggested that the impact of spatial range was closely related to the displacement radius (Fronterrè et al., 2018, Altay et al., 2022), which was why we fixed the spatial range and varied the displacement radius. It is recommended to investigate whether parameter inferences and spatial predictions are affected by a smaller or larger spatial range. Additionally, exploring mesh size could be valuable and should be included in a simulation study. It should be noted that finer meshes lead to heavier computational burden. Prior sensitivity analysis can also be conducted, as there were concerns about potential overfitting towards the base model with PC priors (Bakka et al., 2018).

Overall, the results from the case study and simulation study pointed to the superiority of `GeoAdjust` against the naive when geomasking was introduced in the data. The best scenarios for `GeoAdjust` were data with larger sample sizes, larger displacements, and larger spatial variance. Most of these scenarios were not explored before by the authors of the method. Therefore, analysts of geomasked data should consider the sample size, displacement, and if possible spatial variation of the data before applying `GeoAdjust`. Additionally, spatial data owners who wish to apply geomasking should exercise caution. Our results indicated that to recover as much spatial information as possible, the optimal scenarios were necessary.

In the current literature, we found that `GeoAdjust` was the better and faster option, but also there were other fast techniques like the composite likelihood approach proposed by Fronterrè et al. (2018). The composite likelihood approach was limited to point estimates of the parameters and did not have an associated `R` package. Future development by the authors could provide more options for analyzing geomasked data. Additionally, new methods can be developed to address some of the weaknesses of `GeoAdjust`. `GeoAdjust` was designed for DHS datasets, which meant it is constrained to the DHS displacement procedure only. Researchers can use our findings to develop these new methods. Beyond the displacement over a circular region considered in this thesis and by the methods presented, various other forms of geomasking are also applied in the real world. Significant research gaps still exist, presenting opportunities for further investigation.

In conclusion, we reviewed current methods for geomasking and selected `GeoAdjust` for further analysis. Through a case study and simulation study, we obtained interesting results and identified scenarios where `GeoAdjust` performed best. We also provided recommendations and suggested potential directions for further research. The findings of this research offer valuable insights into the broader study of geomasking and location error.

Bibliography

Albertsen, C. M., Whoriskey, K., Yurkowski, D., Nielsen, A., & Flemming, J. M. (2015). Fast fitting of non-gaussian state-space models to animal movement data via template model builder.

Allshouse, W. B., Fitch, M. K., Hampton, K. H., Gesink, D. C., Doherty, I. A., Leone, P. A., Serre, M. L., & Miller, W. C. (2010). Geomasking sensitive health data and privacy protection: An evaluation using an e911 database. *Geocarto international*, *25*(6), 443–452.

Altay, U., Paige, J., Riebler, A., & Fuglstad, G. (2023). Submitted to R Journal on 21.03. 2023. *Geostatistical Analysis of DHS Data: Accounting for Random Displacement of Survey Locations*, 75.

Altay, U., Paige, J., Riebler, A., & Fuglstad, G.-A. (2022). Fast geostatistical inference under positional uncertainty: Analysing dhs household survey data. *arXiv preprint arXiv:2202.11035*.

Altay, U., Paige, J., Riebler, A., & Fuglstad, G.-A. (2023). Geoadjust: Adjusting for positional uncertainty in geostatistial analysis of dhs data. *arXiv preprint arXiv:2303.12668*.

Altay, U., Paige, J., Riebler, A., & Fuglstad, G.-A. (2024). Impact of jittering on raster-and distance-based geostatistical analyses of dhs data. *Statistical Modelling*, 1471082X231219847.

Arbia, G., Ghiringhelli, C., & Nardelli, V. (2023). Effects of confidentiality-preserving geo-masking on the estimation of semivariogram and of the kriging variance. *Geographical Analysis*, *55*(3), 466–481.

Armstrong, M. P., Rushton, G., & Zimmerman, D. L. (1999). Geographically masking health data to preserve confidentiality. *Statistics in Medicine*, *18*(5), 497–525.

Auger-Méthé, M., Albertsen, C. M., Jonsen, I. D., Derocher, A. E., Lidgard, D. C., Studholme, K. R., Bowen, W. D., Crossin, G. T., & Flemming, J. M. (2017). Spatiotemporal modelling of marine movement data using template model builder (tmb). *Marine Ecology Progress Series*, *565*, 237–249.

Bakka, H., Rue, H., Fuglstad, G.-A., Riebler, A., Bolin, D., Illian, J., Krainski, E., Simpson, D., & Lindgren, F. (2018). Spatial modeling with r-inla: A review. *Wiley Interdisciplinary Reviews: Computational Statistics*, *10*(6), e1443.

Bertino, E., Thuraisingham, B., Gertz, M., & Damiani, M. L. (2008). Security and privacy for geospatial data: Concepts and research directions. *Proceedings of the SIGSPATIAL ACM GIS 2008 International Workshop on Security and Privacy in GIS and LBS*, 6–19.

Bevilacqua, M., & Gaetan, C. (2015). Comparing composite likelihood methods based on pairs for spatial gaussian random fields. *Statistics and Computing*, *25*, 877–892.

Blangiardo, M., Cameletti, M., Baio, G., & Rue, H. (2013). Spatial and spatio-temporal models with R-INLA. *Spatial and Spatio-temporal Epidemiology*, *4*, 33–49.

Burgert, C. R., Colston, J., Roy, T., & Zachary, B. (2013). *Geographic displacement procedure and georeferenced data release policy for the demographic and health surveys.* ICF International.

Chipeta, M. G., Terlouw, D. J., Phiri, K. S., & Diggle, P. J. (2016). Adaptive geostatistical design and analysis for prevalence surveys. *Spatial Statistics*, *15*, 70–84.

Corsi, D. J., Neuman, M., Finlay, J. E., & Subramanian, S. (2012). Demographic and health surveys: A profile. *International Journal of Epidemiology*, *41*(6), 1602–1613.

Cressie, N., & Kornak, J. (2003). Spatial statistics in the presence of location error with an application to remote sensing of the environment. *Statistical Science*, 436–456.

Diggle, P. J., & Giorgi, E. (2019). *Model-based geostatistics for global public health: Methods and applications.* Chapman; Hall/CRC.

Diggle, P. J., Menezes, R., & Su, T.-l. (2010). Geostatistical inference under preferential sampling. *Journal of the Royal Statistical Society Series C: Applied Statistics*, *59*(2), 191–232.

Diggle, P. J., Ribeiro, P. J., & Christensen, O. F. (2003). An introduction to model-based geostatistics. *Spatial statistics and computational methods*, 43–86.

Diggle, P. J., Tawn, J. A., & Moyeed, R. A. (1998). Model-based geostatistics. *Journal of the Royal Statistical Society Series C: Applied Statistics*, *47*(3), 299–350.

Fanshawe, T., & Diggle, P. (2011). Spatial prediction in the presence of positional error. *Environmetrics*, *22*(2), 109–122.

Fronterrè, C., Giorgi, E., & Diggle, P. (2018). Geostatistical inference in the presence of geomasking: A composite-likelihood approach. *Spatial Statistics*, *28*, 319–330.

Fuglstad, G.-A., Simpson, D., Lindgren, F., & Rue, H. (2019). Constructing priors that penalize the complexity of gaussian random fields. *Journal of the American Statistical Association*, *114*(525), 445–452.

Gabrosek, J., & Cressie, N. (2002). The effect on attribute prediction of location uncertainty in spatial data. *Geographical Analysis*, *34*(3), 262–285.

Gao, S., Rao, J., Liu, X., Kang, Y., Huang, Q., & App, J. (2019). Exploring the effectiveness of geomasking techniques for protecting the geoprivacy of twitter users. *Journal of Spatial Information Science*, (19), 105–129.

Gelfand, A. E., & Banerjee, S. (2017). Bayesian modeling and analysis of geostatistical data. *Annual Review of Rtatistics and its Application*, *4*, 245–266.

Goldberg, D. W., & Cockburn, M. G. (2012). The effect of administrative boundaries and geocoding error on cancer rates in california. *Spatial and Spatio-temporal Epidemiology*, *3*(1), 39–54.

Gómez-Rubio, V., & Rue, H. (2018). Markov chain monte carlo with the integrated nested laplace approximation. *Statistics and Computing*, *28*, 1033–1051.

Jacquez, G. M. (2012). A research agenda: Does geocoding positional error matter in health gis studies? *Spatial and spatio-temporal epidemiology*, *3*(1), 7–16.

Kelly, C., Hulme, C., Farragher, T., & Clarke, G. (2016). Are differences in travel time or distance to healthcare for adults in global north countries associated with an impact on health outcomes? a systematic review. *BMJ Open*, *6*(11), e013059.

Kinnee, E. J., Tripathy, S., Schinasi, L., Shmool, J. L., Sheffield, P. E., Holguin, F., & Clougherty, J. E. (2020). Geocoding error, spatial uncertainty, and implications for exposure assessment

and environmental epidemiology. *International Journal of Environmental Research and Public Health*, *17*(16), 5845.

Kristensen, K., Nielsen, A., Berg, C., Skaug, H., & Bell, B. (2015). Template model builder TMB. *Journal of Statistical Software*, *70*, 1–21.

Matheson, J. E., & Winkler, R. L. (1976). Scoring rules for continuous probability distributions. *Management Science*, *22*(10), 1087–1096.

Moraga, P. (2019). *Geospatial health data: Modeling and visualization with R-INLA and Shiny.* Chapman; Hall/CRC.

Moraga, P. (2023). *Spatial Statistics for Data Science: Theory and Practice with R.* CRC Press.

Osgood-Zimmerman, A., & Wakefield, J. (2023). A statistical review of template model builder: A flexible tool for spatial modelling. *International Statistical Review*, *91*(2), 318–342.

Pfeiffer, D. U., Robinson, T. P., Stevenson, M., Stevens, K. B., Rogers, D. J., & Clements, A. C. (2008). *Spatial analysis in epidemiology.* OUP Oxford.

PhilAtlas. (2024). National Capital Region (NCR) [Accessed: 2024-05-28]. https://www.philatlas.com/luzon/ncr.html

Philippine Statistics Authority and ICF. (2022). 2022 Philippine National Demographic and Health Survey (NDHS): Key Indicators Report.

Philippine Statistics Authority and ICF. (2023). 2022 Philippine National Demographic and Health Survey (NDHS): Final Report.

Porio, E. E., Yulo-Loyzaga, A., & Uy, C. (2019). Metro manila.

Redding, S. J., & Rossi-Hansberg, E. (2017). Quantitative spatial economics. *Annual Review of Economics*, *9*, 21–58.

Righetto, A. J., Faes, C., Vandendijck, Y., & Ribeiro Jr, P. J. (2020). On the choice of the mesh for the analysis of geostatistical data using r-inla. *Communications in Statistics-Theory and Methods*, *49*(1), 203–220.

Simpson, D., Rue, H., Riebler, A., Martins, T. G., & Sørbye, S. H. (2017). Penalising model component complexity: A principled, practical approach to constructing priors.

Stein, M. L., Chi, Z., & Welty, L. J. (2004). Approximating likelihoods for large spatial data sets. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, *66*(2), 275–296.

The DHS Program. (2024). Sustainable development goals (sdgs) [Accessed: 2024-05-22]. https://dhsprogram.com/topics/sdgs/index.cfm

Varin, C., Reid, N., & Firth, D. (2011). An overview of composite likelihood methods. *Statistica Sinica*, 5–42.

White, E. P., Ernest, S. M., Adler, P. B., Hurlbert, A. H., & Lyons, S. K. (2010). Integrating spatial and temporal approaches to understanding species richness. *Philosophical Transactions of the Royal Society B: Biological Sciences*, *365*(1558), 3633–3643.

Wilson, K., & Wakefield, J. (2021). Estimation of health and demographic indicators with incomplete geographic information. *Spatial and Spatio-temporal Epidemiology*, *37*, 100421.

Zandbergen, P. A., et al. (2014). Ensuring confidentiality of geocoded health data: Assessing geographic masking strategies for individual-level data. *Advances in Medicine*, *2014*.

Appendices

Appendix A

Additional Results for the Case Study

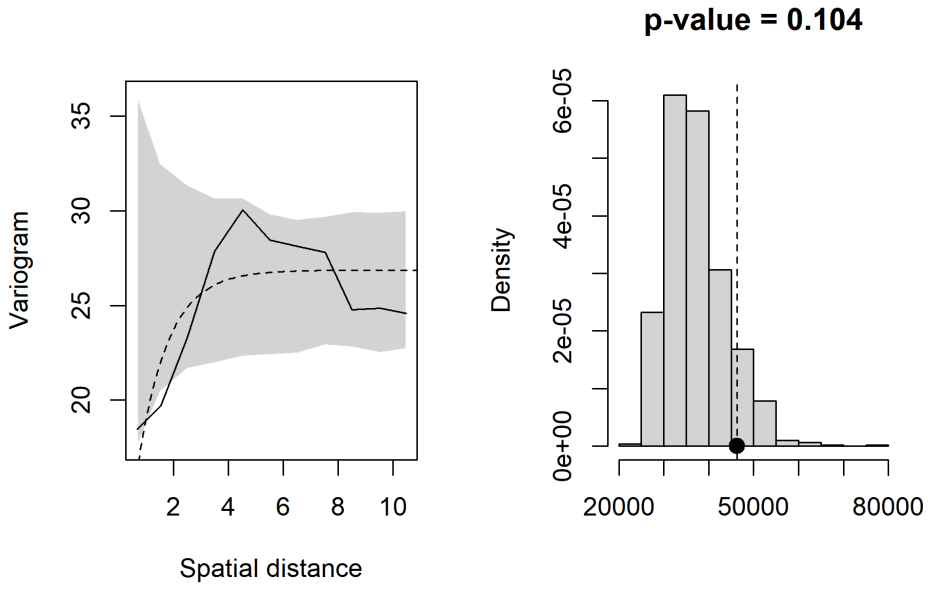

Figure A.1: Results for test of residual spatial correlation: 11km

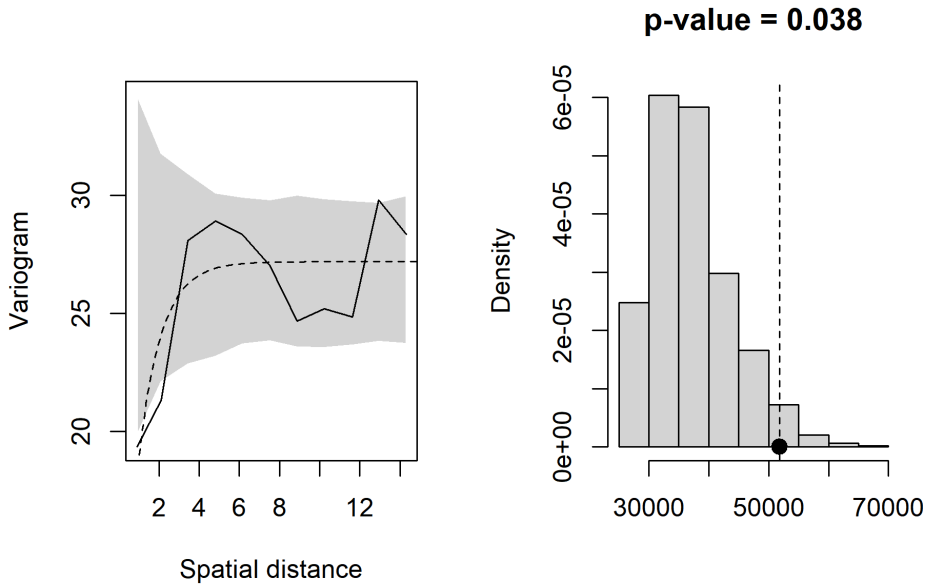

Figure A.2: Results for test of residual spatial correlation: 15km

Parameter	Mesh A		Mesh B	
	Naive	**GeoAdjust**	**Naive**	**GeoAdjust**
μ	2.6022 (2.49, 2.71)	2.5966 (2.48, 2.71)	2.6025 (2.49, 2.71)	2.5968 (2.49, 2.70)
σ^2	0.0351	0.0289	0.0348	0.0290
ϕ	4.7818	6.2253	4.8797	6.1042
τ^2	0.1109	0.1196	0.1113	0.1164
Time elapsed (in sec)	16.69	42.14	20.77	92.32

Table A.1: Parameter estimates for the Naive and `GeoAdjust` model with different meshes: Mesh A (a=1, b=3, c=0.5) and Mesh B (a=1, b=3, c=0.35)

Parameter	Prior A		Prior B	
	Naive	**GeoAdjust**	**Naive**	**GeoAdjust**
μ	2.6022 (2.49, 2.71)	2.5966 (2.48, 2.71)	2.6022 (2.49, 2.71)	2.5966 (2.48, 2.71)
σ^2	0.0351	0.0289	0.0351	0.0289
ϕ	4.7818	6.2253	4.7822	6.2196
τ^2	0.1109	0.1196	0.1108	0.1196
Time elapsed (in sec)	16.69	42.14	14.38	45.06

Table A.2: Parameter estimates for the Naive and `GeoAdjust` model under different priors: Prior A (σ_0=τ_0=1, $\rho_0 = 4$) and Prior B (σ_0=τ_0=1.2, $\rho_0 = 6$)

