THE HONG KONG
POLYTECHNIC UNIVERSITY
香港理工大學
Pao Yue-kong Library
包玉剛圖書館

Copyright Undertaking

This thesis is protected by copyright, with all rights reserved.

By reading and using the thesis, the reader understands and agrees to the following terms:

1. The reader will abide by the rules and legal ordinances governing copyright regarding the use of the thesis.
2. The reader will use the thesis for the purpose of research or private study only and not for distribution or further reproduction or any other purpose.
3. The reader agrees to indemnify and hold the University harmless from and against any loss, damage, cost, liability or expenses arising from copyright infringement or unauthorized usage.

IMPORTANT

If you have reasons to believe that any materials in this thesis are deemed not suitable to be distributed in this form, or a copyright owner having difficulty with the material being included in our database, please contact lbsys@polyu.edu.hk providing details. The Library will look into your claim and consider taking remedial action upon receipt of the written requests.

Pao Yue-kong Library, The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong

http://www.lib.polyu.edu.hk

A BAYESIAN COMPARISON IN STAN AND NIMBLE BY TRIMMED MEAN REGRESSION

LULU ZHANG

MPhil

The Hong Kong Polytechnic University

2021

The Hong Kong Polytechnic University

Department of Applied Mathematics

A Bayesian comparison in Stan and NIMBLE by trimmed mean regression

Lulu Zhang

A thesis submitted in partial fulfilment of the requirements

for the degree of Master of Philosophy

July 2021

Certificate of Originality

I hereby declare that this thesis is my own work and that, to the best of my knowledge and belief, it reproduces no material previously published or written, nor material that has been accepted for the award of any other degree or diploma, except where due acknowledgement has been made in the text.

_____(Signature)

Lulu Zhang (Name of student)

Abstract

The Bayesian statistical paradigm has successful applications across various research fields, including medicine, machine learning, artificial intelligence, and more. Motivated by the arising impact of Bayesian computing, the thesis compares two contemporary Bayesian specialized computational tools, *Stan* and *NIMBLE*. Both have remained under active development, although they are enjoying the merit of freeing the practitioners and analysts from complicated statistical posterior inference by automating the construction of samplers.

The comparison between Stan and NIMBLE is focused on the samplers. Their performances are illustrated by the implementation of weakly informative and informative Bayesian estimation under the trimmed mean regression model by numerical studies, respectively. The informative estimation requires a resampling scheme. We replace Stan with **R** in comparison since resampling is problematic in Stan. We assess performance of Bayesian inference in both parameter estimation and MCMC diagnostics, for the comparison among Stan, NIMBLE, and **R** program.

We conclude that, both Bayesian computing tools can automate posterior approximation accurately and conveniently compared with pure **R** programming by parameters hand-tuning plus mathematical derivation. RStan is efficient in parallel computing but needs contrivance tackling discrete parameters owning to Hamiltonian Monte Carlo sampling. NIMBLE aims to serving users who are accustomed to **R** software but less efficient.

Acknowledgements

As a Master of Philosophy student at Hong Kong Polytechnic University, many people have helped me in various aspects during the past two years. I want to express my sincere gratitude here.

My utmost gratitude goes to my chief supervisor Dr. Binyan JIANG and co-supervisor, Dr. Catherine LIU. I am grateful to Dr. JIANG for his monitoring and warm encouragement at all stages of my study. I am indebted to the painstaking supervision by Dr. Liu for her insight and suggestions. She has continued to train me toward critical thinking, impacting far beyond my acquired academic knowledge. I have benefited a lot from her responsibility and professionalism.

I appreciate the tremendous discussion from my academic brother Mr. Chong ZHONG in Bayesian computing. He has selflessly shared his programming experience and knowledge with me. My thanks go to my another academic elder brother Dr. Sheng XU for his guidance in my early period of learning programming. I acknowledge the help from all team members of our nonparametric Bayesian seminar: Dr. Junshan SHEN, Dr. Xu ZHANG, and Dr. Zhihua MA for their insightful advice and warm discussions. I would like to thank Mr. Qinyi ZHANG for his encouragement and brotherhood during my study.

I would like to thank the exam committee, Prof. Yangxin HUANG, Prof. Jian Qing SHI, Dr. Ting Kei PONG for their constructive suggestions and comments. My thanks also go to Dr. Aki Vehtari, Prof. Perry de Valpine, and particularly to

Dr. Martin Modrák for his timely feedback and comment through the Stan forum. The thesis is partially supported by General Research Funding 15301519, Research Grants Council, University Grants Council, Hong Kong, China.

I want to thank my many lab fellows Miss. Rui ZHOU, Miss. Shu MA, Miss. Changyu LIU, Miss. Peiran YU, Mr. Run ZHENG, Mr. Zhengqi ZHANG, Mr. Chaoyu LIU, and Mr. Shisen LIU. In the past two years, they have given me enthusiastic companionship and encouragement. I thank my considerate roommates Miss. Kedi WANG and Miss. Shuqi WANG, and many other friends.

Last but not least, I want to say "thank you" to my dearest parents and my sister. Your unconditional love and support are always accompanied with me. I keep in mind what Papa said when I was a little girl, "only knowledge and wisdom are always in your mind, and no one can take them away". Thanks again to my parents for having nurtured and educated me. Being their child makes me feel happy.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Machine learning (ML) and Artificial Intelligence (AI) pose challenges and opportunities to all statistical methods dealing with uncertainty in the current big data era. Fortunately, Bayesian models have commonly been used for dealing with uncertainty. Bayesian statistics have been applied successfully to a broad range of fields related to ML and AI as an important branch of computing statistics. The companion of Bayesian computing is becoming more and more important. Also, recent years have seen a surge in Bayesian computing methods to handle a massive data set.

With the trend of ML and AI, we may see further developments in Bayesian computing in the next few years. However, as an essential branch of computing statistics, Bayesian computing has been under full development, covering a wide range of Bayesian awareness. Using specific programming languages to implement statistical models is still one of the biggest obstacles to embracing the Bayesian method. It is noticed that two contemporary programming languages, *Stan* and *NIMBLE*, are constantly evolving, and the underlying algorithms are continually improving. This paper compares these two computing tools for Bayesian computing which can free participants and analysts from the complicated statistical inference of posterior distributions.

R might be the most prevailing language or environment for data analysis and

visualization with an unlimited framework that can write any sampler. R requires users to have appropriate mathematical and statistical training and certain parameter tuning ability. In short, R language provides flexibility for Bayesian programming but needs mathematical and statistical training for inference of posterior distributions to some extend. The traditional Bayesian language **BUGS** (Bayesian inference Using Gibbs Sampling. WinBugs initial released in 1997) and its extension **JAGS** (Just Another Gibbs sampler) have tried to integrate with R, generating packages like *rjags, runjags, BRugs, R2WinBUGS* for implementation of Markov chain Monte Carlo (MCMC) calculations and Bayesian simulations. (Gelfand et al. (1990) and Plummer et al. (2003)). **Stan**, initially released in 2012, is named after Stanislaw Ulam (1909-1984) in memory of the pioneer of Monte Carlo methods. It directly uses C++ dialogue for programming to shorten the compilation time and implements Bayesian sampling by the powerful and efficient Hamiltonian Monte Carlo (HMC) algorithm (Gelman et al. (2015)). **NIMBLE** is initially released in 2015 as a package of R and developed for Bayesian and Likelihood Estimation. It extends and absorbs the advantages of BUGS and JAGS programming languages and proposes a new user-adaptable Metropolis-Hastings (MH) sampling method (de Valpine et al. (2017)). At the same time, NIMBLE has strong adaptability, which reduces the requirements for tuning parameters.

Both Stan and NIMBLE are flexible and do not require users to have solid knowledge of Bayesian and mathematical statistics. This merit is appealing and leads to arising research interests. At the time of writing the thesis, we have searched out over ten publications in using Stan though HMC for posterior sampling (Si et al. (2015), Benavoli et al. (2017), Ghosh et al. (2018), Yao et al. (2018), Buchholz et al. (2021), Gao et al. (2021), Gelman et al. (2020), Weber et al. (2018), Gelman and Vákár (2021), Korner-Nievergelt et al. (2015), McElreath (2018)); there are also several studies applying NIMBLE for posteior sampling (Wehrhahn et al. (2018), Ma

and Chen (2020), Ponisio et al. (2020), Risser and Turek (2020)).

Unfortunately, few works compare their characteristics for beginners to follow up. One may search out a casual wealth of experience sharing casually (Link1 (2020), Link2 (2021) and Kruschke (2014)).

For the purpose of comparison, we demonstrate the difference between Stan and NIMBLE by estimating the heteroscedastic trimmed mean regression with unknown model error distribution from a nonparametric Bayesian perspective. Trimmed mean regression is a more general robust regression tool than quantile regression when analyzing data with heavy tails, outliers, long tails, skewness, and/or other aberrant characteristics. However, there is only frequentist work on trimmed mean regression in the literature, although quantile regression has been studied widely from Bayesian insight. It is nontrivial to develop a Bayesian type estimation procedure considering the heteroscedasticity and many constraints owing to the complex data structure. As a byproduct, we also review robust Bayesian, particularly robust Bayesian regression, since there is little comprehensive review work in the past decade.

We discuss the difference between Stan and NIMBLE based on their samplers. MCMC, also named Markov chain simulation, is a general method to draw the marginal posterior density of the parameter vector. The mathematical theory of MCMC guarantees that the infinite chain will realize the perfect representation of posterior distribution. Instead of directly computing the true posterior density, MCMC allows people to draw samples from an approximate distribution and correct the samples to approximate the true posterior density. The Markov chain enables people to draw samples sequentially, for example, the to-update draws are fully based on the latest draws. Within the trimmed mean regression model setting, we assess mainly the accuracy and efficiency of the estimation procedure. We set the same MCMC scenario implemented in Stan, NIMBLE and R. From the results, all three tools convergence to the same MCMC scenario. We compare MCMC computational

burden based on a new concept which is yet under construction in **R** CRAN.

We conclude that thanks to the benefit brought by HMC, Stan is efficient with a higher effective sample size and MCMC efficiency, especially when parallel computing is implemented. On the other side of the coin, Stan suffers from discrete parameters and randomness of the posterior caused by the resampling scheme during the sampling procedure owning to the gradient element of HMC. NIMBLE aims to serving users who are accustomed to the R software and the use of both MH and Gibbs sampler enables it to adjust to various models. But it is less efficient compared to Stan in the indices like effective sample size and MCMC efficiency.

The rest of the thesis is organized as follows. In Chapter 2 we review the literature in Bayesian robust regression. In Chapter 3, We discuss the difference between Stan and NIMBLE, and including **R** in some situations, based on their samplers and Bayesian inference under the trimmed mean regression setting. In Chapter 4 we have a summary flow.

Chapter 2

Bayesian robust regression

2.1 Robust Bayesian analysis

Robust Bayesian analysis studies the sensitivity analysis of the impact of subjective input on output in a specific range (Insua and Ruggeri (2012)). MCMC inherits the basic statistical concept of sample inference population. With the development of MCMC, people can analyze how different prior information affects posterior distribution. Therefore, it is necessary to discuss Bayesian robustness analysis. Robust Bayesian analysis focuses on the impact of input changes on output. So people begin to pay attention to the robustness of the likelihood function or loss function. The purpose is to find a general method for robust analysis of all components in the Bayesian paradigm. We will review some research status on Bayesian robustness analysis on priors, models and loss functions. Compared with traditional Bayesian analysis, robust Bayesian has lower requirements for a prior. In the next chapter, we will discuss the weakly informative prior and informative prior.

Robust analyses on priors

According to Ferguson (1973), the priors should have two suitable properties: 1) the support set should be large to include all beliefs, and 2) when a sample is given, the posterior distribution is analytically treatable. Most sensitivity studies focused on

the function form of priors. The typical choice is conjugate priors, including Gauss, Beta, Poisson, and others. Berger (2013) mentioned that the flat-tailed distribution might be more robust than the standard conjugate selection. Goldstein (1980) considered a prior with the mean and variance, while Ruggeri (1990) considered quantile classes. One way to distinguish a prior is to select a baseline prior and see what happens when another prior is chosen in a specific neighborhood (chapter 21, Dey and Rao (2005)). A comprehensive method is *global robustness*, which considers all input values compatible with a prior, and calculates the robustness measure when the input in the class changes. Another approach is to use the derivative to study the *local robustness* of the change rate of Bayesian decision-making.

Robust analyses on models

When estimating the parameters and giving the posterior probability of the model parameters, the model class is considered in all parametric reasoning problems. Similar to robust analysis on prior, one may try different models and measure the changes.

Robust analyses on loss

Although a prior choice will lead to some losses, some are still related to the loss function. Therefore, loss robustness focuses on a class of loss functions. The loss robustness measure is defined by taking multiple loss functions and calculating a posterior range.

Bayesian robust regression and Bayesian robust analysis are entirely different concepts. The former focuses on the robustness of sensitivity, while the latter focuses on the resistance of abnormal data. A robust regression model can reduce the impact of aberrant data.

2.2 Bayesian robust regression

Wang and Blei (2018) mentioned that the goal of robust statistics aim to prevent deviations that are difficult to diagnose. The classical ordinary linear regression is "non-robust" to outliers, seriously affected by abnormal data. Robust regression can be considered as an alternative to regression error to normal distribution. Data with outliers, heavy tails, skewness, or heteroscedasticity are ubiquitous, so it is of great significance to study robust regression in the Bayesian paradigm. We introduce the characteristics of these abnormal data through the following examples.

Outliers

Rousseeuw and Yohai (1987) collected data on the average brain and body weight of terrestrial animals. Figure 2.1 is a scatter diagram of logarithmic conversion data. We can see that the species in the graph have a roughly linear relationship with the brain and body weight. There are three points named "Triceratops", "Dipliodocus" and "Brachiosaurus," which are different from the linear parameter. These three species may be outliers in the data.

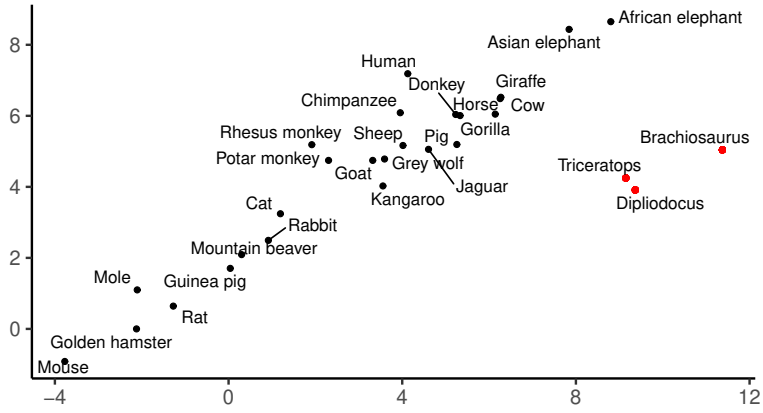


Figure 2.1: Scatter plot of the relationship between brain and body weight of terrestrial animals

Heavy tails

Outliers may also cause the *heavy tail distribution* of the data (Resnick et al. (1997)). Afify et al. (2020) analyzed a heavy-tailed real data set from the insurance field. The data is a monthly indicator of unemployment insurance in the United States from July 2008 to April 2013. Figure 2.2 shows the probability density diagram of 58 observations, from which we can see that the image presents a long-tail property.

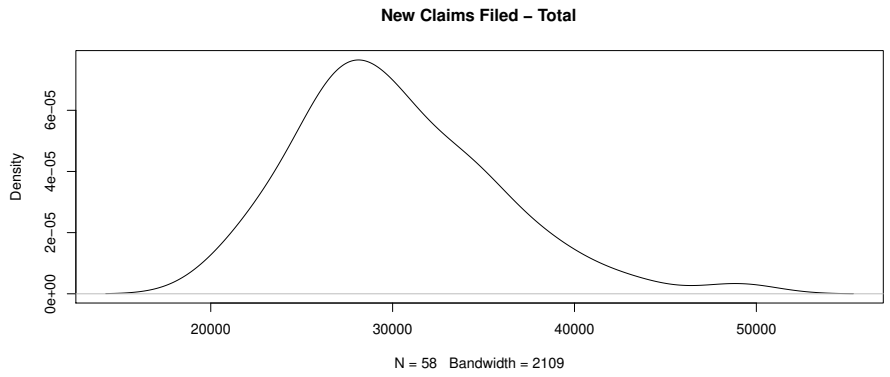


Figure 2.2: The probability density plot of 58 observations of monthly indicators of unemployment insurance in the United States

Skewness

Skewness measures the shape and asymmetry of univariate continuous distribution based on third-order moments. We drew the histogram, probability density diagram and block diagram of the length of stay (LOS) data set. The New York state government health data website recorded the data set of more than 2.3 million patients in 1-20 days. As shown in Figure 2.3, the density distribution is right (positive) skew, and the box diagram also presents asymmetric results.

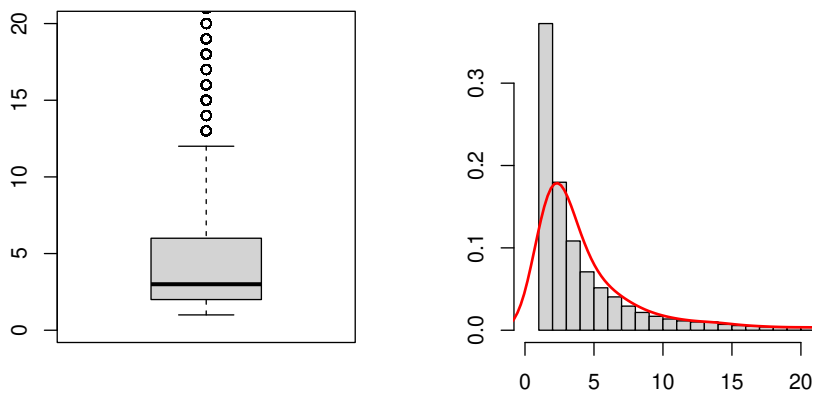


Figure 2.3: The box plot of the length of stay in the hospital (left panel) and the histogram plot (right panel)

heteroscedasticity

Heteroskedasticity occurs when the variance for all observations in a data set is not the same. In Chapter 3 of (Hill et al., 2018, page 298), the author studies the relationship between mean household expenditure on food expenditure and household income. Figure 2.4 shows that the higher the income, the more scattered the observations. Thus, the equal variance is not satisfied.

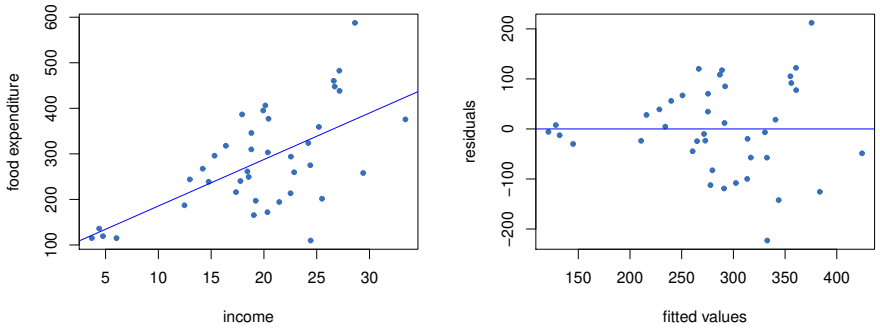


Figure 2.4: Scatter plot of food expenditure and income (left panel), residual plot of food expenditure and income (right panel)

There are lots of articles on robust regression from the perspective of Bayesian. Box and Tiao (1968) used the Bayesian method to solve outlier problems. They considered a linear model, and they proposed that each error could draw from either one or two distributions. They allowed the error term from a distribution with contaminants. Therefore, they believe that each error term may draw from a mixed normal distribution, one component for the consists of a "nonoutliers" normal distribution $N(0, \sigma^2)$ and a "outliers" one $N(0, k^2\sigma^2)$, where k is a constant. West (1984) assumed the error term ε_i is a set of zero-mean exchangeable random variables with standard distribution continuous on $\mathbb{R}$, unimodal and symmetric. He used the heavy-tailed distribution to model the error, constructed as a scale mixture of normals, including the student distribution. Verdinelli and Wasserman (1991) showed that the Gibbs sampler provides a simple method to calculate the posterior distribution, allows the probability of outliers to be unknown, and introduces an additional parameter into the model. They used Gaussian distribution, t distribution and other regression to illustrate their point of view. Peña et al. (2009) showed that the t distribution is not robust when the outliers reach infinity. They proposed a heteroscedasticity model in which the weight of each observation decreases with the distance between observation and data. Ruggeri (2010) assumed that the sampling distribution is from a Dirichlet process and consider Bayesian robust regression from a nonparametric perspective. Gagnon et al. (2020) used the assumption of super-heavy tailed (log-Pareto-tailed normal) distribution instead of traditional assumptions to ensure overall robustness. They showed that the error modeling with fat tail distribution eliminated the influence of infinite outliers.

2.3 Bayesian quantile regression

Although the least-squares (LS) estimator of the parameter vector is efficient when the error follows normal distribution, the estimator is inefficient when the distribution has a heavier tail than the Gaussian distribution. The estimator is highly sensitive to spurious observation. Therefore, it is not appropriate to use conventional regression techniques to deal with aberrant data. Thus, to find some robust models and estimations against outliers, lower or upper conditional quantiles might be estimated instead. Through quantile regression, a more detailed description of the relationship between variables can be obtained. As a supplement of the ordinary mean regression model, quantile regression is commonly used in statistics for its strong robustness against outliers.

Following Koenker and Bassett Jr (1978), the τ-th ($0 < \tau < 1$) conditional quantile function of Y_i given X_i defined as

$$Q_{Y_i}(\tau|X_i) = F_{Y_i}^{-1}(\tau|X_i) = \xi_i(\tau) = X^\top \beta(\tau) \tag{2.1}$$

where β is coefficient for the τ-th quantile level.

2.3.1 Literature review of Bayesian quantile regression

The idea of quantile regression is to model the conditional quantiles of response variables. I roughly divide the literature into the following parts.

Parametric

Yu and Moyeed (2001) utilized the **asymmetric Laplace likelihood (ASL)** function for error term to develop a Bayesian method for quantile regression. The most significant difference from the view of frequentists is that the actual distribution of the data is not considered. The Bayesian inference of quantile regression is performed by forming a likelihood function based on the asymmetric Laplace distribution. They

set Bayesian quantiles in the generalized linear model. They found that choosing an inappropriate uniform prior to parameters resulted in a proper joint posterior distribution. Lee and Neocleous (2010) extended Yu and Moyeed (2001)'s method for **count data** and applied their methodologies in environmental epidemiology. Lancaster and Jae Jun (2010) considered the **empirical likelihood** of tilted exponential to Bayesian quantile regression and gave an unambiguous form and comparison of the posterior density of the quantile. Also, note that combining multiple quantile values in the Lancaster and Jun framework is very simple. Yuan and Yin (2010) proposed a shared-parameter Bayesian quantile regression model of the longitudinal process with the **missing data** model. They assumed that the missing data is associated with the longitudinal outcome process through potential shared random effects. Kozumi and Kobayashi (2011) considered a pseudo asymmetric Laplace distribution for the error term and proposed a Gibbs algorithm based on the **position-scale hybrid** representation. Their method can easily include a scale parameter and can be directly extended to Tobit quantile regression. Yang et al. (2012) considered the quantile regression model and then used the Bayesian **empirical likelihood** to show that the resultant posterior from any fixed prior is asymptotically normal. They focused on estimating several quantiles together and use the empirical likelihood (EL). Luo et al. (2012) assumed that the error term followed the ASL distribution and established a hierarchical Bayesian quantile regression inference model for **longitudinal data**. They explained the dependence between the data by adding random effects to the model. They used Metropolis Hastings algorithm and Gibbs sampling to perform MCMC simulations. Sriram et al. (2013) proved that under the assumption of **ASL misspecification**, the method of asymptotic property and empirical verification can still be widely used. They studied the posterior behavior of a misspecification ASL model with independent but non identically distributed responses. Rahman (2016) considered Bayesian analysis of quantile regression models for **ordered univariate**

data. They assumend the error term follows the normal–exponential mixture representation of the ASL distribution. Bernardi et al. (2016) extended the Bayesian quantile regression framework of the asymmetric Laplace distribution and used the **skewed exponential power (SEP) distribution** to explain the fat tail. They used linear and generalized additive models (GAM) with penalty splines to show the flexibility of SEP in the context of Bayesian quantile regression. Yang et al. (2016) proposed **adjusting the posterior covariance** based on the ASL likelihood function, in the case of complete data and fixed censored data. This adjustment can make posterior reasoning more effective. They pointed out that through simple adjustments, misspecified ASL likelihood can also derive the correct posterior. Zhang and Tang (2017) estimated the parameters and latent varibles based on the Bayesian **empirical likelihood** method. Tong et al. (2021) proposed a Bayesian robust **growth curve modeling** method using the conditional median. They transformed the estimation problem into the maximum likelihood estimation problem of the transformation model by using asymmetric Laplace distribution. Moreover, they used **RStan** to implement their model.

