university of groningen

faculty of science and engineering

# A Comparative analysis of Bayesian and Frequentist approaches to linear regression

Bachelor's Project Mathematics
July 2021
Student: O.D.J Kolkman
First supervisor: dr. M.A. Grzegorczyk
Second assessor: dr. C.P. Hirsch

# Abstract

There are two different methods to estimate the parameters of a linear regression model: the frequentist and the Bayesian approach. This paper aims to present a more comprehensive examination of these two methods. The paper first considers the theoretical foundation of the linear regression model and the frequentist technique. Thereafter the fundamentals of Bayesian statics and the Bayesian approach are discussed. Furthermore two ways to do variable selection for the frequentist approach and the principle of cross-validation will be reviewed. Using the Boston housing data set a simulation study will be performed in which two the following estimators will be compared: ordinary least squares, Lasso, ordinary least squares in combination with backward stepwise model selection using AIC, Gibbs sampling with an uninformative prior and Gibbs sampling with an informative prior. The methods are compared using ten different sample sizes with the mean absolute deviation, the root mean square error and the mean squared error as the measures of fit.

# Contents

# Acknowledgments

First and foremost, I want to express my gratitude to my first supervisor Prof. Marco A. Grzegorczyk for assisting me with my questions, providing valuable ideas and recommending possible improvements. Furthermore I would like to thank my second assessor Prof. Christian C. Hirsch for his time and the pleasant cooperation. In addition, I would like to thank both supervisors for their flexibility.

# 1 Introduction

Linear regression modelling is a highly effective data analysis technique and a very important statistical tool. Forcasting, parameter estimation, and data explanation are only a few examples of applications of this technique. The concept of regression was introduced in 1886 by Sir Francis Galton in his paper Regression Towards Mediocrity in Hereditary Stature [7]. In this study Galton tried to discover the relation between the height of fathers and their sons. Another pivotal moment in the development of linear regression models was the discovery of the method of least squares by Adrien Marie Legendre and Carl Friedrich Gauss [14, 25]. In combination with the concept of regression this would eventually form the basis for the first linear regression model.

Linear regression models are used in a great variety of fields ranging from environmental sciences to economics to epidemiology [22, 4, 26]. The goal of these models is to figure out how the variable in question, referred to as the dependent variable, and several other variables, referred to as the explanatory variables, are related. How do variables such as crime rate and average number of rooms, for example, affect the full-value property-tax rates? To answer questions like these one can use two different methods to estimate the coefficients: the frequentist and the Bayesian approach.

The frequentist approach for linear regression assumes the coefficients to be fixed constants that maximize the likelihood. The likelihood is defined as the probability of the observed data given the model's coefficients. In the Bayesian approach the coefficients are considered to be random variables. These random variables are assumed to follow a certain prior distribution in advance. The goal of the Bayesian approach is to update this prior distribution using the observed data.

This thesis aims to present a broad analysis of the two methods and their relation based on more than one estimator for different sample sizes. The central question in this thesis is: "What is the theoretical foundation for Bayesian and frequentist estimation methods in linear regression and how do these approaches relate in a simulation study?"

Chapter 2 gives an overview of the literature about the two estimation approaches and some comparison studies. Chapter 3 will focus on the theory behind the frequentist technique for the multiple linear regression model. The chapter will begin with an overview of the multiple linear regression model and the underlying assumptions. Thereafter the ordinary least squares (OLS) estimator will be discussed. In chapter 4 the Bayesian approach will be studied. The concept of Bayesian statics will be introduced using Bayes' theorem together with some information on the prior distribution. In this section two different prior distributions will be proposed for performing Bayesian regression analysis. To approximate the posterior distribution a Markov chain Monte Carlo method called Gibbs sampling will be considered. Before performing the simulation study we will shortly review some model selection and comparison methods in chapter 5. Chapter 5 will first focus on the method of cross-validation to split the data in a training and a testing set. Then two frequentist variable selection methods called the Lasso and the backward stepwise model selection using AIC will be discussed. At the end of this section we will introduce the mean absolute deviation and the root mean square error to measure the model fit. Chapter 6 gives an overview of the data that was used and provides an exploratory data analysis. In chapter 7 the results of the simulation will be presented. The chapter consists of a model analysis and a model comparison part. In the first part of the chapter we will consider five parameter estimation techniques: OLS, Lasso, OLS in combination with backward stepwise model selection using AIC, a gibbs sampler in combination with an uninformative prior and a gibbs

sampler in combination with an informative prior. For these estimators we will check the parameter estimates, credible/confidence intervals and p-values (if available). Furthermore we will study the fitted regression values. In the second part we will compare the model fit of the five models for ten different sample sizes using the mean absolute deviation, the root mean square error and the mean squared error as measure of fit statistics. The conclusion of the paper will be given in chapter 8 after which a discussion with critical aspects and future improvements follows in chapter 9.

# 2 Literature Review

A lot of books and articles have been written about the frequentist and the Bayesian approach to linear regression. The frequentist method to regression analysis is addressed in the book by Dobson and Barnett as well as the book by Hayashi [5, 11]. The book by Dobson and Barnett covers the theoretical background of generalized linear models and techniques to analyse certain types of data. The book by Hayashi reviews the classical linear regression model with the corresponding assumptions. In both books an estimator called the ordinary least squares estimator is proposed to estimate the regression coefficients. Multiple properties of the estimator such as its expectation and variance and ways to test hypothesis about the coefficients are discussed. A disadvantage of the OLS estimator is that is likely to overfit the data. As a possible solution one can use an estimator called the Lasso which is explained in [27] and is closely related to the OLS. The Lasso sets some of the regression coefficients to zero. In [10] a backward covariate selection for the OLS is introduced which can remove some of the variables from the regression model.

In the book by Hoff [12] the basics of Bayesian statistics and the Bayesian approach to linear regression are reviewed. For the multiple linear regression model two different prior distributions are presented: a so-called flat prior distribution and a conjugate prior distribution. Furthermore the derivations of the posterior distributions are provided. The book describes a frequently used algorithm to approximate the joint posterior distribution called Gibbs sampling which is a special case of the Metropolis-Hastings algorithm. The goal of the algorithm is to generate a sequence of samples that eventually converges to a sample of the posterior distribution. The book by Dobson and Barnett also contains a few sections on the Bayesian approach. In one of these sections a backwards elimination procedure using the deviance information criterion (DIC) is illustrated to obtain a sparse regression model.

Two papers that compare the frequentist OLS estimator with the Bayesian Gibbs sampler with a conjugate prior distribution are [2] and [21]. In the paper by Hussein and Kadhim the frequentist and Bayesian estimation method for linear regression were used to forecast future observations for the unemployment rates in Iraq. To compare the model fit of the two models the root mean square error (RMSE) and median absolute deviation (MAD) were used. These criteria showed that the linear regression model with the Bayesian technique performed better than the model with the frequentist technique. The paper by Permai and Tanty's considers a linear regression model for the energy performance of residential buildings. In addition to the RMSE and MAD, the mean absolute percentage error (MAPE) was utilized to compare the models. The outcome of this study was again that the Bayesian method outperformed the frequentist method.