Appendix B

Additional Results for the Simulation Study

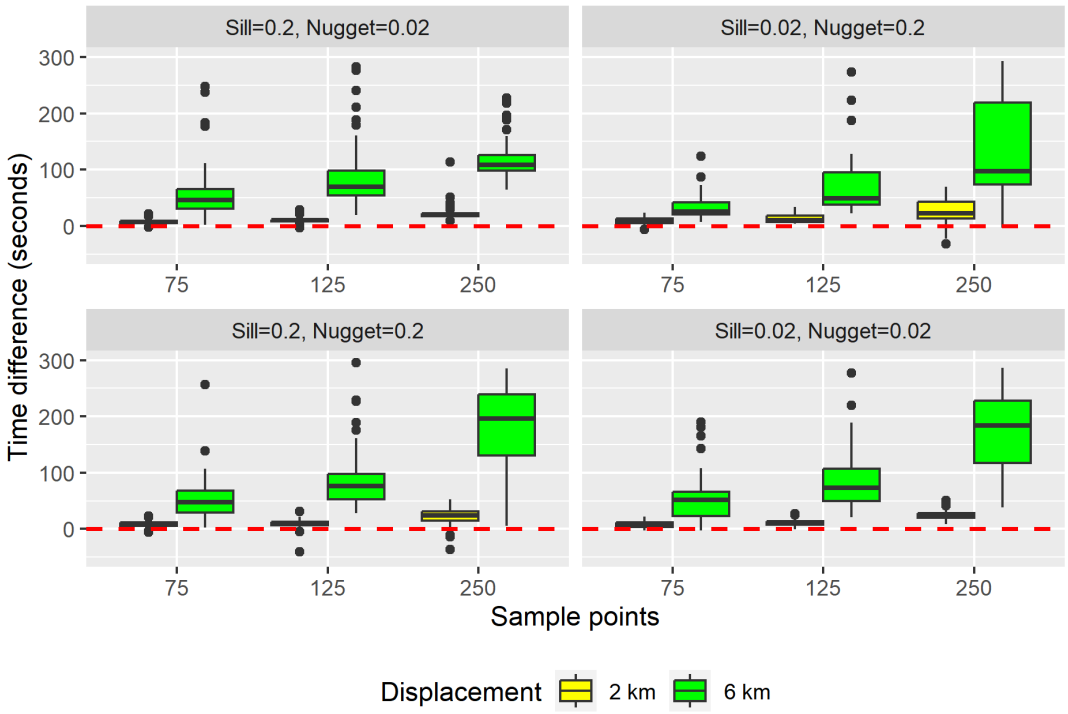

Figure B.1: Time difference of the two methods (in seconds)