Semiparametric and nonmiparametric

Kottas and Gelfand (2001) proposed two Bayesian modeling methods for the error distribution: semiparametric and completly nonparametric. They considered nonparametric median zero distribution as median regression of error term in linear regression model. Dunson and Taylor (2005) proposed a substitution likelihood characterized by a vector of quantiles and found that it has excellent frequency operation characteristics for several real distribution shapes. Kottas and Krnjajić (2009) considered nonparametric working likelihoods, the Dirichlet process mixture models. Chen and Yu (2009) used regression quantiles to create Markov chains to estimate quantile curves instead of drawing samples from the posterior. Reich et al.

(2010) proposed that the error term distribution follows a unspecified distribution, and the infinite mixture of Gaussian density is considered as the likelihood. Reich et al. (2011) proposed a Bayesian **spatial model**. Their model does not assume the response is Gaussian by such setting and allows complex relationships between covariates and response. Reich (2012) developed a spatiotemporal model that allows the entire distribution of responses to change over time and space. They took the Reich et al. (2011) model and allow the closed form expression of response distribution, so that the Bayesian model can be applied to large spatiotemporal data sets. Reich and Smith (2013) proposed a Bayesian quantile regression model for processing a **censored survival data**. They adopted a semi-parametric method to represent the quantile process as a linear combination of basis functions. Hu et al. (2013) They proposed Bayesian quantile regression for the single-index model. They used the Gaussian process prior of unknown nonparametric link function and the Laplacian distribution on the index vector to deal with high-latitude nonparametric and proposed a method to deal with high-dimensional nonparametric Effective methods of estimating problems. Feng et al. (2015) considered using linear interpolation of quantiles to approximate the likelihood.

longitudinal/ missing/ censored/ special data

Lee and Neocleous (2010) proposed a Bayesian quantile regression model. Their model combined the Yu and Moyeed (2001) method of processing continuous data based on MCMC simulation. Yu et al. (2012) developed a flexible Bayesian framework for regularization in the quantile regression model, similar to Reich et al. (2010), but introducing a hierarchical model framework makes the unimportant coefficient of precise reasoning and contraction zero. They assumed that the error distribution is an infinite mixture of Gaussian densities.d Alhamzawi and Ali (2018) proposed a random effects ordinal quantile regression model to analyze longitudinal data with

ordinal results. They assumed that the error term followed a location-scale mixture representation of the skewed double-exponential distribution and gave an effective the Gibbs algorithm. Xu et al. (2019) used Bayesian quantile regression to analyze macro data, and they conducted data research on the impact of industrial emissions on health in China. Huang (2016) proposed a semiparametric nonlinear mixed effect (QR-SPNLME) model based on Quantile Regression to solve the simultaneous impact of all these typical data features on reasoning in longitudinal research under the Bayesian framework. Huang and Chen (2016) proposed a nonlinear mixed-effects joint (QR-NLMEJ) model based on Quantile Regression. They assumed the covariate model error following a multivariate skew-t distribution. Tian et al. (2016) discussed Bayesian joint quantile regression for mixed effect models. A Bayesian hierarchical model is established under the assumption of asymmetric Laplace error distribution, and the posterior distribution of all unknown parameters is derived based on the Gibbs sampling algorithm. Huang et al. (2017) studied the longitudinal data of the QR based nonlinear mixed effect (NLME) joint model and the covariates of non center position and outliers and / or heavy tail response, non normality and measurement error under the Bayesian framework. They assumed the covariate model error following a multivariate skew-normal distribution. Zhang et al. (2019) established a partially linear mixed-effects joint model (QRPLMJM). They used the covariate measurement error process of skew-normal and skew-t distribution.

variable selection for quantile regression

Li et al. (2010) introduced Bayesian regularized quantile regression and generally treated three different types of penalties: lasso, elastic net penalty and group lasso. Bayesian hierarchical models for each regularized quantile regression problem and Gibbs sampling are derived. Their results show that Bayesian quantile regression is not sensitive to the ASL assumption, even if it is generated from other distributions.

Alhamzawi and Yu (2013) proposed a quantile-related conjugate prior distribution, and their method is based on the regression coefficients of the conditional conjugate prior distribution. They used the bending percentage correlation to obtain an appropriate prior in the model space. Fabrizi et al. (2020) introduced the Bayesian analysis small region estimation of quantile regression model to predict the limited population description. This small area estimation complements from the specific region sample and all other regions in the sample. Their distributional assumptions are very flexible but keep normality, which often plays a central role in small area estimation, as a particular case. They used **JAGS** to implement their model.

2.3.2 Flexible Bayesian quantile regression

From Reich et al. (2010), I have learned that they assumed the data is $(\boldsymbol{x}_i, y_i)$, $i = 1, .., n$, and the heteroskedastic the linear regression model

$$y_i = \boldsymbol{x}_i^\top \beta + \boldsymbol{x}_i^\top \boldsymbol{\gamma} \varepsilon \tag{2.2}$$

Then the y_i's τth qunatile,

$$\tau = F_Y(\xi_\tau) = \mathbb{P}(Y \leq \xi_\tau) = \mathbb{P}(\boldsymbol{x}\boldsymbol{\beta} + (\boldsymbol{x}\boldsymbol{\gamma})\varepsilon \leq \xi_\tau) = \mathbb{P}(\varepsilon \leq \frac{\xi_\tau - \boldsymbol{x}\boldsymbol{\beta}}{\boldsymbol{x}\boldsymbol{\gamma}}) := \Psi_\varepsilon(\frac{\xi_\tau - \boldsymbol{x}\boldsymbol{\beta}}{\boldsymbol{x}\boldsymbol{\gamma}}),$$

where Ψ_ε denotes the cumulative distribution function of ε. Then we have

$$\Psi_\varepsilon^{-1}(\tau) = \frac{\xi_\tau - \boldsymbol{x}\boldsymbol{\beta}}{\boldsymbol{x}\boldsymbol{\gamma}} \Rightarrow \boldsymbol{x}\boldsymbol{\gamma}\Psi_\varepsilon^{-1}(\tau) = \xi_\tau - \boldsymbol{x}\boldsymbol{\beta} \Rightarrow \xi_\tau = \boldsymbol{x}\boldsymbol{\beta} + \boldsymbol{x}\boldsymbol{\gamma}\Psi_\varepsilon^{-1}(\tau)$$

Since

$$y = \boldsymbol{x}\boldsymbol{\beta} + \boldsymbol{x}\boldsymbol{\gamma}\varepsilon \text{ and } F_Y^{-1}(\tau) = \boldsymbol{x}\boldsymbol{\beta} + \boldsymbol{x}\boldsymbol{\gamma}\Psi_\varepsilon^{-1}(\tau)$$

then

$$y = F_Y^{-1}(\tau) + \boldsymbol{x}\boldsymbol{\gamma}(\varepsilon - \Psi_\varepsilon^{-1}(\tau)) \Rightarrow y = \boldsymbol{x}\boldsymbol{\beta}^{(\tau)} + \boldsymbol{x}\boldsymbol{\gamma}^{(\tau)}\varepsilon^{(\tau)}$$

Then model (2.2) may be rewritten as

$$y_i = \boldsymbol{x_i}\boldsymbol{\beta}^{(\tau)} + \boldsymbol{x_i}\boldsymbol{\gamma}^{(\tau)}\varepsilon_i^{(\tau)}, \quad i = 1, \ldots, n \tag{2.3}$$

where $\varepsilon_i^{(\tau)} = \varepsilon_i - \Psi_\varepsilon^{-1}(\tau)$, It is worth noting here that $\varepsilon_i^{(\tau)}$ has τ^{th} quantile equal to zero. since,

$$\begin{aligned}\tau = \Psi_\varepsilon(\xi_\tau) = \mathbb{P}(\varepsilon \le \xi_\tau) = \mathbb{P}(\varepsilon - \Psi_\varepsilon^{-1}(\tau) \le \xi_\tau - \Psi_\varepsilon^{-1}(\tau)) = \mathbb{P}(\varepsilon^{(\tau)} \le \xi_\tau - \Psi_\varepsilon^{-1}(\tau)) \\ = \mathbb{P}(\varepsilon^{(\tau)} \le 0) = \Psi_{\varepsilon^{(\tau)}}(0)\end{aligned}$$

Through simple algebraic calculations, we have the following conclusions, model (2.2) may be written as (2.3), and the error term $e_i^{(\tau)}$ has τth quantile equal to zero.

A flexible residual distribution h established as an infinite mixture of simple densities,

$$h(\varepsilon|\boldsymbol{\mu}, \boldsymbol{\sigma^2}) = \sum_{k=1}^{\infty} p_k f(\varepsilon|\boldsymbol{\mu}_i, \boldsymbol{\sigma}_k^2, q_k) = \sum_{k=1}^{\infty} p_k (q_k \phi_{(\mu_{1k}, \sigma_{1k}^2)} + (1 - q_k)\phi_{(\mu_{2k}, \sigma_{2k}^2)})$$

where $q_k = \{\Phi(-\mu_{1k}/\sigma_{1k}) - \Phi(-\mu_{2k}/\sigma_{2k})\}^{-1}\{\tau - \Phi(-\mu_{2k}/\sigma_{2k})\}$, and $p_k = V_k(1 - \sum_{j<k} p_j)$, with Φ being the cumulative distribution function of standard normal.

They took $\mu_{1k}, \mu_{2k} \overset{iid}{\sim} ASL : \lambda^{-1}\exp[-\mu\lambda^{-1}\{\tau - I(\mu \le 0)\}]$, and $\sigma_{1k}, \sigma_{2k} \sim \mathrm{U}(0, c_1)$ for some sizeable constant c_1. This leads to the truncated prior

$$\begin{aligned}P(\mu_{1k}, \mu_{2k}, \sigma_{1k}, \sigma_{2k}|\lambda, \tau, c_1) \propto \exp\{-\frac{\mu_{1k}}{\lambda}(\tau - I[\mu_{1k} \le 0]) - \frac{-\mu_{2k}}{\lambda}(\tau - I[\mu_{2k} \le 0])\} \\ \times I[0 \le \sigma_1 \le c_1] \times I[0 \le \sigma_2 \le c_1] \times I[0 \le q_k \le 1]\end{aligned} \tag{2.4}$$

Recall that the multivariate normal distribution has density

$$f_{\boldsymbol{X}}(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_k) = \frac{1}{\sqrt{(2\pi)^k|\boldsymbol{\sigma}|}}\exp\{-\frac{1}{2}(\boldsymbol{x} - \boldsymbol{\mu})^\top \boldsymbol{\sigma}^{-1}(\boldsymbol{x} - \boldsymbol{\mu})\}$$

where $\boldsymbol{x}$ is a real k-dimensional column vector and $|\boldsymbol{\sigma}| \equiv \det \boldsymbol{\sigma}$ is the determinant of $\boldsymbol{\sigma}$. Following Savage (2016) the typical workflow and inference can be shown as follow. The full conditional distribution for β.

For model:

$$Y_{n\times 1} = \boldsymbol{X_{n\times p}}\boldsymbol{\beta_{n\times 1}} + (\boldsymbol{X_{n\times p}}\boldsymbol{\gamma_{p\times 1}}^{\mathsf{T}}\mathrm{diag}(\varepsilon_{n\times 1}))^{\top}$$

where $Y = \begin{pmatrix} y_1 \\ \dots \\ y_n \end{pmatrix}$, $X = \begin{pmatrix} x_{11} & \dots & x_{1p} \\ \dots & \dots & \dots \\ x_{n1} & \dots & x_{np} \end{pmatrix}$, $\beta = \begin{pmatrix} \beta_1 \\ \dots \\ \beta_n \end{pmatrix}$, $\gamma = \begin{pmatrix} \gamma_1 \\ \dots \\ \gamma_n \end{pmatrix}$, $\varepsilon = \begin{pmatrix} \varepsilon_1 \\ \dots \\ \varepsilon_n \end{pmatrix}$

The prior

- $\beta_j \sim \mathrm{N}(0, c_2)$, $j = 1, ..., p$,
  $\beta_{p\times 1} \sim \mathrm{N}(\mathbf{0}, \boldsymbol{\Sigma_{C_2}})$, where $\Sigma_{C_2} = \mathrm{diag}(c_2)$,
- $\varepsilon_i \sim \mathrm{N}(\mu_\varepsilon, \sigma_\varepsilon^2)$, $i = 1, ..., n$
  $\varepsilon_{n\times 1} \sim \mathrm{N}(\mu_\varepsilon, \Sigma_\varepsilon)$, where $\Sigma_\varepsilon = \ \mathrm{diag}(\sigma_\varepsilon^2)$.
- $\gamma \sim \mathrm{Gamma}(0.1, 0.1)$.
- $y_i \sim \mathrm{N}(x_i^{\top}\beta + x_i\gamma\mu_\varepsilon, (x_i^{\top}\gamma\sigma_\varepsilon)^2)$, $i = 1, ..., n$.
- $G_i \in \{1, 2, 3, ...\}$, $G_i \sim \mathrm{Categorical}(p_1, P_2, ...)$
- $H_i \in \{1, 2\}$, $h_i \sim \mathrm{Categorical}(qG_i, 1 - qG_i)$.
- $Y \sim \mathrm{N}(X\beta + X\gamma\mu_\varepsilon, \Sigma_M)$, $i = 1, ..., n$, $\Sigma_M = \mathrm{diag}((X\gamma)^{\top})\Sigma_\varepsilon\mathrm{diag}(X\gamma)$.
- $f(y_1, ..., y_n|x_1, ..., x_n, \beta, \gamma, \varepsilon) = \Pi_{i=1}^{n}\frac{1}{x_i\gamma\sigma_\varepsilon\sqrt{2\pi}}\exp\{-\frac{(y_i - x_i\beta - x_i\gamma\mu_\varepsilon)^2}{2(x_i\gamma\sigma_\varepsilon^2)^2}\}$
- $f(Y|X, \beta, \gamma, \varepsilon) = (2\pi)^{-\frac{n}{2}}((X\gamma)^{\top}|\Sigma_\varepsilon|X\gamma)^{-\frac{1}{2}}\exp\{-\frac{1}{2}(Y - X\beta - X\gamma\mu_\varepsilon)^{\top}(\Sigma_M)^{-1}(Y - X\beta - X\gamma\mu_\varepsilon)\}$
- $p(\beta_1, ..., \beta_n) = \Pi_{j=1}^{p}\frac{1}{\sqrt{2\pi c_2}}\exp\{-\frac{\beta_j^2}{2c_2}\}$

- $p(\beta) = (2\pi)^{-\frac{p}{2}}|\Sigma_{c_2}|^{-\frac{1}{2}}\exp\{-\frac{1}{2}\beta^\top\Sigma_{c_2}^{-1}\beta\}$

The inference procedure:

$$
\begin{aligned}
p(\beta|X,Y,\gamma,\varepsilon) &= f(Y|X,\beta,\gamma,\varepsilon)P(\beta)P(\gamma)P(\varepsilon) \\
&\propto f(Y|X,\beta,\gamma,\varepsilon)P(\beta) \\
&= (2\pi)^{-\frac{n}{2}}((X\gamma)^\top|\Sigma_\varepsilon|X\gamma)^{-\frac{1}{2}}\exp\{-\frac{1}{2}(Y-X\beta-X\gamma\mu_\varepsilon)^\top(\Sigma_M)^{-1} \\
&\qquad (Y-X\beta-X\gamma\mu_\varepsilon)\}(2\pi)^{-\frac{p}{2}}|\Sigma_{c_2}|^{-\frac{1}{2}}\exp\{-\frac{1}{2}\beta^\top\Sigma_{c_2}^{-1}\beta\} \\
&\propto \exp\{-\frac{1}{2}(Y-X\beta-X\gamma\mu_\varepsilon)^\top(\Sigma_M)^{-1}(Y-X\beta-X\gamma\mu_\varepsilon)\}\exp\{-\frac{1}{2}\beta^\top\Sigma_{c_2}^{-1}\beta\} \\
&= \exp\{-\frac{1}{2}(R-X\beta)^\top(\Sigma_M)^{-1}(R-X\beta)\}\exp\{-\frac{1}{2}\beta^\top\Sigma_{c_2}^{-1}\beta\}\ (R=Y-X\gamma\mu_\varepsilon) \\
&= \exp\{-\frac{1}{2}(R^\top-\beta^\top X^\top)(\Sigma_M)^{-1}(R-X\beta)\}\exp\{-\frac{1}{2}\beta^\top\Sigma_{c_2}^{-1}\beta\} \\
&= \exp\{-\frac{1}{2}[R^\top(\Sigma_M)^{-1}R-R^\top(\Sigma_M)^{-1}X\beta-\beta^\top X^\top(\Sigma_M)^{-1}R+ \\
&\qquad \beta^\top X^\top(\Sigma_M)^{-1}X\beta)]\}\exp\{-\frac{1}{2}\beta^\top\Sigma_{c_2}^{-1}\beta\} \\
&\text{then } R^\top(\Sigma_M)^{-1}X\beta = (\beta^\top X^\top(\Sigma_M)^{-1}R)^\top(\text{since M is diagonal matrix},M^\top=M) \\
&\propto \exp\{-\frac{1}{2}[\beta^\top X^\top(\Sigma_M)^{-1}X\beta-2\beta^\top X^\top(\Sigma_M)^{-1}R]\}\exp\{-\frac{1}{2}\beta^\top\Sigma_{c_2}^{-1}\beta\}
\end{aligned}
$$

Mean: $(X^\top(\Sigma_M)^{-1}X+\Sigma_{c_2}^{-1})^{-1}X^\top(\Sigma_M)^{-1}R$

Variance: $(X^\top(\Sigma_M)^{-1}X+\Sigma_{c_2}^{-1})^{-1}$

Chapter 3

Comparison of Stan with NIMBLE: illustrated by Bayesian trimmed mean regression

Bayesian methods are often associated with a large number of calculations. It is not an inherent property of the Bayesian paradigm. Bayesian computing is designed to estimate the posterior distribution, analogous to frequentist computing estimating the sampling distribution. The MCMC produces a large sample of representative values, and the sample size of the data does not limit the accuracy of the approximation. However, the MCMC approximation can be arbitrarily precise even for non-Gaussian posteriors by increasing the computational effort. For multivariate models with $p > 1$ parameters, the samples $\theta_j^{(1)}, \ldots, \theta_j^{(S)}$ follow the marginal posterior distribution of θ_j, $p(\theta_j|\mathbf{Y})$. Critically, we do not need to analytically integrate $p(\theta_j|\mathbf{Y}) = \int f(\boldsymbol{\theta}|\mathbf{Y}) d\theta_1 \ldots d\theta_{j-1} d\theta_{j+1} \ldots d\theta_p$. Because each sample consists of a random draw from all parameters, MC sampling automatically produces samples from the marginal distribution θ_j accounting for uncertainty in the other parameters. Once we have posterior samples, summarizing the posterior or even complicated functions of the posterior is straightforward and this is one of the appeals of MC sampling. However, generating valid samples from the posterior distribution is not

always straightforward. We will focus on three sampling algorithms in Section 3.2.

3.1 Why Bayesian trimmed mean regression

3.1.1 What is Trimmed mean

The trimmed mean is a statistical measure that takes advantage of the mean and quantile. It shows the primary trend and very robust. The basic idea of the trimmed mean is discarding parts of the sample or distribution. Welsh et al. (1987) defined the sample trimmed mean and assumed the median regression is exactly equal to mean regression. Serfling (2009) and Dhar and Chaudhuri (2012) gave the definition of trimmed mean under symmetric conditions distributions. We give a general definition as follows.

Definition 3.1 (Trimmed mean)**.** *A random variable $X \sim F$, where F is the cumulative distribution function. Denote the τ_i-th quantile of F to be $F^{-1}(\tau_i) = \inf\{t : F(t) \geq \tau_i\}$ for $i = 1, 2$. Let $0 \leq \tau_1 \leq \frac{1}{2} \leq \tau_2 \leq 1$. The $(\tau_1, \tau_2)-$ trimmed mean for a random variable X with distribution F is written as*

$$T_{\tau_1,\tau_2}(F) := T(\tau_1, \tau_2; F) = \frac{1}{\tau_2 - \tau_1} \int_{F^{-1}(\tau_1)}^{F^{-1}(\tau_2)} t dF(t).$$

For a trimmed mean regression, it is general and flexible to include both quantile regression and general mean regression. The dependent variable can be regarded as the sum of two parts: (1) the linear function of the independent variable, (2) the random error. Let us look into the main mean regression part as follows. Taking a continuous random variable with density function $f(y) = F'(y)$ for instance. Let $\tau_2 = \tau, \tau_1 \to \tau$. By the mean value theorem, we have

$$\lim_{\tau_1 \to \tau} T_{\tau_1,\tau_2}(F) = \lim_{\tau_1 \to \tau} \frac{[F^{-1}(\tau) - F^{-1}(\tau_1)]\xi f(\xi)}{\tau - \tau_1}, \quad \xi \in [F^{-1}(\tau_1), F^{-1}(\tau)].$$

If $f[F^{-1}(\tau)] \neq 0$, it is easy to see that this limit is $F^{-1}(\tau)$ based on the facts $[F^{-1}(\tau) - F^{-1}(\tau_1)](\tau - \tau_1)^{-1} \to f[F^{-1}(\tau)]^{-1}$, $\xi \to F^{-1}(\tau)$ and $f(\xi) \to f[F^{-1}(\tau)]$. This shows that trimmed mean regression can deduce quantile regression. In addition, trimmed mean regression may also reduce to the general mean regression if we do not trim on both sides. Therefore, trimmed mean regression is a robust regression. This robustness can handle many abnormal data, which is widespread in the field of economics, social sciences, and biomedical, (Dolmas et al. (2005), Atkinson et al. (2016), Pusparum et al. (2017), Rydell et al. (2009), Chahal et al. (2020), Hovik et al. (2016)). A real data example can be found in Johnson et al. (2007). This book provides data on the mineral content of the arm bones of 25 subjects. Figure 3.1 shows the residual plot based on l_1-norm. From the residual plot we may conclude that there may be some outliers in the data, such as subjects with numbers 19 and 23. This data will be discussed later in the Section 3.4.