# 3 The Frequentist Approach to Linear Regression

## 3.1 Multiple Linear Regression

In many researches, regression models with more than one explanatory variable are used. This type of regression model is called the multiple regression model. The multiple linear regression model is usually expressed as follows:

$$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_k x_{ik} + \varepsilon_i, \ (i = 1, 2, \ldots, n; \ n > k), \tag{3.1}$$

where the $x_i$'s are the explanatory variables, $y_i$ the dependent variable, $\varepsilon_i$ the random error and $\beta_1, \beta_2, \ldots, \beta_k$ the unknown parameters. The random error term represents the component of the dependent variable $y$ that cannot be explained by the explanatory variable $x$. The model can also be written in matrix notation which turns out to be convenient for deriving the least squares estimator. A general form of the matrix notation of the multiple linear regression model is given by

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}, \tag{3.2}$$

where

$$\underset{(n\times 1)}{\mathbf{y}} = \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix}, \ \underset{(n\times k)}{\mathbf{X}} = \begin{bmatrix} \mathbf{x_1'} \\ \vdots \\ \mathbf{x_n'} \end{bmatrix} = \begin{bmatrix} x_{11} & \ldots & x_{1k} \\ \vdots & \ddots & \vdots \\ x_{n1} & \ldots & x_{nk} \end{bmatrix}, \ \underset{(k\times 1)}{\boldsymbol{\beta}} = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_k \end{bmatrix}, \ \underset{(n\times 1)}{\boldsymbol{\varepsilon}} = \begin{bmatrix} \varepsilon_1 \\ \vdots \\ \varepsilon_n \end{bmatrix}.$$

The vector $\mathbf{y}$ and the matrix $\mathbf{X}$ are referred to as the data vector and the data matrix since the rows correspond to the observations [20, 30, 5, 11].

### 3.1.1 Model assumptions

The linear regression model is based on a set of assumptions that have to be satisfied in order for the model to produce reliable/unbiased results. This section will only provide a brief overview of the assumptions. A more in-depth explanation of the assumptions can be found in the book by Hayashi [11] and the article by Águila and Benítez-Parejo [24].

**Linearity**
The first assumption underlying the simple linear regression model is linearity. The dependent variable can be written as a linear combination of the explanatory variables as presented in equation (3.9). The dependent and explanatory variables may also be transformations of the original variables. You could for example define the variable logy $= log(y)$ or x1squared $= x_1^2$ and include the transformed variable in the linear regression.

**Strict Exogeneity**
Another assumption that needs to be satisfied is strict exogeneity. This restriction states that the expectation of the error term conditioned on the explanatory variables for all observations should be equal to 0 for every observation or written down mathematically:

$$E(\varepsilon_i \mid \mathbf{X}) = 0, \ \ (i = 1, 2, \ldots, n). \tag{3.3}$$

If the strict exogeneity assumption is violated the coefficient estimates might be biased and inconsistent.

**No Multicollinearity**
The third assumption that must be met is the absence of multicollinearity. Multicollinearity occurs when multiple explanatory variables are correlated with each other. Formally stated the rank of the $n \times k$ data matrix, $\mathbf{X}$, should be $k$ with probability 1.

**Spherical Error Variance**
Spherical error variance is the assumption that there is no correlation between the different error terms and that the conditional second moment of the error term is constant which is called homoskedasticity. Homoskedasticity is denoted as follows:

$$E({\varepsilon_i}^2 \mid \mathbf{X}) = \sigma^2 > 0, \;\; (i = 1, 2, \ldots, n) \tag{3.4}$$

and the no correlation between observations assumption as:

$$E(\varepsilon_i \, \varepsilon_j \mid \mathbf{X}) = 0, \;\; (i, j = 1, 2, \ldots, n; \; i \neq j). \tag{3.5}$$

In combination with strict exogeneity, the spherical error variance assumption is equivalent to:

$$\mathrm{Var}(\varepsilon_i \mid \mathbf{X}) = \mathbf{E}(\boldsymbol{\varepsilon_i}^{\mathbf{2}} \mid \mathbf{X}) - \mathbf{E}(\boldsymbol{\varepsilon_i} \mid \mathbf{X})^{\mathbf{2}} \underset{(\mathbf{3.3})}{=} \mathbf{E}(\boldsymbol{\varepsilon_i}^{\mathbf{2}} \mid \mathbf{X}) \underset{(\mathbf{3.4})}{=} \boldsymbol{\sigma}^{\mathbf{2}} > \mathbf{0}$$

and

$$\mathrm{Cov}(\varepsilon_i, \varepsilon_j \mid \mathbf{X}) = \mathbf{E}(\boldsymbol{\varepsilon_i} \, \boldsymbol{\varepsilon_j} \mid \mathbf{X}) - \mathbf{E}(\boldsymbol{\varepsilon_i} \mid \mathbf{X}) \mathbf{E}(\boldsymbol{\varepsilon_j} \mid \mathbf{X}) \underset{(\mathbf{3.3})}{=} \mathbf{E}(\boldsymbol{\varepsilon_i} \, \boldsymbol{\varepsilon_j} \mid \mathbf{X}) \underset{(\mathbf{3.5})}{=} \mathbf{0}$$

**Normality of Error Term**
The final assumption that needs to be satisfied is that the error $\varepsilon$ conditional on $\mathbf{X}$ is jointly normal distributed. Recall from probability theory that the normal distribution depends just on the mean and the variance. From the strict exogeneity and spherical error variance assumptions we already know that $E(\varepsilon_i \mid \mathbf{X}) = \mathbf{0}$ and $\mathrm{Var}(\varepsilon_i \mid \mathbf{X}) = \boldsymbol{\sigma}^{\mathbf{2}}$ for $i = 1, 2, \ldots, n$ which implies that:

$$\boldsymbol{\varepsilon} \mid \mathbf{X} \sim \mathbf{N}(\mathbf{0}, \; \boldsymbol{\sigma}^{\mathbf{2}} \, \mathbf{I_n}), \tag{3.6}$$

where $\varepsilon$ is the random error vector.

### 3.1.2 The Least Squares Estimator

If we want to model a situation using linear regression we need an estimate for the regression coefficients. To estimate the regression coefficients the least squares principle will be applied. The goal of the least squares principle is to find an estimator $\hat{\boldsymbol{\beta}}$ for the coefficient vector $\boldsymbol{\beta}$ that minimizes the sum of squared residuals (SSR). Here the residual is the difference between the actual response $y_i$ and their predicted response $\hat{y}_i$ $(= \mathbf{x}_i' \boldsymbol{\beta})$. The ordinary least squares estimator is defined as follows

$$\hat{\boldsymbol{\beta}} = \underset{\boldsymbol{\beta}}{argmin} \; SSR(\boldsymbol{\beta}) = \underset{\boldsymbol{\beta}}{argmin} \sum_{i=1}^{n} (y_i - \mathbf{x}_i' \boldsymbol{\beta})^2 = \underset{\boldsymbol{\beta}}{argmin} (\mathbf{y} - \mathbf{X}\boldsymbol{\beta})' (\mathbf{y} - \mathbf{X}\boldsymbol{\beta}). \tag{3.7}$$

Observe that the final term can be rewritten as

$$(\mathbf{y}-\mathbf{X}\boldsymbol{\beta})'(\mathbf{y}-\mathbf{X}\boldsymbol{\beta}) = \mathbf{y}'\mathbf{y}-\boldsymbol{\beta}'\mathbf{X}'\mathbf{y}-\mathbf{y}'\mathbf{X}\boldsymbol{\beta}+\boldsymbol{\beta}'\mathbf{X}'\mathbf{X}\boldsymbol{\beta} = \mathbf{y}'\mathbf{y}-\mathbf{2}\,\mathbf{y}'\mathbf{X}\boldsymbol{\beta}+\boldsymbol{\beta}'\mathbf{X}'\mathbf{X}\boldsymbol{\beta} = \mathbf{y}'\mathbf{y}-\mathbf{2}\,\mathbf{a}'\boldsymbol{\beta}+\boldsymbol{\beta}'\mathbf{A}\boldsymbol{\beta},$$

where $\mathbf{a} = \mathbf{X}'\mathbf{y}$ and $\mathbf{A} = \mathbf{X}'\mathbf{X}$. The estimator that minimizes *SSR* is obtained by differentiating *SSR* with respect to $\boldsymbol{\beta}$ and solving the system

$$\frac{\partial SSR\,(\boldsymbol{\beta})}{\partial\,\boldsymbol{\beta}} = 0.$$

Recall from linear algebra [16] that for a vector $\mathbf{a}$ and a symmetric matrix $\mathbf{A}$

$$\frac{\partial\,\mathbf{a}'\boldsymbol{\beta}}{\boldsymbol{\beta}} = \mathbf{A} \text{ and } \frac{\partial\,\boldsymbol{\beta}'\mathbf{A}\boldsymbol{\beta}}{\boldsymbol{\beta}} = \mathbf{2A}\boldsymbol{\beta}.$$