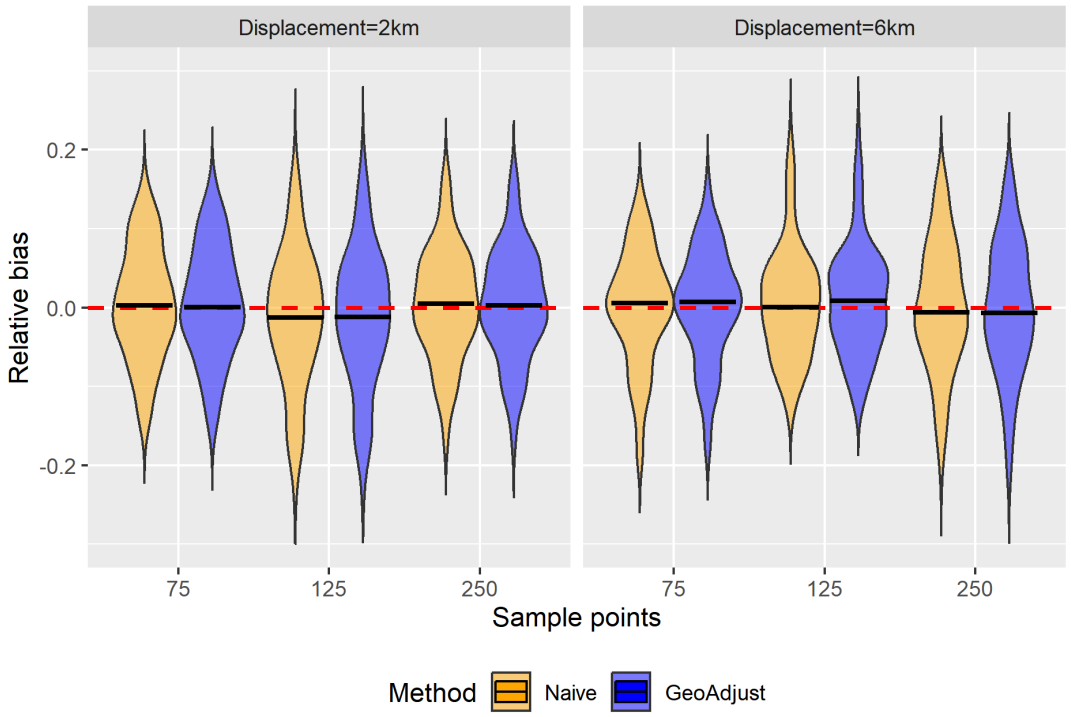

Figure B.2: Relative bias for μ: Sill=0.2, Nugget=0.02

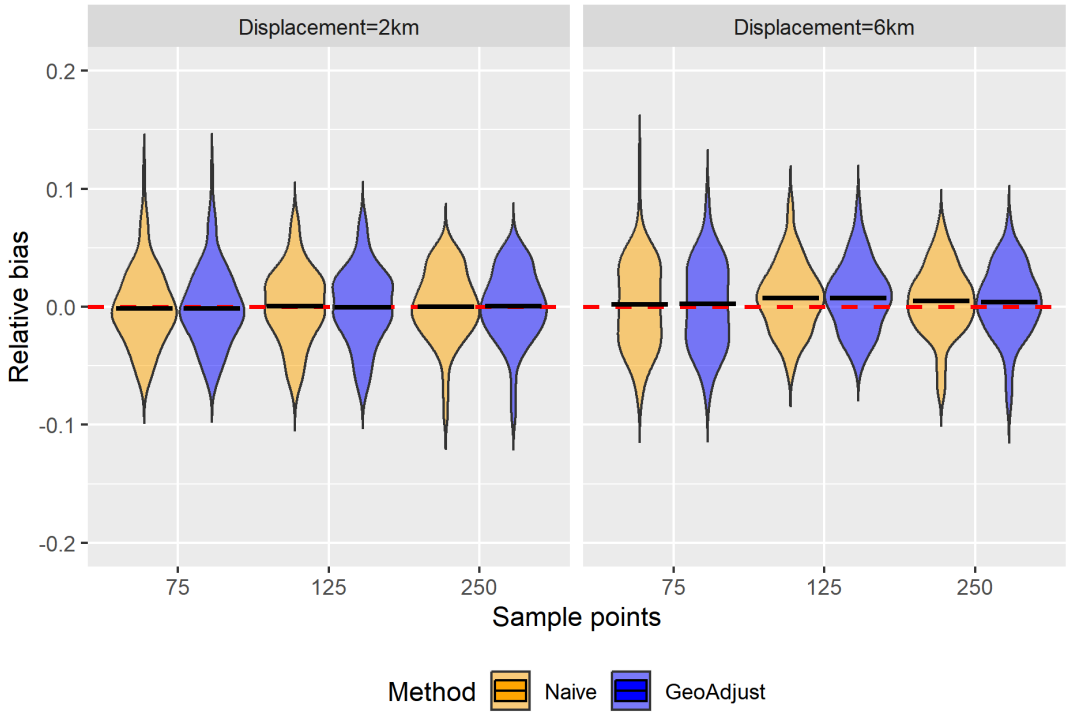

Figure B.3: Relative bias for μ: Sill=0.02, Nugget=0.2

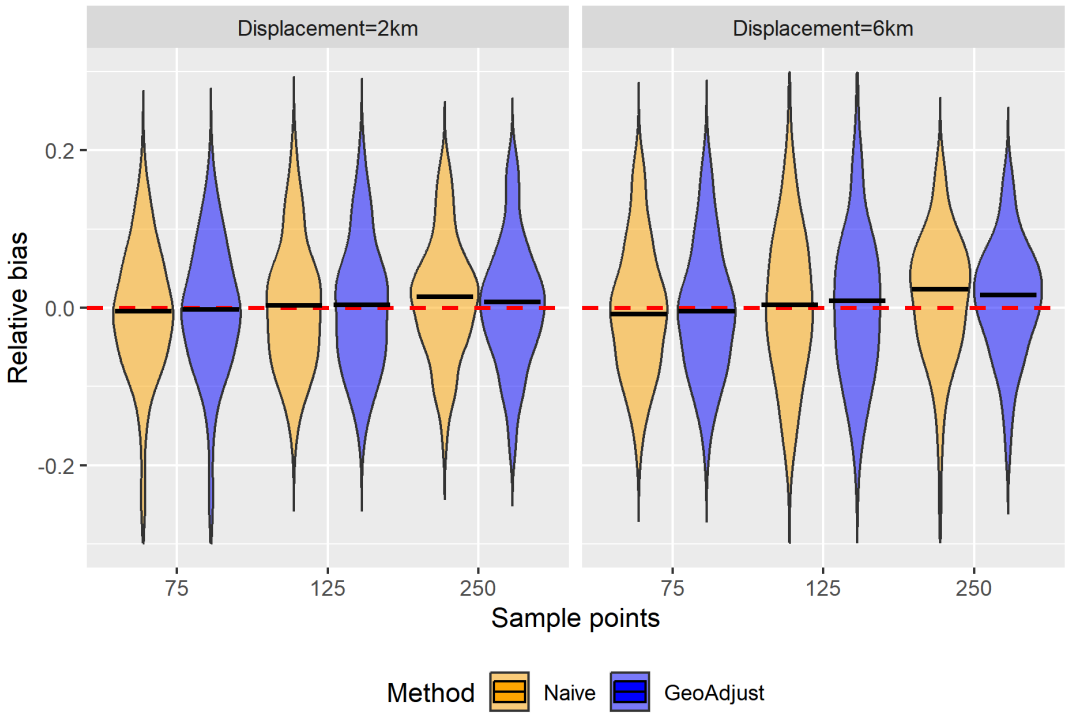

Figure B.4: Relative bias for μ: Sill=0.2, Nugget=0.2

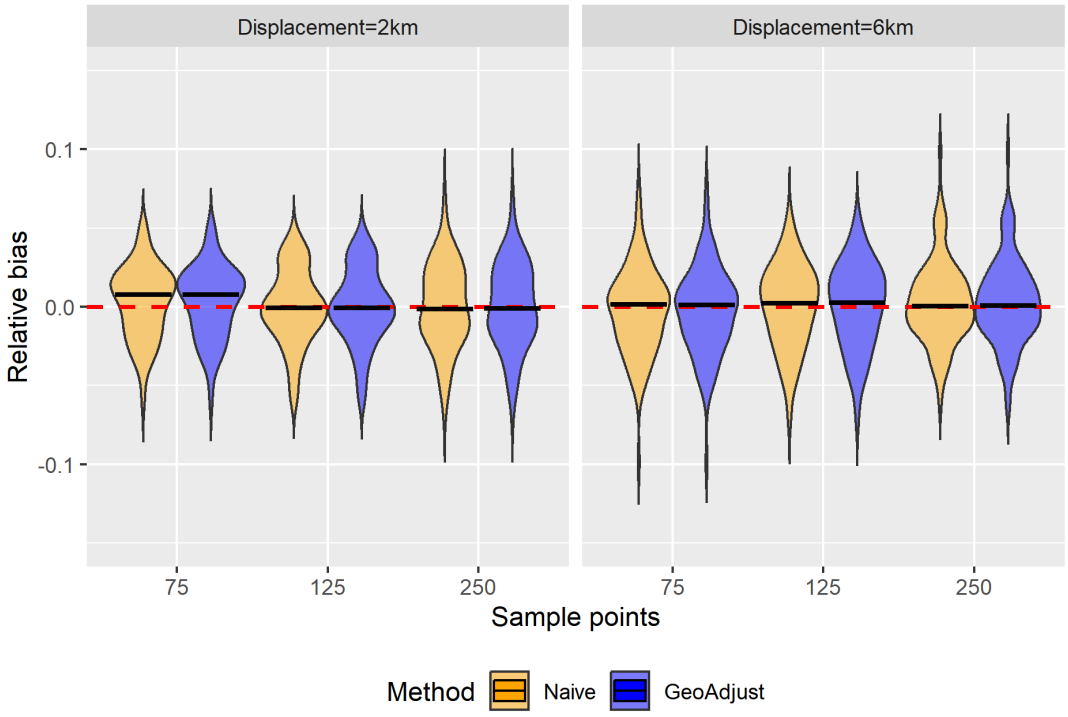

Figure B.5: Relative bias for μ: Sill=0.02, Nugget=0.02

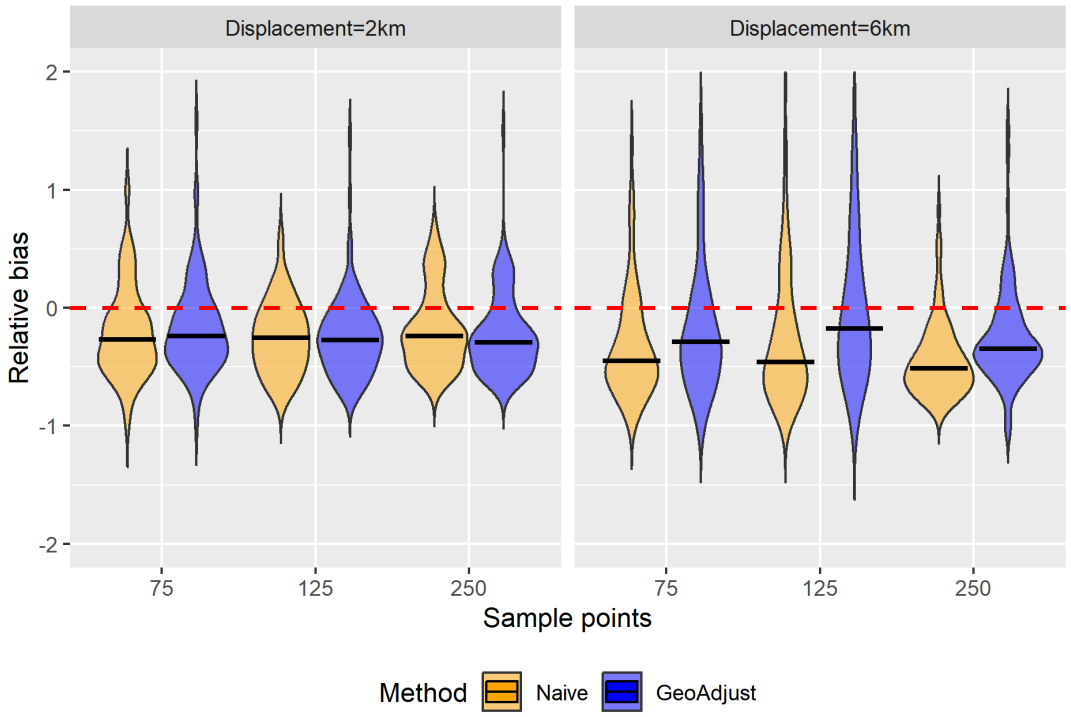

Figure B.6: Relative bias for σ^2: Sill=0.2, Nugget=0.02

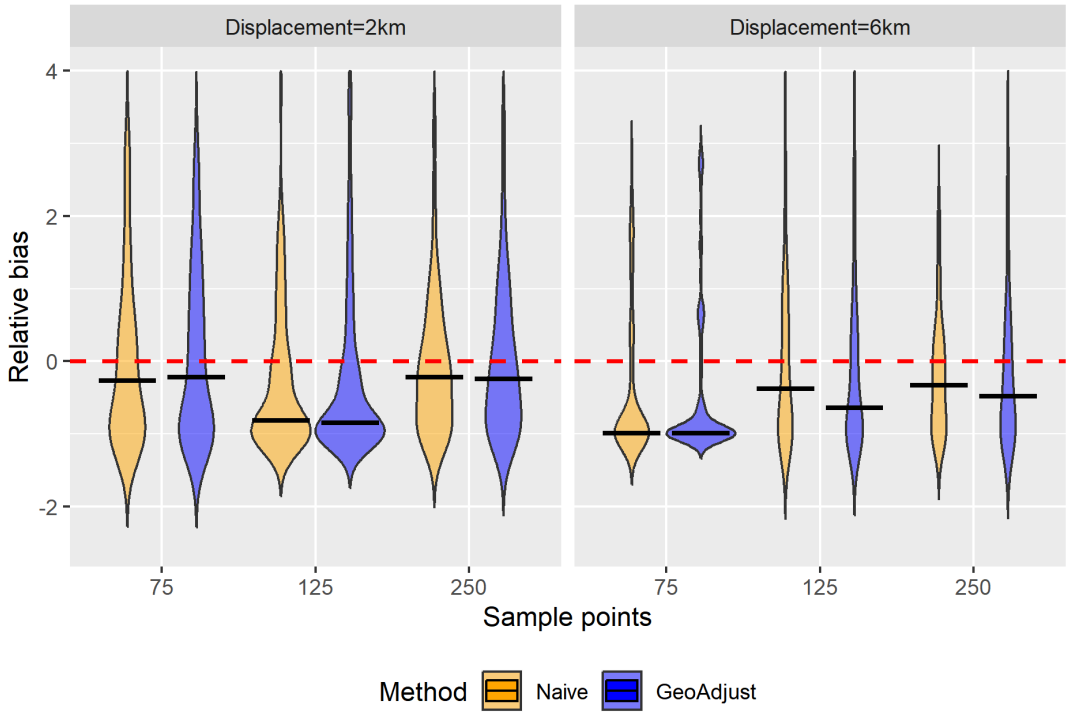

Figure B.7: Relative bias for σ^2: Sill=0.02, Nugget=0.2

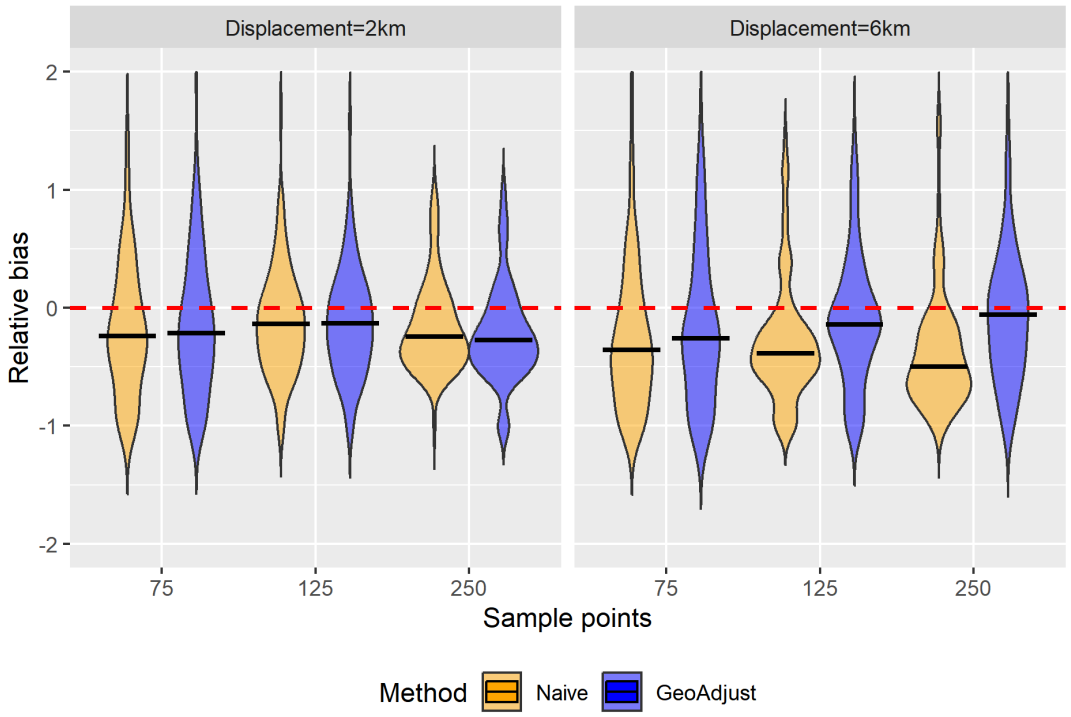

Figure B.8: Relative bias for σ^2: Sill=0.2, Nugget=0.2

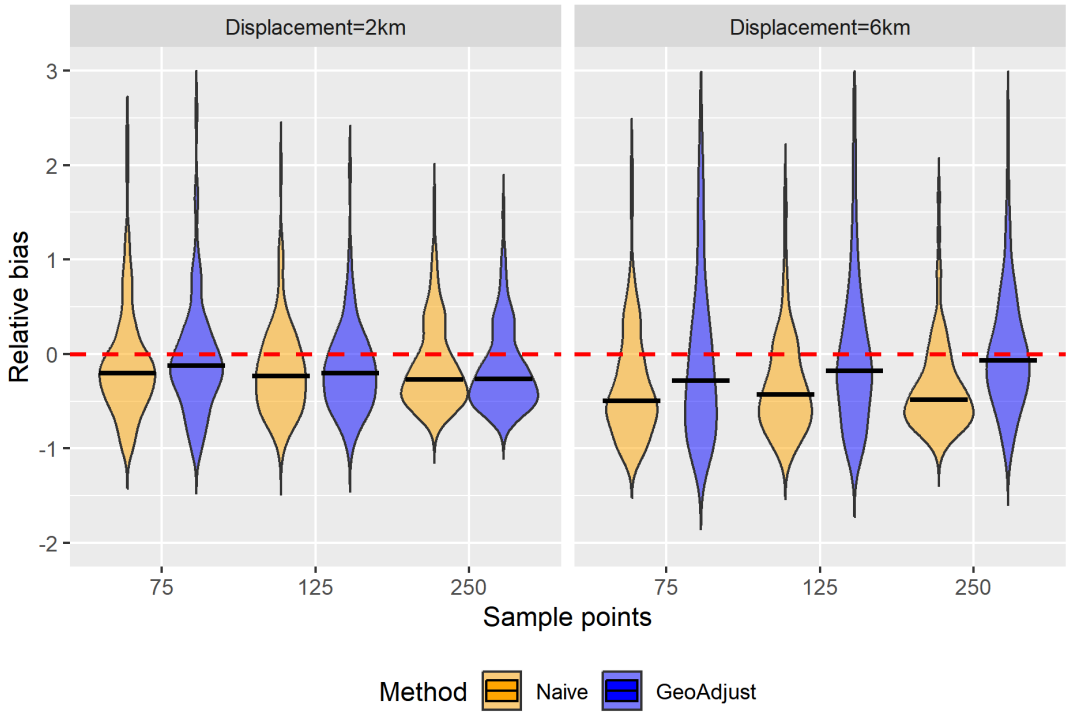

Figure B.9: Relative bias for σ^2: Sill=0.02, Nugget=0.02

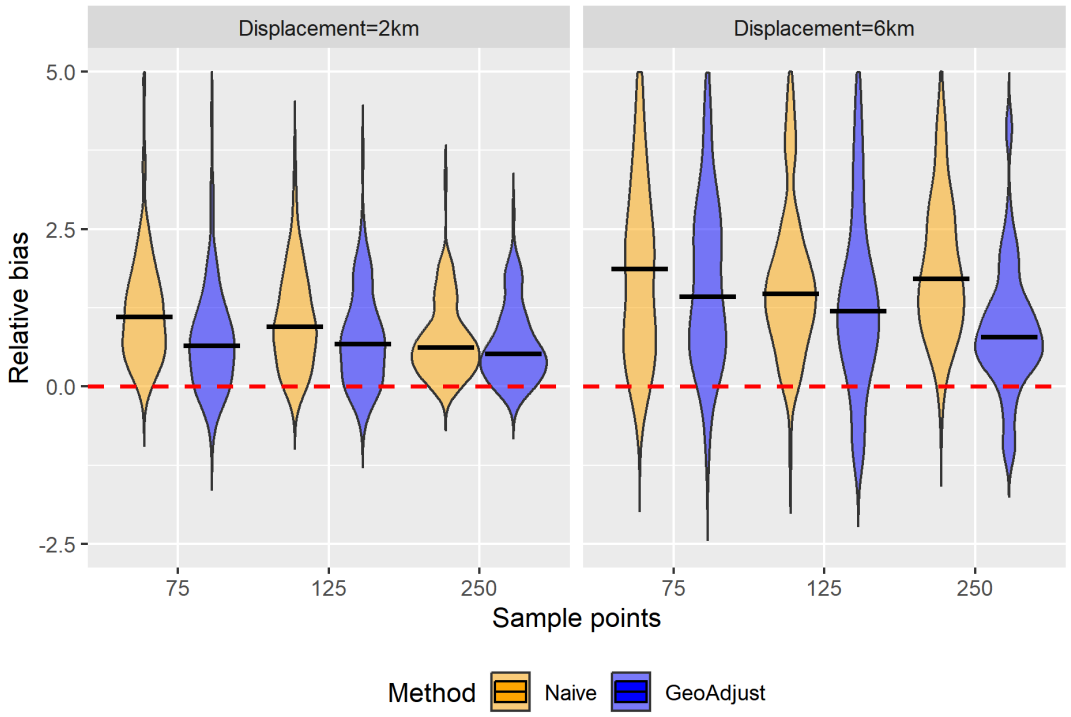

Figure B.10: Relative bias for ϕ: Sill=0.2, Nugget=0.02

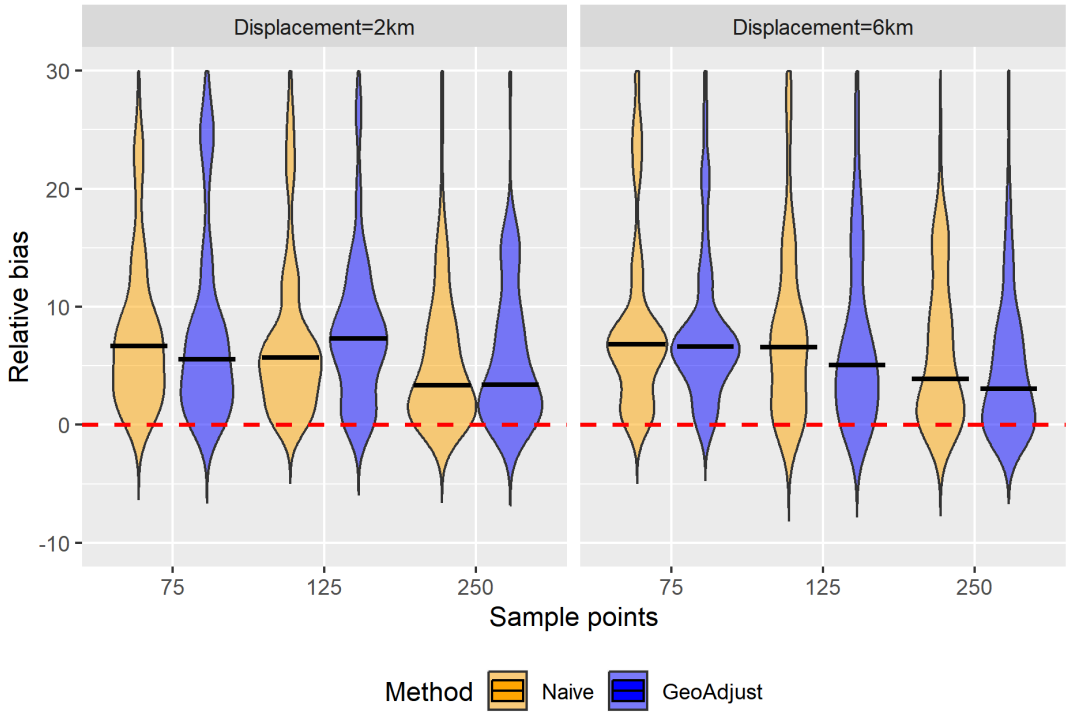

Figure B.11: Relative bias for ϕ: Sill=0.02, Nugget=0.2

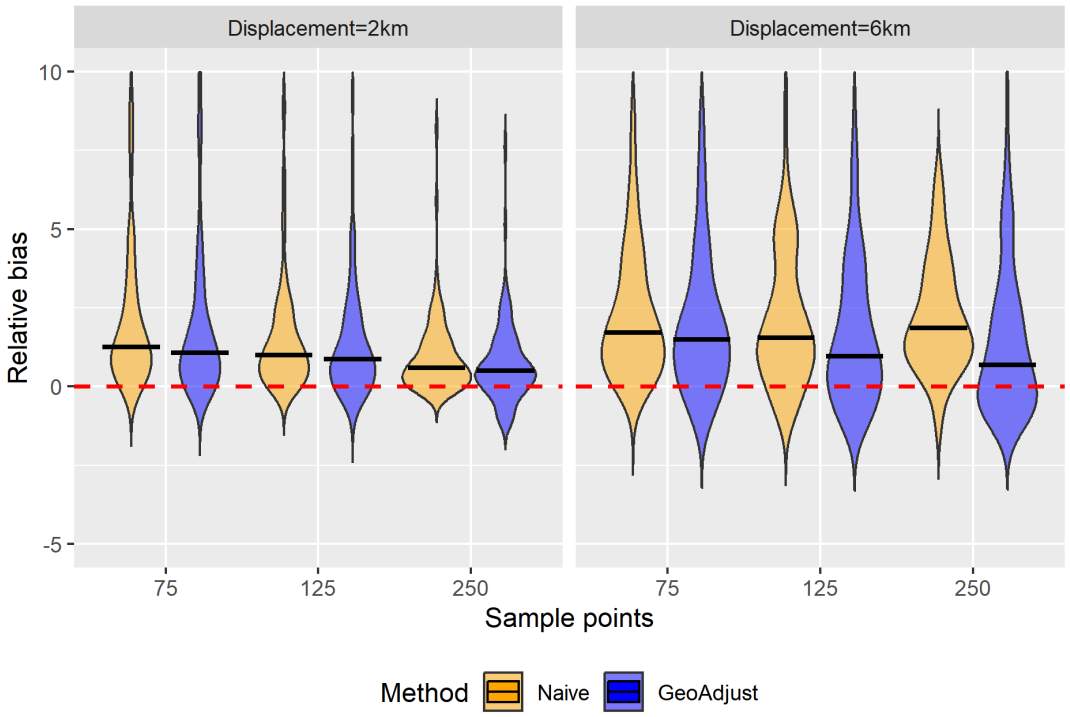

Figure B.12: Relative bias for ϕ: Sill=0.2, Nugget=0.2

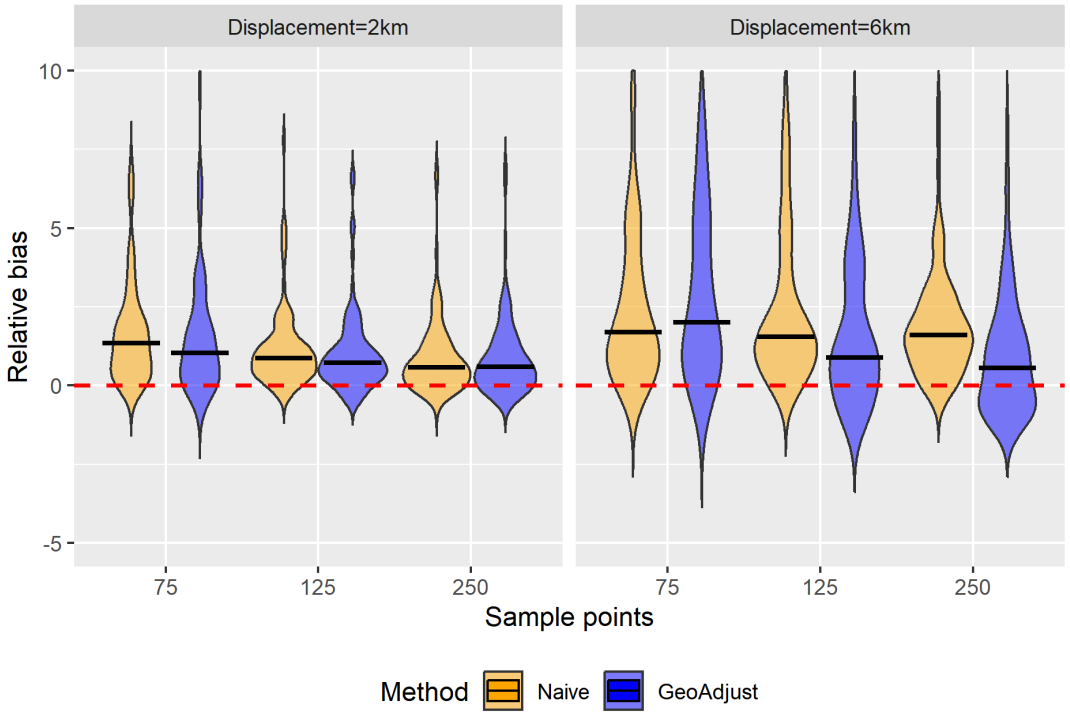

Figure B.13: Relative bias for ϕ: Sill=0.02, Nugget=0.02

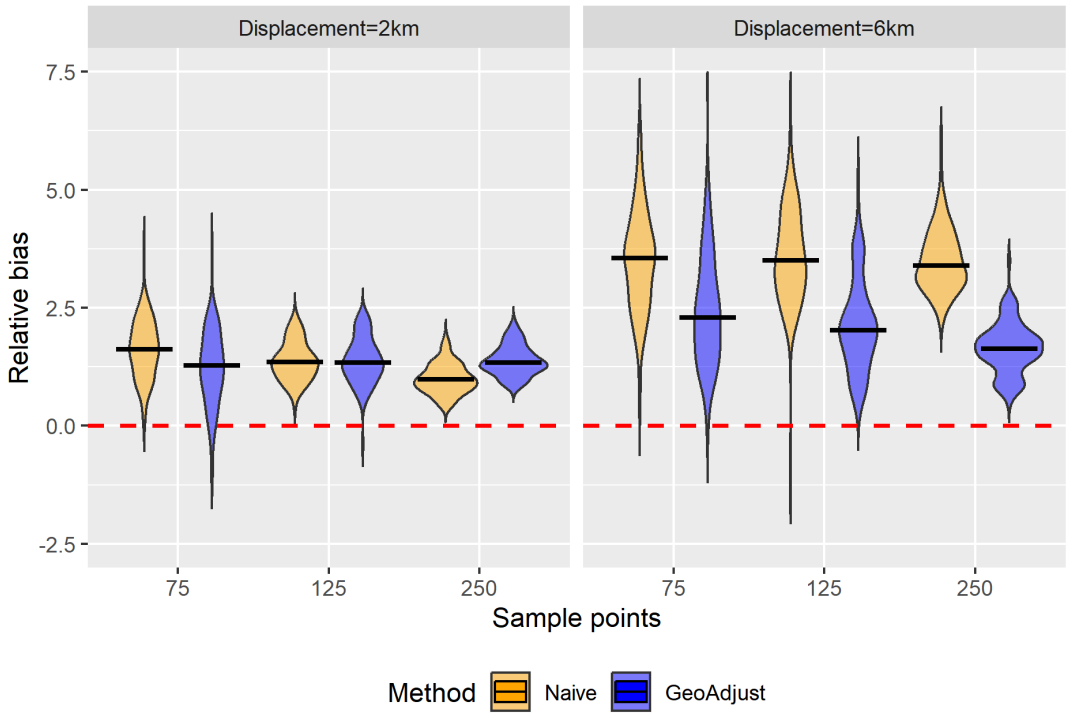

Figure B.14: Relative bias for τ^2: Sill=0.2, Nugget=0.02

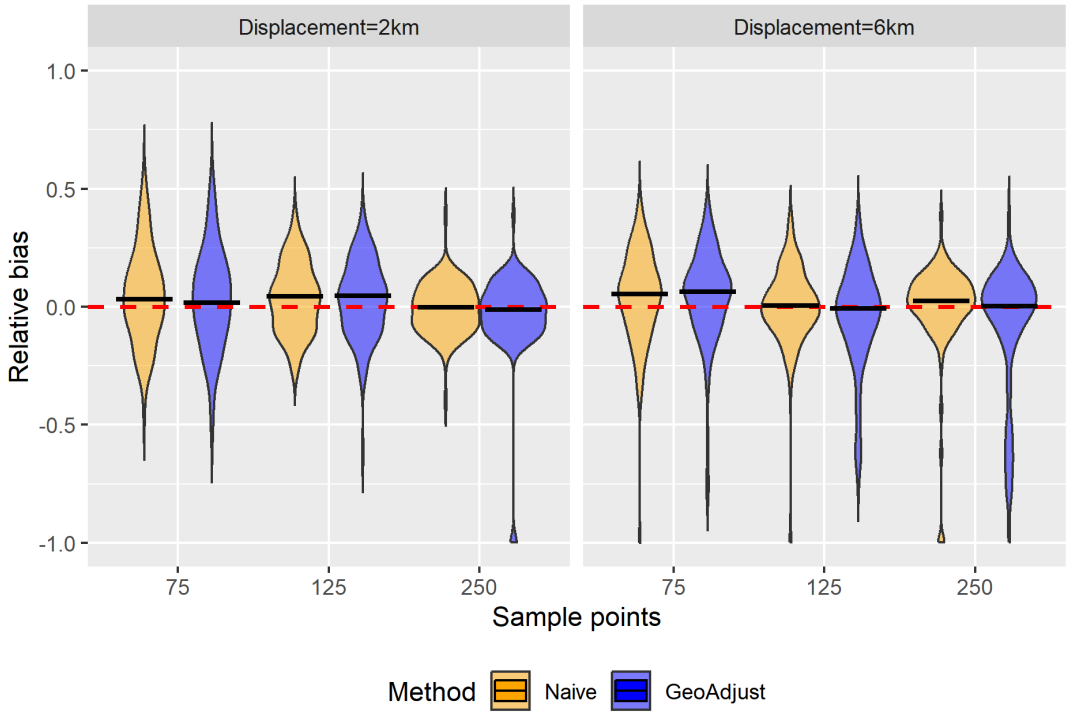

Figure B.15: Relative bias for τ^2: Sill=0.02, Nugget=0.2

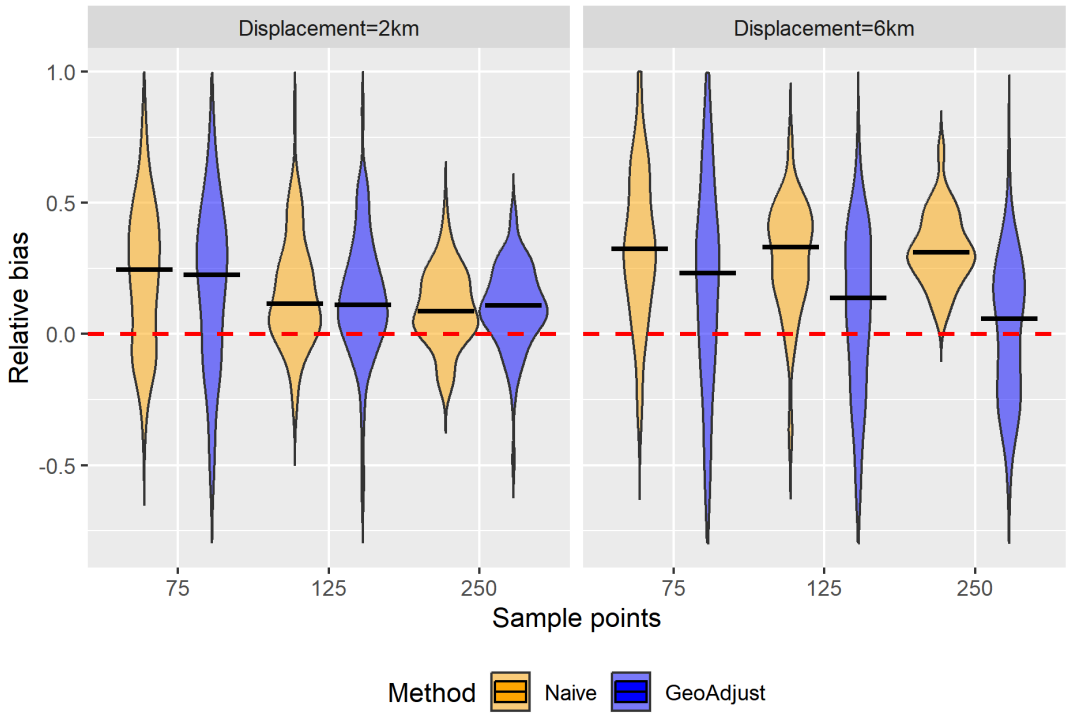

Figure B.16: Relative bias for τ^2: Sill=0.2, Nugget=0.2

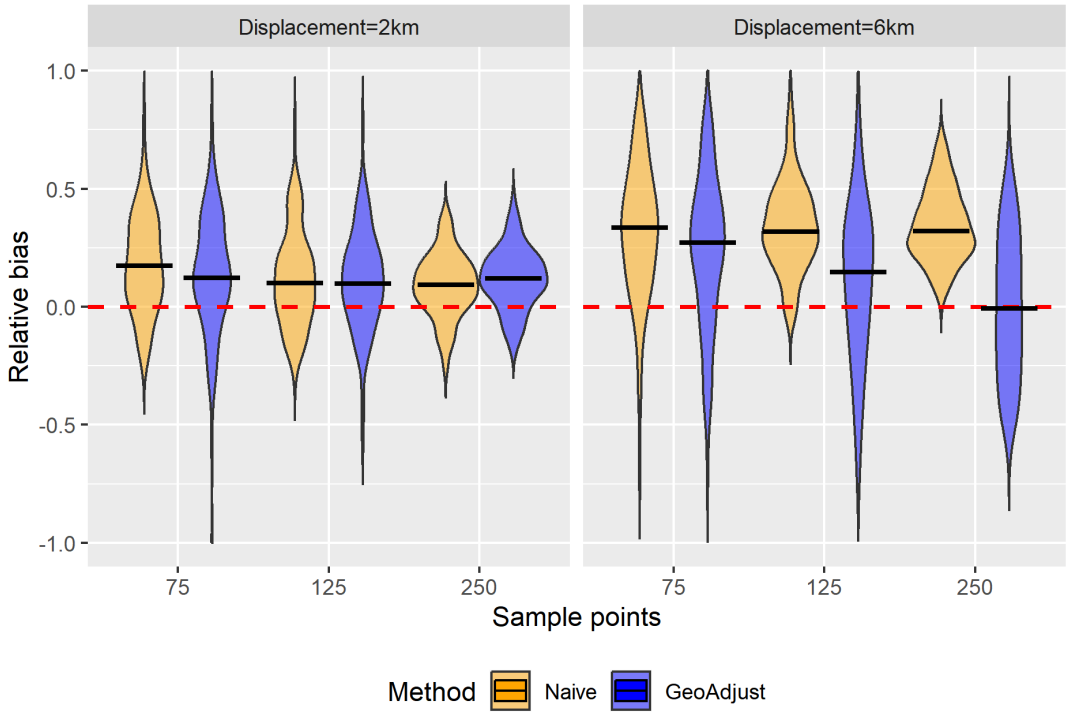

Figure B.17: Relative bias for τ^2: Sill=0.02, Nugget=0.02

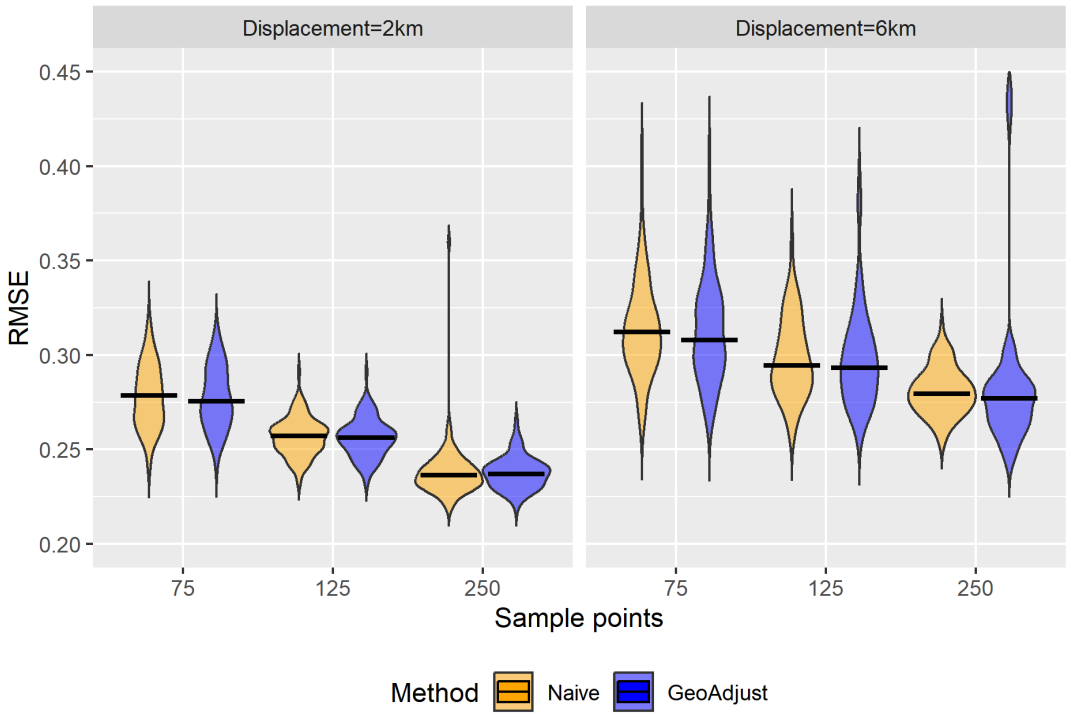

Figure B.18: RMSE: Sill=0.2, Nugget=0.02

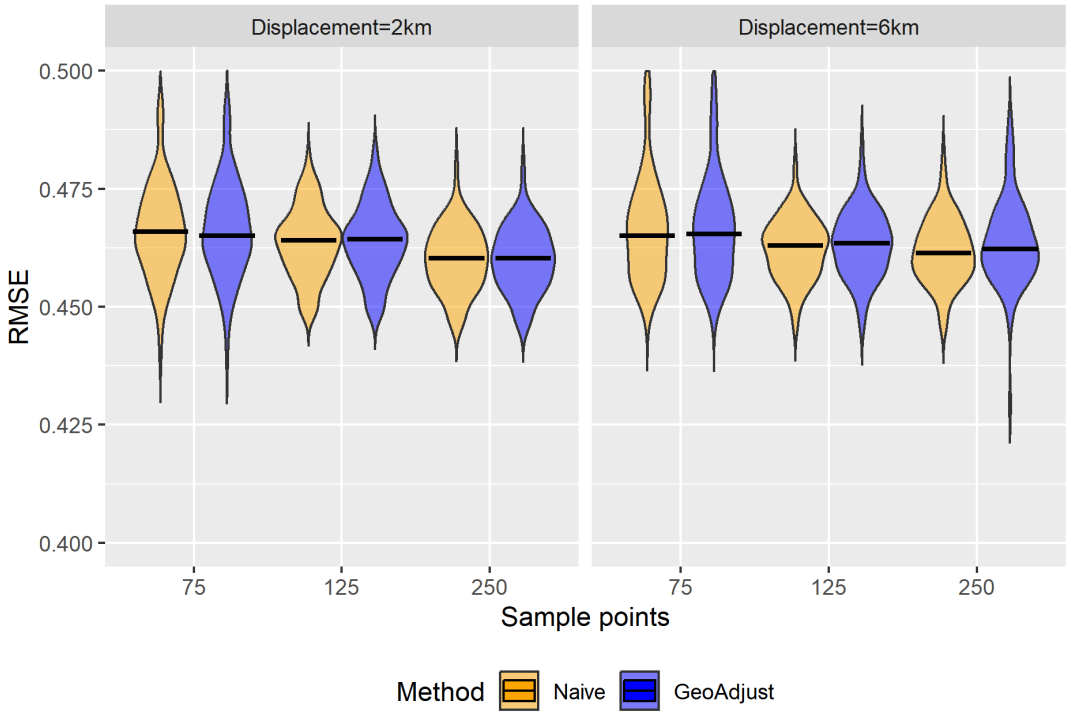

Figure B.19: RMSE: Sill=0.02, Nugget=0.2

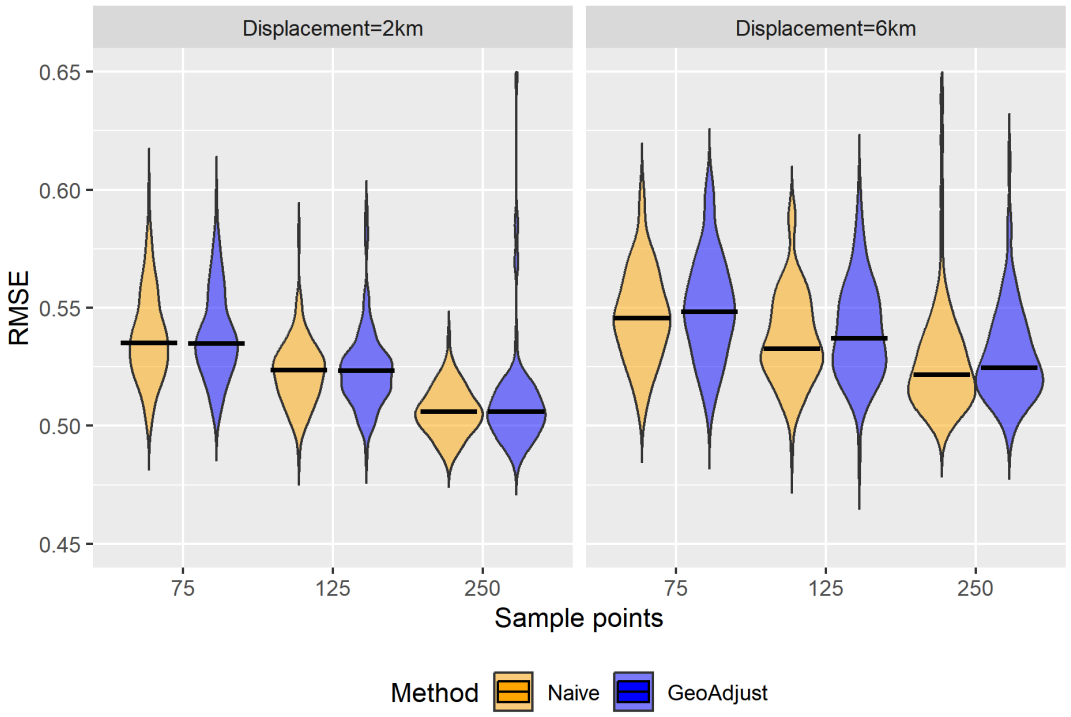

Figure B.20: RMSE: Sill=0.2, Nugget=0.2

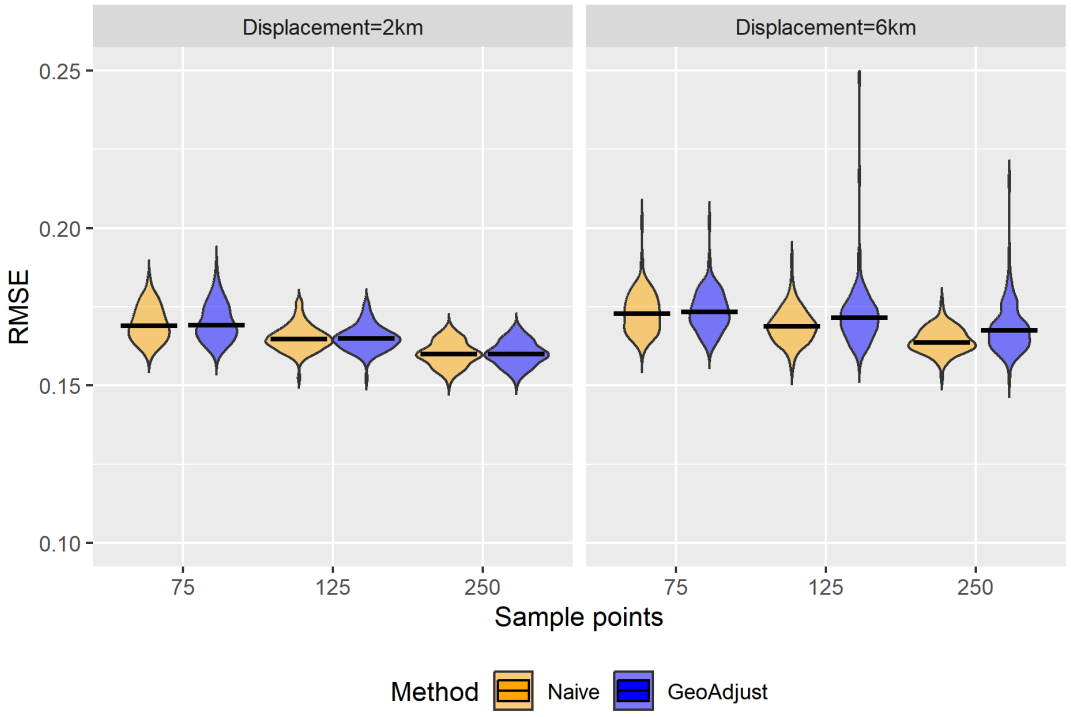

Figure B.21: RMSE: Sill=0.02, Nugget=0.02

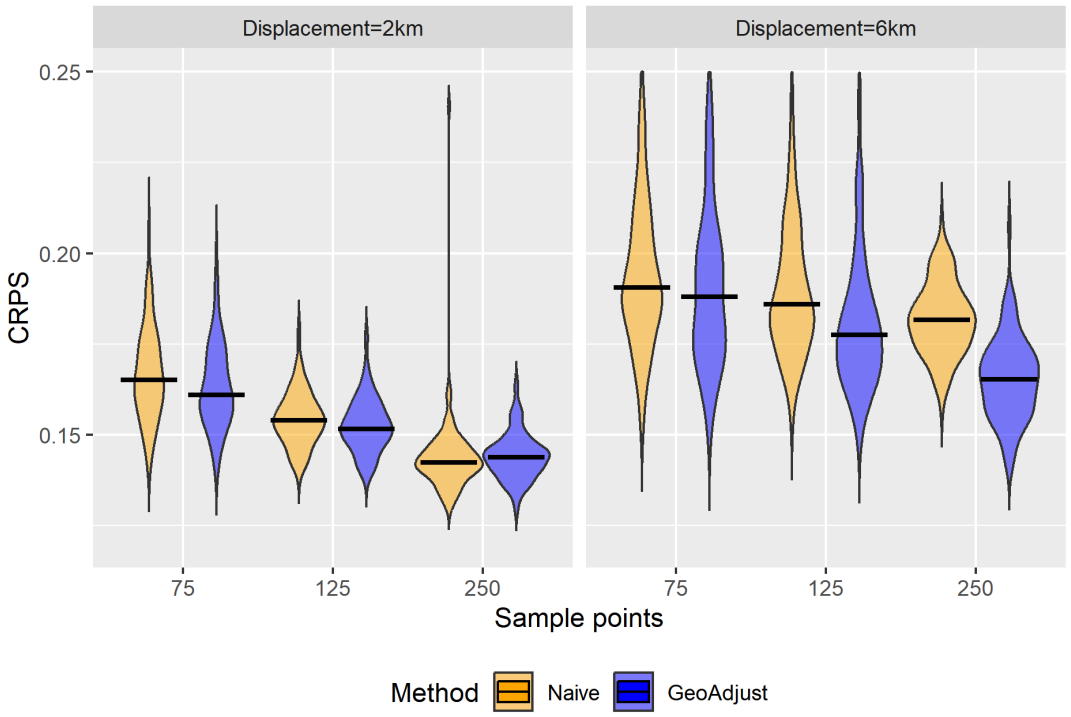

Figure B.22: CRPS: Sill=0.2, Nugget=0.02

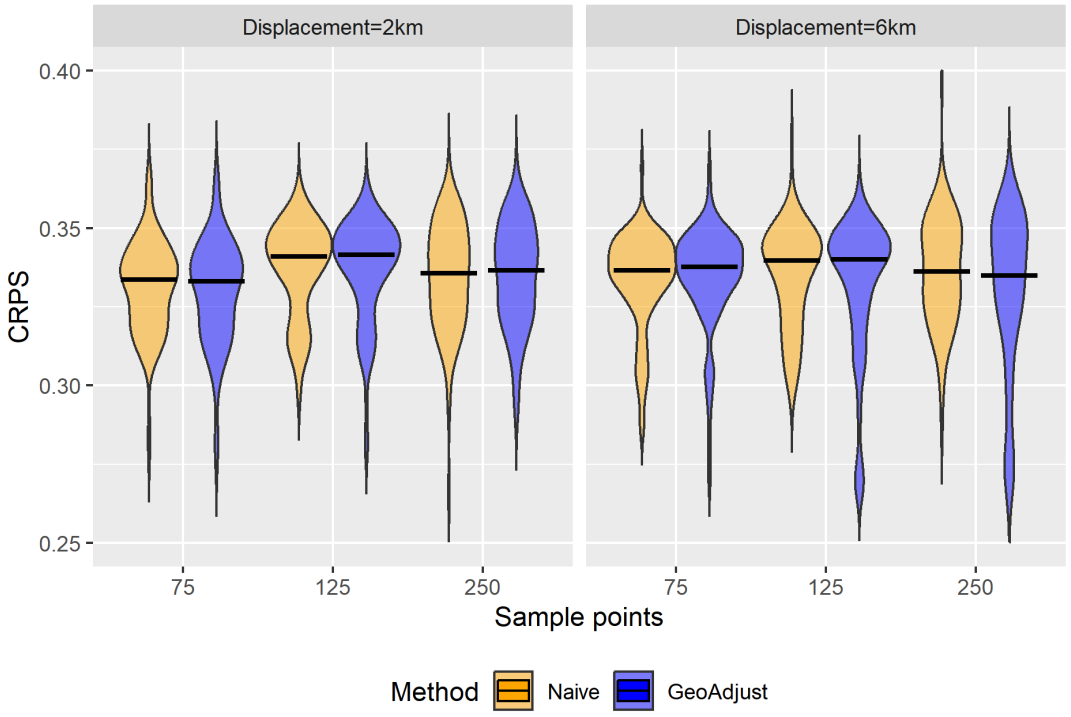

Figure B.23: CRPS: Sill=0.02, Nugget=0.2

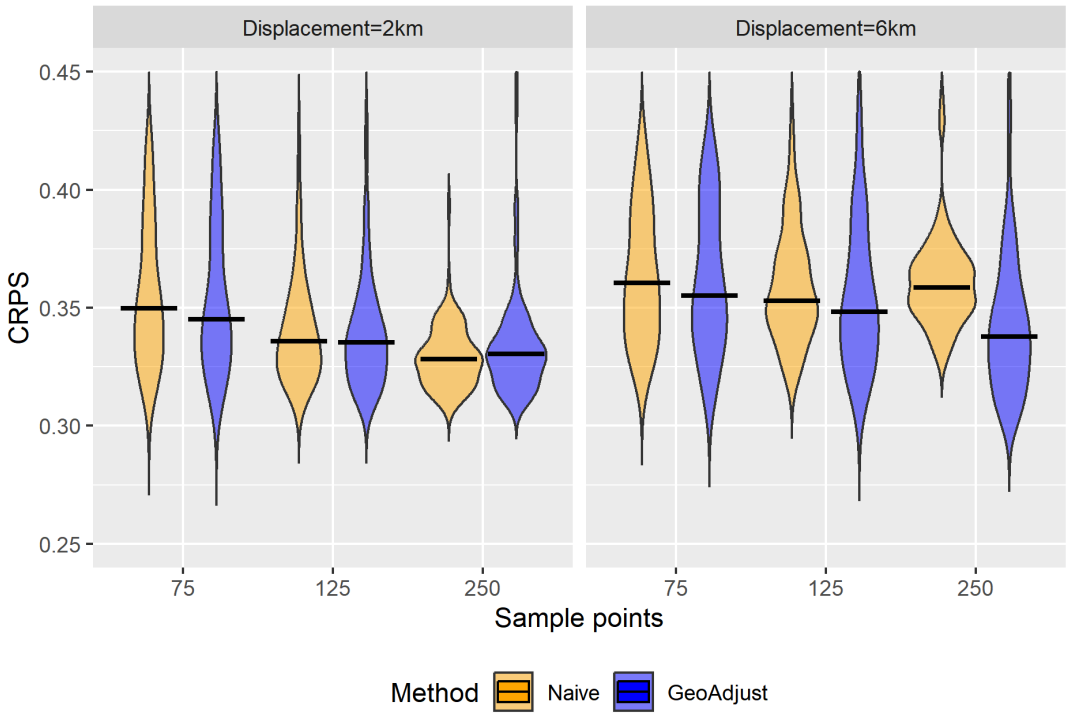

Figure B.24: CRPS: Sill=0.2, Nugget=0.2

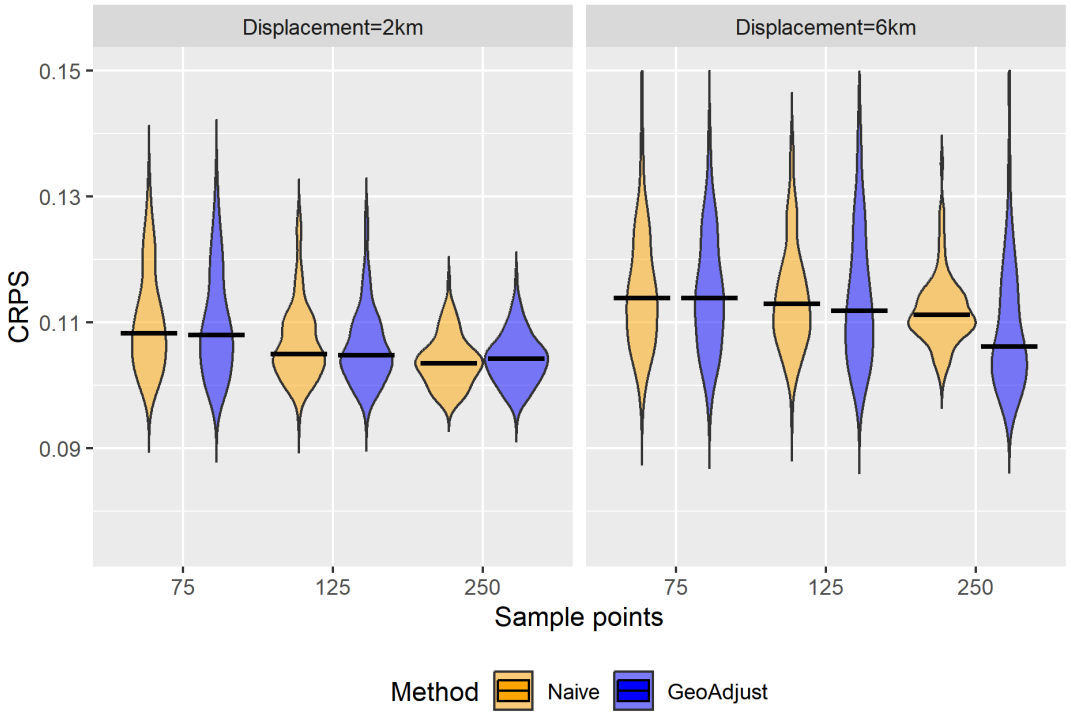

Figure B.25: CRPS: Sill=0.02, Nugget=0.02

Appendix C

R Codes

```
#-----------------------------------------------------------------------
# CASE STUDY: PHILIPPINES DHS
#-----------------------------------------------------------------------

##### Set working directory
rm(list=ls())
setwd("C:/Users/Roel Jude Bagaforo/Documents/Personal Files/Academic
Files/Masters/Thesis/Codes and Outputs/Data")

##### Load libraries
library(tidyverse)   # most variable creation here uses tidyverse
library(tidyselect)  # used to select variables
library(haven)       # used for Haven labelled DHS variables
library(labelled)    # used for Haven labelled variable creation
library(expss)       # for creating tables with Haven labelled data
library(xlsx)        # for exporting to excel
library(sf)          # for reading shp files
library(PrevMap)     # for standard inferences
library(geoR)        # for standard inferences
library(patchwork)   #for arranging plots
library(raster)      # for calculating distances between the points
library(GeoAdjust)   # GeoAdjust package

##### Load the dataset
IRdata <- read_dta("PHIR82FL.dta") #Individual recode data
minhc <- IRdata[c("v001","v002","v483a")] %>% group_by(v001) %>%
  summarise(mean_min=mean(v483a), sd_min=sd(v483a), total_women=n())
names(minhc)[names(minhc) == "v001"] <- "clustid" #renaming cluster id
#shapefile
boundaries <- st_read("ncr_map2_projected.shp") #NCR boundaries ncr_map2
boundgeom <- boundaries$geometry %>% fortify() #for plotting
samplelocs <- read.csv("ncr2_clustloc_projected.csv") #NCR cluster locations
names(samplelocs)[names(samplelocs) == "DHSCLUST"] <- "clustid" #rename
#we also want to know the distances between the locations
locsdist <- pointDistance(samplelocs[, c("X", "Y")], lonlat=FALSE)
```

```
range(replace(locsdist,locsdist==0,NA),na.rm=T)/1000

#merge data and shp
gausample <- merge(samplelocs, minhc, by="clustid", all=F)
gausample <- gausample[c("clustid","X","Y","LATNUM","LONGNUM","URBAN_RURA",
                         "mean_min","total_women")]
gausample$log_min <- log(gausample$mean_min)

# Plot of samples
ggplot() +
  geom_sf(data=boundgeom, fill="grey") +
  geom_point(data=gausample,aes(X,Y, color=log_min),size=3) +
  labs(x = "Longitude", y= "Latitude", color="Log Mean Minutes") +
  theme_minimal() + theme(legend.position = "bottom") +
  scale_color_gradient(limits=c(1,4))

# spatcorr test
check.spatgau.11 <- spat.corr.diagnostic(log_min~1,
                                         coords = ~I(X/1000) + I(Y/1000),
                                         data=gausample,likelihood = "Gaussian",
                                         lse.variogram = T)
check.spatgau.15 <- spat.corr.diagnostic(log_min~1,
                                         coords = ~I(X/1000) + I(Y/1000),
                                         data=gausample,likelihood = "Gaussian",
                                         lse.variogram = T)

# GeoAdjust
set.seed(888)
crs_Degrees = "+proj=longlat +datum=WGS84" #the original CRS of the data
crs_KM = "+units=km +proj=utm +zone=51" #target CRS

mesh.s <- meshCountry(admin0= boundaries, max.edge = c(1,3), offset = -.02,
          cutoff=0.5, target_crs = crs_KM)
png(filename = "mesh0.5.png", width = 6.27, height =  9.69/2.25, units = "in", res = 300)
plot(mesh.s)
dev.off()

locObs = data.frame(long = gausample$LONGNUM, lat = gausample$LATNUM)
locObs = sf::st_as_sf(locObs, coords=c("long","lat"), crs = crs_Degrees)

#gaussian outcome
system.time(inputDatagau <- prepareInput(response = list(ys=gausample$log_min),
                locObs = locObs,
                likelihood = 0, jScale = 1, urban = gausample$URBAN_RURA,
                mesh.s = mesh.s, adminMap = boundaries, covariateData = NULL,
                target_crs = crs_KM))

system.time(gau.naive <- estimateModel(data=inputDatagau, options=list(random=0,
                    covariates=0),
```

```
                    priors=list(beta=c(0,100), range=4), USpatial=1,
                    alphaSpatial=0.05, UNugget=1, alphaNug=0.05, n.sims=1000))

system.time(gau.geoadjust <- estimateModel(data=inputDatagau, options=list(random=1,
                    covariates=0),
                    priors=list(beta=c(0,100), range=4), USpatial=1,
                    alphaSpatial=0.05, UNugget=1, alphaNug=0.05, n.sims=1000))

print(gau.naive$res)
print(gau.geoadjust$res)

# predictions
newlocs <- st_sample(x=boundaries, size=10000, type="regular")
pred.naive <- predRes(obj=gau.naive[["obj"]], predCoords=newlocs/1000,
                    draws=gau.naive[["draws"]], mesh.s=mesh.s, flag=0, covariateData=NULL)
pred.geoadjust <- predRes(obj=gau.geoadjust[["obj"]], predCoords=newlocs/1000,
                    draws=gau.geoadjust[["draws"]], mesh.s=mesh.s, flag=0, covariateData=NULL)
plotpred.naive <- as.data.frame(cbind(X=st_coordinates(newlocs)[,1],
              Y=st_coordinates(newlocs)[,2],
              pred=pred.naive[,1], sd= pred.naive[,3], cv=pred.naive[,1]/pred.naive[,3]*100))

ggplot() +
  geom_sf(data=boundgeom, fill="grey") +
  geom_point(data=plotpred.naive,aes(X,Y, color=pred)) +
  labs(x = "Longitude", y= "Latitude", color="Log Mean Minutes") +
  scale_color_gradient(limits=c(2.3,2.9), breaks=c(2.3,2.45,2.60,2.85,2.90)) +
  theme_minimal() + theme(legend.position = "bottom")

ggplot() +
  geom_sf(data=boundgeom, fill="grey") +
  geom_point(data=plotpred.naive,aes(X,Y, color=sd)) +
  labs(x = "Longitude", y= "Latitude", color="SD") +
  scale_color_gradient(limits=c(0.1,0.22),low="yellow",high="red",
  breaks=c(0.1,0.14,0.18,0.22)) +
  theme_minimal() + theme(legend.position = "bottom")

plotpred.geoadjust <- as.data.frame(cbind(X=st_coordinates(newlocs)[,1],
              Y=st_coordinates(newlocs)[,2],
              pred=pred.geoadjust[,1], sd= pred.geoadjust[,3],
              cv=pred.geoadjust[,3]/pred.geoadjust[,1]*100))

ggplot() +
  geom_sf(data=boundgeom, fill="grey") +
  geom_point(data=plotpred.geoadjust,aes(X,Y, color=pred)) +
  labs(x = "Longitude", y= "Latitude", color="Log Mean Minutes") +
  scale_color_gradient(limits=c(2.3,2.9), breaks=c(2.3,2.45,2.60,2.85,2.90)) +
  theme_minimal() + theme(legend.position = "bottom")

ggplot() +
```

```
  geom_sf(data=boundgeom, fill="grey") +
  geom_point(data=plotpred.geoadjust,aes(X,Y, color=sd)) +
  labs(x = "Longitude", y= "Latitude", color="SD") +
  scale_color_gradient(limits=c(0.1,0.22),low="yellow",high="red",
  breaks=c(0.1,0.14,0.18,0.22)) +
  theme_minimal() + theme(legend.position = "bottom")

# mesh sensitivity
mesh.s2 <- meshCountry(admin0= boundaries, max.edge = c(1,3),
          offset = -.02,
          cutoff=0.35, target_crs = crs_KM)
plot(mesh.s2)

system.time(inputDatagau <- prepareInput(response =
list(ys=gausample$log_min), locObs = locObs,
                likelihood = 0, jScale = 1, urban = gausample$URBAN_RURA,
                mesh.s = mesh.s2, adminMap = boundaries, covariateData = NULL,
                target_crs = crs_KM))

system.time(gau.naive <- estimateModel(data=inputDatagau,
options=list(random=0,covariates=0),
                    priors=list(beta=c(0,100), range=4), USpatial=1,
                    alphaSpatial=0.05, UNugget=1, alphaNug=0.05, n.sims=1000))

system.time(gau.geoadjust <- estimateModel(data=inputDatagau,
options=list(random=1,covariates=0),
                    priors=list(beta=c(0,100), range=4), USpatial=1,
                    alphaSpatial=0.05, UNugget=1, alphaNug=0.05, n.sims=1000))

print(gau.naive$res)
print(gau.geoadjust$res)

# prior sensitivity
system.time(gau.naive <- estimateModel(data=inputDatagau,
                    options=list(random=0,covariates=0),
                    priors=list(beta=c(0,100), range=6), USpatial=1.2,
                    alphaSpatial=0.05, UNugget=1.2, alphaNug=0.05, n.sims=1000))

system.time(gau.geoadjust <- estimateModel(data=inputDatagau,
                    options=list(random=1,covariates=0),
                    priors=list(beta=c(0,100), range=6), USpatial=1.2,
                    alphaSpatial=0.05, UNugget=1.2, alphaNug=0.05, n.sims=1000))

print(gau.naive$res)
print(gau.geoadjust$res)
```

```
#-----------------------------------------------------------------------
# SIMULATION STUDY
#-----------------------------------------------------------------------

#SIMULATE GEOMASKED DATA OVER THE STUDY REGION
# Empty the environment
rm(list=ls(all=TRUE))

# Load libraries
library("sf") #for shp manipulation
library("geoR") #to simulate GRF

# Load the shapefile/boundary
boundaries <- st_read("ncr_map2_projected.shp") #NCR boundaries ncr_map2
crs_KM = "+units=km +proj=utm +zone=51" #target CRS
boundaries <- st_transform(boundaries, crs=crs_KM)

## Function to simulate data for geomasked sample locations and un-
geomasked prediction locations