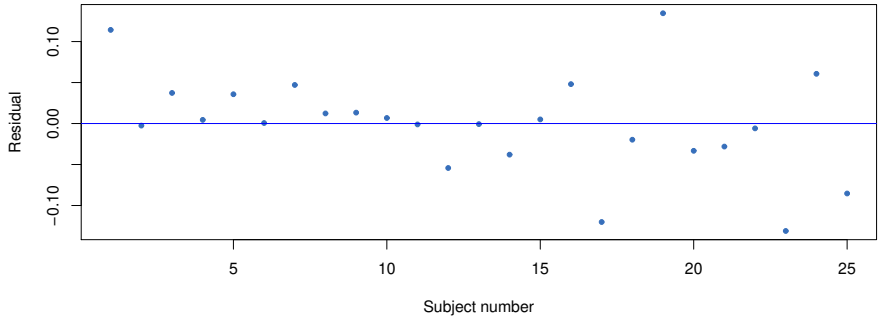


Figure 3.1: Residual plot of the mineral context of arm bones of 25 subjects based on l_1-norm.

3.1.2 Literature review of trimmed mean regression

The trimmed mean regression has emerged as a valuable supplement to ordinary mean regression. The popularity of the trimmed mean seems attributable to both theoretical and practical contestations. Tukey and McLaughlin (1963) put forward the idea of the trimmed mean. They defined the trimmed mean estimator based

on ordering deviations. They also mentioned Winsorized mean which means the arithmetic mean of the n values obtained by replacing each of the g lowest values by the value y_g of the nearest other value y_{g+1}, and each of the highest value y_{g+h} of the nearest other value y_{n-g}. Bickel et al. (1965) defined the α-trimmed mean of the sample $X_1, \ldots, X_n$ based on order statistics. Similarly, they defined the α-Winsorized mean and extend the method to higher dimensions. Bickel (1973) considered the general linear regression model with independent symmetric errors. He constructed an estimator based on a preliminary estimate and has good asymptotic properties. He considered a linear regression and supposed the error term is independent and identically distributed with common density f concerning the Lebesgue measure. He discussed three different classes estimate for a location model, M estimates, linear combinations of order statistics and rank tests proposed. Koenker and Bassett Jr (1978) proposed the trimmed mean to be the least-squares estimator calculated after discarding those observations. Ruppert and Carroll (1980) proposed trimmed least squares estimation in the linear model. They used residuals from a preliminary estimator and estimator defied by Koenker and Bassett Jr (1978) to define trimmed mean. Welsh et al. (1987) considered a linear regression model and assumed the error term is independent and identically distributed draw from F, and without loss of generality suppose that $F(0) = \frac{1}{2}$. He examined the structure of the estimator $T_n = T(F_n)$ defined in

$$T(G) = (\beta - \alpha)^{-1} \int_{\alpha}^{\beta} G^{-1}(t)dt,$$

where $G^{-1}(t) = \inf\{s : G(s)\}$. Then $T_n = T(F_n)$. Chen (1997) considered the linear regression model and assumed that the error term independent and identically distributed with a distribution function F of zero mean and constant variance. He constructed the weighted trimmed mean through the symmetric quantile. Following Chen and Chiang (1996), defined the weighted trimmed mean based on symmet-

ric quantile function. When non-normal data are not symmetric, researchers have proposed that data be trimmed asymmetrically (De Wet and Van Wyk (1979) and Hogg (1974)). Accordingly, rather than trim an equal amount from each tail of the distribution, they suggest that different amounts of data should be trimmed from the right and left tails of the distribution. Furthermore, the number of observations to be trimmed from each tail is determined by the characteristics of the sample data. Accordingly, these estimators are referred to as adaptive robust estimators. However, adaptive estimators that deal with estimating regression parameters must estimate a score function, including the derivative of the logarithm of an unknown density function, which makes them computationally complicated. Moreover, unlike most nonadaptive estimators, the adaptive estimators cannot naturally be generalized to other statistical problems, especially when the Fisher information is unknown.

3.1.3 Bayesian trimmed mean regression (BTMR)

We consider a heteroscedasticity model similar to Reich et al. (2010) and He (1997).

$$y = \boldsymbol{x}^\top \boldsymbol{\beta} + \exp(\boldsymbol{x}^\top \boldsymbol{\gamma})\varepsilon \tag{3.1}$$

where coefficients $\boldsymbol{\beta}$ and $\boldsymbol{\gamma}$ are two vectors of parameters and $\boldsymbol{\varepsilon}_i$ are independent and identically distributed (i.i.d.) follow the unknown distribution $F_\varepsilon(s)$ with density $f_\varepsilon(s)$. By adding an exponential operation, the models do not place restrictions on $x^\top \gamma > 0$.

We build a semiparametric model in which the treatment effect of response variable y_i depends on covariates variable $\boldsymbol{x}_i$ is reflected in mean and variance. For mean function, we use $\boldsymbol{\beta}$ to describe the mean function relationship between the independent and dependent variable. In terms of variance function, we use the function of $\boldsymbol{\gamma}$ and covariates to describe. For parameterizations, as long as we estimate the value of $\boldsymbol{\beta}$ and $\boldsymbol{\gamma}$, we can know how the response variable depends on the covariates for

mean and variance. However, the difficulty of nonparametric part is that the residual distribution is unknown.

We consider a flexible density $f_\varepsilon(s)$ for errors $\boldsymbol{\varepsilon}_i$ as an infinite mixture of simple densities $f_{\boldsymbol{\eta}}(s)$ that each satisfy the desired trimmed constraint, which is τ_1 and τ_2 trimmed-mean-zero. We assume density $f_\varepsilon(s)$ can be represented with a mixture form with the kernel $f_{\boldsymbol{\eta}}(s)$ and mixing distribution $G(\boldsymbol{\eta})$:

$$f_\varepsilon(s) = \int f_{\boldsymbol{\eta}}(s) dG(\boldsymbol{\eta}). \tag{3.2}$$

The kernel $f_{\boldsymbol{\eta}}(s)$ should be designed to satisfy the equations

$$F_{\boldsymbol{\eta}}^{-1}(\tau_1) = \theta_1, \quad F_{\boldsymbol{\eta}}^{-1}(\tau_2) = \theta_2, \quad E_{\tau_1,\tau_2}(F_{\boldsymbol{\eta}}) = 0, \tag{3.3}$$

which is equivalent to

$$\int_{-\infty}^{\theta_1} f_{\boldsymbol{\eta}}(s)dx = \tau_1, \quad \int_{-\infty}^{\theta_2} f_{\boldsymbol{\eta}}(s)dx = \tau_2, \quad \int_{\theta_1}^{\theta_2} x f_{\boldsymbol{\eta}}(s)dx = 0, \tag{3.4}$$

where $F_{\boldsymbol{\eta}}(s)$ is the distribution of $f_{\boldsymbol{\eta}}(s)$. It means that the τ-trimmed mean of $F_{\boldsymbol{\eta}}$ is zero. Here parameters θ_1 and θ_2 are the τ_1- and τ_2- quantiles of $F_{\boldsymbol{\eta}}(s)$ for any $\boldsymbol{\eta} = (\boldsymbol{\mu}, \boldsymbol{\sigma^2})$. All those constraints make sure that the density $f_\varepsilon(s)$ is also τ_1 and τ_2-trimmed mean-zero and θ_1, θ_2 are its τ_1 and τ_2-quantiles.

Constraint 3.1. $\sum_{i=1}^4 \alpha_i \Phi_{\sigma_i}(\theta_1 - \mu_i) = \tau_1$

Constraint 3.2. $\sum_{i=1}^4 \alpha_i \Phi_{\sigma_i}(\theta_2 - \mu_i) = \tau_2$

Constraint 3.3. $\sum_{i=1}^4 \alpha_i = 1$

Constraint 3.4. $\sum_{i=1}^4 \alpha_i d_i = 0$

where Φ is the standard normal distribution function and $d_i = \int_{\theta_1}^{\theta_2} s\Phi_{\sigma_i}(s - \mu_i)ds$.

Table 3.1: Comparison of quantile regression and trimmed mean regression

	Quantile regression	Trimmed mean regression
model	$y = \boldsymbol{x}^\top \boldsymbol{\beta}^{(\tau)} + (\boldsymbol{x}^\top \boldsymbol{\gamma}^{(\tau)})\varepsilon^{(\tau)}$	$y = \boldsymbol{x}^\top \boldsymbol{\beta} + \exp(\boldsymbol{x}^\top \boldsymbol{\gamma})\varepsilon$
$f_\varepsilon(s) \sim$ DPM	$\sum_{k=1}^{\infty} p_k f_\eta(s)$	$\int f_{\boldsymbol{\eta}}(s) dG(\boldsymbol{\eta})$
$f_\eta(s)$	$\sum_{i=1}^{2} q_{ik} \varphi_{\sigma_{ik}}(s - \mu_{ik})$	$\sum_{i=1}^{4} \alpha_i \varphi_{\sigma_i}(s - \mu_i)$
constraints	$\sum_{i=1}^{2} q_{ik} = 1,\ 0 < q_{ik} < 1$	$\sum_{i=1}^{4} \alpha_i = 1,\ 0 < \alpha_i < 1$
	$q_{ik} = \frac{\tau - \Phi(-\mu_{2k}/\sigma_{2k})}{\Phi(-\mu_{1k}/\sigma_{1k}) - \Phi(-\mu_{2k}/\sigma_{2k})}$	$\int_{-\infty}^{\theta_1} f_{\boldsymbol{\eta}}(s) ds = \tau_1,$
		$\int_{-\infty}^{\theta_2} f_{\boldsymbol{\eta}}(s) ds = \tau_2,$
		$\int_{\theta_1}^{\theta_2} s f_{\boldsymbol{\eta}}(s) ds = 0,$

Trimmed mean expression can be regarded as a supplement to the quantile regression, Reich et al. (2010) assumed that the residual distribution function is an infinite Gaussian mixture, and we assume density $f_\varepsilon(s)$ can be represented with a mixture form with the kernel $f_{\boldsymbol{\eta}}(s)$ and mixing distribution $G(\boldsymbol{\eta})$. In both QR and TMR models, the probability distribution of the error term is unspecified. But the method here is more complicated since the kernel has four components determined by an equation system without a close form of a solution, whereas the QR in Reich et al. (2010) has only one equation with a close form of solution. Also, in the Bayesian framework, Reich et al. (2010) assumed prior is stochastically centered on the asymmetric Laplace density. Reich and Ghosh (2019) adjusted the model so that it has only one restriction. However, to satisfy the goal of trimmed-mean-zero, we establish four constraints.

3.2 Stan vs. NIMBLE: MCMC samplers

The question that a Bayesian analyst is most interested in is how to approximate the posterior distribution. When accuracy and stability of the approximation are the most concerned, simulation-based (stochastic) methods, which are aimed to generate samples from the posterior distribution, are widely used (Wang and Park, 2020). Particularly, the MCMC uses a Markov process to obtain collections of dependent variables. When implementing an MCMC process, selecting a proper MCMC sampler from the posterior distribution is important to Bayesian statistics (van de Schoot et al., 2021).

The MCMC technique plays an important role in both Stan and NIMBLE. The essential difference between Stan and **NIMBLE** is the MCMC samplers on which their MCMC are based. Stan's MCMC techniques are based on two samplers, one is Hamiltonian Monte Carlo (HMC) and the other is the No-U-Turn sampler (NUTS), a modification of HMC, which indicates the execution of Stan should be taken under the HMC framework. However, NIMBLE provides different MCMC samplers, including MH and Gibbs samplers, except HMC. The use of different samplers makes a great difference in the application of these two languages, including model specification, model estimation, and posterior inference. The selection of a good MCMC sampler from the posterior distribution is important to Bayesian statistics (van de Schoot et al., 2021). Therefore, to compare the difference between Stan and NIMBLE, we should first discuss the MCMC samplers.

3.2.1 A brief introduction to Stan and NIMBLE

Stan

Users can find a detailed introduction about Stan on its official website (Link3 (2021)). Stan was designed specifically for defining and fitting statistical models.

Stan must be called from another more general-purpose language such as **R**, MATLAB, Julia, or Python. Stan designed an interface at **R**, and users can use *rstan* to call Stan in **R** (Link4 (2021)). For faster compilation speed, stan is written in C++ syntax. You can download its user manual (Stan Development Team (2020)) on the official website, which contains a large number of examples using Stan to complete Bayesian inference.

NIMBLE

NIMBLE is also a contemporary Bayesian programming tool. NIMBLE is a package that can only interface with R. NIMBLE can be used for statistical calculations of general model structures, especially hierarchical models. Users can find detailed information on its website (Link5 (2021)). The emergence of NIMBLE has increased the flexibility of Bayesian programming. Users can balance between programming algorithm languages of different models and advanced programmability and execution efficiency. NIMBLE sets the setup function to be executed in **R** but not compiled, and one or more run functions complete MCMC iteration. NIMBLE inherits the syntax of JAGS and BUGS, which is friendly to traditional users.

***R** software environment*

We briefly discuss the use of **R** in MCMC and Bayesian statistics. **R** is a flexible platform that can call Stan and NIMBLE. More than this, **R** is a strong programming language that users can directly construct different MCMC samplers, including the aforementioned MH, Gibbs, and HMC. Programs written in **R** using appropriate samplers may be more effective for specific problems than using Stan or NIMBLE, such as Zhou et al. (2020). However, for users who are not well trained in Bayesian analysis and computational science, it is always difficult to write a well **R** program. Therefore, Stan and nimble may be more attractive from an application perspective.

3.2.2 Metropolis-Hastings sampler

MH algorithm was developed by Metropolis et al. (1953) and subsequently generalized by Hastings (1970). We first briefly introduce the algorithm of the MH sampler in Algorithm 1. MH sampling replaces the exact full conditional distribution with a draw from a candidate distribution followed by an accept/ reject step. Here, we call the distribution q a jump distribution that the parameter θ jumps from state $t-1$ to state t. In each iteration, we make a random walk from the previous state to jump to an updated state. Then one should determine whether this state is suitable to correct the previous state by a rejection-acceptance step. The acceptance rate is the ratio of a posterior distribution in a consecutive state. The higher the ratio, the more likely it is to accept the update status. Once the update state is rejected, the Markov chain will remain in the previous state.

Algorithm 1 Metropolis-Hastings algorithm sampler

Initialize $\theta^{(0)} \sim q(\theta)$
for iteration $t = 1, 2, \ldots$ **do**
  Propose: $\theta^{cand} \sim q(\theta^{(t)}|\theta^{(t-1)})$
  Acceptance Probability:
    $\alpha(\theta^{cand}|\theta^{(t-1)}) = \min\{1, \frac{q(\theta^{(t-1)}|\theta^{cand})\pi(\theta^{cand})}{q(\theta^{cand}|\theta^{(t-1)})\pi(\theta^{(t-1)})}\}$
  $u \sim$ Uniform $(0, 1)$
  if $u < \alpha$ **then**
    Accept the proposal $\theta^{(t-1))} \leftarrow \theta^{cand}$
  else
    Reject the proposal $\theta^{(t-1))} \leftarrow \theta^{cand}$
  end if
end for

The sampler, also known as random walking MH sampler, is designed based on MCMC convergence theory. There are two steps to prove the convergence of sample chain. One is to prove that the sample chain is a Markov chain and the stationary distribution is unique. The other is to prove that the stationary distribution is

the target marginal posterior distribution. If the transition between two states is a random walk on an appropriate distribution, we can easily get the proof of the first step. The term "appropriate" means that the transition distribution, or the jump distribution at the t-th iteration, has a positive probability of jumping to all states. The second step of the proof is a bit complicated, and one can find details in Gelman et al. (2013). However, we can conclude that the reject accept step in the algorithm is related to the second step.

The MH sampler is quite simple since the transition step relies only on the random walk without any information of the form of the posterior density. In particular, it is flexible to select the appropriate jump distribution. In most cases, the Gaussian distribution is the proper choice for continuous variables, and the standard deviation is the tuning parameter (Reich and Ghosh, 2019). The tuning parameter plays a role of "jump size" in the transition, and its selection is important. Since the Gaussian distribution is symmetrical, the MH sampler is simplified to a metropolis sampler, a simpler algorithm. However, enjoying the simplicity needs to pay effort in computational efficiency. The acceptance rate of the MH sampler is low (Hoffman and Gelman, 2014), which requires long iterations to achieve convergence and extract enough effective samples. The word "effective samples" means weak auto-correlation. They are used as analogous independent samples. MH sampler has no mechanism to improve the acceptance rate other than tuning the jumping distribution. Thus, a longer chain is required when using the MH sampler, especially a complicated model.

3.2.3 Gibbs sampler

The Gibbs sampling is a special case of MH with careful selection of the candidate distributions. The methods to determine the transition distribution for state $t-1$ to t are various. To derive the marginal density of a parameter $\theta_j \in \boldsymbol{\theta}$, an intuitive way is to fix other parameters $\boldsymbol{\theta}_{-j}$ at a certain value and therefore take the conditional

posterior distribution given other parameters and data as the transition distribution. The Gibbs sampler is based on this idea. It is first proposed by Geman and Geman (1984) illustrating by image-processing models.

Algorithm 2 Gibbs sampler

1: Initialize $\boldsymbol{\theta}^{(0)} \sim q(\boldsymbol{\theta})$
2: **for** iteration $t = 1, 2, \ldots$ **do**
3: $\quad \theta_1^{(t)} \sim p\left(\boldsymbol{\theta}_1 = \theta_1 \mid \boldsymbol{\theta}_2 = \theta_2^{(t-1)}, \boldsymbol{\theta}_3 = \theta_3^{(t-1)}, \ldots, \boldsymbol{\theta}_p = \theta_p^{(t-1)}\right)$
4: $\quad \theta_2^{(t)} \sim p\left(\boldsymbol{\theta}_2 = \theta_2 \mid \boldsymbol{\theta}_1 = \theta_1^{(t)}, \boldsymbol{\theta}_3 = \theta_3^{(t-1)}, \ldots, \boldsymbol{\theta}_p = \theta_p^{(t-1)}\right)$
5: $\quad \vdots$
6: $\quad \theta_p^{(t)} \sim p\left(\boldsymbol{\theta}_p = \theta_p \mid \boldsymbol{\theta}_1 = \theta_1^{(t)}, \boldsymbol{\theta}_2 = \theta_2^{(t)}, \ldots, \boldsymbol{\theta}_p = \theta_{p-1}^{(t)}\right)$
7: **end for**

We write the Gibbs sampler algorithm in Algorithm 2. The generation of initialization is similar to the MH sampler. In this algorithm, each iteration is implemented by p steps. In the j-th step, θ_j is updated by a conditional posterior distribution on other parameters fixed in the current state. Then the $(j+1)$-th step is to update the parameter θ_{j+1} in the same way. The iteration is finished until all parameters are updated. In a word, each parameter θ_j is updated conditional on the latest values of the other components of $\boldsymbol{\theta}$, which are the iteration t values for the components already updated and the iteration $t-1$ values for the others. The advantage of this algorithm is that it simplifies the sampling problem of multivariate distribution into a sequence of simple univariate problems. This assumes that the full conditional distributions are easy to sample. Nevertheless, even for high-dimensional large problems, the full conditional distribution usually follows the common conjugate pairs conducive to sampling.

Unlike MH, Gibbs sampler avoids the rejection procedure. The Gibbs sampler requires one to derive the full conditional posterior distribution. Ideally, the conditional posterior density is considered to come from a parametric family. An important para-

metric family is a conjugate family, which allows the posterior distribution to follow the same parametric form as the prior distribution. If all parameters are assumed to come from a conjugate family, the derivation of a conditional posterior density is very straightforward. Therefore, even if the dimension of the parameter space goes higher, the update procedure can be effectively calculated using the closed-form. Consequently, the sample parameter chains are expected to converge quickly. The commonly used conjugate families include Gaussian, Gamma, Beta, and others for continuous parameters. Poisson and Dirichlet are also conjugate in terms of discrete parameters. In the simulation research in the next section, we will use Gaussian prior and Beta prior in our MCMC algorithm.

In many cases, it is not easy or even impossible to derive a full conditional posterior density. Therefore, the Gibbs sampler is not suitable for these situations. Like MH sampler, Gibbs sampler has no mechanism to avoid high autocorrelation between samples. If one wants to obtain more effective samples, it is not satisfactory.

3.2.4 Hamiltonian Monte Carlo

The MH sampler and Gibbs sampler can be regarded as samplers based on a random walk because the MH takes a random walk on the jump distribution. In contrast, the Gibbs sampler "walks" on the conditional posterior distribution. However, the acceptance rate of random walk behavior is low, resulting in a long time to generate more effective samples. To overcome this problem, HMC takes a series of steps notified by gradient information. Here we briefly introduce the principle of HMC based on Brooks et al. (2011) and Stan reference manual (Link6 (2020)).

The motivation of HMC is again to draw samples from the posterior distribution $p(\boldsymbol{\theta}|D)$. Notice that here we do not use sample parameters marginally but jointly. For simplicity, we note the posterior as $p(\boldsymbol{\theta})$ in this section. Then by introducing an auxiliary momentum vector ρ, we have the joint distribution of (ρ, θ_j) be $p(\rho, \boldsymbol{\theta}) =$

$p(\rho|\boldsymbol{\theta})p(\boldsymbol{\theta})$. Generally, independent of $\boldsymbol{\theta}$, the auxiliary density of ρ is assumed to be a multivariate Gaussian with mean 0 and covariance matrix Σ scaled to the Hessian of $\log p(\boldsymbol{\theta})$. In many cases, including Stan, Σ is set to be identity. The joint density $p(\rho, \boldsymbol{\theta})$ defines a Hamiltonian joint system:

$$H(\rho, \boldsymbol{\theta}) = -\log p(\rho, \boldsymbol{\theta}) = -\log p(\rho|\boldsymbol{\theta}) - \log p(\boldsymbol{\theta}) = T(\rho, \boldsymbol{\theta}) + V(\boldsymbol{\theta}),$$

where T is so-called "kinetic energy" and V is so-called "potential energy". By definition, our interest is the potential energy. The potential energy in the form of $-\log(p(\theta))$ is considered by HMC and also in Stan. The Hamiltonian dynamics systems of equations with respect to the time t are:

$$\frac{\partial \boldsymbol{\theta}}{\partial t} = \frac{\partial T}{\partial \rho};$$

$$\frac{\partial \rho}{\partial t} = \frac{-\partial V}{\partial \boldsymbol{\theta}}.$$

Therefore, it is transformed to solving differential equations. The solving algorithm is approximated by a "leapfrog" algorithm. Finally, the update status is again rejected or accepted by the Metropolis rejection or acceptance process. We summarize the steps of the algorithm into Algorithm 3.

Brooks et al. (2011) proved that theoretically, the time cost in sampling an independent/ effective sample with p-dimension of $\boldsymbol{\theta}$ is roughly $O(p^{5/4})$, whereas the cost of MH is $O(p^2)$. However, HMC was not widely popularized until the advent of No-U-Turn-Sampler (NUTS) Hoffman and Gelman (2014). The traditional HMC is very sensitive to ε, the "step size" of leapfrog algorithm and L, the depth of leapfrog and one has to tune these two parameters to get better performance, which is very time-consuming. The NUTS does not need to tune these two parameters, which significantly improves the efficiency. The details of NUTS can be found in Hoffman and Gelman (2014).

Algorithm 3 Hamiltonian Monte Carlo

Given $\theta^0, \varepsilon, L, P, M$:
for $m = 1$ to M **do**
  Sample $\rho^0 \sim N(0, I)$
  Set $\theta^m \leftarrow \theta^{m-1}, \tilde{\theta} \leftarrow \theta^{m-1}, \tilde{\rho} \leftarrow \rho^0$
  for $t = 1$ to L **do**
    Set $\tilde{\theta}, \tilde{\rho} \leftarrow \text{Leapfrog}(\tilde{\theta}, \tilde{\rho}, \varepsilon)$.
  end for
  with probability $\alpha = \min\{1, \frac{\exp\{P(\tilde{\theta}) - \frac{1}{2}\tilde{\rho}\cdot\tilde{\rho}\}}{\exp\{P(\theta^{m-1}) - \frac{1}{2}\rho^0\cdot\rho^0\}}\}$, set $\theta^m \leftarrow \tilde{\theta}, \rho^m \leftarrow -\tilde{\rho}$.
end for Leapfrog θ,ρ,ε
Set $\tilde{\rho} \leftarrow \rho + (\varepsilon/2)\nabla_\theta P(\theta)$.
Set $\tilde{\theta} \leftarrow \theta + \varepsilon\tilde{\rho}$
$\tilde{\rho} \leftarrow \tilde{\rho} + (\varepsilon/2)\nabla_\theta P(\tilde{\theta})$
return $\tilde{\theta}, \tilde{\rho}$.