From this follows that

$$\frac{\partial SSR\,(\boldsymbol{\beta})}{\partial\,\boldsymbol{\beta}} = \frac{\partial}{\partial\,\boldsymbol{\beta}}(\mathbf{y}'\mathbf{y}-\mathbf{2}\,\mathbf{a}'\boldsymbol{\beta}+\boldsymbol{\beta}'\mathbf{A}\boldsymbol{\beta}) = -\mathbf{2}\,\mathbf{a}+\mathbf{2A}\boldsymbol{\beta} = \mathbf{2}(\mathbf{A}\boldsymbol{\beta}-\mathbf{a}) = \mathbf{2}(\mathbf{X}'\mathbf{X}\boldsymbol{\beta}-\mathbf{X}'\mathbf{y}) = \mathbf{0}.$$

Hence we conclude

$$\mathbf{X}'\mathbf{X}\boldsymbol{\beta} = \mathbf{X}'\mathbf{y}. \tag{3.8}$$

(3.8) are called the normal equations. To solve these equations and obtain the least squares estimate of $\boldsymbol{\beta}$ both sides need to be multiplied by the inverse of $\mathbf{X}'\mathbf{X}$. This inverse only exists if the matrix is non-singular. By the no multicollinearity assumption the matrix $\mathbf{A} = \mathbf{X}'\mathbf{X}$ is positive definite which implies that the matrix is non-singular. Therefore the ordinary least squares estimator of $\boldsymbol{\beta}$ is given by

$$\hat{\boldsymbol{\beta}} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y} \tag{3.9}$$

By checking the second derivative of *SSR* with respect to $\boldsymbol{\beta}$ it can be easily verified that $\hat{\boldsymbol{\beta}}$ indeed minimizes *SSR*.

The vector of fitted values for the multiple linear regression is $\hat{\mathbf{y}} = \mathbf{X}\hat{\boldsymbol{\beta}} = \mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y} = \mathbf{H}\mathbf{y}$, where $\mathbf{H}$ is the so-called hat matrix. Thus, the vector of regression residuals is

$$\mathbf{e} = \mathbf{y}-\hat{\mathbf{y}} = \mathbf{y}-\mathbf{X}\hat{\boldsymbol{\beta}} = \mathbf{y}-\mathbf{H}\mathbf{y} = (\mathbf{I_n}-\mathbf{H})\mathbf{y} = \mathbf{M}\mathbf{y}$$

with $\mathbf{M} = \mathbf{I_n}-\mathbf{H}$ the annihilator matrix.

### 3.1.3 Properties of the Least Squares Estimator

The least squares estimators $\hat{\boldsymbol{\beta}}$ also has a number of important properties. The unbiasedness of the estimators will be addressed first.

**Theorem 3.1.** *The least squares estimator $\hat{\boldsymbol{\beta}}$ is an unbiased estimator of the coefficient vector $\boldsymbol{\beta}$.*

*Proof.*

$$\begin{aligned}
E(\hat{\boldsymbol{\beta}} \mid \mathbf{X}) &= E((\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y} \mid \mathbf{X}) \\
&= E((\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'(\mathbf{X}\boldsymbol{\beta}+\boldsymbol{\varepsilon}) \mid \mathbf{X}) \\
&= E((\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{X}\boldsymbol{\beta} + (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\boldsymbol{\varepsilon} \mid \mathbf{X}) \\
&= E(\boldsymbol{\beta} \mid \mathbf{X}) + ((\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}')\mathbf{E}(\boldsymbol{\varepsilon} \mid \mathbf{X}) \\
&\underset{(3.3)}{=} E(\boldsymbol{\beta} \mid \mathbf{X}) + \mathbf{0} = \boldsymbol{\beta} \\
&\text{Hence }, E(\hat{\boldsymbol{\beta}} \mid \mathbf{X}) = \boldsymbol{\beta}.
\end{aligned}$$

□

Another important property of the least squares estimator $\hat{\boldsymbol{\beta}}$ is the variance which can be calculated using the model assumptions.

**Theorem 3.2.** *The variance of* $\hat{\boldsymbol{\beta}}$ *equals* $\sigma^2 \cdot (\mathbf{X}'\mathbf{X})^{-1}$.

*Proof.*

$$\begin{aligned}
Var(\hat{\boldsymbol{\beta}} \mid \mathbf{X}) &= Var((\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y} \mid \mathbf{X}) \\
&= ((\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}')\mathbf{Var}(\mathbf{y} \mid \mathbf{X})((\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}')' \\
&= ((\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}')\mathbf{Var}(\mathbf{X}\boldsymbol{\beta}+\boldsymbol{\varepsilon} \mid \mathbf{X})((\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}')' \\
&= ((\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}')\mathbf{Var}(\boldsymbol{\varepsilon} \mid \mathbf{X})((\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}')' \\
&\underset{(3.6)}{=} ((\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}')(\sigma^2\,\mathbf{I_n})((\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}')' \\
&= \sigma^2\,((\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}) \\
&= \sigma^2 \cdot (\mathbf{X}'\mathbf{X})^{-1} \\
&\text{Hence, } Var(\hat{\boldsymbol{\beta}}) = \sigma^2 \cdot (\mathbf{X}'\mathbf{X})^{-1}.
\end{aligned}$$

□

### 3.1.4 Estimation of $\sigma^2$

Theorem 3.2 shows that the variance of the error term in the simple linear regression model $\sigma^2$ is included in the variances of $\hat{\boldsymbol{\beta}}$. Moreover $\sigma^2$ is needed to build interval estimates for the regression model and perform hypothesis testing. Unfortunately the variance of the error term is unknown which means that this also requires an estimate. The most common estimator of $\sigma^2$ is the residual mean square $s^2$

$$s^2 = \frac{SSR}{n-k} = \frac{\mathbf{e}'\mathbf{e}}{n-k}, \tag{3.10}$$

where $n-k$ are the degrees of freedom with $n$ the number of observations and $k$ the number of estimated parameters. To prove that $s^2$ is an unbiased estimator of $\sigma^2$ it will be shown that $E(\mathbf{e}'\mathbf{e}|\mathbf{X}) = (\mathbf{n}-\mathbf{k})\sigma^2$.