sim_geodata <- function(npoints=npoints, mean=mean, sill=sill,
range=range, nugget=nugget, delta=delta){
#set.seed(123)
n <- 3000+npoints
locs <- st_sample(x=boundaries, size=n, type="random")
coords <- st_coordinates(locs)
#set.seed(123)
obs <- grf(n=n, grid=coords, cov.pars=c(sill,range), nugget=nugget,
mean=mean)$data
data <- data.frame(cbind(coords,obs))
data$urbanrural <- "U"

#un-geomasked prediction locations
predlocs <- st_as_sf(data, coords=c("X", "Y"), crs=crs_KM)[(1+npoints):
(n-1),]

#geomasked sample locations
samplelocs <-st_as_sf(data, coords=c("X", "Y"), crs=crs_KM)[1:npoints,]
#function to apply geomasking on the the locations but making sure that
the new locations fall within the study region
displace_within_region <- function(point, polygon=boundaries,
delta=delta) {
  while (TRUE) {
    # Generate random displacement within max_distance
    dx <- runif(1, min=0, max=delta)*cos(runif(1, min = 0, max=2*pi))
    dy <- runif(1, min=0, max=delta)*sin(runif(1, min = 0, max=2*pi))

    # Displace the point
    new_point <- st_coordinates(point) + c(dx, dy)
    new_point <- st_point(new_point)
```

```
    # Check if the new point is within the polygon
    if (st_within(new_point, polygon, sparse=F)) {
      return(st_coordinates(new_point))
    }
  }
}

geomasked_points <- do.call(rbind.data.frame,lapply(samplelocs$geometry,
displace_within_region,
                            polygon=boundaries, delta=delta))
colnames(geomasked_points) <- c("maskedX", "maskedY")

samplelocs <- cbind(data[1:npoints,], geomasked_points)
samplelocs <-st_as_sf(samplelocs, coords=c("maskedX", "maskedY"),
crs=crs_KM)

#check if samplelocs are inside boundaries
st_within(samplelocs, boundaries, sparse=F)


# return the locs and corresponding data
out <- list(samplelocs=samplelocs, predlocs=predlocs, jScale=delta/2)
return(out)
}

#system.time(trial <- sim_geodata(npoints=250, mean=0.25, sill=0.009,
range=8, nugget=0.001, delta=6))


args = commandArgs(trailingOnly = T)
setting = as.integer(args[1])
npoints = as.integer(args[2])
mean = as.numeric(args[3])
sill = as.numeric(args[4])
range = as.integer(args[5])
nugget = as.numeric(args[6])
delta = as.integer(args[7])
run = as.integer(args[8])

data <- sim_geodata(npoints, mean, sill, range, nugget, delta)

filename_save <- sprintf("/vsc-hard-mounts/leuven-
data/356/vsc35665/roel/spdata_%02d_%03d", setting, run)

save("data", file = sprintf("%s.RData", filename_save))

#ANALYZE THE SIMULATED DATA (NAIVE AND GEOADJUST)
# Empty the environment
```

```
rm(list=ls(all=TRUE))

# Load libraries
library("sf") #for shp manipulation
library("GeoAdjust") #for analysis
library("scoringRules") #for CRPS


# Analyze the data
analyze <- function(data){
jScale <- data$jScale
  #build mesh
crs_Degrees = "+proj=longlat +datum=WGS84" #the original CRS of the data
crs_KM = "+units=km +proj=utm +zone=51" #target CRS
boundaries <- st_read("ncr_map2_projected.shp") #NCR boundaries ncr_map2
boundaries <- st_transform(boundaries, crs=crs_KM)
mesh.s <- meshCountry(admin0= boundaries, max.edge = c(1,3), offset =
-.02,
                      cutoff=0.5, target_crs = crs_KM)
#load data locations
locObs = st_as_sf(st_transform(data$samplelocs$geometry, crs =
crs_Degrees))

#prepare input data, build integration points
inputData <- prepareInput(response = list(ys=data$samplelocs$obs),
                                          locObs = locObs,
                                          likelihood = 0, jScale =
                                          jScale, urban =
                                          data$samplelocs$urbanrural,
                                          mesh.s = mesh.s, adminMap =
                                          boundaries, covariateData =
                                          NULL,
                                          target_crs = crs_KM)
#naive estimation
naive_time<-system.time(naive <- estimateModel(data=inputData,
options=list(random=0,covariates=0),
                                       priors=list(beta=c(0,100),
                                       range=8), USpatial=1,
                                       alphaSpatial=0.05, UNugget=1,
                                       alphaNug=0.05, n.sims=1000))
#geoadjust
geoadjust_time<-system.time(geoadjust <- estimateModel(data=inputData,
options=list(random=1,covariates=0),
                                          priors=list(beta=c(0,100),
                                          range=8), USpatial=1,
                                          alphaSpatial=0.05, UNugget=1,
                                          alphaNug=0.05, n.sims=1000))

#save parameter
```

```
#print(naive$res)[,1:2]
#print(geoadjust$res)[,1:2]

#predictions
pred.naive <- predRes(obj=naive[["obj"]],
predCoords=data$predlocs$geometry,
                      draws=naive[["draws"]], mesh.s=mesh.s, flag=0,
                      covariateData=NULL)
pred.geoadjust <- predRes(obj=geoadjust[["obj"]],
predCoords=data$predlocs$geometry,
                          draws=geoadjust[["draws"]], mesh.s=mesh.s,
                          flag=0, covariateData=NULL)
  #prediction RMSE and mean CRPS
RMSE_naive <- sqrt(sum((data$predlocs$obs-
pred.naive[,1])^2)/length(data$predlocs$obs))
RMSE_geoadjust <- sqrt(sum((data$predlocs$obs-
pred.geoadjust[,1])^2)/length(data$predlocs$obs))
CRPS_naive <- mean(mapply(FUN=crps_norm, y=data$predlocs$obs,
mean=pred.naive[,1], sd=pred.naive[,3]))
CRPS_geoadjust <- mean(mapply(FUN=crps_norm, y=data$predlocs$obs,
mean=pred.geoadjust[,1], sd=pred.geoadjust[,3]))

#return the values
return(list(parameters_naive=print(naive$res)[,1:2],
parameters_geoadjust=print(geoadjust$res)[,1:2],
            RMSE_naive=RMSE_naive, CRPS_naive=CRPS_naive,
            RMSE_geoadjust=RMSE_geoadjust,
            CRPS_geoadjust=CRPS_geoadjust, naive_time=naive_time,
            geoadjust_time=geoadjust_time))
}


args = commandArgs(trailingOnly = T)
setting = as.integer(args[1])
npoints = as.integer(args[2])
mean = as.numeric(args[3])
sill = as.numeric(args[4])
range = as.integer(args[5])
nugget = as.numeric(args[6])
delta = as.integer(args[7])
run = as.integer(args[8])

# Load the simulated data
load(sprintf("/vsc-hard-mounts/leuven-
data/356/vsc35665/roel/spdata_%02d_%03d.RData", setting, run))

analysis <- analyze(data)

filename_save <- sprintf("/vsc-hard-mounts/leuven-
```

```
data/356/vsc35665/roel/output_%02d_%03d", setting, run)

save("analysis", file = sprintf("%s.RData", filename_save))

#SUMMARY OF SIMULATION RESULTS
# Empty the environment
rm(list=ls(all=TRUE))

# Change working directory
setwd("C:/Users/Roel Jude Bagaforo/Documents/Personal Files/Academic
Files/Masters/Thesis/Codes and Outputs/Simulation Study")

# Load libraries
library(dplyr)
library(tidyr)
library(ggplot2)

# Load the data and simulation results
sim_data <- readRDS("gen_data_ls.R")
sim_results <- readRDS("analysis_setting1_24_inc2.R")
sim_results <- readRDS("analysis_1_32.R")

# Function to extract parameters from a single run and create a dataframe
extract_parameters <- function(run) {
  # Extract parameters from each list
  parameters_naive <- run$parameters_naive$estimates
  parameters_geoadjust <- run$parameters_geoadjust$estimates
  RMSE_naive <- run$RMSE_naive
  CRPS_naive <- run$CRPS_naive
  RMSE_geoadjust <- run$RMSE_geoadjust
  CRPS_geoadjust <- run$CRPS_geoadjust
  naive_time <- run$naive_time
  geoadjust_time <- run$geoadjust_time

  # Create dataframe for parameters
  df <- cbind(
    range_naive = parameters_naive[1],
    sigma2_naive = parameters_naive[2]^2,
    tau2_naive = parameters_naive[3]^2,
    intercept_naive = parameters_naive[4],
    RMSE_naive = RMSE_naive,
    CRPS_naive = CRPS_naive,
    time_naive = naive_time["elapsed"],
    range_geoadjust = parameters_geoadjust[1],
    sigma2_geoadjust = parameters_geoadjust[2]^2,
    tau2_geoadjust = parameters_geoadjust[3]^2,
    intercept_geoadjust = parameters_geoadjust[4],
    RMSE_geoadjust = RMSE_geoadjust,
    CRPS_geoadjust = CRPS_geoadjust,
```

```
    time_geoadjust = geoadjust_time["elapsed"]
  )
  return(as.data.frame(df))
}

# Apply the function to each run
all_runs_df <- list()
for (i in 1:24) {
  for (j in 1:100) {
    all_runs_df[[length(all_runs_df) + 1]] <-
    extract_parameters(sim_results[[i]][[j]])
  }
}

# Make as one dataframe
combined_df <- bind_rows(all_runs_df)
rownames(combined_df) <- NULL
# Add setting variable considering the failed runs
sim_summary_wide <- rbind(combined_df[1:1592, ], NA,
combined_df[1593:2034, ],
                     NA, combined_df[2035:2119, ], NA,
                     combined_df[2120:2397, ])
rownames(sim_summary_wide) <- NULL
sim_summary_long <- pivot_longer(sim_summary_wide,
                          cols = starts_with(c("range", "sigma2",
                          "tau2", "intercept", "RMSE", "CRPS", "time")),
                          names_to = c(".value", "method"),
                          names_sep = "_")
sim_summary_long$setting <- rep(1:24, each=200)

# Incorporate parameter settings
par_settings <- read.csv("par_settings_summary.csv")
sim_summary <- merge(sim_summary_long, par_settings, by="setting",
all.x=T)

# Calculate relative bias for parameters
sim_summary$intbias <- (sim_summary$intercept-
sim_summary$mean)/sim_summary$mean
sim_summary$rangebias <- (sim_summary$range-
sim_summary$true_range)/sim_summary$true_range
sim_summary$sillbias <- (sim_summary$sigma2-
sim_summary$sill)/sim_summary$sill
sim_summary$nuggetbias <- (sim_summary$tau2-
sim_summary$nugget)/sim_summary$nugget

sim_summary$setting <- ifelse(sim_summary$sill == 0.2 &
sim_summary$nugget == 0.02, 1,
                        ifelse(sim_summary$sill == 0.02 &
                        sim_summary$nugget == 0.2, 2,
```

```
                        ifelse(sim_summary$sill == 0.2 &
                        sim_summary$nugget == 0.2, 3, 4)))

# Subset the data into four scenarios
# 1 Sill greater than nugget
summary_1 <- sim_summary %>% filter(sill == 0.2 & nugget == 0.02)
# 2 Nugget greater than sill
summary_2 <- sim_summary %>% filter(sill == 0.02 & nugget == 0.2)
# 3 Equal sill and nugget
summary_3 <- sim_summary %>% filter(sill == 0.2 & nugget == 0.2)
# 4 Equal sill and nugget (small)
summary_4 <- sim_summary %>% filter(sill == 0.02 & nugget == 0.02)

### PLOTS
## Parameter bias
# mean
png(filename = "mean1.png", width = 6.27, height =  9.69/2.25, units =
"in", res = 300)
summary_1 %>%
  ggplot(aes(fill = method, y = intbias, x = factor(npoints))) +
  geom_violin(position = "dodge", alpha = 0.5, trim=F) +
  geom_hline(yintercept = 0, color = "red", linetype = "dashed", size =
  0.8) +
  stat_summary(fun = median, geom = "crossbar", width = 0.75, fatten =
  2, color = "black", position = position_dodge(width = 0.9)) +
  labs(x = "Sample points",y = "Relative bias",fill = "Method") +
  scale_fill_manual(values = c("naive" = "blue", "geoadjust" =
  "orange"), labels = c("Naive", "GeoAdjust")) +