As an algorithm using gradient information, HMC requires all parameters to be continuous. This can be problematic when some parameters are discrete. Possible solutions include smoothing discrete parameters and setting them as tuning parameters. Another problem that HMC may encounter is the nature of a posterior density. Suppose some parameters have "bad" partial gradients that are not easy to sample from leapfrog. For example, in the conditional function that arises discontinuity or not well-defined gradients, the leapfrog may either fail to run or be forced to take very long depth by these parameters. The algorithm will be possibly inefficient and unstable. And since the HMC is computed in a somewhat "black box" procedure, one cannot know how to modify the algorithm exactly.

3.2.5 Summary

Since the MCMC techinques in Stan and NIMBLE are based on different MCMC samplers, they have to share the pleasant and unpleasant points of MCMC samplers. *The use of the HMC and its modification, the NUTS, enables Stan to be computationally efficient with many effective samples. But it also constraints that all parameters have to be continuous with well-defined posteriors.* NIMBLE is flexible to both Gibbs

and MH samplers. But to ease the deriving of full conditional posterior, *NIMBLE can only automatically assign Gibbs sampler to some models with a known close form of conditional posterior, such as the stick-breaking process. Other parameters are automatically assigned the MH sampler with the jump distribution tuned by NIMBLE itself.* That makes NIMBLE flexible to various models but not demonstrating the acceptance rate and the number of effective samples. Details can be found on page 72, version 0.11.1, *NIMBLE user manual* (de Valpine et al. (2021)). The next subsection will discuss the implementation of trimmed mean regression model in Stan and NIMBLE using a simulation study. The character of the MCMC samplers used by them is well illustrated through posterior inspection.

3.3 Stan vs. NIMBLE: Bayesian trimmed mean regression

This subsection compares Stan with NIMBLE by estimating the regression coefficients in the trimmed mean regression model (3.1). We also include the comparison with an **R** software program. A simulation study is designed, and widely used assessments of the posterior will be computed as the comparison standard. We demonstrate this under different posterior scenarios through the following two aspects: *Estimation results*, *MCMC diagnosis and efficiency.* In contemporary statistics, visualization plays an essential role in Bayesian analysis, especially when accessing posterior inference. For example, the trace plot of posterior sample chains is a good choice to evaluate the acceptance rate in which the denser sticks of samples are obtained. The higher acceptance rate is computed. In evaluating the auto-correlation of samples, the number of lags of auto-correlation function (ACF) plot makes it quite intuitive.

3.3.1 Simulation settings and evaluation

This subsection will conduct a simulation study implementing model (3.1) in Stan, NIMBLE, and R for comparison. In our simulation setting, we set the sample size $n = 50$, indicating a small sample size. The dimension of covariates is set to be $p = 2$ and $X_i = (x_{i1}, x_{i2})^\top$, where $x_{ij} \sim U(-2, 2)$ for $j = 1, 2$, which yields non Gaussian variables. The distribution for error term is set as $\varepsilon \sim t(5)$, leading to a heavy tail. We here simply set the regression $\beta = (1, 1)^\top$ and the heteroscedasticity coefficients $\gamma = (0.5, 0.5)^\top$. We generate 200 independent data sets and for each data set, we run MCMC in Stan, NIMBLE, and **R**. On each data set, we run 4 MCMC chains, and each of the chains takes $5,000$ iterations with the first $2,500$ times burn-in, aggregating to a total of $20,000$ iterations with the first $10,000$ times burn-in. This is sufficient in terms of convergence. To compare the MCMC performance of different tools, we set the same MCMC scenario in each tool. In Stan's software, we set "chains = 4, iter = 5000, warmup = 2500, thin = 1" to achieve the purpose of parallel calculation. In each calculation, MCMC simulation calculation generates 4 chains. Each chain generates 5000 samples and saves the 2501-5000-th samples. NIMBLE package and **R** software rely on `foreach` package to conduct parallel computing. we set "niter = 5000, nburnin = 2500, thin = 1".

In the simulation study, all three tools converge under the same MCMC scenario. We then focus on the result of estimation, MCMC diagnosis and MCMC efficiency. The result of estimation reflects how well an MCMC chain approximates the posterior distribution. The convergence diagnosis of MCMC is used to decide whether the simulated posterior is reliable or not. The MCMC efficiency measures the efficiency of the MCMC process for generating posterior samples. It depends on how well the MCMC chains are mixed and how fast they compute.

Estimation results

In terms of the estimation results, the frequency type assessments , bias and square

root of mean square error (RMSE) are always considered. A bias of unknown parameter θ is defined as $\text{bias}(\theta) = T^{-1}\sum_{t=1}^{T}(\hat{\theta}_t - \theta_0)$. where θ_0 is the true value of an unknown parameter, T is the total number of simulations and $\hat{\theta}_t$ is the t-th replication of θ. The root of the estimator's mean squared error (RMSE) is the average of the squares of the measurement errors, $\text{RMSE}^2(\theta) = T^{-1}\sum_{t=1}^{T}(\hat{\theta}_t - \theta_0)^2$. Generally speaking, the more efficient the lower RMSE. The *effective sample size (ESS)* is calculated by, $\text{ESS} = N/[1 + 2\sum_{t=1}^{T}\text{ACF}(t)]$, Where N is the number of total MCMC samples, T is a truncation number, and $\text{ACF}(t)$ is the chain's auto-correlation at lag t (Kass et al., 1998, p.99).

MCMC diagnosis and efficiency

Visualization always plays an important role in the MCMC diagnosis. The first is to take a view of the trace plot of the MCMC chains. A graph of sampled parameter values as a function of the step length in the chain is called a trace graph. Ideally, a converged MCMC chain should be horizontal with no trend, and the length of sticks of samples is expected to be distributed around a certain value. Thus convergence is often assessed by visual inspection of the trace plots (Reich and Ghosh, 2019). Generally, in an MCMC procedure, a user follows the chain until it has converged and discards all previous samples from its burn-in period. Thus we only plot the trace of the after burn-in iterations in the following pages. Another term people are concerned with is the *MCMC representativeness*, which evaluates whether the MCMC samples are representative to the posterior. If so, the different initial values of MCMC chains will not affect the target distribution. Therefore, to check for the representativeness, one can create multiple independent chains (say 4 in our case) and see whether they are well-mixed. The goodness of the mixture of the chains from visualization implies the goodness of the MCMC representativeness. In the

simulation studies, 4 independent chains are included in each simulation parallelly.

The posterior samples are not independent since they are generated from a Markov chain. But since the central limiting theorem requires independent samples, we would expect the posterior samples to be weakly dependent or weakly correlated. A common way to describe the correlation between samples is the ACF. Thus we will use ACF plots to evaluate the dependence between samples. A group of samples with lower ACF is considered "better" than that with higher ACF.

Guided by the same philosophy, we need to count the "effective number of independent simulation draws" in an MCMC chain. We call this number ESS, an important quantity in Bayesian analysis. In general, a larger ESS indicates better MCMC performance. As Stan development team suggested, the **R** package **coda** is not recommended especially when multiple chains are considered, so we turn to use the *ess_bulk* function provided by Stan to compute the "bulk" ESS. The bulk ESS estimates ESS of the "bulk" (the body of the density except for the tail) of posterior samples after rank normalization.

When we evaluate the MCMC efficiency, the ESS generated per second can be used as an assessment. The higher MCMC efficiency implies the higher ability of a computing tool to simulate an effective sample. Since an MCMC chain contains the burn-in period to be discarded, we have to omit the ESS by burn-in period and the time consumed in the burn-in period. One can easily compute the MCMC efficiency in Stan and **R** environment. But, in NIMBLE, particularly in a replicate Monte Carlo study, the MCMC efficiency might be a little bit underestimated. The reason is that in NIMBLE, when a new data set is imported, the computer needs to recompile the NIMBLE code into a C++ file, which takes a long time. We cannot ignore the compiling time when computing MCMC efficiency, but the compile-time also covers the burn-in time; thus, the MCMC efficiency for NIMBLE will be slightly underestimated. To improve Bayesian computation, one can run several independent

chains parallely in Stan, NIMBLE and R. Stan itself can easily activate MCMC parallel computing. Still, NIMBLE needs help from other parallel packages in **R**. We use the *parallel* and `foreach` package in the **R** to accomplish the parallel MCMC sampling in NIMBLE guided by de Valpine et al. (2021).

3.3.2 The weakly informative kernel

In the trimmed mean regression model, in order to characterize the heteroscedasticity, we introduce a Dirichlet process mixture(DPM) model with a mixture kernel of four components of Gaussian densities as the nonparametric prior. We here conclude two key points of the prior: one is the kernel that is made up by a mixture 4 Gaussian densities and the other is the 4 densities and their weights should satisfy the constraints 3.1 to 3.4. One can first generate either the weight α or location and scale parameters $(\mu_1, \ldots, \mu_4, \sigma_1, \ldots, \sigma_4)^\top$ for Gaussian densities and use the generated one to specify the other. But this procedure may suffers difficult y that their supports are not isometric to each other and one should be cautious to the sampling scheme due to this problem, which will be discussed in the next subsection.

Dirichlet prior for α

The aforementioned difficulty can be eased when the distribution of the random error is believed to be symmetric. By simple algebra, the median regression is the same as the mean regression for symmetric density. In this case, the equation $\tau_2 = 1 - \tau_1$ always holds, such as the trimmed mean between τ_1 and τ_2 will always be the exact expectation. In Section 3 we have shown that the trimmed mean zero constraint is guaranteed by a mixture of 4 Gaussian densities with constraint 3.1 to constraint 3.4. However, since in this case the selection of τ_1 is arbitrary, both constraint 3.1 and constraint 3.2 can be eliminated. To fulfill constraint 3.3, an intuitive way is to draw α randomly from a Dirichlet distribution. This prior of α yields a kind of

weakly informed kernel for the nonparametric prior, which limits the posterior kernel with a possibly quadra-modal shape but unconstrained trimmed mean. In other words, using this weakly informative prior, the problem is transformed to the point estimation of the trimmed mean of the random error, where the choice of the priors for the other parameters are unconstrained. Another justification to this weakly informed prior is pointed by (Gelman et al., 2013, page 55), "in general any problem has some natural constraints that would allow a weakly informative model", and in our simulation study this prior is proved to be computationally efficient with satisfactory estimation accuracy. We demonstrate the simulation by Stan (Listing 3.1) and NIMBLE (Listing 3.2).

Listing 3.1: Stan program with weakly informative prior for simulation

```
data {
  int<lower=1> n; //sample size
  int<lower=1> p; //dim of beta
  int<lower=1> q; //dim of eta
  int<lower=1> L;//length of the truncated  Dirichlet process
  real<lower=0> alpha; // mass para
  vector[n] Y; // response
  matrix[n,p] X; // covariates
  vector[p] beta_init0;
  vector[L-1] w_init0;
}
parameters {
  vector[p] beta;
  vector[q] eta;
  matrix[L, 4] mu;
  matrix<lower = 0>[L, 4] sigma;
  vector<lower=0, upper=1>[L-1] w;
  simplex[4] v[L]; // prior of weight alpha
 }
transformed parameters {
  simplex[L] DP_weights;
  DP_weights[1] = w[1];
  for (s in 2:(L-1))  {
  DP_weights[s] = w[s] * prod(1 - w[1:(s - 1)]);
  }
  DP_weights[L] = 1- sum(DP_weights[1:(L-1)]);
}
model {
  beta ~ normal(beta_init0, 100); // non informative
  eta ~ normal(0, 1);
  for(l in 1:L){
```

```
32     mu[l, 1:4] ~ normal(0, 1);
33     sigma[l, 1:4] ~ inv_gamma(1, 1);
34     v[l] ~ dirichlet(rep_vector(1, 4)); // dirichlet prior
35   }
36   for (j in 1:n){
37    real eps;
38    vector[L] lp_ik;
39    eps = (Y[j] - X[j, 1:p]*beta) * exp(-X[j, 1:p]*eta);
40      for(l in 1:L){
41        vector[4] lp_piece;
42        lp_ik[l] = log(DP_weights[l]) ;
43          for(cat in 1:4){
44          lp_piece[cat] = normal_lpdf(eps|mu[l, cat], sigma[l, cat])+
     log(v[l,cat]);
45          }
46        lp_ik[l] += log_sum_exp(lp_piece);
47      }
48    target += log_sum_exp(lp_ik);
49   }
50 }
```

Listing 3.2: NIMBLE program with weakly informative prior for simulation

```
1  TMRcode <- nimbleCode({
2    for (i in 1:N) {
3      y[i] ~ dnorm(mu_y[i], sd = sigma_y[i])
4      exp_tem[i] <- exp( gamma[1] * x1[i] + gamma[2] * x2[i])
5      mu_y[i] <-  beta[1] * x1[i] + beta[2] * x2[i] + exp_tem[i] * mu[
     h[i], g[i]]
6      sigma_y[i] <- exp_tem[i] * sigma[h[i], g[i]]
7      g[i] ~ dcat(prob[1:M])
8      h[i] ~ dcat(alpha[1:4, g[i]])
9    }
10   for (j in 1:p) {
11     beta[j] ~ dnorm(0, sd = 100)
12     gamma[j] ~ dunif(0, 1)
13   }
14   prob[1:M] <- stick_breaking(v[1:(M-1)])
15   for (j in 1:(M-1)) {
16     v[j] ~ dbeta(1, v_alpha)
17   }
18   v_alpha ~ dgamma(1,1)
19   ## truncated normal prior for theta1 and theta2, the inverse of
      the cdf
20   theta1 ~ T(dnorm(0,0.01),,0)
21   theta2 ~ T(dnorm(0,0.01),0,)
22   for (j in 1:M) {
23     alpha[1:4, j] ~ ddirch(aa[1:4]) ## Dir prior
24   }
25   for (i in 1:4) {
26     for (j in 1:M) {
27       mu[i, j] ~ ddexp(0, 1)
```

```
28        sigma[i, j] ~ dunif(min_sig, max_sig)
29      }
30    }
31 })
```

Estimation results

In the trimmed mean regression model, we simply implement the Dirichlet prior for α in Stan and NIMBLE, without constraints 3.1 to 3.4. Thus, the other parameters $(\mu_1, \ldots, \mu_4, \sigma_1, \ldots, \sigma_4)^\top$ are independently given an non-informative prior. The estimation results are given in Table 3.2.

Table 3.2: The estimation results using the weakly informative kernel

Parameters	Stan					NIMBLE				
	BIAS	RMSE	SSD	ESD	ESS	BIAS	RMSE	SSD	ESD	ESS
β_1	0.006	0.11	0.11	0.101	9362	-0.013	0.106	0.106	0.103	308
β_2	-0.008	0.1	0.1	0.101	9444	0.015	0.099	0.098	0.1	318
γ_1	-0.037	0.227	0.224	0.144	10679	-0.014	0.124	0.124	0.134	167
γ_2	-0.015	0.214	0.214	0.143	10828	-0.001	0.112	0.112	0.135	164

BIAS, the average bias; RMSE, square root of mean square error; SSD, sample standard deviation; ESD, the average estimated standard error; ESS, effective sample size

It can be seen from Table 3.2 that the results by both tools closed to each other and the bias is acceptable, which means that the Dirichlet prior for α is suitable. We use the Dirichlet prior for α as the weakly informative prior because α is a weight and the Dirichlet distribution is a natural choice. In terms of RMSE, NIMBLE has lower RMSE in all parameters than Stan, which indicates that the parametric estimation of NIMBLE is more efficient. NIMBLE has lower RMSE because NIMBLE has lower SSD, and thus it seems that the point estimator given by NIMBLE is more robust.

MCMC diagnosis and efficiency

The first indicator for evaluating MCMC samples is whether the chains are well integrated. Figure 3.2 displays the trace plots of the samples generated by Stan and

NIMBLE. All the trace plots show that all four chains mix quite well in both Stan and NIMBLE. It illustrates that after enough length of burn-in period, the posterior is representative. In addition, the trajectory is horizontal and has no trend, so we conclude that all the chains have converged. It is worth mentioning that the trace plots of samples by NIMBLE are significantly more "sparse" than Stan. In other words, the count of successful transition times is less than that in Stan, which implies a lower acceptance rate. A possible reason may be that the gradient of the posterior is easy-computing, which makes samples generated by NUTS easier to converge.

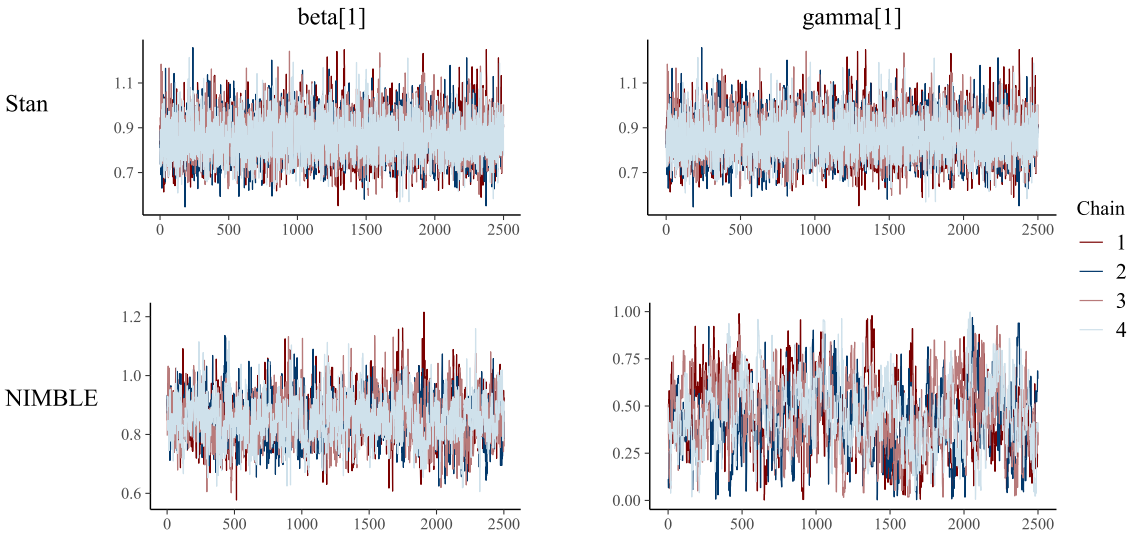
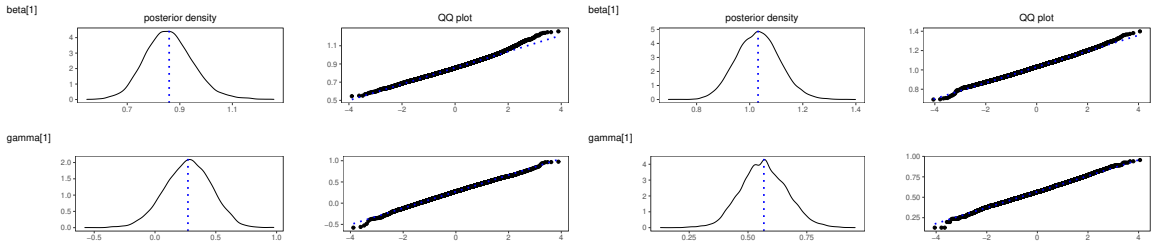


Figure 3.2: The MCMC trace plots of samples for parameters using weakly informative kernel simulated by Stan (upper panel) and NIMBLE (down panel)

The large sample theory (Gelman et al., 2013, page 87) illustrates that as the sample size n goes to infinity, the posterior density is asymptotically normal. In our simulation setup, we have four parameters. We present the posterior density and Q-Q plots of β_1 and γ_1 in Figures 3.3(a) and 3.3(b). When the chains converge, the improved Gelman-Rubin (GR) statistic is close to 1 (Gelman and Rubin (1992) Brooks and Gelman (1998), Vehtari et al. (2021)). The trace plots show that the MCMC chains overlap very well, which corresponds to the improved Gelman-Rubin statistic very close to 1. We further give the dynamic plot of improved GR statistics

in Figure 3.4, where one can find that with the progress of iteration, the improved GR statistic of all chains becomes close to 1 in both Stan and NIMBLE, but Stan converges even faster than NIMBLE.



(a) Stan (b) NIMBLE

Figure 3.3: The density plot and Q-Q plots for samples of parameters using weakly informative kernel simulated by Stan and NIMBLE

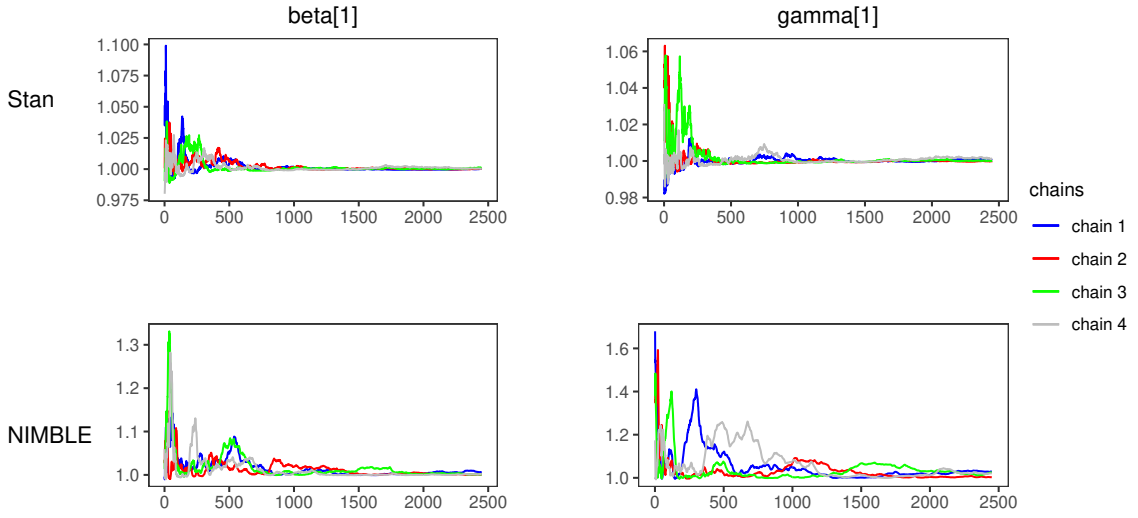


Figure 3.4: The dynamic improved Gelman-Rubin plot of samples by weakly informative kernel simulated by Stan and NIMBLE

Samples generated by MCMC are impossible to be independent since they are in a Markov chain. People are more concerned about the weak-dependence samples in the chains, which can be viewed by the plot of the auto-correlation function (ACF).

It can be seen from Figure 3.5 that the autocorrelation of samples of both β and γ generated by Stan is almost equal to 0. For comparison, the samples of β generated

by NIMBLE share significant autocorrelations until the time lag exceeds 20, and the autocorrelation of samples of γ in NIMBLE is significant within 40 time lags. It indicates that the ESS of Stan is larger than that of NIMBLE, and each successive step simulated by NIMBLE is partially redundant with the previous step.