**Theorem 3.3.** *The residual mean square estimator $s^2$ is an unbiased estimator of the error variance $\sigma^2$*

*Proof.* First notice that the term $\mathbf{e}'\mathbf{e}$ can be rewritten as

$$\begin{aligned}
\mathbf{e}'\mathbf{e} &= ((\mathbf{I_n}-\mathbf{H})\mathbf{y})'(\mathbf{I_n}-\mathbf{H})\mathbf{y} \\
&= \mathbf{y}'(\mathbf{I_n}-\mathbf{H})'(\mathbf{I_n}-\mathbf{H})\mathbf{y} \\
&= \mathbf{y}'(\mathbf{I_n}-\mathbf{H})\mathbf{y} \\
&= (\mathbf{X\beta}+\boldsymbol{\varepsilon})'(\mathbf{I_n}-\mathbf{H})(\mathbf{X\beta}+\boldsymbol{\varepsilon}) \\
&= (\boldsymbol{\beta}'\mathbf{X}'+\boldsymbol{\varepsilon}')(\mathbf{I_n}-\mathbf{H})(\mathbf{X\beta}+\boldsymbol{\varepsilon}) \\
&= \boldsymbol{\varepsilon}'(\mathbf{I_n}-\mathbf{H})\boldsymbol{\varepsilon}+\boldsymbol{\beta}'\mathbf{X}'(\mathbf{I_n}-\mathbf{H})\mathbf{X\beta} \\
&= \boldsymbol{\varepsilon}'(\mathbf{I_n}-\mathbf{H})\boldsymbol{\varepsilon}+\boldsymbol{\beta}'\mathbf{X}'\mathbf{I_n}-\boldsymbol{\beta}'\mathbf{X}'\mathbf{X}(\mathbf{X}'\mathbf{X})^{-\mathbf{1}}\mathbf{X}'+\mathbf{I_n}\mathbf{X\beta}-\mathbf{X}(\mathbf{X}'\mathbf{X})^{-\mathbf{1}}\mathbf{X}'\mathbf{X\beta} \\
&= \boldsymbol{\varepsilon}'(\mathbf{I_n}-\mathbf{H})\boldsymbol{\varepsilon}+\boldsymbol{\beta}'\mathbf{X}'-\boldsymbol{\beta}'\mathbf{X}'+\mathbf{X\beta}-\mathbf{X\beta} \\
&= \boldsymbol{\varepsilon}'(\mathbf{I_n}-\mathbf{H})\boldsymbol{\varepsilon} = \boldsymbol{\varepsilon}'\mathbf{M}\boldsymbol{\varepsilon}.
\end{aligned}$$

Before we can calculate the expectation of $\mathbf{e}'\mathbf{e}$ we first need to show that

$$\begin{aligned}
\text{trace}(\mathbf{M}) &= \text{trace}(\mathbf{I_n}-\mathbf{H}) \\
&= \text{trace}(\mathbf{I_n})-\text{trace}(\mathbf{H}) \\
&= n-\text{trace}(\mathbf{H}) \\
&= n-\text{trace}(\mathbf{X}(\mathbf{X}'\mathbf{X})^{-\mathbf{1}}\mathbf{X}') \\
&= n-\text{trace}(\mathbf{X}'\mathbf{X}(\mathbf{X}'\mathbf{X})^{-\mathbf{1}}) \;\; (\text{since trace}(\mathbf{AB})=\text{trace}(\mathbf{BA})) \\
&= n-\text{trace}(\mathbf{I_k}) = n-k. \\
&\text{So trace}(\mathbf{M}) = n-k.
\end{aligned}$$

This property will be used to evaluate the expectation

$$\begin{aligned}
E(\mathbf{e}'\mathbf{e} \mid \mathbf{X}) &= E(\boldsymbol{\varepsilon}'\mathbf{M}\boldsymbol{\varepsilon} \mid \mathbf{X}) \\
&= \sum_{i=1}^{n}\sum_{j=1}^{n} m_{ij}\, E(\varepsilon_i\varepsilon_j \mid \mathbf{X}) \\
&\underset{(3.4),\,(3.5)}{=} \sum_{i=1}^{n} m_{ii}\,\sigma^2 \\
&= \sigma^2 \sum_{i=1}^{n} m_{ii} \\
&= \sigma^2\cdot\text{trace}(\mathbf{M}) = (n-k)\sigma^2.
\end{aligned}$$

Hence we conclude that $E(s^2) = E\left(\dfrac{\mathbf{e}'\mathbf{e}}{n-k}\right) = \dfrac{1}{n-k}E(\mathbf{e}'\mathbf{e}) = \dfrac{1}{n-k}(n-k)\sigma^2 = \sigma^2$. □

### 3.1.5 Hypothesis Testing

Incorporating unimportant regressors may make the model less reliable. Therefore one might want to check the significance of the regressors. For the multiple linear regression model it is possible to

test hypotheses about individual regressors as well as hypotheses about multiple regressors. In this section the t test will be discussed to evaluate hypotheses about individual regressors.

**t Tests**

The hypothesis for testing the significance of the jth regression coefficient is

$$H_0 : \beta_j = \beta_{j0},\ H_1 : \beta_j \neq \beta_{j0}, \tag{3.11}$$

where $\beta_{j0}$ is a constant. In appendix A it is proven that under the null hypothesis $H_0 : \beta_j = \beta_{j0}$ the statistic $Z_j$ that is given by

$$Z_j = \frac{\hat{\beta}_j - \beta_{j0}}{\sqrt{\sigma^2((\mathbf{X}'\mathbf{X})^{-1})_{jj}}},$$

is standard normal distributed by the assumptions. Here $((\mathbf{X}'\mathbf{X})^{-1})_{jj}$ is the diagonal element of the matrix $(\mathbf{X}'\mathbf{X})^{-1}$ corresponding to $\hat{\beta}_j$. Since $\sigma^2$ is often unknown, we define a new statistic $t_j$ that uses the estimate $s^2$

$$t_j = \frac{\hat{\beta}_j - \beta_{j0}}{se(\hat{\beta}_j)},$$

where $se(\hat{\beta}_j)$ is the standard error of $\hat{\beta}_j$. The standard error of $\hat{\beta}_j$ is defined as

$$se(\hat{\beta}_j) = \sqrt{s^2 \cdot ((\mathbf{X}'\mathbf{X})^{-1})_{jj}} \tag{3.12}$$

The new statistic $t_j$ can be rewritten as

$$t_j = \frac{\hat{\beta}_j - \beta_{j0}}{\sqrt{s^2\sigma^2(n-k)/\sigma^2(n-k)((\mathbf{X}'\mathbf{X})^{-1})_{jj}}} = \frac{\frac{\hat{\beta}_j - \beta_{j0}}{\sqrt{\sigma^2/((\mathbf{X}'\mathbf{X})^{-1})_{jj}}}}{\sqrt{\frac{s^2(n-k)}{\sigma^2}/(n-k)}} = \frac{Z_j}{\sqrt{\frac{s^2(n-k)}{\sigma^2}/(n-k)}}.$$

In appendix A it is proven that under the null hypothesis $t_j$ follows a Student's t-distribution with $n-k$ degrees of freedom. The statistic $t_j$ is used to test hypothesis of the form (3.11). $H_0$ is rejected if

$$|t_j| > t_{\alpha/2,\ n-k},$$

where $t_{\alpha/2,\ n-k}$ is the upper tail of the $\alpha/2$ percentage point of the $t_{n-k}$ distribution. The p-value can also be used to express the decision rule of the t-test. One can calculate the p-value using the expression $p = P(|t_j| > t_{\alpha/2,\ n-k}) \times 2$. The null hypothesis is rejected if $p \leq \alpha$ and failed to reject otherwise.