  ylim(-0.3, 0.3) + facet_wrap(~ delta, scales = "free_x", ncol = 2,
  labeller = labeller(delta = c("2" = "Displacement=2km", "6" =
  "Displacement=6km"))) +
  theme(legend.position = "bottom")
dev.off()

png(filename = "mean2.png", width = 6.27, height =  9.69/2.25, units =
"in", res = 300)
summary_2 %>%
  ggplot(aes(fill = method, y = intbias, x = factor(npoints))) +
  geom_violin(position = "dodge", alpha = 0.5, trim=F) +
  geom_hline(yintercept = 0, color = "red", linetype = "dashed", size =
  0.8) +
  stat_summary(fun = median, geom = "crossbar", width = 0.75, fatten =
  2, color = "black", position = position_dodge(width = 0.9)) +
  labs(x = "Sample points",y = "Relative bias",fill = "Method") +
  scale_fill_manual(values = c("naive" = "blue", "geoadjust" =
  "orange"), labels = c("Naive", "GeoAdjust")) +
  ylim(-0.2, 0.2) + facet_wrap(~ delta, scales = "free_x", ncol = 2,
                           labeller = labeller(delta = c("2" =
                           "Displacement=2km", "6" =
```

```
                            "Displacement=6km"))) +
  theme(legend.position = "bottom")
dev.off()

png(filename = "mean3.png", width = 6.27, height =  9.69/2.25, units =
"in", res = 300)
summary_3 %>%
  ggplot(aes(fill = method, y = intbias, x = factor(npoints))) +
  geom_violin(position = "dodge", alpha = 0.5, trim=F) +
  geom_hline(yintercept = 0, color = "red", linetype = "dashed", size =
  0.8) +
  stat_summary(fun = median, geom = "crossbar", width = 0.75, fatten =
  2, color = "black", position = position_dodge(width = 0.9)) +
  labs(x = "Sample points",y = "Relative bias",fill = "Method") +
  scale_fill_manual(values = c("naive" = "blue", "geoadjust" =
  "orange"), labels = c("Naive", "GeoAdjust")) +
  ylim(-0.3, 0.3) + facet_wrap(~ delta, scales = "free_x", ncol = 2,
                            labeller = labeller(delta = c("2" =
                            "Displacement=2km", "6" =
                            "Displacement=6km"))) +
  theme(legend.position = "bottom")
dev.off()

png(filename = "mean4.png", width = 6.27, height =  9.69/2.25, units =
"in", res = 300)
summary_4 %>%
  ggplot(aes(fill = method, y = intbias, x = factor(npoints))) +
  geom_violin(position = "dodge", alpha = 0.5, trim=F) +
  geom_hline(yintercept = 0, color = "red", linetype = "dashed", size =
  0.8) +
  stat_summary(fun = median, geom = "crossbar", width = 0.75, fatten =
  2, color = "black", position = position_dodge(width = 0.9)) +
  labs(x = "Sample points",y = "Relative bias",fill = "Method") +
  scale_fill_manual(values = c("naive" = "blue", "geoadjust" =
  "orange"), labels = c("Naive", "GeoAdjust")) +
  ylim(-0.15, 0.15) + facet_wrap(~ delta, scales = "free_x", ncol = 2,
                            labeller = labeller(delta = c("2" =
                            "Displacement=2km", "6" =
                            "Displacement=6km"))) +
  theme(legend.position = "bottom")
dev.off()

# sill
png(filename = "sill1.png", width = 6.27, height =  9.69/2.25, units =
"in", res = 300)
summary_1 %>%
  ggplot(aes(fill = method, y = sillbias, x = factor(npoints))) +
  geom_violin(position = "dodge", alpha = 0.5, trim=F) +
  geom_hline(yintercept = 0, color = "red", linetype = "dashed", size =
```

```
  0.8) +
  stat_summary(fun = median, geom = "crossbar", width = 0.75, fatten =
  2, color = "black", position = position_dodge(width = 0.9)) +
  labs(x = "Sample points",y = "Relative bias",fill = "Method") +
  scale_fill_manual(values = c("naive" = "blue", "geoadjust" =
  "orange"), labels = c("Naive", "GeoAdjust")) +
  ylim(-2, 2) + facet_wrap(~ delta, scales = "free_x", ncol = 2,
                           labeller = labeller(delta = c("2" =
                           "Displacement=2km", "6" =
                           "Displacement=6km"))) +
  theme(legend.position = "bottom")
dev.off()

png(filename = "sill2.png", width = 6.27, height =  9.69/2.25, units =
"in", res = 300)
summary_2 %>%
  ggplot(aes(fill = method, y = sillbias, x = factor(npoints))) +
  geom_violin(position = "dodge", alpha = 0.5, trim=F) +
  geom_hline(yintercept = 0, color = "red", linetype = "dashed", size =
  0.8) +
  stat_summary(fun = median, geom = "crossbar", width = 0.75, fatten =
  2, color = "black", position = position_dodge(width = 0.9)) +
  labs(x = "Sample points",y = "Relative bias",fill = "Method") +
  scale_fill_manual(values = c("naive" = "blue", "geoadjust" =
  "orange"), labels = c("Naive", "GeoAdjust")) +
  ylim(-2.5, 4) + facet_wrap(~ delta, scales = "free_x", ncol = 2,
                           labeller = labeller(delta = c("2" =
                           "Displacement=2km", "6" =
                           "Displacement=6km"))) +
  theme(legend.position = "bottom")
dev.off()

png(filename = "sill3.png", width = 6.27, height =  9.69/2.25, units =
"in", res = 300)
summary_3 %>%
  ggplot(aes(fill = method, y = sillbias, x = factor(npoints))) +
  geom_violin(position = "dodge", alpha = 0.5, trim=F) +
  geom_hline(yintercept = 0, color = "red", linetype = "dashed", size =
  0.8) +
  stat_summary(fun = median, geom = "crossbar", width = 0.75, fatten =
  2, color = "black", position = position_dodge(width = 0.9)) +
  labs(x = "Sample points",y = "Relative bias",fill = "Method") +
  scale_fill_manual(values = c("naive" = "blue", "geoadjust" =
  "orange"), labels = c("Naive", "GeoAdjust")) +
  ylim(-2, 2) + facet_wrap(~ delta, scales = "free_x", ncol = 2,
                           labeller = labeller(delta = c("2" =
                           "Displacement=2km", "6" =
                           "Displacement=6km"))) +
  theme(legend.position = "bottom")
```

```
dev.off()

png(filename = "sill4.png", width = 6.27, height =  9.69/2.25, units =
"in", res = 300)
summary_4 %>%
  ggplot(aes(fill = method, y = sillbias, x = factor(npoints))) +
  geom_violin(position = "dodge", alpha = 0.5, trim=F) +
  geom_hline(yintercept = 0, color = "red", linetype = "dashed", size =
  0.8) +
  stat_summary(fun = median, geom = "crossbar", width = 0.75, fatten =
  2, color = "black", position = position_dodge(width = 0.9)) +
  labs(x = "Sample points",y = "Relative bias",fill = "Method") +
  scale_fill_manual(values = c("naive" = "blue", "geoadjust" =
  "orange"), labels = c("Naive", "GeoAdjust")) +
  ylim(-2, 3) + facet_wrap(~ delta, scales = "free_x", ncol = 2,
                           labeller = labeller(delta = c("2" =
                           "Displacement=2km", "6" =
                           "Displacement=6km"))) +
  theme(legend.position = "bottom")
dev.off()


# range
png(filename = "range1.png", width = 6.27, height =  9.69/2.25, units =
"in", res = 300)
summary_1 %>%
  ggplot(aes(fill = method, y = rangebias, x = factor(npoints))) +
  geom_violin(position = "dodge", alpha = 0.5, trim=F) +
  geom_hline(yintercept = 0, color = "red", linetype = "dashed", size =
  0.8) +
  stat_summary(fun = median, geom = "crossbar", width = 0.75, fatten =
  2, color = "black", position = position_dodge(width = 0.9)) +
  labs(x = "Sample points",y = "Relative bias",fill = "Method") +
  scale_fill_manual(values = c("naive" = "blue", "geoadjust" =
  "orange"), labels = c("Naive", "GeoAdjust")) +
  ylim(-2.5, 5) + facet_wrap(~ delta, scales = "free_x", ncol = 2,
                             labeller = labeller(delta = c("2" =
                             "Displacement=2km", "6" =
                             "Displacement=6km"))) +
  theme(legend.position = "bottom")
dev.off()

png(filename = "range2.png", width = 6.27, height =  9.69/2.25, units =
"in", res = 300)
summary_2 %>%
  ggplot(aes(fill = method, y = rangebias, x = factor(npoints))) +
  geom_violin(position = "dodge", alpha = 0.5, trim=F) +
  geom_hline(yintercept = 0, color = "red", linetype = "dashed", size =
  0.8) +
```

```
  stat_summary(fun = median, geom = "crossbar", width = 0.75, fatten =
  2, color = "black", position = position_dodge(width = 0.9)) +
  labs(x = "Sample points",y = "Relative bias",fill = "Method") +
  scale_fill_manual(values = c("naive" = "blue", "geoadjust" =
  "orange"), labels = c("Naive", "GeoAdjust")) +
  ylim(-10, 30) + facet_wrap(~ delta, scales = "free_x", ncol = 2,
                             labeller = labeller(delta = c("2" =
                             "Displacement=2km", "6" =
                             "Displacement=6km"))) +
  theme(legend.position = "bottom")
dev.off()

png(filename = "range3.png", width = 6.27, height =  9.69/2.25, units =
"in", res = 300)
summary_3 %>%
  ggplot(aes(fill = method, y = rangebias, x = factor(npoints))) +
  geom_violin(position = "dodge", alpha = 0.5, trim=F) +
  geom_hline(yintercept = 0, color = "red", linetype = "dashed", size =
  0.8) +
  stat_summary(fun = median, geom = "crossbar", width = 0.75, fatten =
  2, color = "black", position = position_dodge(width = 0.9)) +
  labs(x = "Sample points",y = "Relative bias",fill = "Method") +
  scale_fill_manual(values = c("naive" = "blue", "geoadjust" =
  "orange"), labels = c("Naive", "GeoAdjust")) +
  ylim(-5, 10) + facet_wrap(~ delta, scales = "free_x", ncol = 2,
                             labeller = labeller(delta = c("2" =
                             "Displacement=2km", "6" =
                             "Displacement=6km"))) +
  theme(legend.position = "bottom")
dev.off()

png(filename = "range4.png", width = 6.27, height =  9.69/2.25, units =
"in", res = 300)
summary_4 %>%
  ggplot(aes(fill = method, y = rangebias, x = factor(npoints))) +
  geom_violin(position = "dodge", alpha = 0.5, trim=F) +
  geom_hline(yintercept = 0, color = "red", linetype = "dashed", size =
  0.8) +
  stat_summary(fun = median, geom = "crossbar", width = 0.75, fatten =
  2, color = "black", position = position_dodge(width = 0.9)) +
  labs(x = "Sample points",y = "Relative bias",fill = "Method") +
  scale_fill_manual(values = c("naive" = "blue", "geoadjust" =
  "orange"), labels = c("Naive", "GeoAdjust")) +
  ylim(-5, 10) + facet_wrap(~ delta, scales = "free_x", ncol = 2,
                             labeller = labeller(delta = c("2" =
                             "Displacement=2km", "6" =
                             "Displacement=6km"))) +
  theme(legend.position = "bottom")
dev.off()
```

```
# nugget
png(filename = "nugget1.png", width = 6.27, height =  9.69/2.25, units =
"in", res = 300)
summary_1 %>%
  ggplot(aes(fill = method, y = nuggetbias, x = factor(npoints))) +
  geom_violin(position = "dodge", alpha = 0.5, trim=F) +
  geom_hline(yintercept = 0, color = "red", linetype = "dashed", size =
  0.8) +
  stat_summary(fun = median, geom = "crossbar", width = 0.75, fatten =
  2, color = "black", position = position_dodge(width = 0.9)) +
  labs(x = "Sample points",y = "Relative bias",fill = "Method") +
  scale_fill_manual(values = c("naive" = "blue", "geoadjust" =
  "orange"), labels = c("Naive", "GeoAdjust")) +
  ylim(-2.5, 7.5) + facet_wrap(~ delta, scales = "free_x", ncol = 2,
                                labeller = labeller(delta = c("2" =