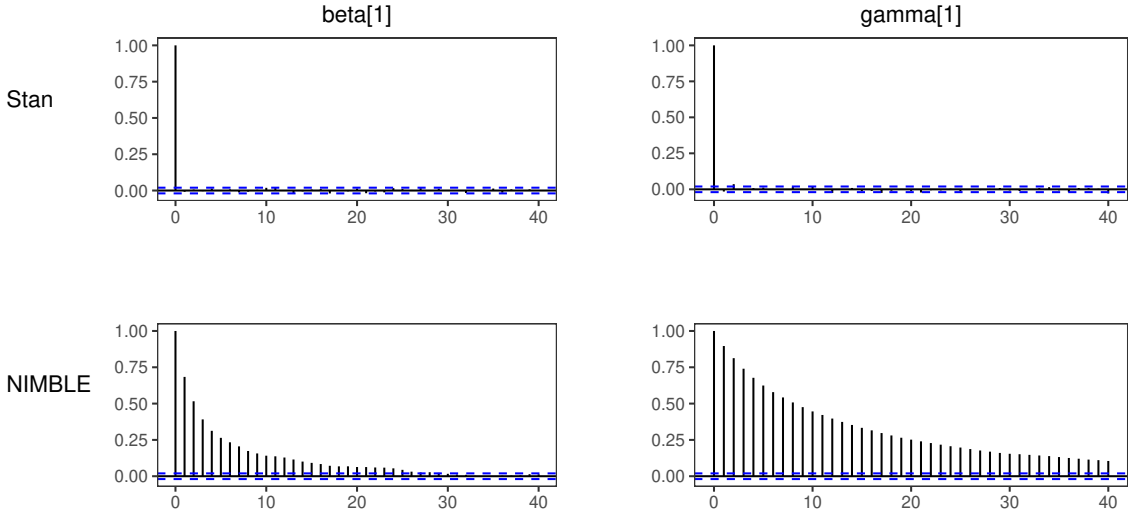
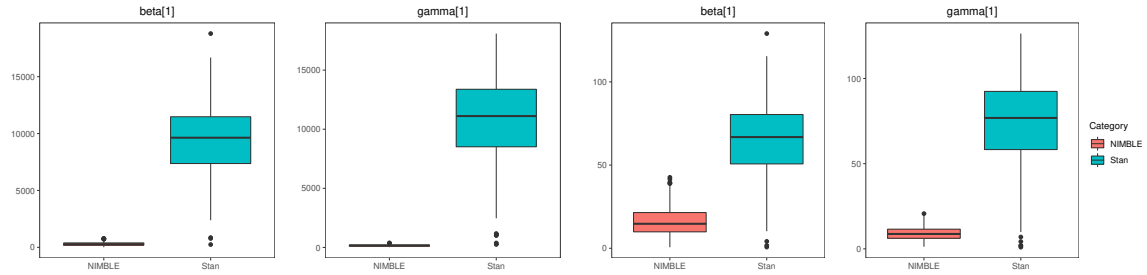


Figure 3.5: The ACF plot of samples by weakly informative kernel simulated by Stan and NIMBLE

When using the weakly informative kernel, the ESS provided by Stan is much higher than that provided by NIMBLE for all parameters (Figure 3.6(a)). The dominating reason is that the NUTS used in Stan is much more possible to generate effective samples than the MH sampler used in NIMBLE, just as mentioned by Hoffman and Gelman (2014). The interesting question is that why in some cases the ESS computed by Stan is larger than 10000, the total number of MCMC iterations. That's because Stan uses an antithetic Markov chain that has negative odd lag auto-correlations (Vehtari et al., 2021). It is clear when we get negative auto-correlations on odd lags in the chain, and the effective sample size can also be larger than the total sample size. Some parameters often have strong auto-correlation in practical applications, so a long chain is needed to achieve sufficient ESS. This is not necessary when using Stan to generate data. This shows that Stan's computational

efficiency is much higher when using a weak information kernel than NIMBLE. (Figure 3.6(b)). However, benefit of high efficiency is limited when one is to estimate some relevant quantities, like the variance. As shown in table, the ESD of γ_1 and γ_2 is underestimated because of the "super-efficiency" of ESS (Vehtari et al., 2021).



(a) ESS (b) MCMC Efficiency

Figure 3.6: Box-plots of ESS and the MCMC efficiency of parameters weakly informative kernel. In each plot, the left box is computed by NIMBLE; and the right box is computed by Stan.

3.3.3 The informative prior

In the above subsection we reported and discussed the result estimated by using a weakly informative nonparametric kernel without constraints 3.1 to 3.4 but to generate α randomly from a Dirichlet distribution. In this subsection we discuss the method using the informative kernel that includes constraints 3.1 to 3.4. We call this kernel "informative" since we have proved that quantile regression is a sub-problem of trimmed mean regression. Thus, the density of random error is surely trimmed mean zero.

Resampling scheme to include constraints 3.1 to 3.4

When defying a weakly informative kernel, the support of the prior of α is the simplex in R^4. In order to include 3.1 to 3.4, one possible way is to determine the parameters vector $(\mu_1, \ldots, \mu_4, \sigma_1, \ldots, \sigma_4)^\top$ by specified α. However, it is known

difficult to determine the 8-dim parameter vector by a specified α since to find the dual space of the 4-dim transformation in an 8-dim linear space is mathematically challenging. Consequently, one may consider to determine α by specified $(\mu_1, \ldots, \mu_4, \sigma_1, \ldots, \sigma_4)^\top$, which indicates the α should be constrained into a subspace of the simplex in R^4 such as the α should be fully determined by the randomly generated parameters $(\mu_1, \ldots, \mu_4, \sigma_1, \ldots, \sigma_4)^\top$ so as to construct a trimmed mean zero kernel. However, this may lead to a contradict that the determined α may not be a legal weight. In other words, with randomly generated vector $(\mu_1, \ldots, \mu_4, \sigma_1, \ldots, \sigma_4)^\top$, the support prior of α is a subspace of R^4, which can not fully cover the support of target distribution of α, the simplex of R^4. For example, when $(\mu_1, \ldots, \mu_4) = (-0.596, 0.785, 1.478, 0.410)^\top$, and $(\sigma_1, \ldots, \sigma_4) = (0.465, 1.484, 0.591, 1.987)^\top$, the determined $\alpha = (152.191, -604.358, 355.420, 97.746)$, which is obviously not a legal weight. This problem calls for resampling when running MCMC procedure. That is, once we determine an α from the sampled $(\mu_1, \ldots, \mu_4, \sigma_1, \ldots, \sigma_4)^\top$, we have to check whether this α is a legal weight and if not, we should go to another sample of $(\mu_1, \ldots, \mu_4, \sigma_1, \ldots, \sigma_4)^\top$ until the α is legal.

Stan does not work!!

This resampling procedure brings uncertainty to the posterior density since we insert an if-else step in the routine sampling procedure. We here point out that this makes an unsolvable problem in Stan or any other HMC based computing tools. Recall that the transition in sampling procedure using HMC is transformed to solve the Hamiltonian dynamics and thus no random number generation is needed when sampling. Particularly, Stan's architecture forbids using the random number generator in sampling procedure such as the model block in Stan program, in order to avoid any possible randomness of posterior, which means resampling is impossible in Stan. One may consider defining an if-else like function to define the kernel but the if-else

condition always brings discontinuity to the posterior and thus brings illness to the gradient of the posterior. We have tried defying such a function in Stan but the algorithm can not converge at all. Thus, we conclude that Stan is unsuitable for realizing the method using an informative kernel and we don't compare the result given by Stan here. We demonstrate the simulation by NIMBLE (Listing 3.3) and **R** (Listing 3.4). We omit the routine part of **R** in the code below.

Listing 3.3: NIMBLE program with informative prior for simulation

```
# function to determine alpha
make.alpha <- function(mu,sig,tau1,tau2,theta1,theta2){
  b <- c(tau1,tau2,1,0)
  C <- pnorm((theta1-mu)/sig,0,1)
  C <- rbind(C,pnorm((theta2-mu)/sig,0,1))
  C <- rbind(C,rep(1,4))
  C <- rbind(C,(dnorm(theta1,mu,sig)-dnorm(theta2,mu,sig))*(sig)^2
  +mu*(pnorm(theta2,mu,sig)-pnorm(theta1,mu,sig)))
  alpha <- ginv(C)%*%b
  c(alpha)
}
# call function from R into NIMBLE
alpha_r <- nimbleRcall(function(mu = double(1), sig = double(1),
    tau1 = double(0),tau2 = double(0), theta1 = double(0), theta2 =
    double(0)){}, Rfun = "make.alpha",returnType = double(1))
TMRcode <- nimbleCode({
  for (i in 1:N) {
    y[i] ~ dnorm(mu_y[i], sd = sigma_y[i])
    exp_tem[i] <- exp( gamma[1] * x1[i] + gamma[2] * x2[i])
    mu_y[i] <-  beta[1] * x1[i] + beta[2] * x2[i] + exp_tem[i] * mu[
    h[i], g[i]]
    sigma_y[i] <- exp_tem[i] * sigma[h[i], g[i]]
    g[i] ~ dcat(prob[1:M])
    h[i] ~ dcat(alpha[1:4, g[i]])
  }
  for (j in 1:p) {
    beta[j] ~ dnorm(0, sd = c2)
    gamma[j] ~ dnorm(0, 1)
  }
  prob[1:M] <- stick_breaking(v[1:(M-1)])
  for (j in 1:(M-1)) {
    v[j] ~ dbeta(1, v_alpha)
  }
  v_alpha ~ dgamma(1,1)
  theta1 ~ T(dnorm(0,0.01),,0)
  theta2 ~ T(dnorm(0,0.01),0,)
  for (j in 1:M) {
    alpha[1:4, j] <- alpha_r(mu[1:4,j], sigma[1:4,j], tau1, tau2,
    theta1, theta2)
```

```
    constraint_data[j] ~ dconstraint(alpha[1,j] > 0 &
                                     alpha[2,j] > 0 &
                                     alpha[3,j] > 0 &
                                     alpha[4,j] > 0)
    ## dconstraint for prior with constraints
  }
  for (i in 1:4) {
    for (j in 1:M) {
      mu[i, j] ~ ddexp(0, 1)
      sigma[i, j] ~ dunif(min_sig, max_sig) }}
})
```

Listing 3.4: R program with informative prior for simulation

```
# This is the R code for MCMC procedure of trimmed mean regression
# beta and stick breaking process are updated by Gibbs sampler and
    others are updated by MH. Here we simply list the key part of the
     code since others are routine.
# solve the equations for informative constraints
  make.alpha <- function(mu,sig,tau1,tau2,theta1,theta2){
  b <- c(tau1,tau2,1,0)
   C <- pnorm((theta1-mu)/sig,0,1)
   C <- rbind(C,pnorm((theta2-mu)/sig,0,1))
   C <- rbind(C,rep(1,4))
   C <- rbind(C,(dnorm(theta1,mu,sig)-dnorm(theta2,mu,sig))*(sig)^2+
   mu*(pnorm(theta2,mu,sig)-pnorm(theta1,mu,sig)))
   alpha <- ginv(C)%*%b
   alpha }
# compute the stick-breaking weights
  makeprobs <- function(v)
  {N <- length(v)
   probs <- v
   probs[2:N] <- probs[2:N]*cumprod(1-v[2:N-1])
   probs }
# resampling scheme to determine alpha
for(k in 1:M) ### M: the number of truncated dirichlet process{
   while(alpha[1,k]<0|alpha[1,k]>1|alpha[2,k]<0|alpha[2,k]>1|alpha
   [3,k]<0|alpha[3,k]>1|alpha[4,k]<0|alpha[4,k]>1)
   { mu[1,k] <- rnorm(1,2*theta1,1)
     mu[2:3,k] <- rnorm(2,0,1)
     mu[4,k] <- rnorm(1,2*theta2,1)
    sig[,k] <- runif(4,mn.sig,mx.sig)
    alpha[,k] <- make.alpha(mu[,k],sig[,k],tau1,tau2,theta1,theta2)
    ind[k] <- ind[k]+1}
  }
  v <- rbeta(M,1,D)
  v[M] <- 1
  probs <- makeprobs(v)
  g <- sample(1:M,n,replace=T,prob=probs)
  h <- rep(0,M)
  for(i in 1:n) {h[i] <- sample(1:4,1,alpha[,g[i]],replace=F)}
```

Estimation results

The MCMC using informative kernel can easily be realized in **R** and NIMBLE. In **R**, we extend the **R** program provided by Reich et al. (2010) that uses the Gibbs sampler for the updating of *beta* and other parameters are updated by MH. In NIMBLE, we simply follow the automatically assigned sampler, such as the stick-breaking parameter q_l in DP is updated by conjugate distribution by the Gibbs sampler, and other parameters are updated by default the MH sampler. We again run both 10000 times of sampling iterations and 10000 warm-up iterations in both **R** and NIMBLE, and their estimation results are given in Table 3.3.

Table 3.3: The estimation results using the informative kernel

Parameters	NIMBLE					R				
	BIAS	RMSE	SSD	ESD	ESS	BIAS	RMSE	SSD	ESD	ESS
β_1	-0.012	0.11	0.11	0.107	260	0.007	0.109	0.109	0.123	110
β_2	0.015	0.099	0.099	0.105	265	-0.003	0.109	0.109	0.126	114
γ_1	-0.013	0.14	0.139	0.152	134	-0.016	0.155	0.155	0.179	37
γ_2	0.007	0.122	0.122	0.155	130	-0.018	0.17	0.169	0.179	35

BIAS, the average bias; RMSE, square root of mean square error; SSD, sample standard deviation; ESD, the average estimated standard error; ESS, effective sample size

From table we find the RMSE for β and γ given by NIMBLE using the informative kernel is lower than using the weakly informative kernel, which means that the trimmed mean information is helpful to improve parametric estimation efficiency. The results by **R** are similar to NIMBLE yielding that the two computing tools have no significant difference in the implementation of this model in this simulation. Again, we find NIMBLE provides lower RMSE for all parameters with lower SSD, which demonstrates the robustness of the estimation by NIMBLE.

MCMC diagnosis and efficiency

From the trace plot, both NIMBLE and **R** can demonstrate the convergence of MCMC chains (Figure 3.7). In contrast, the four chains generated by NIMBLE are better mixed than those generated by R.

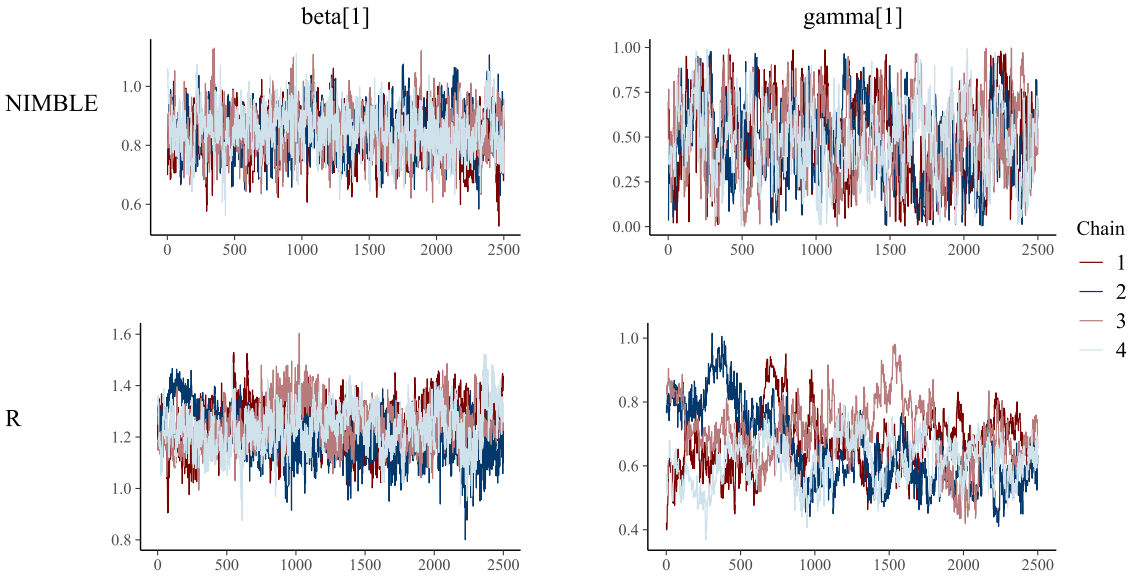
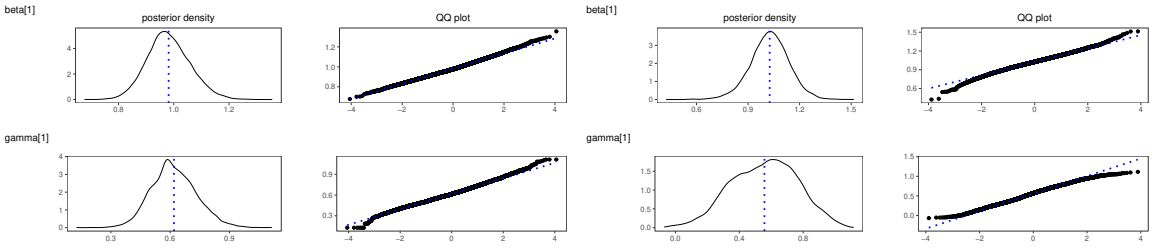


Figure 3.7: The MCMC trace plot of samples for parameters using informative kernel simulated by NIMBLE and R.

The density plot and Q-Q plot of parameters β_1 and γ_1 are shown in Figure 3.8. We find the density of posterior samples generated by NIMBLE is more likely to be normal than that generated by R.



(a) NIMBLE (b) R

Figure 3.8: The density plot and Q-Q plot for samples of parameters using informative kernel simulated by NIMBLE and R.

Figure 3.9 demonstrates that along with the chains, the improved GR statistic converges to 1 in NIMBLE. Nimble converges faster and better than **R**. The posterior samples in **R** share higher auto-correlation than NIMBLE (Figure 3.10). A possibly better choice is to thin the posterior samples by a larger number of lags, but this always requires longer chains of iterations and we do not thin the samples here. For instance, in terms of the thinning of chains, for instance, one can consider drawing a sample in every 20 samples in NIMBLE for the estimation of γ_1, which requires 20×10000 samples, a quite long chain. But in R, one may need an even longer chain as the order of ACF is much higher.

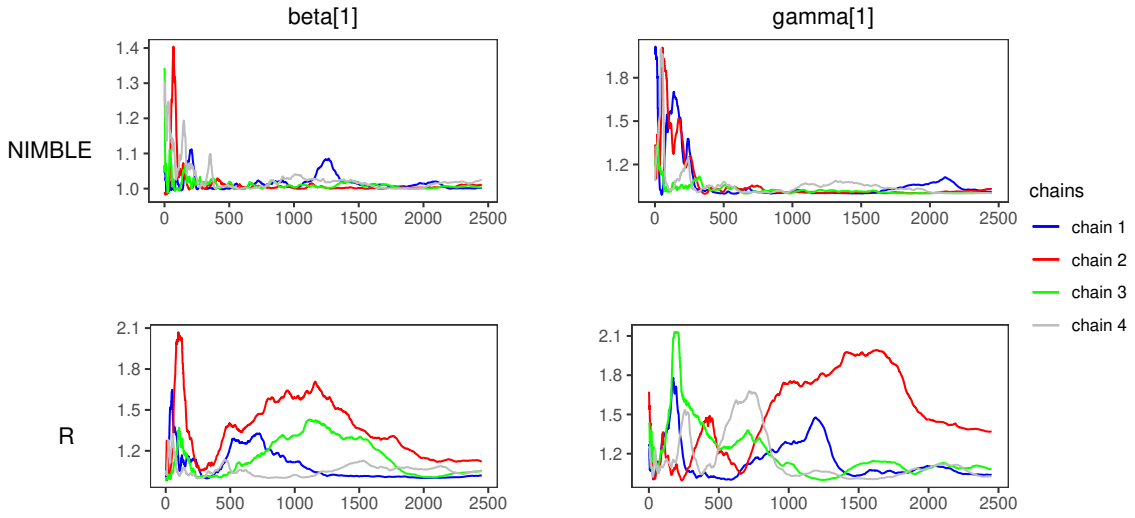


Figure 3.9: The dynamic improved Gelman-Rubin plot of samples by informative kernel simulated by NIMBLE and R

When we focus on β_1, we find that the **R** program provides slightly lower ESS than NIMBLE (Figure 3.11(a)). NIMBLE and **R** use Gibbs sampling algorithm when estimating β, and NIMBLE has a slightly better effect. When estimating γ, NIMBLE and **R** uses MH sampler, NIMBLE produced significantly higher ESS than R. This is may because NIMBLE could tune parameters automatically better than **R**. Recall the ACF plots, we also get the same conclusion. It can be found that the samples generated by R in estimating γ have high autocorrelation than NIMBLE.

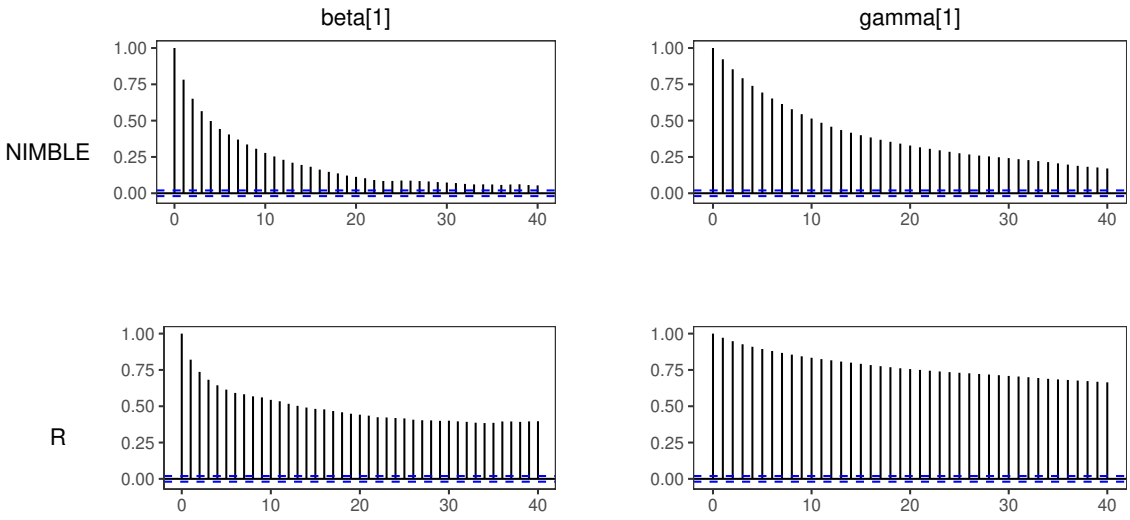
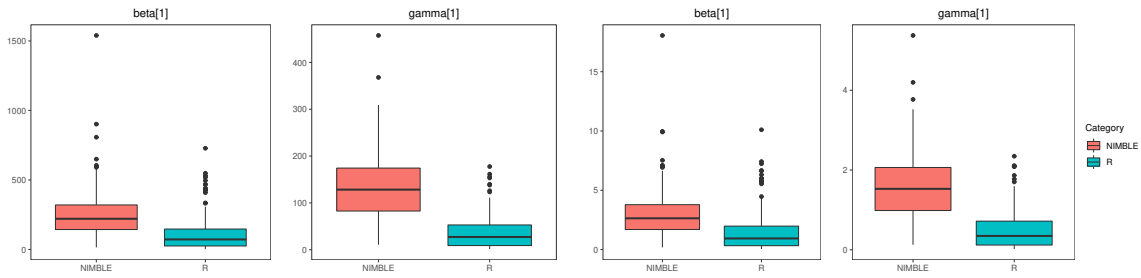


Figure 3.10: The ACF plot of samples by informative kernel simulated by NIMBLE and R



(a) ESS (b) MCMC effiency

Figure 3.11: Box-plot figures of ESS and the MCMC efficiency of parameters (β_1 and γ_1). In each sub figure, the left box is simulated by NIMBLE; and the right box is simulated by R.