### 3.1.6 Confidence Intervals

It is often useful to not only consider the point estimates of the coefficients but also their confidence intervals. The frequentist interpretation of the $100(1-\alpha)$ percent confidence interval is that repeating the experiment will lead to $100(1-\alpha)\%$ of the confidence intervals containing the parameter's true value. The $100(1-\alpha)$ percent confidence interval of $\beta_j$ can be constructed using the previously defined t-statistic. Recall that for hypothesis of the form (3.11) we fail to reject $H_0$ if

$$-t_{\alpha/2,\ n-k} \leq \frac{\hat{\beta}_j - \beta_{j0}}{se(\hat{\beta}_j)} \leq t_{\alpha/2,\ n-k}.$$

Hence the $100(1-\alpha)$ confidence interval of $\beta_j$ is given by

$$\hat{\beta}_j - t_{\alpha/2,\, n-k} \cdot se(\hat{\beta}_j) \leq \beta_{j0} \leq \hat{\beta}_j + t_{\alpha/2,\, n-k} \cdot se(\hat{\beta}_j)$$

or in interval notation

$$\left[\hat{\beta}_j - t_{\alpha/2,\, n-k} \cdot se(\hat{\beta}_j),\ \hat{\beta}_j + t_{\alpha/2,\, n-k} \cdot se(\hat{\beta}_j)\right]. \tag{3.13}$$

# 4 The Bayesian Approach to Linear Regression

So far, we have focused on the Frequentist approach to linear regression, which assumes the parameters to be fixed unknown constants. A different method is to view these parameters as random variables. This procedure is known as the Bayesian approach to linear regression. As a result of this change of perspective it is possible to include prior knowledge into the model. The distinction between Bayesian and Frequentist regression becomes clear using Bayes' theorem.

## 4.1 Bayes' Theorem

Bayesian inference is based on the distribution of the parameter vector $\boldsymbol{\theta}$ conditioning on the observed data $\mathbf{y}$. This conditional distribution is also known as the posterior distribution of $\boldsymbol{\theta}$. The fundamental law of Bayesian regression that establishes a method to calculate this posterior distribution is Bayes' theorem [8, 12]

$$P(\boldsymbol{\theta}|\mathbf{y}) = \frac{P(\mathbf{y}|\boldsymbol{\theta})\ P(\boldsymbol{\theta})}{P(\mathbf{y})}. \tag{4.1}$$

$P(\boldsymbol{\theta})$ is the prior distribution and represents the prior beliefs about $\boldsymbol{\theta}$. $P(\mathbf{y}|\boldsymbol{\theta})$ and $P(\mathbf{y})$ are the likelihood function and the marginal density of $\mathbf{y}$, respectively. By the law of total probability the marginal density of $\mathbf{y}$ is equivalent to

$$P(\mathbf{y}) = \sum P(\mathbf{y}|\boldsymbol{\theta})P(\boldsymbol{\theta}) \text{ in the discrete case}$$

or

$$P(\mathbf{y}) = \int P(\mathbf{y}|\boldsymbol{\theta})P(\boldsymbol{\theta})\ d\boldsymbol{\theta} \text{ in the continuous case.}$$

Since $P(\mathbf{y})$ does not dependent on $\boldsymbol{\theta}$, it may be regarded as a normalizing constant ($c = P(\mathbf{y})$). Hence we can simplify equation (4.1) to

$$P(\boldsymbol{\theta}|\mathbf{y}) \propto P(\mathbf{y}|\boldsymbol{\theta}) \times P(\boldsymbol{\theta}) \tag{4.2}$$

which is equivalent to stating

$$\text{Posterior} \propto \text{Likelihood} \times \text{Prior}.$$

The $\propto$ symbol indicates that the posterior distribution is proportional to the right hand side of the equation. Before we use equation (4.2) to perform Bayesian regression analysis we will first take a closer look at the prior distribution.

## 4.2 The Prior Distribution

In order to perform Bayesian analysis a prior distribution for the unknown parameter needs to be specified. Even if one does not want to incorporate prior knowledge into the model the prior distribution still needs to be specified. The various types of priors can roughly be divided into two categories: informative and uninformative priors.

For most studies there is some prior information accessible. An informative prior tries to express this prior knowledge. One could for example base the prior distribution for a new study on similar Bayesian studies from the past or expert knowledge. Furthermore one might already rule out certain parameter values based on prior beliefs.

An uninformative prior or reference prior is used when there is insufficient prior knowledge available or when one wants to keep the analysis as objective as possible. Critics of the Bayesian approach argue that the latter is impossible because the inclusion of the prior causes the analysis to always be subjective. An uninformative prior assumes all the possible parameter values to have the same probability. A frequently used uninformative prior is the uniform distribution which is also referred to as the flat prior.

The parameter space is frequently very large, especially for problems with many parameters. As a consequence of this evaluating equation (4.1) can become computationally heavy and rather complex. A factor that affects this complexity is the prior specification. Therefore one can sometimes choose a prior that facilitate the computations. An example of this is a class of priors called conjugate priors. Conjugate priors are priors that given a certain likelihood function lead to posteriors of the same family. Those priors can be informative as well as uninformative. In the next subsection we will consider an example of a conjugate prior. Unfortunately it is often not possible to simplify the computations using conjugate priors so in those cases other methods are needed to compute the posterior distribution.

## 4.3 Multiple Linear Regression

In this section we will focus on the multiple linear regression model as described in section 3.1 and the corresponding Bayesian approach to estimate the unknown parameters. Most of the results in this section will be based on the books by Rachev et al and Hoff [23, 12]. Recall that the assumptions of the multiple linear regression model imply that the conditional probability of $\mathbf{y}$ follows a multivariate normal distribution

$$\{\mathbf{y}|\mathbf{X}\} \sim N(\mathbf{X}\boldsymbol{\beta},\ \sigma^2\mathbf{I}_n). \tag{4.3}$$

This is the same as saying that the likelihood function is given by

$$p(\mathbf{y}|\mathbf{X},\boldsymbol{\beta},\sigma^2) = (2\pi\sigma^2)^{-\frac{n}{2}} exp\left\{-\frac{1}{2\sigma^2}(\mathbf{y}-\mathbf{X}\boldsymbol{\beta})'(\mathbf{y}-\mathbf{X}\boldsymbol{\beta})\right\} \propto (\sigma^2)^{-\frac{n}{2}} exp\left\{-\frac{1}{2\sigma^2}RSS(\boldsymbol{\beta})\right\}. \tag{4.4}$$

To obtain the posterior distribution and do Bayesian regression analysis we still have to specify a prior distribution. For the prior distribution two different options will be proposed: an uninformative and an informative prior.
Montgomery

### 4.3.1 Uninformative Prior Distribution

A frequently used uninformative joint prior for the parameters $\boldsymbol{\beta}$ and $\sigma^2$ follows from Jeffreys multiparameter rule [17]. It is the product of a flat prior on $\boldsymbol{\beta}$ and $log(\sigma^2)$. Hence it is proportional to

$$p(\boldsymbol{\beta},\sigma^2) \propto \sigma^{-2}. \tag{4.5}$$

Observe that this prior does not integrate to one and therefore is not a probability density function. Such a prior is called an improper prior. Improper priors might cause issues in the estimation process or at another point of the analysis. This prior is often chosen because it results in a relatively simple posterior distribution. The posterior distributions of $\boldsymbol{\beta}$ and $\sigma^2$ can be calculated using Bayes theorem