                                "Displacement=2km", "6" =
                                "Displacement=6km"))) +
  theme(legend.position = "bottom")
dev.off()

png(filename = "nugget2.png", width = 6.27, height =  9.69/2.25, units =
"in", res = 300)
summary_2 %>%
  ggplot(aes(fill = method, y = nuggetbias, x = factor(npoints))) +
  geom_violin(position = "dodge", alpha = 0.5, trim=F) +
  geom_hline(yintercept = 0, color = "red", linetype = "dashed", size =
  0.8) +
  stat_summary(fun = median, geom = "crossbar", width = 0.75, fatten =
  2, color = "black", position = position_dodge(width = 0.9)) +
  labs(x = "Sample points",y = "Relative bias",fill = "Method") +
  scale_fill_manual(values = c("naive" = "blue", "geoadjust" =
  "orange"), labels = c("Naive", "GeoAdjust")) +
  ylim(-1, 1) + facet_wrap(~ delta, scales = "free_x", ncol = 2,
                                labeller = labeller(delta = c("2" =
                                "Displacement=2km", "6" =
                                "Displacement=6km"))) +
  theme(legend.position = "bottom")
dev.off()

png(filename = "nugget3.png", width = 6.27, height =  9.69/2.25, units =
"in", res = 300)
summary_3 %>%
  ggplot(aes(fill = method, y = nuggetbias, x = factor(npoints))) +
  geom_violin(position = "dodge", alpha = 0.5, trim=F) +
  geom_hline(yintercept = 0, color = "red", linetype = "dashed", size =
  0.8) +
  stat_summary(fun = median, geom = "crossbar", width = 0.75, fatten =
  2, color = "black", position = position_dodge(width = 0.9)) +
```

```
  labs(x = "Sample points",y = "Relative bias",fill = "Method") +
  scale_fill_manual(values = c("naive" = "blue", "geoadjust" =
  "orange"), labels = c("Naive", "GeoAdjust")) +
  ylim(-0.8, 1) + facet_wrap(~ delta, scales = "free_x", ncol = 2,
                                labeller = labeller(delta = c("2" =
                                "Displacement=2km", "6" =
                                "Displacement=6km"))) +
  theme(legend.position = "bottom")
dev.off()

png(filename = "nugget4.png", width = 6.27, height =  9.69/2.25, units =
"in", res = 300)
summary_4 %>%
  ggplot(aes(fill = method, y = nuggetbias, x = factor(npoints))) +
  geom_violin(position = "dodge", alpha = 0.5, trim=F) +
  geom_hline(yintercept = 0, color = "red", linetype = "dashed", size =
  0.8) +
  stat_summary(fun = median, geom = "crossbar", width = 0.75, fatten =
  2, color = "black", position = position_dodge(width = 0.9)) +
  labs(x = "Sample points",y = "Relative bias",fill = "Method") +
  scale_fill_manual(values = c("naive" = "blue", "geoadjust" =
  "orange"), labels = c("Naive", "GeoAdjust")) +
  ylim(-1, 1) + facet_wrap(~ delta, scales = "free_x", ncol = 2,
                                labeller = labeller(delta = c("2" =
                                "Displacement=2km", "6" =
                                "Displacement=6km"))) +
  theme(legend.position = "bottom")
dev.off()

## Prediction
# RMSE
png(filename = "rmse1.png", width = 6.27, height =  9.69/2.25, units =
"in", res = 300)
summary_1 %>%
  ggplot(aes(fill = method, y = RMSE, x = factor(npoints))) +
  geom_violin(position = "dodge", alpha = 0.5, trim=F) +
  stat_summary(fun = median, geom = "crossbar", width = 0.75, fatten =
  2, color = "black", position = position_dodge(width = 0.9)) +
  labs(x = "Sample points",y = "RMSE",fill = "Method") +
  scale_fill_manual(values = c("naive" = "blue", "geoadjust" =
  "orange"), labels = c("Naive", "GeoAdjust")) +
  ylim(0.2, 0.45) + facet_wrap(~ delta, scales = "free_x", ncol = 2,
                                labeller = labeller(delta = c("2" =
                                "Displacement=2km", "6" =
                                "Displacement=6km"))) +
  theme(legend.position = "bottom")
dev.off()

png(filename = "rmse2.png", width = 6.27, height =  9.69/2.25, units =
```

```
"in", res = 300)
summary_2 %>%
  ggplot(aes(fill = method, y = RMSE, x = factor(npoints))) +
  geom_violin(position = "dodge", alpha = 0.5, trim=F) +
  stat_summary(fun = median, geom = "crossbar", width = 0.75, fatten =
  2, color = "black", position = position_dodge(width = 0.9)) +
  labs(x = "Sample points",y = "RMSE",fill = "Method") +
  scale_fill_manual(values = c("naive" = "blue", "geoadjust" =
  "orange"), labels = c("Naive", "GeoAdjust")) +
  ylim(0.4, 0.5) + facet_wrap(~ delta, scales = "free_x", ncol = 2,
                                labeller = labeller(delta = c("2" =
                                "Displacement=2km", "6" =
                                "Displacement=6km"))) +
  theme(legend.position = "bottom")
dev.off()

png(filename = "rmse3.png", width = 6.27, height =  9.69/2.25, units =
"in", res = 300)
summary_3 %>%
  ggplot(aes(fill = method, y = RMSE, x = factor(npoints))) +
  geom_violin(position = "dodge", alpha = 0.5, trim=F) +
  stat_summary(fun = median, geom = "crossbar", width = 0.75, fatten =
  2, color = "black", position = position_dodge(width = 0.9)) +
  labs(x = "Sample points",y = "RMSE",fill = "Method") +
  scale_fill_manual(values = c("naive" = "blue", "geoadjust" =
  "orange"), labels = c("Naive", "GeoAdjust")) +
  ylim(0.45, 0.65) + facet_wrap(~ delta, scales = "free_x", ncol = 2,
                                labeller = labeller(delta = c("2" =
                                "Displacement=2km", "6" =
                                "Displacement=6km"))) +
  theme(legend.position = "bottom")
dev.off()

png(filename = "rmse4.png", width = 6.27, height =  9.69/2.25, units =
"in", res = 300)
summary_4 %>%
  ggplot(aes(fill = method, y = RMSE, x = factor(npoints))) +
  geom_violin(position = "dodge", alpha = 0.5, trim=F) +
  stat_summary(fun = median, geom = "crossbar", width = 0.75, fatten =
  2, color = "black", position = position_dodge(width = 0.9)) +
  labs(x = "Sample points",y = "RMSE",fill = "Method") +
  scale_fill_manual(values = c("naive" = "blue", "geoadjust" =
  "orange"), labels = c("Naive", "GeoAdjust")) +
  ylim(0.10, 0.25) + facet_wrap(~ delta, scales = "free_x", ncol = 2,
                                labeller = labeller(delta = c("2" =
                                "Displacement=2km", "6" =
                                "Displacement=6km"))) +
  theme(legend.position = "bottom")
dev.off()
```

```
# CRPS
png(filename = "crps1.png", width = 6.27, height =  9.69/2.25, units =
"in", res = 300)
summary_1 %>%
  ggplot(aes(fill = method, y = CRPS, x = factor(npoints))) +
  geom_violin(position = "dodge", alpha = 0.5, trim=F) +
  stat_summary(fun = median, geom = "crossbar", width = 0.75, fatten =
  2, color = "black", position = position_dodge(width = 0.9)) +
  labs(x = "Sample points",y = "CRPS",fill = "Method") +
  scale_fill_manual(values = c("naive" = "blue", "geoadjust" =
  "orange"), labels = c("Naive", "GeoAdjust")) +
  ylim(0.12, 0.25) + facet_wrap(~ delta, scales = "free_x", ncol = 2,
                            labeller = labeller(delta = c("2" =
                            "Displacement=2km", "6" =
                            "Displacement=6km"))) +
  theme(legend.position = "bottom")
dev.off()

png(filename = "crps2.png", width = 6.27, height =  9.69/2.25, units =
"in", res = 300)
summary_2 %>%
  ggplot(aes(fill = method, y = CRPS, x = factor(npoints))) +
  geom_violin(position = "dodge", alpha = 0.5, trim=F) +
  stat_summary(fun = median, geom = "crossbar", width = 0.75, fatten =
  2, color = "black", position = position_dodge(width = 0.9)) +
  labs(x = "Sample points",y = "CRPS",fill = "Method") +
  scale_fill_manual(values = c("naive" = "blue", "geoadjust" =
  "orange"), labels = c("Naive", "GeoAdjust")) +
  ylim(0.25, 0.4) + facet_wrap(~ delta, scales = "free_x", ncol = 2,
                            labeller = labeller(delta = c("2" =
                            "Displacement=2km", "6" =
                            "Displacement=6km"))) +
  theme(legend.position = "bottom")
dev.off()

png(filename = "crps3.png", width = 6.27, height =  9.69/2.25, units =
"in", res = 300)
summary_3 %>%
  ggplot(aes(fill = method, y = CRPS, x = factor(npoints))) +
  geom_violin(position = "dodge", alpha = 0.5, trim=F) +
  stat_summary(fun = median, geom = "crossbar", width = 0.75, fatten =
  2, color = "black", position = position_dodge(width = 0.9)) +
  labs(x = "Sample points",y = "CRPS",fill = "Method") +
  scale_fill_manual(values = c("naive" = "blue", "geoadjust" =
  "orange"), labels = c("Naive", "GeoAdjust")) +
  ylim(0.25, 0.45) + facet_wrap(~ delta, scales = "free_x", ncol = 2,
                            labeller = labeller(delta = c("2" =
```

```
                            "Displacement=2km", "6" =
                            "Displacement=6km"))) +
  theme(legend.position = "bottom")
dev.off()


png(filename = "crps4.png", width = 6.27, height =  9.69/2.25, units =
"in", res = 300)
summary_4 %>%
  ggplot(aes(fill = method, y = CRPS, x = factor(npoints))) +
  geom_violin(position = "dodge", alpha = 0.5, trim=F) +
  stat_summary(fun = median, geom = "crossbar", width = 0.75, fatten =
  2, color = "black", position = position_dodge(width = 0.9)) +
  labs(x = "Sample points",y = "CRPS",fill = "Method") +
  scale_fill_manual(values = c("naive" = "blue", "geoadjust" =
  "orange"), labels = c("Naive", "GeoAdjust")) +
  ylim(0.075, 0.15) + facet_wrap(~ delta, scales = "free_x", ncol = 2,
                            labeller = labeller(delta = c("2" =
                            "Displacement=2km", "6" =
                            "Displacement=6km"))) +
  theme(legend.position = "bottom")
dev.off()

# Pairwise differences
# Compute pairwise differences and ratios
sim_summary$id <- rep(1:2400, each=2)
sum_pairwise <- pivot_wider(sim_summary,
                        id_cols = c(id, setting, npoints, delta),
                         names_from = method,
                         values_from = c(intbias, rangebias, sillbias,
                         nuggetbias, RMSE, CRPS, time))

# Pairwise difference
sum_pairwise$int_diff <- sum_pairwise$intbias_geoadjust -
sum_pairwise$intbias_naive
sum_pairwise$sill_diff <- sum_pairwise$sillbias_geoadjust -
sum_pairwise$sillbias_naive
sum_pairwise$range_diff <- sum_pairwise$rangebias_geoadjust -
sum_pairwise$rangebias_naive
sum_pairwise$nugget_diff <- sum_pairwise$nuggetbias_geoadjust -
sum_pairwise$nuggetbias_naive

sum_pairwise$int_ratio <- sum_pairwise$intbias_geoadjust /
sum_pairwise$intbias_naive
sum_pairwise$sill_ratio <- sum_pairwise$sillbias_geoadjust /
sum_pairwise$sillbias_naive
sum_pairwise$range_ratio <- sum_pairwise$rangebias_geoadjust /
sum_pairwise$rangebias_naive
sum_pairwise$nugget_ratio <- sum_pairwise$nuggetbias_geoadjust /
```

```
sum_pairwise$nuggetbias_naive

sum_pairwise$rmse_diff <- sum_pairwise$RMSE_geoadjust -
sum_pairwise$RMSE_naive
sum_pairwise$crps_diff <- sum_pairwise$CRPS_geoadjust -
sum_pairwise$CRPS_naive

sum_pairwise$time_diff <- sum_pairwise$time_geoadjust -
sum_pairwise$time_naive



# plots of pairwise difference
png(filename = "intratio.png", width = 6.27, height =  9.69/2.25, units