The result of MCMC efficiency is given in Figure 3.11(b). Interestingly, the MCMC efficiency of estimation of β by NIMBLE is slightly lower than that of **R**, possibly implying the Gibbs sampler might be the better choice when sampling conjugate distributions. However, the MCMC efficiency of estimation of γ by NIMBLE is much better than that of R. We conjecture that the tuning parameter in the jump distribution of the MH sampler used in NIMBLE is better than that used in R, which is tuned by us.

3.3.4 Thinning v.s. unthinning

In this subsection we discuss the thinning method in MCMC techniques. In case one gets the posterior samples with high autocorrelation, one may consider thinning the sample by saving every k number of samples for a positive integer k. This is an effective way to improve the ESS and to deduct the autocorrelation. Nevertheless, this method does not help improve MCMC efficiency and will bring a higher computing burden.

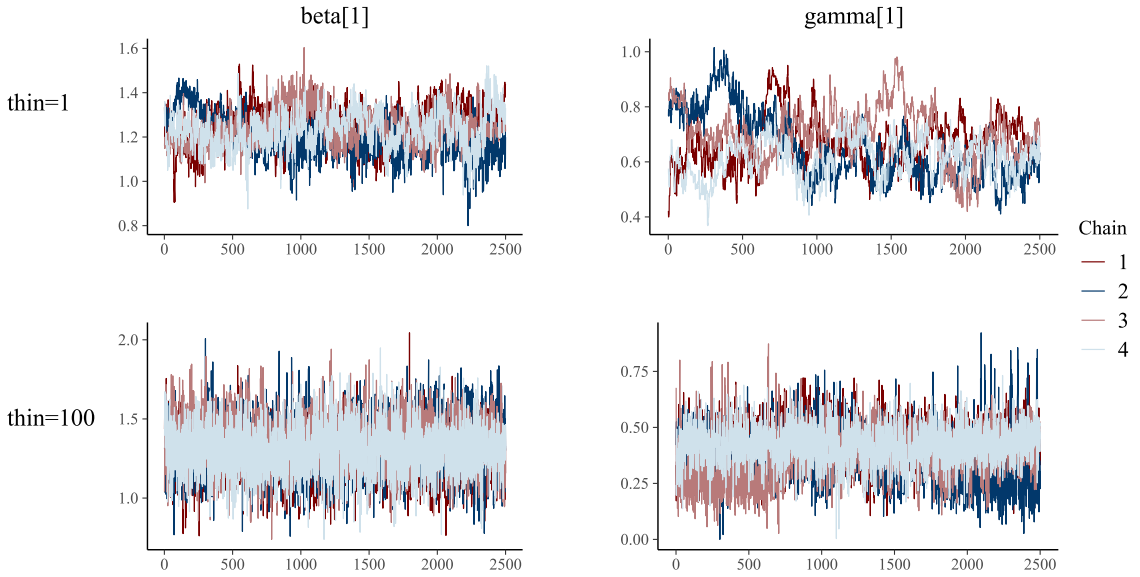
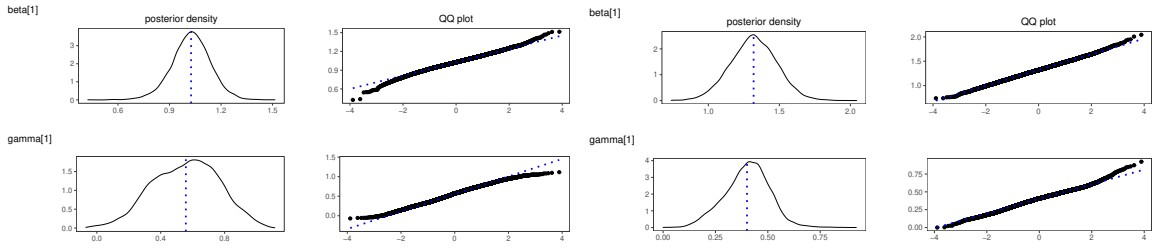


Figure 3.12: The MCMC trace plot of samples for parameters using informative kernel simulated by R (thin=1 and thin =100).

We demonstrate analysis of thinning procedure in **R**. The results of thinning in NIMBLE code are analog to **R** and skipped. Figure 3.12 shows the trace plot of chains after thinning. We find that the sticks of samples in the trace plot of MCMC chains after thinning are denser, which means more weakly-dependant MCMC jumps are taken. Naturally, more weakly-dependant MCMC jumps indicate that the autocorrelation between samples after thinning is lower. Figure 3.13 shows the density and Q-Q plot of posterior samples. The probability density after thin is closer to the normal distribution than unthin.



(a) thin=1 (b) thin=100

Figure 3.13: The posterior density and Q-Q plot of samples for parameters using informative kernel simulated by R (thin=1 and thin =100).

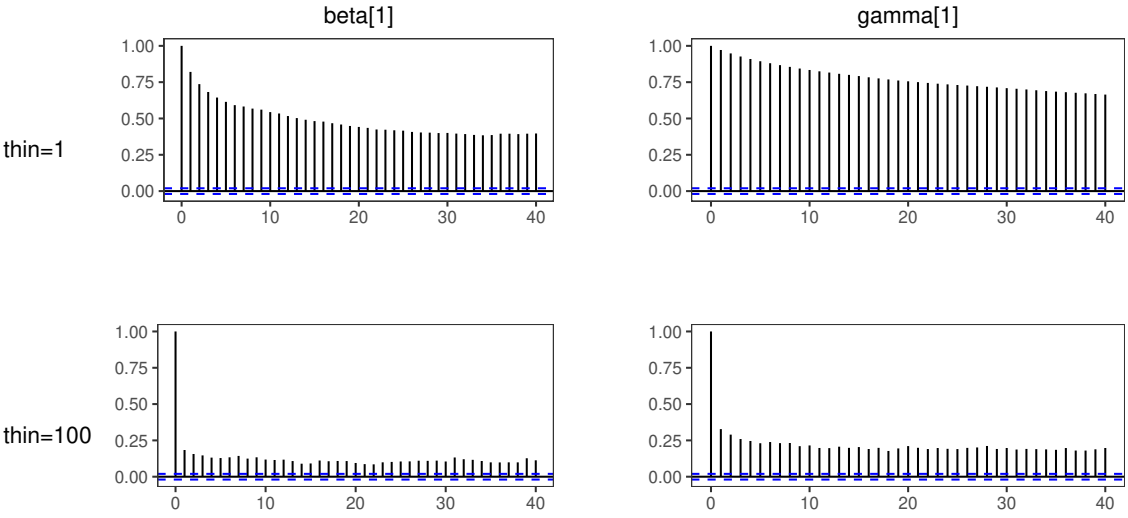


Figure 3.14: The ACF plot of samples for parameters using informative kernel simulated by R (thin=1 and thin =100).

Figure 3.14 shows the ACF plot of posterior samples. We can obviously observe that the auto-correlation between samples with thin = 100 is significantly lower than that with thin = 1. Through thinning the chain of MCMC, the auto-correlation between samples can be effectively reduced. Figure 3.15 shows the dynamic plot of the Gelman-Rubin statistic of different chains. The value of the improved Gelman-Rubin statistic with thin=100 is closer to 1 than that with thin = 1. In summary, one can conclude from these plots that the thinning improves the MCMC performance by giving a better mixture of chains, deducting auto-correlation and speeding up the convergence of chains.

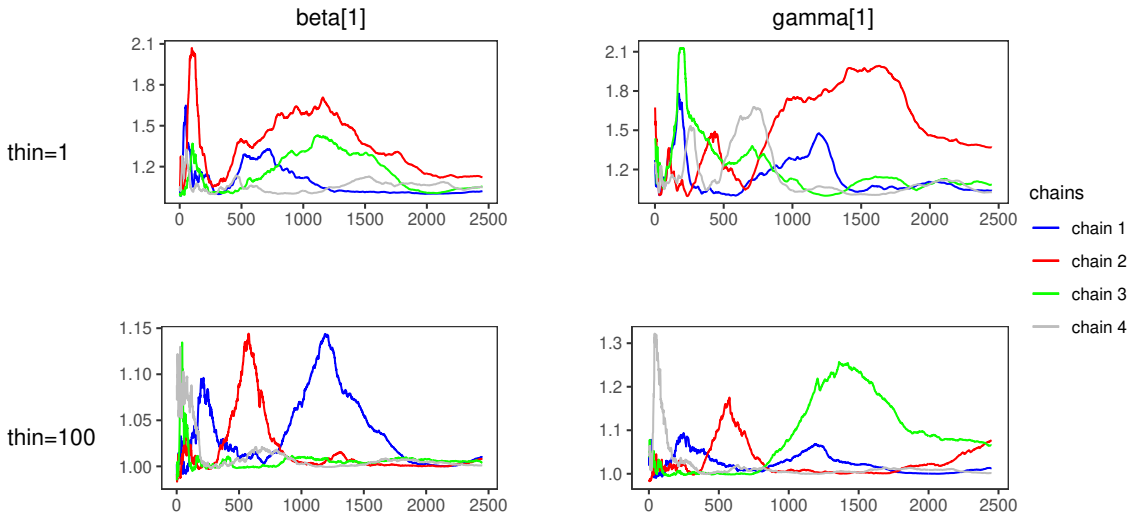


Figure 3.15: The dynamic improved Gelman-Rubin plot of samples by informative kernel simulated by R (thin=1 and thin =100)

3.3.5 Computational burden

Effective sample size (ESS) can measure the accuracy and stability of the approximation of posterior distribution. MCMC efficiency is ESS per second. If we use the computation time as the measurement of computational burden, the computation time is computed by dividing total ESS and MCMC efficiency. That is,

$$\text{Computation time} = \text{ESS} \ / \ \text{MCMC Efficiency}.$$

Another metric of interest is the average time needed to generate each effective sample, called MCMC Pace. Stan team emphasized the importance of MCMC Pace, and the NIMBLE team gives the definition formulae, as the inverse of computational efficiency,

$$\text{MCMC Pace} = \text{Computation time} \ / \ \text{ESS} = 1 \ / \ \text{MCMC Efficiency}.$$

The above two measures have different interpretations when averaging over multiple runs and/or multiple parameters. Here we take the estimation of β_1 as an example to compare the computational burden between Stan and NIMBLE, NIMBLE and R.

Table 3.4: The computation time and MCMC Pace computation based on β_1.

	Weakly informative			Informative		
	Stan	NIMBLE	Ratio 1	NIMBLE	R	Ratio 2
ESS	9362	425	22.03	1284	737	1.74
MCMC efficiency	63.86	6.80	9.39	2.67	2.69	0.99
Computation time	146.60	62.50	2.35	480.90	273.98	1.76
MCMC Pace	0.016	0.147	0.11	0.375	0.372	1.01

Ratio 1: Stan / NIMBLE; Ratio 2: NIMBLE / R

From Table 3.4, we observe when using the weakly informative prior, Stan spends more than twice the total computation time than NIMBLE, but a much less average time (nearly 10%) to generate each effectively independent sample. It seems that Stan has a heavier computational burden. Nevertheless, recall that for fairness, we set the length of the MCMC chains to be the same for all tools in subsection 3.3.1. This comparison implies that Stan may not need chains that are as long as NIMBLE to generate sufficient effective samples. In other words, one can run a shorter chain in Stan to achieve the same MCMC computational efficiency.

In terms of the informative prior, the MCMC efficiency and MCMC Pace for estimation of β_1 in NIMBLE is almost equivalent to in R. This means that it takes NIMBLE and **R** almost the same average time to generate each effectively independent sample. That's not surprising since both **R** and NIMBLE in this example call the resampling scheme, severely reducing the MCMC efficiency in both tools.

In addition, the average time to generate effectively independent samples of NIMBLE under weakly informative prior (MCMC Pace 0.147) is significantly lower than under informative prior (MCMC Pace 0.375). The main reason is the resampling scheme used for informative prior is too inefficient to generate effective samples.

3.3.6 Summary

In this section, we briefly review the use of two computing tools, Stan and NIMBLE. For comparison, a simulation study for the trimmed mean regression model is designed and we compare their results with a well-written **R** program. In terms of the model estimation, we here introduce two nonparametric priors, one is weakly informative and the other is informative. Under the weakly informative prior, both Stan and NIMBLE give similar estimation results. Due to the difference between HMC and MH, *Stan enjoys better MCMC performance than NIMBLE using the weakly informative kernel with much higher ESS and lower ACF.* With more effort paid for generating an effective sample, NIMBLE is able to provide a more efficient parametric estimator. *The informative kernel we defined here calls for a resampling scheme during the MCMC transition but resampling is illegal in Stan and therefore Stan can not be implemented when using an informative kernel.* That means when one is about to use Stan she/he should be cautious about the properties of the posterior and its gradient. Compared to the method using weakly informative prior, *the informative prior provides better estimation results with cost in computation efficiency.* The computing time of using the informative kernel is almost 10 times to the time using the weakly informative kernel in NIMBLE, and the computing time of **R** is similar to the time in NIMBLE. We point out that the *time consumption is mainly caused by the resampling procedure* since it is quite possible to sample a lot of many times to generate suitable $(\mu_1, \ldots, \mu_4, \sigma_1, \ldots, \sigma_4)^\top$ so as to determine the legal α. NIMBLE and **R** program give similar estimation results but the Gibbs sampler called by **R** is a little more efficient to parameters with the conjugate distribution. *Therefore if a conjugate prior is given to a parameter and one can derive its full conditional posterior, Gibbs sampler may be more preferable. The tuning parameter used in MH is well-tuned in NIMBLE and in general better than that tuned by the*

user artificially. Hence when one decides to use the MH sampler (in most cases MH works), especially when the HMC fails, NIMBLE is an ideal choice.

In addition, the performances of parallel computing in Stan, NIMBLE, and **R** are quite different. In the simulation studies, all independent chains are sampled parallelly. The parallel computing is an inherent function of Stan. Thus in Stan the users can run multiple parallel chains without extra setting and effort. However, NIMBLE and **R** rely on packages such as `foreach` in **R** to conduct parallel computing. For NIMBLE, notice that, *it must contain lines of codes for memory protection.* When using the `foreach` function in `foreach` package to engine the parallel computing in a series of repeated Monte Carlo simulations in NIMBLE, we have to repeat the procedure of stopping-registering the cluster of CPUs at each repeated simulation; otherwise the heavy computational burden might slow down the implementation to cease the computing. We conjecture this might be owing to running out of memory of the computer. As an evidence, we search out the example that the amount of memory required by NIMBLE sharply increases as the number of parallel chains increases (page 26, Beraha et al. (2021)). This seems inevitable because NIMBLE has to compile the NIMBLE code into C++ for every parallel chain. On the contrary, in **R**, the parallel computing is efficient and does not suffer from the problem of the consuming of memory.

3.4 Stan vs. NIMBLE: real data analysis

In this subsection, we compare the Bayesian estimation performance under the trimmed mean regression model setting by the tools, Stan, NIMBLE, and **R** for the data set of the mineral content of the arm bones (page 43, Johnson et al. (2007)). The data set was analyzed by Chen et al. (2001) through the so called ALWO estimation in the meaning of the approach of a generalization of the linear Winsorized

mean (Section 5, page 153). The data contain 25 subjects including the dominant radius (y) and 4 covariates, mineral content in the dominant humerus (x_1), mineral content in the remaining humerus (x_2), mineral content in the dominant ulna (x_3), and mineral content in the remaining ulna (x_4).

On the one hand, we compare the fitting results of the proposed weakly informative and informative estimation methods under the TMR model with a) the approach by Reich et al. (2010) under the QR model, and b) the frequency approach ALWO presented by Chen, Welsh and Chan under the multivariate regression model. On the other hand, we make a companion comparison of MCMC performance when implementing the proposed two estimation procedures under the TMR model for the data set aforementioned in Stan, NIMBLE and **R**.

Data aberration checking

First we inspect the possible data features and distribution shape. Figure 3.16 displays the approximate shape of the distribution of the data and residual plot against the fitted value for multivariate regression. The box plot shows that the data have a long lower tail and a heavy upper tail. One outlier is detected in the lower tail. No skewness is detected by box plot since the sample median and the sample mean are almost coincided. Thus we analyze the method by Reich et al. (2010) under the median regression model directly later. The residual plot is not in a horizontal band and hence the variance is not a constant. Furthermore, the subjects of 17, 19, and 23 are highly suspected outliers in both tails. Therefore, heteroscedastic regression is reasonable to model the data.

Next we take a view on the ordinary residual against the subject number and the Q-Q plot in Figure 3.17. The plot of ordinary residuals shows that the traditional multivariate regression is not adequate to fit the data. In the Q-Q plot, the main

centre fits pretty well, whereas the two tails are aberrant with obvious outliers. It looks as if the underlying distribution of the error term is a three piecewise function. Therefore, it is rational to approximate the unknown error distribution by the proposed 4-component mixture (Table 3.1)

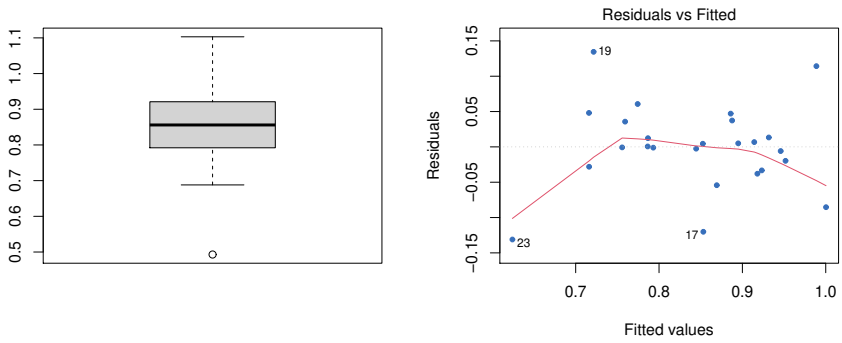


Figure 3.16: The box plot (left panel) and residual plot (right panel) of multivariate regression for the arm bones data set

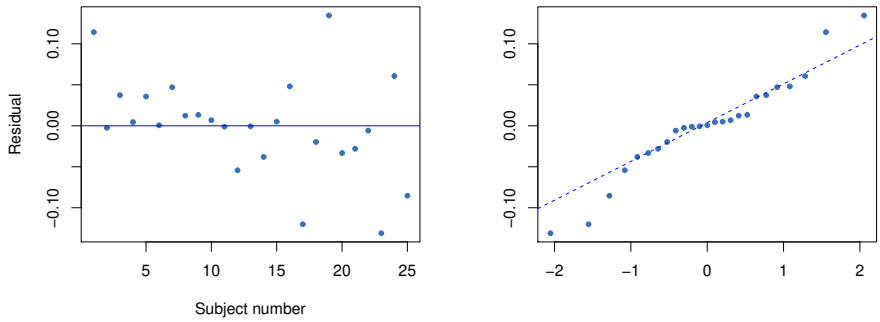


Figure 3.17: Plot of ordinary residual against the subject number and Q-Q plot of residual of multivariate regression the mineral content of the arm bones data set

Taking the data aberration into consideration, we use the following linear regression model to fit the data:

$$y = \beta_0 + x_1\beta_1 + x_2\beta_2 + x_3\beta_3 + x_4\beta_4 + \exp(x_1\gamma_1 + x_2\gamma_2 + x_3\gamma_3 + x_4\gamma_4)\varepsilon, \tag{3.5}$$

where $\beta = (\beta_1, \cdots, \beta_4)^T$ denotes the effect of the regressors on the mean function, β_0 denotes the intercept term and $\gamma = (\gamma_1, \cdots, \gamma_4)^T$ denotes the effect of regressors on the variance.

The MCMC settings and choice of priors

We implement the TMR model in Stan, NIMBLE and **R**. Similarly to the simulation study, in Stan, we use the weakly informative nonparametric prior. In NIMBLE, both weakly informative and informative priors are implemented, whereas we only use the informative prior in R.

Table 3.5: MCMC settings and choice of priors for real data analysis

	Weakly informative		Informative	
	Stan	NIMBLE	NIMBLE	**R**
LoC	6,000	100,000,	30,000	100,000
LoB	1,000	50,000	5,000	50,000
Thin	1	10	5	10
β prior	N(0, 1)	N$(0, 100^2)$	N$(0, 100^2)$	N$(0, 100^2)$
γ prior	N(0, 1)	N(0, 1)	N(0, 1)	N$(0, 10^2)$
μ prior	N(0, 1)	DExp(0,1)	DExp(0,1)	N$(0,2^2)$
σ prior	Inv_Gamma(2, 1)	U(0.01,100)	U(0.01,100)	N$(0, 1)I(x > 0)$

LoC: Length of the MCMC chains; LoB: Length of burn-in steps; DExp: double exponential distribution.

In order to get better estimation performance, we adjust different MCMC settings in different computing tools in this part. The details about the MCMC settings and the choice of priors in different tools are shown in Table 3.5. In NIMBLE, most priors are diffuse except the prior for μ, where we use the double exponential prior as a special case of ASL prior (Reich et al., 2010). We have to point out that in Stan and **R** we have to choose the prior carefully. Otherwise, either the estimation or the MCMC performance would behave even worse. We also set different initial values for MCMC in different tools. In NIMBLE, we simply use the randomly generated number from standard normal distribution as the initials for β, γ and μ and set all $\sigma = 1$ as the initial. However, in Stan and R, we use the estimated β by multiple linear regression as the initial for β. To some extent, in this example, NIMBLE allows the user to choose arbitrary possible but legal priors and initials, whereas in Stan

and R, the user needs to be cautious about the choice of priors and initial values.

We run Stan, NIMBLE and R program in the **R** environment of version 4.0.3. The CPU is a 2.3GHz 8-core Intel Core i9 processor and the platform is the x86_64 Apple. We record the time cost of sampling by MCMC. In weakly informative estimation, the time cost of sampling in Stan and in NIMBLE is about 84 seconds and about 86 seconds, respectively; in informative estimation, the time cost in NIMBLE and in **R** is about 690 seconds and about 683 seconds, respectively.

Estimation performance

Table 3.6: The parametric estimation results of weakly informative estimation

	Stan			NIMBLE		
Parameter	Estimate	SD	ESS	Estimate	SD	ESS
β_0	0.080	0.890	19907	0.236	0.122	151
β_1	0.292	0.855	20593	0.127	0.099	57
β_2	-0.185	0.854	20441	-0.046	0.122	52
β_3	0.351	0.973	19616	0.286	0.150	249
β_4	0.401	0.952	20015	0.380	0.209	74
γ_1	1.187	0.819	19404	-1.630	0.731	253
γ_2	1.226	0.836	19471	-1.073	0.721	282
γ_3	0.539	0.969	19903	-0.528	0.953	1266
γ_4	0.554	0.954	19558	-0.410	0.942	1198

SD: the standard deviation; ESS: the effective sample size .

We use the mean of posterior samples as the estimator of parameters. The parametric estimation results given by the weakly informative estimation method in Stan and NIMBLE are shown in Table 3.6 and the results of the informative estimation method in NIMBLE and **R** are in Table 3.7. Since the underlying truth is unknown, we cannot evaluate the bias or RMSE here. We list ESS here as a reference that a larger ESS indicates a more reliable approximation to the posterior distribution. It can be found that the results of the estimations by different tools are not similar to each

other. We conjecture that the sample size of data is not large enough to cover the uncertainty of the Bayesian estimator, especially when the ratio of sample size over the dimension of parameters is not large.

Table 3.7: The parametric estimation results of informative estimation

	NIMBLE			R		
Parameter	Estimate	SD	ESS	Estimate	SD	ESS
β_0	0.025	0.133	2855	0.100	0.234	3
β_1	0.081	0.126	1149	0.085	0.133	17
β_2	0.076	0.142	1123	0.125	0.134	17
β_3	0.443	0.220	3029	0.057	0.227	2
β_4	0.285	0.227	1967	0.455	0.223	10
γ_1	-1.165	0.717	1297	-0.039	1.085	8
γ_2	-0.313	0.729	1221	0.659	0.848	4
γ_3	-0.326	0.889	6726	-2.8118	5.557	7
γ_4	0.217	0.892	6017	-3.961	3.419	9

SD: the standard deviation; ESS: the effective sample size .