$$
\begin{aligned}
p(\boldsymbol{\beta},\sigma^2|\mathbf{y},\mathbf{X}) &\underset{(\mathbf{4.2})}{\propto} \mathbf{p}(\mathbf{y}|\mathbf{X},\boldsymbol{\beta},\boldsymbol{\sigma^2}) \times \mathbf{p}(\boldsymbol{\beta},\boldsymbol{\sigma^2}) \\
&\propto \left[(\sigma^2)^{-\frac{n}{2}} exp\left\{-\frac{1}{2\sigma^2}(\mathbf{y}-\mathbf{X}\boldsymbol{\beta})'(\mathbf{y}-\mathbf{X}\boldsymbol{\beta})\right\}\right] \times \frac{1}{\sigma^2} \\
&= (\sigma^2)^{-\frac{n+2}{2}} exp\left\{-\frac{1}{2\sigma^2}(\mathbf{y}-\mathbf{X}\boldsymbol{\beta})'(\mathbf{y}-\mathbf{X}\boldsymbol{\beta})\right\} \\
&= (\sigma^2)^{-\frac{n+2}{2}} exp\left\{-\frac{1}{2\sigma^2}((\boldsymbol{\beta}-\hat{\boldsymbol{\beta}})'\mathbf{X}'\mathbf{X}(\boldsymbol{\beta}-\hat{\boldsymbol{\beta}})+\mathbf{SSE})\right\} \\
&= \left[(\sigma^2)^{-(\frac{n-k}{2}+1)} exp\left\{-\frac{1}{2\sigma^2}SSE\right\}\right] \times \left[(\sigma^2)^{-\frac{k}{2}} exp\left\{-\frac{1}{2\sigma^2}(\boldsymbol{\beta}-\hat{\boldsymbol{\beta}})'\mathbf{X}'\mathbf{X}(\boldsymbol{\beta}-\hat{\boldsymbol{\beta}})\right\}\right] \\
&= \left[(\sigma^2)^{-(\frac{n-k}{2}+1)} exp\left\{-\frac{1}{2\sigma^2}(n-k)s^2\right\}\right] \times \left[(\sigma^2)^{-\frac{k}{2}} exp\left\{-\frac{1}{2\sigma^2}(\boldsymbol{\beta}-\hat{\boldsymbol{\beta}})'\mathbf{X}'\mathbf{X}(\boldsymbol{\beta}-\hat{\boldsymbol{\beta}})\right\}\right].
\end{aligned}
$$

Recall that by the definition of conditional probability

$$p(\boldsymbol{\beta},\sigma^2|\mathbf{y},\mathbf{X}) = \mathbf{p}(\boldsymbol{\beta}|\mathbf{y},\mathbf{X},\boldsymbol{\sigma^2})\mathbf{p}(\boldsymbol{\sigma^2}|\mathbf{y},\mathbf{X}).$$

Hence we can conclude that

$$\{\boldsymbol{\beta}|\mathbf{y},\mathbf{X},\boldsymbol{\sigma^2}\} \sim \mathrm{N}(\hat{\boldsymbol{\beta}},\boldsymbol{\sigma^2}(\mathbf{X}'\mathbf{X})^{-\mathbf{1}}) \tag{4.6}$$

and

$$\{\sigma^2|\mathbf{y},\mathbf{X}\} \sim \text{Inv-}\chi^2(n-k,\ s^2), \tag{4.7}$$

where $s^2$ is the mean squared error, $\hat{\boldsymbol{\beta}}$ the least squares estimator of $\boldsymbol{\beta}$ and $\sigma^2(\mathbf{X}'\mathbf{X})^{-\mathbf{1}}$ the variance of $\hat{\boldsymbol{\beta}}$. As a result, we may expect a similar outcome as with the Frequentist method, but then with a Bayesian interpretation. The posterior distribution of $\boldsymbol{\beta}$ depends on the variance $\sigma^2$. To get the marginal posterior distribution of $\boldsymbol{\beta}$ that is independent of $\sigma^2$ we must integrate $\sigma^2$ out of the joint posterior distribution.

$$
\begin{aligned}
p(\boldsymbol{\beta}|\mathbf{y},\mathbf{X}) &= \int_0^\infty p(\boldsymbol{\beta},\sigma^2|\mathbf{y},\mathbf{X})\ \mathbf{d\sigma^2} \\
&\propto \int_0^\infty (\sigma^2)^{-\frac{n+2}{2}} exp\left\{-\frac{1}{2\sigma^2}((\boldsymbol{\beta}-\hat{\boldsymbol{\beta}})'\mathbf{X}'\mathbf{X}(\boldsymbol{\beta}-\hat{\boldsymbol{\beta}})+\mathbf{SSE})\right\}\ d\sigma^2.
\end{aligned}
$$

To proceed we do a change of variables $\gamma = 1/\sigma^2$

$$p(\boldsymbol{\beta}|\mathbf{y},\mathbf{X}) \propto \int_0^\infty \gamma^{\frac{n-2}{2}} exp\left\{-\frac{\gamma}{2}((\boldsymbol{\beta}-\hat{\boldsymbol{\beta}})'\mathbf{X}'\mathbf{X}(\boldsymbol{\beta}-\hat{\boldsymbol{\beta}})+\mathbf{SSE})\right\}\ d\gamma.$$

We then do another change of variables $u = -\frac{\gamma}{2}((\boldsymbol{\beta}-\hat{\boldsymbol{\beta}})'\mathbf{X}'\mathbf{X}(\boldsymbol{\beta}-\hat{\boldsymbol{\beta}})+\mathbf{SSE})$

$$
\begin{aligned}
p(\boldsymbol{\beta}|\mathbf{y},\mathbf{X}) &\propto ((\boldsymbol{\beta}-\hat{\boldsymbol{\beta}})'\mathbf{X}'\mathbf{X}(\boldsymbol{\beta}-\hat{\boldsymbol{\beta}})+\mathbf{SSE})^{-\frac{\mathbf{n}}{\mathbf{2}}}\int_{\mathbf{0}}^{\infty}\mathbf{u}^{\frac{\mathbf{n-2}}{\mathbf{2}}}\mathbf{e}^{-\mathbf{u}}\,\mathbf{du}\\
&= ((\boldsymbol{\beta}-\hat{\boldsymbol{\beta}})'\mathbf{X}'\mathbf{X}(\boldsymbol{\beta}-\hat{\boldsymbol{\beta}})+\mathbf{SSE})^{-\frac{\mathbf{n}}{\mathbf{2}}}\Gamma(\tfrac{\mathbf{n}}{\mathbf{2}})\\
&\propto ((\boldsymbol{\beta}-\hat{\boldsymbol{\beta}})'\mathbf{X}'\mathbf{X}(\boldsymbol{\beta}-\hat{\boldsymbol{\beta}})+\mathbf{SSE})^{-\frac{\mathbf{n}}{\mathbf{2}}}\\
&\propto \left[1+\frac{1}{n-k}\frac{(\boldsymbol{\beta}-\hat{\boldsymbol{\beta}})'\mathbf{X}'\mathbf{X}(\boldsymbol{\beta}-\hat{\boldsymbol{\beta}})}{SSE/n-k}\right]^{-\frac{n}{2}}\\
&= \left[1+\frac{(\boldsymbol{\beta}-\hat{\boldsymbol{\beta}})'\mathbf{X}'\mathbf{X}(\boldsymbol{\beta}-\hat{\boldsymbol{\beta}})}{(n-k)s^2}\right]^{-\frac{n}{2}}.
\end{aligned}
$$

Hence we can conclude that

$$\{\boldsymbol{\beta}|\mathbf{y},\mathbf{X}\}\sim T_{n-k}(\hat{\boldsymbol{\beta}},s^2(\mathbf{X}'\mathbf{X})^{-1}). \tag{4.8}$$