= "in", res = 300)
sum_pairwise %>%
  ggplot(aes(fill = factor(delta), y = int_ratio, x = factor(npoints))) +
  geom_boxplot(position = "dodge") +
 # stat_summary(fun = median, geom = "crossbar", width = 0.75, fatten =
 2, color = "black", position = position_dodge(width = 0.9)) +
  geom_hline(yintercept = 1, color = "red", linetype = "dashed", size = 0.8) +
  labs(x = "Sample points",y = "Bias ratio",fill = "Displacement") +
  scale_fill_manual(values = c("2" = "yellow", "6" = "green"), labels =
  c("2 km", "6 km")) +
  ylim(0, 2) + facet_wrap(~ setting, scales = "free_x", ncol = 2,
  labeller = labeller(setting = c("1" = "Sill=0.2, Nugget=0.02", "2" =
  "Sill=0.02, Nugget=0.2",
                                  "3" = "Sill=0.2, Nugget=0.2", "4" =
                                  "Sill=0.02, Nugget=0.02"))) +
  theme(legend.position = "bottom")
dev.off()

png(filename = "sillratio.png", width = 6.27, height =  9.69/2.25, units
= "in", res = 300)
sum_pairwise %>%
  ggplot(aes(fill = factor(delta), y = sill_ratio, x = factor(npoints))) +
  geom_boxplot(position = "dodge") +
  geom_hline(yintercept = 1, color = "red", linetype = "dashed", size = 0.8) +
  labs(x = "Sample points",y = "Bias ratio",fill = "Displacement") +
  scale_fill_manual(values = c("2" = "yellow", "6" = "green"), labels =
  c("2 km", "6 km")) +
  ylim(-0.4, 1.6) + facet_wrap(~ setting, scales = "free_x", ncol = 2,
                                   labeller = labeller(setting = c("1" =
                                   "Sill=0.2, Nugget=0.02", "2" =
                                   "Sill=0.02, Nugget=0.2",
                                    "3" = "Sill=0.2, Nugget=0.2", "4" =
                                    "Sill=0.02, Nugget=0.02"))) +
  theme(legend.position = "bottom")
dev.off()
```

```
png(filename = "nuggetratio.png", width = 6.27, height =  9.69/2.25,
units = "in", res = 300)
sum_pairwise %>%
  ggplot(aes(fill = factor(delta), y = nugget_ratio, x =
  factor(npoints))) +
  geom_boxplot(position = "dodge") +
#  stat_summary(fun = median, geom = "crossbar", width = 0.75, fatten =
2, color = "black", position = position_dodge(width = 0.9)) +
  geom_hline(yintercept = 1, color = "red", linetype = "dashed", size = 0.8) +
  labs(x = "Sample points",y = "Bias ratio",fill = "Displacement") +
  scale_fill_manual(values = c("2" = "yellow", "6" = "green"), labels = c("2 km", "6 km")) +
  ylim(-0.4, 1.6) + facet_wrap(~ setting, scales = "free_x", ncol = 2,
                               labeller = labeller(setting = c("1" =
                               "Sill=0.2, Nugget=0.02", "2" =
                               "Sill=0.02, Nugget=0.2",
                                "3" = "Sill=0.2, Nugget=0.2", "4" =
                                "Sill=0.02, Nugget=0.02"))) +
  theme(legend.position = "bottom")
dev.off()

png(filename = "rangeratio.png", width = 6.27, height =  9.69/2.25,
units = "in", res = 300)
sum_pairwise %>%
  ggplot(aes(fill = factor(delta), y = range_ratio, x =
  factor(npoints))) +
  geom_boxplot(position = "dodge") +
#  stat_summary(fun = median, geom = "crossbar", width = 0.75, fatten =
2, color = "black", position = position_dodge(width = 0.9)) +
  geom_hline(yintercept = 1, color = "red", linetype = "dashed", size = 0.8) +
  labs(x = "Sample points",y = "Bias ratio",fill = "Displacement") +
  scale_fill_manual(values = c("2" = "yellow", "6" = "green"), labels = c("2 km", "6 km")) +
  ylim(-0.4, 1.6) + facet_wrap(~ setting, scales = "free_x", ncol = 2,
                               labeller = labeller(setting = c("1" =
                               "Sill=0.2, Nugget=0.02", "2" =
                               "Sill=0.02, Nugget=0.2",
                                "3" = "Sill=0.2, Nugget=0.2", "4" =
                                "Sill=0.02, Nugget=0.02"))) +
  theme(legend.position = "bottom")
dev.off()

png(filename = "rmsediff.png", width = 6.27, height =  9.69/2.25, units
= "in", res = 300)
sum_pairwise %>%
  ggplot(aes(fill = factor(delta), y = rmse_diff, x = factor(npoints))) +
  geom_boxplot(position = "dodge") +
 # stat_summary(fun = median, geom = "crossbar", width = 0.75, fatten =
 2, color = "black", position = position_dodge(width = 0.9)) +
  geom_hline(yintercept = 0, color = "red", linetype = "dashed", size = 0.8) +
```

```
  labs(x = "Sample points",y = "RMSE difference",fill = "Displacement") +
  scale_fill_manual(values = c("2" = "yellow", "6" = "green"), labels = c("2 km", "6 km")) +
  ylim(-0.01, 0.01) + facet_wrap(~ setting, scales = "free_x", ncol = 2,
                                 labeller = labeller(setting = c("1" =
                                 "Sill=0.2, Nugget=0.02", "2" =
                                 "Sill=0.02, Nugget=0.2",
                                  "3" = "Sill=0.2, Nugget=0.2", "4" =
                                  "Sill=0.02, Nugget=0.02"))) +
  theme(legend.position = "bottom")
dev.off()

png(filename = "crpsdiff.png", width = 6.27, height =  9.69/2.25, units
= "in", res = 300)
sum_pairwise %>%
  ggplot(aes(fill = factor(delta), y = crps_diff, x = factor(npoints))) +
  geom_boxplot(position = "dodge") +
  geom_hline(yintercept = 0, color = "red", linetype = "dashed", size = 0.8) +
  labs(x = "Sample points",y = "CRPS difference",fill = "Displacement") +
  scale_fill_manual(values = c("2" = "yellow", "6" = "green"), labels = c("2 km", "6 km")) +
  ylim(-0.01, 0.01) + facet_wrap(~ setting, scales = "free_x", ncol = 2,
                                   labeller = labeller(setting = c("1" =
                                   "Sill=0.2, Nugget=0.02", "2" =
                                   "Sill=0.02, Nugget=0.2",
                                   "3" = "Sill=0.2, Nugget=0.2", "4" =
                                   "Sill=0.02, Nugget=0.02"))) +
  theme(legend.position = "bottom")
dev.off()

## ratios
png(filename = "intdiff.png", width = 6.27, height =  9.69/2.25, units =
"in", res = 300)
sum_pairwise %>%
  ggplot(aes(fill = factor(delta), y = int_diff, x = factor(npoints))) +
  geom_boxplot(position = "dodge") +
  # stat_summary(fun = median, geom = "crossbar", width = 0.75, fatten =
  2, color = "black", position = position_dodge(width = 0.9)) +
  geom_hline(yintercept = 0, color = "red", linetype = "dashed", size = 0.8) +
  labs(x = "Sample points",y = "Bias difference",fill = "Displacement") +
  scale_fill_manual(values = c("2" = "yellow", "6" = "green"), labels = c("2 km", "6 km")) +
  ylim(-0.05, 0.05) + facet_wrap(~ setting, scales = "free_x", ncol = 2,
                                   labeller = labeller(setting = c("1" =
                                   "Sill=0.2, Nugget=0.02", "2" =
                                   "Sill=0.02, Nugget=0.2",
                                   "3" = "Sill=0.2, Nugget=0.2", "4" =
                                   "Sill=0.02, Nugget=0.02"))) +
  theme(legend.position = "bottom")
dev.off()

png(filename = "silldiff.png", width = 6.27, height =  9.69/2.25, units
```

```
= "in", res = 300)
sum_pairwise %>%
  ggplot(aes(fill = factor(delta), y = sill_diff, x = factor(npoints))) +
  geom_boxplot(position = "dodge") +
  geom_hline(yintercept = 0, color = "red", linetype = "dashed", size = 0.8) +
  labs(x = "Sample points",y = "Bias difference",fill = "Displacement") +
  scale_fill_manual(values = c("2" = "yellow", "6" = "green"), labels = c("2 km", "6 km")) +
  ylim(-0.1, 0.1) + facet_wrap(~ setting, scales = "free_x", ncol = 2,
                               labeller = labeller(setting = c("1" =
                               "Sill=0.2, Nugget=0.02", "2" =
                               "Sill=0.02, Nugget=0.2",
                               "3" = "Sill=0.2, Nugget=0.2", "4" =
                               "Sill=0.02, Nugget=0.02"))) +
  theme(legend.position = "bottom")
dev.off()

png(filename = "nuggetdiff.png", width = 6.27, height =  9.69/2.25,
units = "in", res = 300)
sum_pairwise %>%
  ggplot(aes(fill = factor(delta), y = nugget_diff, x =
  factor(npoints))) +
  geom_boxplot(position = "dodge") +
  #  stat_summary(fun = median, geom = "crossbar", width = 0.75, fatten
  = 2, color = "black", position = position_dodge(width = 0.9)) +
  geom_hline(yintercept = 0, color = "red", linetype = "dashed", size = 0.8) +
  labs(x = "Sample points",y = "Bias difference",fill = "Displacement") +
  scale_fill_manual(values = c("2" = "yellow", "6" = "green"), labels = c("2 km", "6 km")) +
  ylim(-2.5, 1) + facet_wrap(~ setting, scales = "free_x", ncol = 2,
                             labeller = labeller(setting = c("1" =
                             "Sill=0.2, Nugget=0.02", "2" = "Sill=0.02, Nugget=0.2",
                             "3" = "Sill=0.2, Nugget=0.2", "4" =
                             "Sill=0.02, Nugget=0.02"))) +
  theme(legend.position = "bottom")
dev.off()

png(filename = "rangediff.png", width = 6.27, height =  9.69/2.25, units
= "in", res = 300)
sum_pairwise %>%
  ggplot(aes(fill = factor(delta), y = range_diff, x = factor(npoints))) +
  geom_boxplot(position = "dodge") +
  #  stat_summary(fun = median, geom = "crossbar", width = 0.75, fatten
  = 2, color = "black", position = position_dodge(width = 0.9)) +
  geom_hline(yintercept = 0, color = "red", linetype = "dashed", size = 0.8) +
  labs(x = "Sample points",y = "Bias difference",fill = "Displacement") +
  scale_fill_manual(values = c("2" = "yellow", "6" = "green"), labels = c("2 km", "6 km")) +
  ylim(-0.5, 0.5) + facet_wrap(~ setting, scales = "free_x", ncol = 2,
                               labeller = labeller(setting = c("1" =
                               "Sill=0.2, Nugget=0.02", "2" =
                               "Sill=0.02, Nugget=0.2",
```

```
                                "3" = "Sill=0.2, Nugget=0.2", "4" =
                                "Sill=0.02, Nugget=0.02"))) +
  theme(legend.position = "bottom")
dev.off()

png(filename = "timediff.png", width = 6.27, height =  9.69/2.25, units
= "in", res = 300)
sum_pairwise %>%
  ggplot(aes(fill = factor(delta), y = time_diff, x = factor(npoints))) +
  geom_boxplot(position = "dodge") +
  #  stat_summary(fun = median, geom = "crossbar", width = 0.75, fatten
  = 2, color = "black", position = position_dodge(width = 0.9)) +
  geom_hline(yintercept = 0, color = "red", linetype = "dashed", size = 0.8) +
  labs(x = "Sample points",y = "Time difference (seconds)",fill = "Displacement") +
  scale_fill_manual(values = c("2" = "yellow", "6" = "green"), labels = c("2 km", "6 km")) +
  ylim(-50, 300) + facet_wrap(~ setting, scales = "free_x", ncol = 2,
                                labeller = labeller(setting = c("1" =
                                "Sill=0.2, Nugget=0.02", "2" =
                                "Sill=0.02, Nugget=0.2",
                                "3" = "Sill=0.2, Nugget=0.2", "4" =
                                "Sill=0.02, Nugget=0.02"))) +
  theme(legend.position = "bottom")
dev.off()
```