To evaluate the estimation results, similar to the simulation study, we use the mean square error (MSE) and median of absolute distance (MAD) as the assessments for estimation performance. The comparison with other methods is shown in Table 3.8. Since no skewness is detected from the box plot, we compare with the median regression given by Reich et al. (2010) by fixing $\tau = 0.5$ (MDR for short).

From Table 3.8 we find the estimation given by trimmed mean regression has the ultra best performance in MSE. That's not surprising since the trimmed mean is expected to minimize the square type error. Unlike the frequency approaches which simply remove the Winsorized observations out of the data, the proposed method analyzes all data together, which efficiently adjusts the influence of the outliers. In terms of the MAD, both the ALWO estimator and the weakly informative estimation in NIMBLE have the lowest MAD. ALWO performs well in MAD since it aims to minimize the l_1 error. But the results given by weakly informative estimation under

the TMR model are comparable. We find the result of weakly informative estimation in NIMBLE performs overwhelmingly better than other methods for the Bayesian trimmed mean regression model in both MSE and MAD, indicating NIMBLE is the preferable choice to analyze this data set.

Table 3.8: Evaluation of the estimation performance

	TMR				QR	Frequency
	W+S	W+N	I+N	I+R	MDR	ALWO
MSE	0.005	**0.004**	0.005	0.005	0.04	0.04
MAD	0.046	**0.021**	0.047	0.031	0.027	**0.021**

W+S: Weakly informative prior by Stan;
W+N: Weakly informative prior by NIMBLE;
I+N: Informative priorr by NIMBLE;
I+R: Informative prior by R;
MDR: median regression;
ALWO: the method by Chen, Welsh and Chan;
MSE: mean square error;
MAD: the median of the absolute distance.

MCMC performance

In this subsection we conclude the MCMC performance of the implementation of trimmed mean regression model for the real data example in Stan, NIMBLE and R by visualization including the trace plots, density plots, Q-Q plots, ACF plots and dynamic plots of improved Gelman-Rubin statistics.

Weakly informative estimation

We first take a glance at Figure 3.18 to view the trace plot of MCMC chains for β_1 and γ_1 in Stan and NIMBLE using the weakly informative prior. All the chains in the picture converge, but the sticks of samples generated by Stan are much denser than that of NIMBLE and the chains have a better mixture. The density plots of posterior samples by Stan and NIMBLE using weakly informative prior are shown in figure 3.19. We find that the posteriors produced by Stan and NIMBLE are all

bell-shaped. The posterior produced by Stan is more nearly normal since the ESS by Stan is much higher than that of NIMBLE and therefore by MCMC central limit theorem, the posterior density is more likely to be normal.

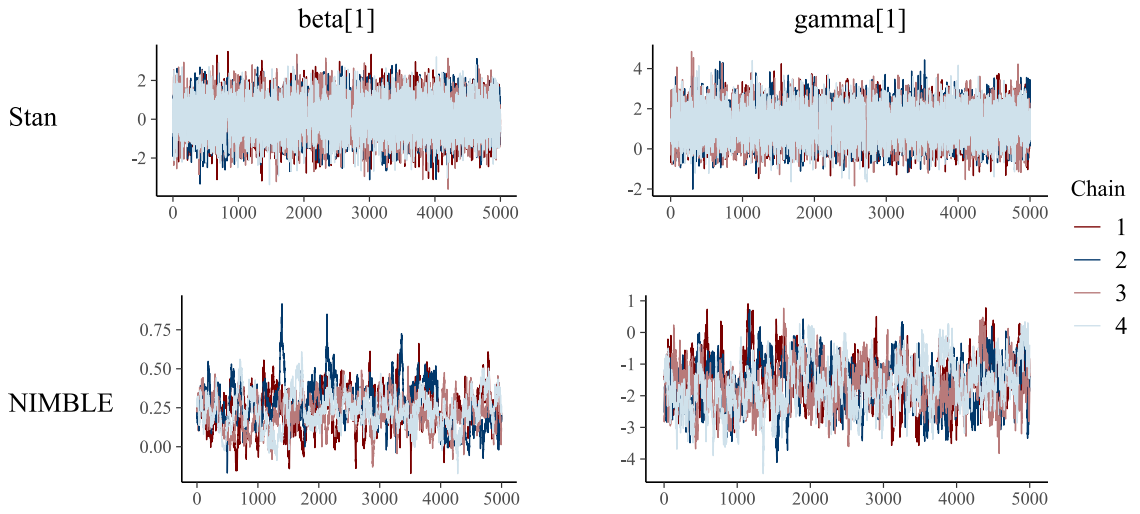
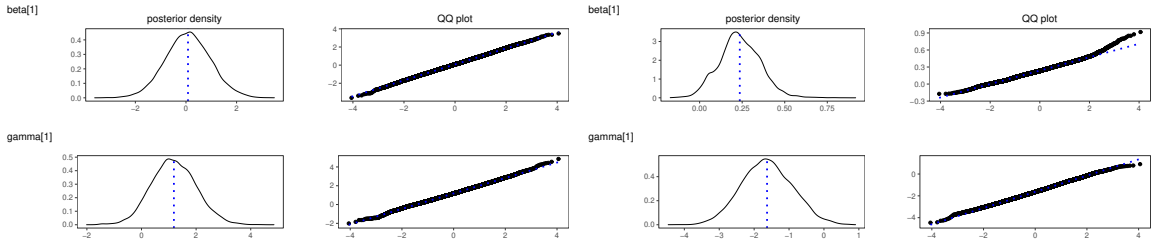


Figure 3.18: The MCMC trace plots of the mineral content of the arm bones data set for parameters (β_1 and γ_1) using weakly informative kernel simulated by Stan and NIMBLE



(a) Stan (b) NIMBLE

Figure 3.19: The density plot and Q-Q plots of the mineral content of the arm bones data set of parameters (β_1 and γ_1) using weakly informative kernel simulated by Stan and NIMBLE

The ACF plots in Figure 3.20 also demonstrate that the ACF between the samples generated by NIMBLE is much higher than that of Stan. That implies the NUTS used by Stan is more powerful to generate effective samples in the sampling procedure than the MH used by NIMBLE. The power of generating effective samples will affect the speed of convergence of MCMC chains. From Figure 3.21, we can find the chains

generated by Stan converges quite well in the early period of the chains, whereas NIMBLE pays more effort to achieve the convergence.

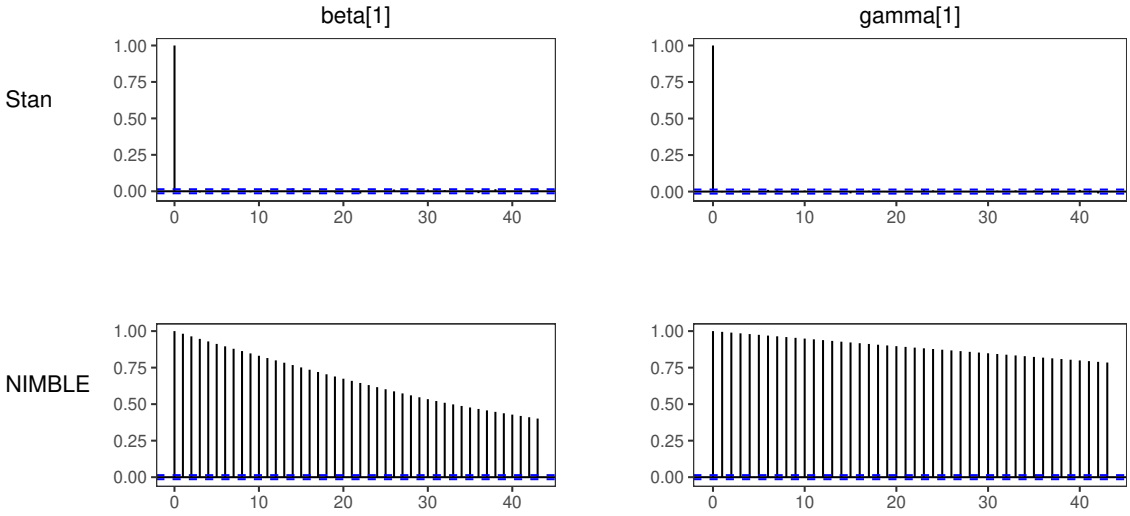


Figure 3.20: The ACF plot of the mineral content of the arm bones data set of parameters (β_1 and γ_1) using weakly informative kernel simulated by Stan and NIMBLE

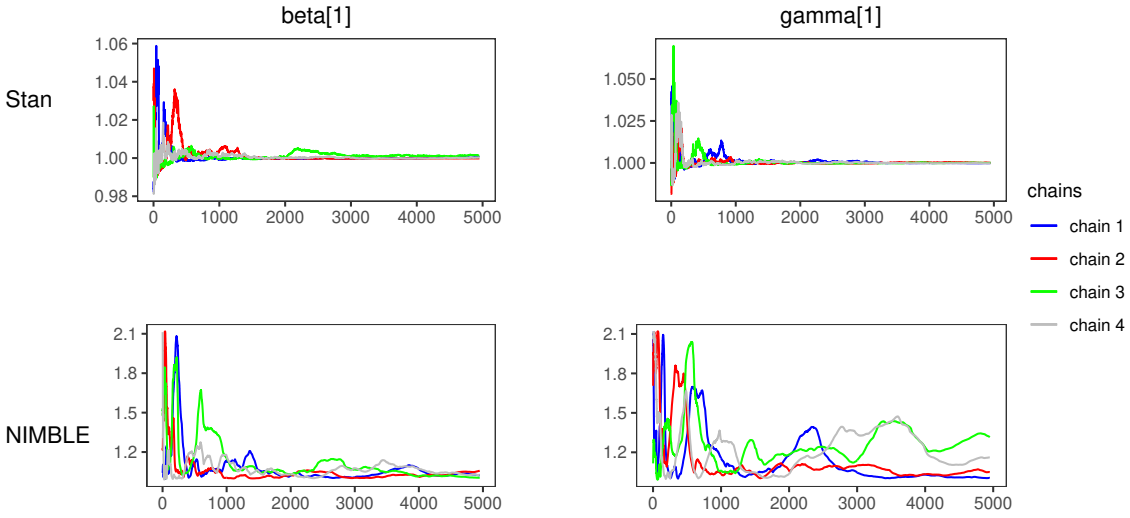


Figure 3.21: The dynamic improved Gelman-Rubin plot of the arm bones data set of parameters (β_1 and γ_1) using weakly informative kernel simulated by Stan and NIMBLE

Informative estimation

Here we compare the MCMC performance of the implementation of trimmed mean regression model for the real data example in NIMBLE and R. The trace plot of

MCMC chains for β_1 and γ_1 in NIMBLE and in R are shown in Figures 3.22. From that we find the chains generated by NIMBLE are well mixed and convergent but those generated by R are not. We point out that for this data example, we have to tune the standard deviation parameter in the MH sampler in R by ourselves, which is fully experience based. By the time of the submission of this thesis, we cannot find ideal tuning parameters s.t all the chains are well-mixed. We think it is a general problem to practitioners of Bayesian when using R, that they might struggle it with the tuning process even though they have correctly modeled the data.

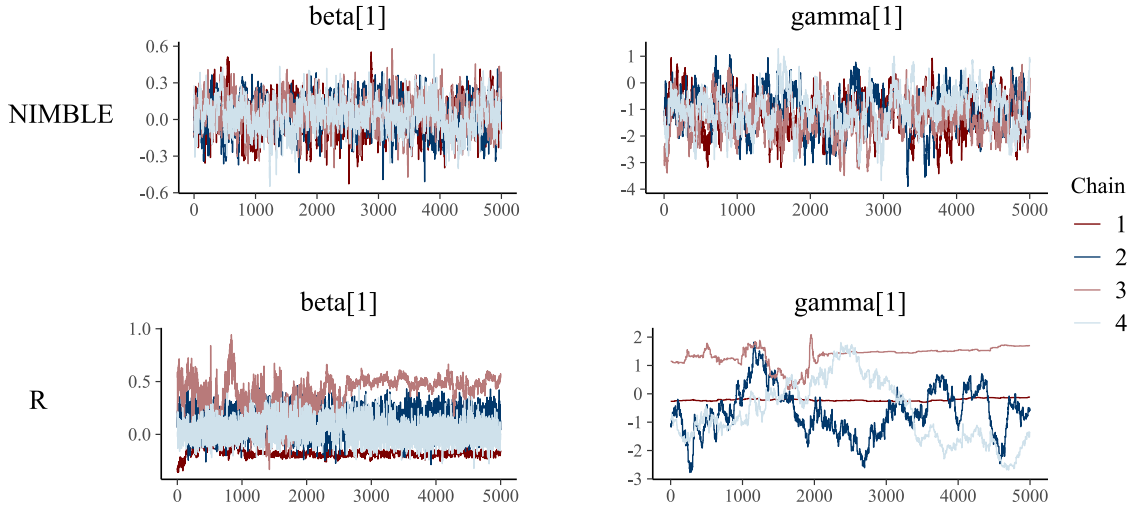
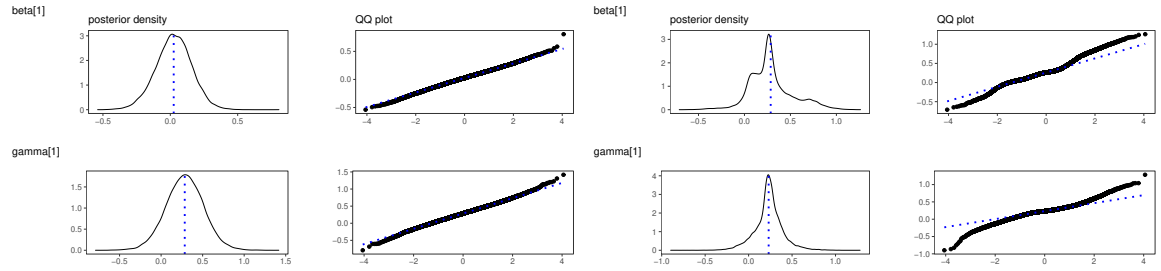


Figure 3.22: The MCMC trace plots of the mineral content of the arm bones data set for parameters (β_1 and γ_1) using informative kernel simulated by NIMBLE and R



(a) NIMBLE (b) R

Figure 3.23: The density plot and Q-Q plots of the mineral content of the arm bones data set of parameters (β_1 and γ_1) using informative kernel simulated by NIMLE and R

The posterior density of samples produced by NIMBLE is very close to the normal distribution, but the posterior density produced by R software is irregular and has double peaks (Figures: 3.23(a) and 3.23(b)). The main reason is that the chains generated by R contain very few effective samples. Thus, the sample distribution is not guaranteed to be Gaussian.

The auto-correlation of samples generated by R is very high, which implies that we might take higher thinning parameters (Figure 3.24). However, the thinning process seriously slows down the program with low efficiency. Therefore, we do not try higher thinning here. By examining the dynamic plot of improved G-R statistics (Figure 3.25), we find that the statistics finally converge to 1 in NIMBLE for all chains. However, it is worth mentioning that the statistics of γ in R do not converge to 1, which indicates the posterior of γ simulated by **R** may not be reliable. We conjecture this is because we don't tune the parameters in MH jump distributions well. By the time we submitted the thesis, we had not tuned it well. Actually, tuning such parameters in **R** is not friendly to practitioners of Bayesian who are not experienced experts.

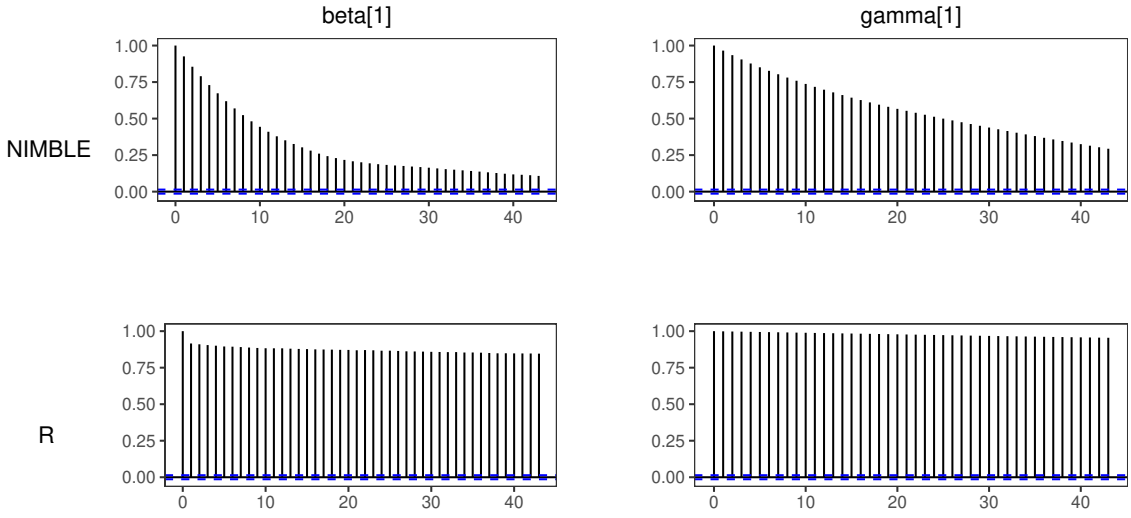


Figure 3.24: The ACF plot of the mineral content of the arm bones data set of parameters (β_1 and γ_1) using informative kernel simulated by NIMBLE and R

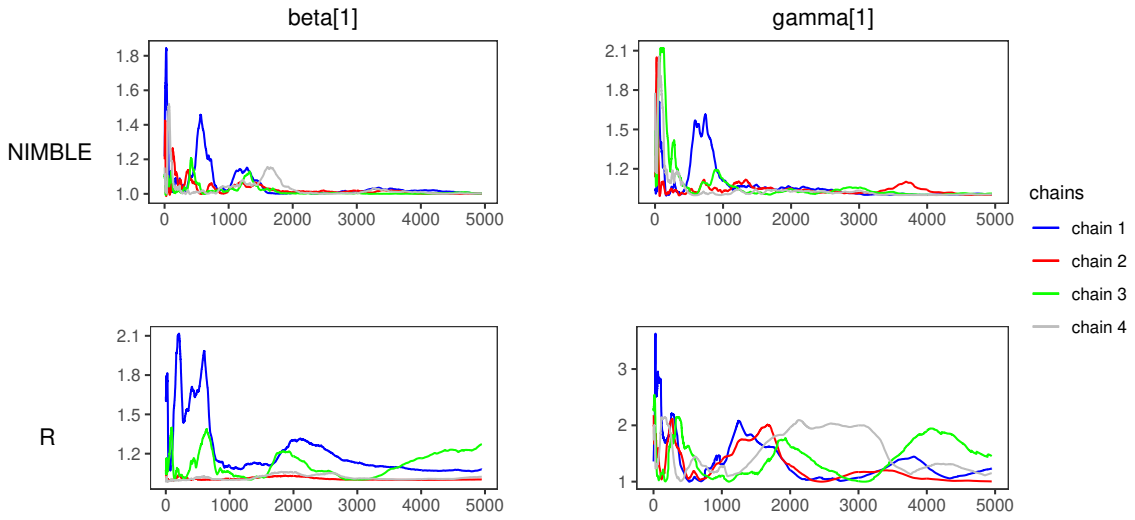


Figure 3.25: The dynamic improved Gelman-Rubin plot of the arm bones data set of parameters (β_1 and γ_1) using informative kernel simulated by NIMBLE and R

Chapter 4

Conclusion

4.1 The miscellaneous

In addition to the official website, there are some useful user forums for user to discuss about installation, programming, or debugging. On Link7 (2021) the users or Research and development staff will answer some questions for you. Also there are some useful tutorials on how to use the above programming tools. (Link1 (2020)). Here we have listed some user comments.

- "Stan is easy to troubleshoot which saves the user time. "(Link8 (2021)).
- "I am using NIMBLE on two different terminals at the same time on Mac. I am very much looking forward to the parallel NIMBLE to take advantage of superclusters although I am very satisfied with NIMBLE so far." (Link9 (2021))
- "Don't get the impression that NIMBLE wins because it shows the fastest mean MCMC Efficiency. What matters more is that Stan is much more efficient at sharing good mixing among all parameters, as shown by its faster minimum MCMC Efficiency." (Link2 (2021))

4.2 Discussion: our view

In this thesis, we compare the Stan with NIMBLE, the two recent statistical packages for Bayesian analysis. In conclusion, we plot the relationship between the popular

computing tools and the three widely used MCMC samplers in Figure 4.1. The random walk behavior allows a resampling scheme to be available when using MH and Gibbs sampler. However, resampling might be unsolvable to the gradient-based sampler HMC and NUTS, and possibly a reparameterization is needed. NIMBLE extends the BUGS and JAGS since it can call both MH and Gibbs sampler.

We treat R as a higher level of programming language since all the other tools in Figure 4.1 can be called from R directly and R is freely to any kind of samplers. But deriving of posterior or parameter tuning is required. The tuning and deriving free computing tools in the picture dramatically ease the application of MCMC in practice and in most time their performance is appealing.

The basic MCMC sampler used by Stan is the HMC and NUTS, which are gradient-based and avoid random walk behavior, leading to high computational efficiency and nice properties of simulated samples. One cannot think of many sensible use cases where randomness in the posterior density would be desirable. Alternatively, if one is just trying to "invert" the R code, that is, to build a model that takes the resulting data and can make inference about some unobserved parameters, than one would usually "replace random number generator with sampling statements". But one may need something else, such as the program background, data features, data-based inference of interest.. As pointed out by a Stan developer at the Stan forum, "Stan needs the posterior density to be completely deterministic, without any randomness", and thus when we call for a resampling scheme in the MCMC, just like the example we discussed in Section 3.3.3, Stan cannot handle. Actually, Stan replaces the process that draws a sample from distribution into a process solving differential equations based on the gradient of posterior, which denies any kind of discreteness. When discrete variables are considered, a smoothing procedure might be indispensable to call Stan.

In contrast, NIMBLE, which assigns Gibbs and MH sampler to different type of

parameters, is not so strict with the properties of posterior. In the trimmed mean regression example, the resampling scheme is simply activated by the *dconstraint* function in NIMBLE. NIMBLE is friendly to users of BUGS, JAGS and R, but one possible challenge that NIMBLE will bring to the user is the definition of random number generation function (rFUN). As de Valpine et al. (2021) points out, if the user simulates the samples from a user defined distribution, the rFUN is necessary. Generally used mixture model such as the DPM can be easily expressed as a categorical variable nested with a certain density, which has been defined in NIMBLE already, and thus this will not trouble user. This challenge may occur when a complicated nested model is considered.

From our perspective, we may conjecture the philosophy of Stan, NIMBLE are totally different. Although both Stan and NIMBLE are fully Bayesian, their target users are different. Stan is so ambitious that it can be called from other platforms (such as R, Python, Julia, Matlab), which enables it to solve even large-scaled data. Stan and its relative package such as *Bayesplot* (Gabry and Mahr, 2021) makes the visualization of MCMC and posterior shiny and charming. That is reason why Stan is popularly welcomed by Bayesianists. But NIMBLE is actually an R package and is R-oriented such that any user of R can pick it up easily. For example, the "RCall" command in NIMBLE makes it easy to call various functions defined in R and thus the R user can realize some key computation in R rather than in NIMBLE code. Further more, the BUGS style of language used in NIMBLE is easily understood by an R user such that one is able to construct an MCMC conveniently even though the user is not a Bayesian statistician.