### 4.3.2 Informative Prior Distribution

In a situation where there is prior knowledge available we can use a conjugate prior distribution for the parameters. Since we assumed the likelihood function to be a multivariate normal distribution we can show that the conjugate priors are

$$\boldsymbol{\beta}\sim N(\boldsymbol{\beta}_0,\Sigma_0) \tag{4.9}$$

and

$$\gamma\sim\Gamma(\tfrac{\nu_0}{2},\ \tfrac{\nu_0\sigma_0^2}{2}), \tag{4.10}$$

with $\gamma=1/\sigma^2$. The hyperparameters $\boldsymbol{\beta}_0,\Sigma_0$ ,$\nu_0$ and $\sigma_0^2$ need to be chosen in advance. Here $\Sigma_0$ is the covariance matrix that contains the covariance between each pair of $\beta_i$'s. Under the conjugate prior the posterior distribution of $\boldsymbol{\beta}$ is

$$
\begin{aligned}
p(\boldsymbol{\beta}\mid\mathbf{y},\mathbf{X},\boldsymbol{\sigma^2}) &\propto p(\mathbf{y}|\mathbf{X},\boldsymbol{\beta},\boldsymbol{\sigma^2})\times\mathbf{p}(\boldsymbol{\beta})\\
&\propto \left[exp\left\{-\frac{1}{2}(-2\boldsymbol{\beta}'\mathbf{X}'\mathbf{y}/\boldsymbol{\sigma^2}+\boldsymbol{\beta}'\mathbf{X}'\mathbf{X}\boldsymbol{\beta}/\boldsymbol{\sigma^2})\right\}\right]\times\left[exp\left\{-\frac{1}{2}(-2\boldsymbol{\beta}'\Sigma_0^{-1}\boldsymbol{\beta_0}+\boldsymbol{\beta}'\Sigma_0^{-1}\boldsymbol{\beta})\right\}\right]\\
&= exp\left\{\boldsymbol{\beta}'(\Sigma_0^{-1}\boldsymbol{\beta_0}+\mathbf{X}'\mathbf{y}/\boldsymbol{\sigma^2})-\frac{\mathbf{1}}{\mathbf{2}}\boldsymbol{\beta}'(\Sigma_{\mathbf{0}}^{-\mathbf{1}}+\mathbf{X}'\mathbf{X}/\boldsymbol{\sigma^2})\boldsymbol{\beta})\right\},
\end{aligned}
$$

which is proportional to multivariate normal distribution. Therefore the posterior distribution of $\boldsymbol{\beta}$ is given by

$$\{\boldsymbol{\beta}\mid\mathbf{y},\mathbf{X},\boldsymbol{\sigma^2}\}\sim\mathbf{N}((\Sigma_{\mathbf{0}}^{-\mathbf{1}}+\mathbf{X}'\mathbf{X}/\boldsymbol{\sigma^2})^{-\mathbf{1}}(\Sigma_{\mathbf{0}}^{-\mathbf{1}}\boldsymbol{\beta_0}+\mathbf{X}'\mathbf{y}/\boldsymbol{\sigma^2}),\ (\Sigma_{\mathbf{0}}^{-\mathbf{1}}+\mathbf{X}'\mathbf{X}/\boldsymbol{\sigma^2})^{-\mathbf{1}}). \tag{4.11}$$

Using a similar approach, the posterior distribution of $\gamma$ may be derived as follows

$$
\begin{aligned}
p(\gamma\mid\mathbf{y},\mathbf{X},\boldsymbol{\beta}) &\propto p(\mathbf{y}|\mathbf{X},\boldsymbol{\beta},\gamma)\times\mathbf{p}(\gamma)\\
&\propto \left[\gamma^{-\frac{n}{2}}exp\left\{-\frac{1}{2\sigma^2}SSE(\boldsymbol{\beta})\right\}\right]\times\left[\gamma^{\frac{\nu_0}{2}-1}exp\left(-\gamma\times\tfrac{\nu_0\gamma_0}{2}\right)\right]\\
&= \gamma^{\frac{\nu_0+n}{2}-1}exp\left(-\gamma\times\tfrac{\nu_0\gamma_0+SSE(\boldsymbol{\beta})}{2}\right).
\end{aligned}
$$

Observe that this expression is proportional to a gamma density so the posterior distribution of $\sigma^2$ is

$$\{\sigma^2 \mid \mathbf{y}, \mathbf{X}, \boldsymbol{\beta}\} \sim \Gamma^{-1}((\nu_0 + \mathbf{n})/\mathbf{2},\ (\nu_0\sigma_0^2 + \mathbf{SSE}(\boldsymbol{\beta}))/\mathbf{2}). \tag{4.12}$$

Therefore the proposed prior distributions are indeed conjugate priors.

## 4.4 Markov Chain Monte Carlo Methods

Calculating the joint posterior distribution analytically may be challenging or even impossible. In this scenario simulation algorithms known as Markov Chain Monte Carlo (MCMC) methods can be used to get a posterior approximation. This section will briefly go over the basic notions behind MCMC methods before focusing on an MCMC procedure known as Gibbs sampling. In addition, we will investigate the convergence of the Markov chain Monte Carlo method.

### 4.4.1 Monte Carlo Simulation

Monte Carlo simulation [12] is a random sampling-based integration approach. Let $\theta$ be the parameter of interest and $y_1, \ldots, y_n$ a sample from the likelihood function $p(y_1, \ldots, y_n \mid \theta)$. Assume that we can directly draw from the posterior distribution to obtain the following sample

$$\theta^{(1)}, \ldots, \theta^{(M)} \underset{i.i.d}{\sim} p(\theta \mid y_1, \ldots, y_n).$$

This sample's empirical distribution is called the Monte Carlo approximation to the target distribution $p(\theta \mid y_1, \ldots, y_n)$ which improves as $M$ becomes larger. Sufficiently large independent Monte Carlo samples can be used to estimate interesting properties of the posterior distribution. One could for example calculate The expectation of any function of $\theta$ with the law of large numbers

$$\tfrac{1}{M} \sum_{m=1}^{M} g(\theta^{(m)}) \to E[g(\theta) \mid y_1, \ldots, y_n] \text{ as } M \to \infty. \tag{4.13}$$

Besides the expectation it is also possible to estimate other aspects of the posterior distribution such as the variance.

### 4.4.2 Markov Chain

As described in [29] a Markov chain is a sequence of discrete or continuous random variables $\theta^{(0)}, \theta^{(1)}, \ldots$ that satisfies the Markovian property

$$p(\theta^{(n+1)} \in A \mid \theta^{(n)} = x, \theta^{(n-1)} = x_{n-1}, \ldots, \theta^{(0)} = x_0) = p(\theta^{(n+1)} \in A \mid \theta^{(n)} = x)$$

for any subset $A$ of the state space $\mathcal{S}$ and all $n \geq 0$.

So the future state is solely determined by the current state and not by the past. The Markov chain is said to be homogeneous if

$$p(\theta^{(n+1)} \in A \mid \theta^{(n)} = x) = p(\theta^{(1)} \in A \mid \theta^{(0)} = x)\ \forall\ n \geq 0.$$

The likelihood of generating a value in $A$ while beginning in $x$ is described by the transition kernel density $p(x, y)$

$$p(\theta^{(1)} \in A \mid x) = \int_A p(x, y)\, dy.$$

A probability density function $\pi$ on $\mathcal{S}$ that satisfies

$$\pi(y) = \int_{\mathcal{S}} \pi(x)p(x,y)\,dx$$

is called the stationary distribution of the Markov chain on $\mathcal{S}$. Once you are sampling from the stationary distribution the next steps of the Markov chain are also sampled from the stationary distribution. The limiting distribution of the Markov chain can be linked to stationary distribution $\pi$ using the Ergodic theorem. For this theorem to hold the Markov chain needs to be irreducible, aperiodic and recurrent. A Markov chain is irreducible if there is a positive probability to go from any state to any other state in the state space in a finite number of steps. A state is periodic if it can only be visited after a regular number of steps. The Markov chain is called periodic if it contains at least one periodic state. Otherwise the Markov Chain is said to be aperiodic. Finally a Markov chain is recurrent if it visits every state an infinite number of times. If all three criteria are satisfied we can use the Ergodic theorem [12]

**Theorem 4.1.** *(Ergodic Theorem) If* $\{\theta^{(0)}, \theta^{(1)}, \dots\}$ *is an irreducible, aperiodic and recurrent Markov chain, then there is a unique probability distribution* $\pi$ *such that as* $n \to \infty$*,*

1. $p(\theta^{(n)} \in A) \to \pi(A)$ *for any set A;*
2. $\frac{1}{n}\sum g(\theta^{(n)}) \to \int g(\theta)\pi(\theta)\,dx = E(g(\theta))$.