Acknowledgements

I would like to express my sincere gratitude to everyone who supported me throughout the process of completing this thesis.

First and foremost, I am deeply grateful to my supervisor, Prof. dr. Thomas Neyens, for his guidance, support, and invaluable feedback throughout this research. His expertise and critical thinking have been hugely instrumental in shaping this work. I will always treasure his advice and the lessons he has taught me directly and indirectly, from our course in Spatial Epidemiology up to this thesis. His passion in teaching and research is truly admirable.

I extend my heartfelt thanks to Liz Limpoco for her unwavering support in all forms, which has been integral to the completion of this thesis, particularly for her assistance in the simulation study. I would also like to acknowledge Phuphu and Riri for their insights and emotional support throughout, especially during the challenging times.

To my support system here in Belgium, the Filipino community, fellow international students, and friends, you made the duration of this thesis and my master studies bearable. I would also like to express my appreciation to my family, friends, and colleagues back home for their unwavering encouragement and belief in me.

Last but not least, I would like to express my profound gratitude to UHasselt and VLIR-UOS for providing me with this once-in-a-lifetime opportunity to study abroad and accomplish such research. To all individuals who provided assistance in various forms, including administrative support and technical assistance, to ensure my stay here was worthwhile, thank you very much.

This thesis would not have been possible without the contributions of each and every one of you. Thank you for being part of this journey. *Madamo gid nga salamat*! (Thank you!)