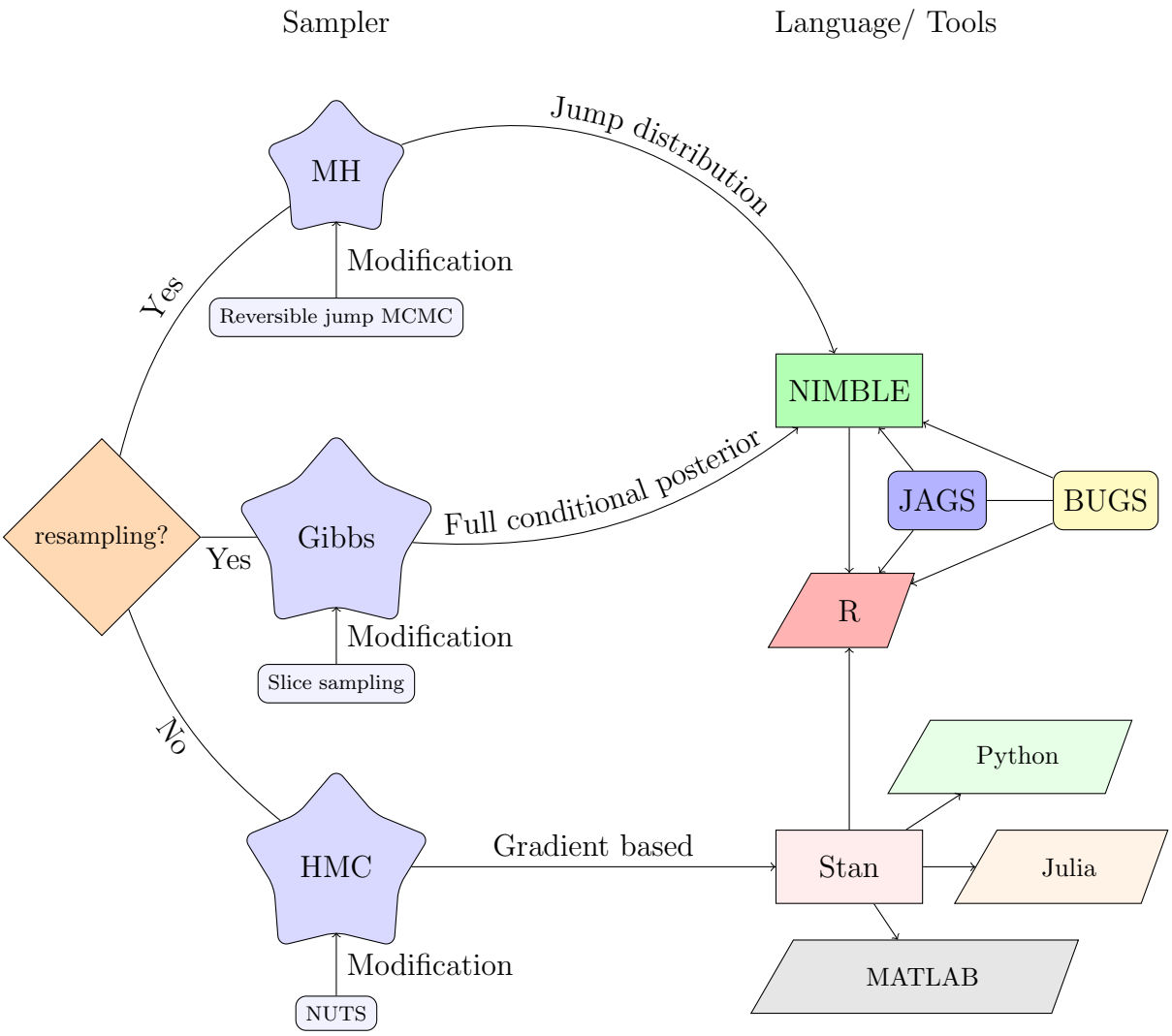


Figure 4.1: Relation of MCMC samplers to Bayesian programming language/ software tools. On the left, It is an unnecessary decision to decide whether resampling is allowed or not. The items in the middle are MCMC samplers. The third column is the various programming languages/ packages/ software or tools.

Bibliography

Afify, A. Z., Gemeay, A. M. and Ibrahim, N. A. (2020) The heavy-tailed exponential distribution: Risk measures, estimation, and application to actuarial data. *Mathematics*, **8**, 1276.

Alhamzawi, R. and Ali, H. T. M. (2018) Bayesian quantile regression for ordinal longitudinal data. *Journal of Applied Statistics*, **45**, 815–828.

Alhamzawi, R. and Yu, K. (2013) Conjugate priors and variable selection for bayesian quantile regression. *Computational Statistics & Data Analysis*, **64**, 209–219.

Atkinson, A. C., Riani, M. and Torti, F. (2016) Robust methods for heteroskedastic regression. *Computational Statistics & Data Analysis*, **104**, 209–222.

Benavoli, A., Corani, G., Demšar, J. and Zaffalon, M. (2017) Time for a change: a tutorial for comparing multiple classifiers through bayesian analysis. *The Journal of Machine Learning Research*, **18**, 2653–2688.

Beraha, M., Falco, D. and Guglielmi, A. (2021) Jags, nimble, stan: a detailed comparison among bayesian mcmc software. *arXiv preprint arXiv:2107.09357*.

Berger, J. O. (2013) *Statistical decision theory and Bayesian analysis*. Springer Science & Business Media.

Bernardi, M., Bottone, M. and Petrella, L. (2016) Bayesian robust quantile regression. *arXiv preprint arXiv:1605.05602*.

Bickel, P. J. (1973) On some analogues to linear combinations of order statistics in the linear model. *The Annals of Statistics*, 597–616.

Bickel, P. J. et al. (1965) On some robust estimates of location. *The Annals of Mathematical Statistics*, **36**, 847–858.

Box, G. E. and Tiao, G. C. (1968) A bayesian approach to some outlier problems. *Biometrika*, **55**, 119–129.

Brooks, S., Gelman, A., Jones, G. and Meng, X.-L. (2011) *Handbook of markov chain monte carlo*. CRC press.

Brooks, S. P. and Gelman, A. (1998) General methods for monitoring convergence of iterative simulations. *Journal of computational and graphical statistics*, **7**, 434–455.

Buchholz, A., Chopin, N. and Jacob, P. E. (2021) Adaptive tuning of hamiltonian monte carlo within sequential monte carlo. *Bayesian Analysis*, **1**, 1–27.

Chahal, R., Gotlib, I. H. and Guyer, A. E. (2020) Research review: Brain network connectivity and the heterogeneity of depression in adolescence–a precision mental health perspective. *Journal of Child Psychology and Psychiatry*.

Chen, C. and Yu, K. (2009) Automatic bayesian quantile regression curve fitting. *Statistics and Computing*, **19**, 271–281.

Chen, L., Welsh, A. and Chan, W. (2001) Estimators for the linear regression model based on winsorized observations. *Statistica Sinica*, 147–172.

Chen, L.-A. (1997) An efficient class of weighted trimmed means for linear regression models. *Statistica Sinica*, 669–686.

Chen, L.-A. and Chiang, Y.-C. (1996) Symmetric quantile and symmetric trimmed mean for linear regression model. *Journal of Nonparametric Statistics*, **7**, 171–185.

de Valpine, P., Paciorek, C., Turek, D., Michaud, N., Anderson-Bergman, C., Obermeyer, F., Wehrhahn Cortes, C., Rodrìguez, A., Temple Lang, D. and Paganin, S. (2021) *NIMBLE User Manual*. URL https://r-nimble.org. R package manual version 0.11.1.

De Wet, T. and Van Wyk, J. (1979) Efficiency and robustness of hogg's adaptive trimmed means. *Communications in statistics-theory and methods*, **8**, 117–128.

Dey, D. K. and Rao, C. R. (2005) Handbook of statistics. In *Bayesian Thinking: Modeling and Computation* (eds. D. Dey and C. Rao), vol. 25. Elsevier. URL https://www.sciencedirect.com/science/article/pii/S0169716105250216.

Dhar, S. S. and Chaudhuri, P. (2012) On the derivatives of the trimmed mean. *Statistica Sinica*, 655–679.

Dolmas, J. et al. (2005) Trimmed mean pce inflation. *Federal Reserve Bank of Dallas Working Paper*, **506**.

Dunson, D. B. and Taylor, J. A. (2005) Approximate bayesian inference for quantiles. *Journal of Nonparametric Statistics*, **17**, 385–400.

Fabrizi, E., Salvati, N. and Trivisano, C. (2020) Robust bayesian small area estimation based on quantile regression. *Computational Statistics & Data Analysis*, **145**, 106900.

Feng, Y., Chen, Y. and He, X. (2015) Bayesian quantile regression with approximate likelihood. *Bernoulli*, **21**, 832–850.

Ferguson, T. S. (1973) A bayesian analysis of some nonparametric problems. *The annals of statistics*, 209–230.

Gabry, J. and Mahr, T. (2021) bayesplot: Plotting for bayesian models. URL https://mc-stan.org/bayesplot/. R package version 1.8.1.

Gagnon, P., Desgagné, A., Bédard, M. et al. (2020) A new bayesian approach to robustness against outliers in linear regression. *Bayesian Analysis*, **15**, 389–414.

Gao, Y., Kennedy, L., Simpson, D., Gelman, A. et al. (2021) Improving multilevel regression and poststratification with structured priors. *Bayesian Analysis*.

Gelfand, A. E., Hills, S. E., Racine-Poon, A. and Smith, A. F. (1990) Illustration of bayesian inference in normal data models using gibbs sampling. *Journal of the American Statistical Association*, **85**, 972–985.

Gelman, A., Carlin, J. B., Stern, H. S., Dunson, D. B., Vehtari, A. and Rubin, D. B. (2013) *Bayesian data analysis*. CRC press.

Gelman, A., Lee, D. and Guo, J. (2015) Stan: A probabilistic programming language for bayesian inference and optimization. *Journal of Educational and Behavioral Statistics*, **40**, 530–543.

Gelman, A. and Rubin, D. B. (1992) Inference from iterative simulation using multiple sequences. *Statistical science*, **7**, 457–472.

Gelman, A. and Vákár, M. (2021) Slamming the sham: A bayesian model for adaptive adjustment with noisy control data. *Statistics in medicine*.

Gelman, A., Vehtari, A., Simpson, D., Margossian, C. C., Carpenter, B., Yao, Y., Kennedy, L., Gabry, J., Bürkner, P.-C. and Modrák, M. (2020) Bayesian workflow. *arXiv preprint arXiv:2011.01808*.

Geman, S. and Geman, D. (1984) Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *IEEE Transactions on pattern analysis and machine intelligence*, 721–741.

Ghosh, J., Li, Y. and Mitra, R. (2018) On the use of cauchy prior distributions for bayesian logistic regression. *Bayesian Analysis*, **13**, 359–383.

Goldstein, M. (1980) The linear bayes regression estimator under weak prior assumptions. *Biometrika*, **67**, 621–628.

Hastings, W. K. (1970) Monte carlo sampling methods using markov chains and their applications.

He, X. (1997) Quantile curves without crossing. *The American Statistician*, **51**, 186–192.

Hill, R. C., Griffiths, W. E. and Lim, G. C. (2018) *Principles of econometrics*. John Wiley & Sons.

Hoffman, M. D. and Gelman, A. (2014) The no-u-turn sampler: adaptively setting path lengths in hamiltonian monte carlo. *J. Mach. Learn. Res.*, **15**, 1593–1623.

Hogg, R. V. (1974) Adaptive robust procedures: A partial review and some suggestions for future applications and theory. *Journal of the American Statistical Association*, **69**, 909–923.

Hovik, K. T., Plessen, K. J., Skogli, E. W., Andersen, P. N. and Øie, M. (2016) Dissociable response inhibition in children with tourette's syndrome compared with children with adhd. *Journal of attention disorders*, **20**, 825–835.

Hu, Y., Gramacy, R. B. and Lian, H. (2013) Bayesian quantile regression for single-index models. *Statistics and Computing*, **23**, 437–454.

Huang, Y. (2016) Quantile regression-based bayesian semiparametric mixed-effects models for longitudinal data with non-normal, missing and mismeasured covariate. *Journal of Statistical Computation and Simulation*, **86**, 1183–1202.

Huang, Y. and Chen, J. (2016) Bayesian quantile regression-based nonlinear mixed-effects joint models for time-to-event and longitudinal data with multiple features. *Statistics in medicine*, **35**, 5666–5685.

Huang, Y., Chen, J. and Qiu, H. (2017) Bayesian quantile regression for nonlinear mixed-effects joint models for longitudinal data in the presence of mismeasured covariate errors. *Journal of biopharmaceutical statistics*, **27**, 741–755.

Insua, D. R. and Ruggeri, F. (2012) *Robust Bayesian Analysis*, vol. 152. Springer Science & Business Media.

Johnson, R. A., Wichern, D. W. et al. (2007) *Applied multivariate statistical analysis*, vol. 5. Prentice hall Upper Saddle River, NJ.

Kass, R. E., Carlin, B. P., Gelman, A. and Neal, R. M. (1998) Markov chain monte carlo in practice: a roundtable discussion. *The American Statistician*, **52**, 93–100.

Koenker, R. and Bassett Jr, G. (1978) Regression quantiles. *Econometrica: journal of the Econometric Society*, 33–50.

Korner-Nievergelt, F., Roth, T., Von Felten, S., Guélat, J., Almasi, B. and Korner-Nievergelt, P. (2015) *Bayesian data analysis in ecology using linear models with R, BUGS, and Stan*. Academic Press.

Kottas, A. and Gelfand, A. E. (2001) Bayesian semiparametric median regression modeling. *Journal of the American Statistical Association*, **96**, 1458–1468.

Kottas, A. and Krnjajić, M. (2009) Bayesian semiparametric modelling in quantile regression. *Scandinavian Journal of Statistics*, **36**, 297–319.

Kozumi, H. and Kobayashi, G. (2011) Gibbs sampling methods for bayesian quantile regression. *Journal of statistical computation and simulation*, **81**, 1565–1578.

Kruschke, J. (2014) Doing bayesian data analysis: A tutorial with r, jags, and stan.

Lancaster, T. and Jae Jun, S. (2010) Bayesian quantile regression methods. *Journal of Applied Econometrics*, **25**, 287–307.

Lee, D. and Neocleous, T. (2010) Bayesian quantile regression for count data with application to environmental epidemiology. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, **59**, 905–920.

Li, Q., Lin, N. and Xi, R. (2010) Bayesian regularized quantile regression. *Bayesian Analysis*, **5**, 533–556.

Link1 (2020) Applied bayesian statistics using stan and r: R-bloggers. URL https://www.r-bloggers.com/2020/01/applied-bayesian-statistics-using-stan-and-r/.

Link2 (2021) How we make mcmc comparisons. URL https://nature.berkeley.edu/~pdevalpine/MCMC_comparisons/nimble_MCMC_comparisons.html.

Link3 (2021) URL https://mc-stan.org/.

Link4 (2021) URL https://mc-stan.org/rstan/.

Link5 (2021) URL https://r-nimble.org/.

Link6 (2020) URL https://mc-stan.org/docs/2_27/reference-manual/index.html.

Link7 (2021) URL https://discourse.mc-stan.org/.

Link8 (2021) URL https://discourse.mc-stan.org/t/selling-stan/3693/2.

Link9 (2021) URL https://groups.google.com/g/nimble-users/c/HQI64qN8tBc/m/2naPM9jEAwAJ.

Luo, Y., Lian, H. and Tian, M. (2012) Bayesian quantile regression for longitudinal data models. *Journal of Statistical Computation and Simulation*, **82**, 1635–1649.

Ma, Z. and Chen, G. (2020) Bayesian semiparametric latent variable model with dp prior for joint analysis: Implementation with nimble. *Statistical Modelling*, **20**, 71–95.

McElreath, R. (2018) *Statistical rethinking: A Bayesian course with examples in R and Stan*. Chapman and Hall/CRC.

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H. and Teller, E. (1953) Equation of state calculations by fast computing machines. *The journal of chemical physics*, **21**, 1087–1092.

Peña, D., Zamar, R. and Yan, G. (2009) Bayesian likelihood robustness in linear models. *Journal of statistical planning and inference*, **139**, 2196–2207.

Plummer, M. et al. (2003) Jags: A program for analysis of bayesian graphical models using gibbs sampling. In *Proceedings of the 3rd international workshop on distributed statistical computing*, vol. 124, 1–10. Vienna, Austria.

Ponisio, L. C., de Valpine, P., Michaud, N. and Turek, D. (2020) One size does not fit all: Customizing mcmc methods for hierarchical models using nimble. *Ecology and evolution*, **10**, 2385–2416.

Pusparum, M., Kurnia, A. and Alamudi, A. (2017) Winsor approach in regression analysis with outlier. *Applied Mathematical Sciences*, **11**, 2031–2046.

Rahman, M. A. (2016) Bayesian quantile regression for ordinal models. *Bayesian Analysis*, **11**, 1–24.

Reich, B. J. (2012) Spatiotemporal quantile regression for detecting distributional changes in environmental processes. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, **61**, 535–553.

Reich, B. J., Bondell, H. D. and Wang, H. J. (2010) Flexible bayesian quantile regression for independent and clustered data. *Biostatistics*, **11**, 337–352.

Reich, B. J., Fuentes, M. and Dunson, D. B. (2011) Bayesian spatial quantile regression. *Journal of the American Statistical Association*, **106**, 6–20.

Reich, B. J. and Ghosh, S. K. (2019) *Bayesian statistical methods*. CRC Press.

Reich, B. J. and Smith, L. B. (2013) Bayesian quantile regression for censored data. *Biometrics*, **69**, 651–660.

Resnick, S. I. et al. (1997) Heavy tail modeling and teletraffic data: special invited paper. *Annals of statistics*, **25**, 1805–1869.

Risser, M. D. and Turek, D. (2020) Bayesian inference for high-dimensional nonstationary gaussian processes. *Journal of Statistical Computation and Simulation*, **90**, 2902–2928.

Rousseeuw, P. and Yohai, V. (1987) Robust regression by means of s-estimators in robust and nonlinear time series analysis: 256-272, edited by j. franke, w. härdle and d. martin.

Ruggeri, F. (1990) Posterior ranges of functions of parameters under priors with specified quantiles. *Communications in Statistics-Theory and Methods*, **19**, 127–144.

Ruggeri, F. (2010) Nonparametric bayesian robustness. *Chilean Journal of Statistics*, **2**, 51–68.

Ruppert, D. and Carroll, R. J. (1980) Trimmed least squares estimation in the linear model. *Journal of the American Statistical Association*, **75**, 828–838.

Rydell, A.-M., Diamantopoulou, S., Thorell, L. B. and Bohlin, G. (2009) Hyperactivity, shyness, and sex: Development and socio-emotional functioning. *British Journal of Developmental Psychology*, **27**, 625–648.

Savage, J. (2016) An introduction to Bayesian modelling in Stan for economists. URL https://rstudio-pubs-static.s3.amazonaws.com/156108_e848a25b3bb348e1a3e42853731fd1be.html.

van de Schoot, R., Depaoli, S., King, R., Kramer, B., Märtens, K., Tadesse, M. G., Vannucci, M., Gelman, A., Veen, D., Willemsen, J. et al. (2021) Bayesian statistics and modelling. *Nature Reviews Methods Primers*, **1**, 1–26.

Serfling, R. J. (2009) *Approximation theorems of mathematical statistics*, vol. 162. John Wiley & Sons.

Si, Y., Pillai, N. S. and Gelman, A. (2015) Bayesian nonparametric weighted sampling inference. *Bayesian Analysis*, **10**, 605–625.

Sriram, K., Ramamoorthi, R. and Ghosh, P. (2013) Posterior consistency of bayesian quantile regression based on the misspecified asymmetric laplace density. *Bayesian analysis*, **8**, 479–504.

Stan Development Team (2020) RStan: the R interface to Stan. URL http://mc-stan.org/. R package version 2.21.2.

Tian, Y., Tian, M. et al. (2016) Bayesian joint quantile regression for mixed effects models with censoring and errors in covariates. *Computational Statistics*, **31**, 1031–1057.

Tong, X., Zhang, T. and Zhou, J. (2021) Robust bayesian growth curve modelling using conditional medians. *British Journal of Mathematical and Statistical Psychology*, **74**, 286–312.

Tukey, J. W. and McLaughlin, D. H. (1963) Less vulnerable confidence and significance procedures for location based on a single sample: Trimming/winsorization 1. *Sankhyā: The Indian Journal of Statistics, Series A*, 331–352.

de Valpine, P., Turek, D., Paciorek, C. J., Anderson-Bergman, C., Lang, D. T. and Bodik, R. (2017) Programming with models: writing statistical algorithms for general model structures with nimble. *Journal of Computational and Graphical Statistics*, **26**, 403–413.

Vehtari, A., Gelman, A., Simpson, D., Carpenter, B. and Bürkner, P.-C. (2021) Rank-normalization, folding, and localization: An improved r for assessing convergence of mcmc. *Bayesian analysis*, **1**, 1–28.

Verdinelli, I. and Wasserman, L. (1991) Bayesian analysis of outlier problems using the gibbs sampler. *Statistics and Computing*, **1**, 105–117.

Wang, C. and Blei, D. M. (2018) A general method for robust bayesian modeling. *Bayesian Analysis*, **13**, 1163–1191.

Wang, M. Y. and Park, T. (2020) A brief tour of bayesian sampling methods. *Bayesian Inference on Complicated Data*, 17.

Weber, S., Gelman, A., Lee, D., Betancourt, M., Vehtari, A. and Racine-Poon, A. (2018) Bayesian aggregation of average data: An application in drug development. *The Annals of Applied Statistics*, **12**, 1583–1604.

Wehrhahn, C., Rodriguez, A. and Paciorek, C. (2018) Bayesian nonparametric mixture models using nimble. In *NeurIPS workshop on nonparametric Bayesian models*.

Welsh, A. et al. (1987) The trimmed mean in the linear model. *The Annals of Statistics*, **15**, 20–36.

West, M. (1984) Outlier models and prior distributions in bayesian linear regression. *Journal of the Royal Statistical Society: Series B (Methodological)*, **46**, 431–439.

Xu, X., Xu, Z., Chen, L. and Li, C. (2019) How does industrial waste gas emission affect health care expenditure in different regions of china: an application of bayesian quantile regression. *International journal of environmental research and public health*, **16**, 2748.

Yang, Y., He, X. et al. (2012) Bayesian empirical likelihood for quantile regression. *The Annals of Statistics*, **40**, 1102–1131.

Yang, Y., Wang, H. J. and He, X. (2016) Posterior inference in bayesian quantile regression with asymmetric laplace likelihood. *International Statistical Review*, **84**, 327–344.

Yao, Y., Vehtari, A., Simpson, D. and Gelman, A. (2018) Using stacking to average bayesian predictive distributions (with discussion). *Bayesian Analysis*, **13**, 917–1007.

Yu, K., Alkenani, A. and Alhamzawi, R. (2012) Penalized flexible bayesian quantile regression.

Yu, K. and Moyeed, R. A. (2001) Bayesian quantile regression. *Statistics & Probability Letters*, **54**, 437–447.

Yuan, Y. and Yin, G. (2010) Bayesian quantile regression for longitudinal studies with nonignorable missing data. *Biometrics*, **66**, 105–114.

Zhang, H., Huang, Y., Wang, W., Chen, H. and Langland-Orban, B. (2019) Bayesian quantile regression-based partially linear mixed-effects joint models for longitudinal data with multiple features. *Statistical methods in medical research*, **28**, 569–588.

Zhang, Y. and Tang, N. (2017) Bayesian empirical likelihood estimation of quantile structural equation models. *Journal of Systems Science and Complexity*, **30**, 122–138.

Zhou, H., Hanson, T. and Zhang, J. (2020) spBayesSurv: Fitting Bayesian spatial survival models using R. *Journal of Statistical Software*, **92**, 1–33.