The goal of Markov chain Monte Carlo methods is to build an irreducible, aperiodic Markov chain with a stationary distribution that equals the target distribution. An MCMC method starts with an initial guess after which the other elements of the Markov chain are generated using Monte Carlo techniques. There are two phases in the procedure of proposing a new sample:

1. A proposal for the new sample is obtained by adding a small perturbation to the old sample
2. A decision rule is applied to either accept or reject the new sample. If accepted, the old sample will be replaced with the new sample. Otherwise, the old sample will be kept [28].

### 4.4.3 Gibbs Sampling

Gibbs sampling is a well-known MCMC technique that can be applied when one is able to sample directly from all the full conditional posterior distributions of the parameters. If the parameter vector is $\boldsymbol{\theta} = (\theta_1, \theta_2, \dots, \theta_q)$ then the full conditional posterior distribution of $\theta_i$ $(i = 1, \dots, q)$ is given by

$$p(\theta_i \mid \theta_1, \dots, \theta_{i-1}, \theta_{i+1}, \dots \theta_q, \mathbf{y}) = p(\theta_i \mid \boldsymbol{\theta_{-i}}, \mathbf{y}).$$

The Gibbs sampling algorithm is as follows:

1. Choose an initial value for all the parameters $\theta_i^{(0)}$ $(i = 1, \dots, q)$ to initialise the Markov chain;
2. At iteration t, obtain $\boldsymbol{\theta^{(t)}} = (\theta_1^{(t)}, \theta_2^{(t)}, \dots, \theta_q^{(t)})$ as follows:
   - Draw an observation, $\theta_1^{(t)}$ from $p(\theta_1^{(t)} \mid \theta_2^{(t-1)}, \theta_3^{(t-1)}, \dots, \theta_q^{(t-1)}, \mathbf{y})$
   - Draw an observation, $\theta_2^{(t)}$ from $p(\theta_2^{(t)} \mid \theta_1^{(t)}, \theta_3^{(t-1)}, \dots, \theta_q^{(t-1)}, \mathbf{y})$
   - Cycle through the rest of the components, $\theta_3^{(t)}, \dots, \theta_q^{(t)}$, in a similar way

3. Repeat step (2) $B$ times until convergence is achieved;
4. Run step (2) $M$ more times to generate $\{\boldsymbol{\theta}^{(B+1)}, \ldots, \boldsymbol{\theta}^{(B+S)}\}$;
5. discard $\{\boldsymbol{\theta}^{(1)}, \ldots, \boldsymbol{\theta}^{(B)}\}$ and use the empirical distribution of $\{\boldsymbol{\theta}^{(B+1)}, \ldots, \boldsymbol{\theta}^{(B+M)}\}$ to approximate $p(\boldsymbol{\theta} \mid \mathbf{y})$.

The first $B$ iterations are known as the "burn-in" period which can be shortened by choosing an initial value closer to the true value. For Gibbs sampling the decision rule is to always accept the new sample [12, 23].

To approximate the joint posterior distribution of the coefficients we can use our knowledge about the Gibbs sampler. Note that under the uninformative prior we can sample from both $p(\boldsymbol{\beta}|\mathbf{y}, \mathbf{X}, \sigma^{\mathbf{2}})$ and $p(\sigma^2|\mathbf{y}, \mathbf{X})$. A sample $\{\boldsymbol{\beta}, \sigma^2\}$ from $p(\boldsymbol{\beta}, \sigma^2|\mathbf{y}, \mathbf{X})$ can be created as follows:

1. Sample $\sigma^2 \sim \text{Inv-}\chi^2(n-k,\ s^2)$;
2. Sample $\boldsymbol{\beta} \sim \text{N}(\hat{\boldsymbol{\beta}}, \sigma^2(\mathbf{X}'\mathbf{X})^{-\mathbf{1}})$, where $\sigma^2$ is the sample from step 1 [12].

The empirical distribution of the samples can be used to approximate the joint posterior distribution of the coefficients.

When it comes to the informative prior distribution we can sample from both the full conditional posterior distributions. Hence we are able to design a Gibbs sampler to approximate the joint posterior distribution $p(\boldsymbol{\beta}, \sigma^2|\mathbf{y}, \mathbf{X})$. Given the old sample $\{\boldsymbol{\beta}^{(s)}, \sigma^{2(s)}\}$, the new sample $\{\boldsymbol{\beta}^{(s+1)}, \sigma^{2(s+1)}\}$ can be generated by:

1. Updating $\boldsymbol{\beta}$
   (a) Compute $\mathbf{m} = \mathbf{E}[\boldsymbol{\beta} \mid \mathbf{y}, \mathbf{X}, \sigma^{\mathbf{2(s)}}]$ and $\mathbf{V} = \mathbf{Var}[\boldsymbol{\beta} \mid \mathbf{y}, \mathbf{X}, \sigma^{\mathbf{2(s)}}]$
   (b) Sample $\boldsymbol{\beta}^{(s+1)} \sim N(\mathbf{m}, \mathbf{V})$
2. Updating $\sigma^2$
   (a) Compute $SSE(\boldsymbol{\beta}^{(s+1)})$
   (b) Sample $\sigma^{2(s+1)} \sim \Gamma^{-1}((\nu_0 + n)/2,\ (\nu_0\sigma_0^2 + SSE(\boldsymbol{\beta}^{(s+1)}))/2)$ [12]

After a "burn-in" period the empirical distribution of the samples can again be used to approximate the joint posterior distribution of the coefficients.

### 4.4.4 Convergence Diagnostics

Before we can study the results of a Markov Chain Monte Carlo method we should first check if the Markov chain has reached its stationary distribution. If the Markov chain has not reached its stationary distribution the results of an MCMC method are not reliable. This section we will cover two frequently used graphical MCMC diagnostics that were discussed in [12, 17]: trace plots and autocorrelation plots.

Trace plots are the easiest way to check convergence of the Markov chain. A trace plot is a plot of the sample value against the iteration number. In general a trace plot is made for every parameter.

Some features indicating stationarity that can be found in the plot are an approximately constant mean and variance. It is frequently stated that a good trace plot should resemble a hairy caterpillar. If you observe a different pattern this demonstrates dependence of the chain on the initial state or another convergence problem.

The dependence between the different states is also an interesting factor that we can study. If the future states of the Markov Chain are highly correlated with the current state then the chain is said to have a low mixing rate. This mixing rate is determined by the autocorrelations of the lags. The autocorrelation of lag $k$ is characterized as the correlation between state $\theta^t$ and state $\theta^{t+k}$. The autocorrelation of the different lags can be plotted in an autocorrelation plot. If the autocorrelation decreases fast with increasing lag this indicates a high mixing rate. There exist converging Markov Chains with a low mixing rate and vice versa. Therefore an autocorrelation plot does not guarantee convergence of the Markov Chain. The only thing that a low mixing rate suggests is that there are more samples needed for a proper inference.

## 4.5 Credible Intervals and Bayesian P-values

For the frequentist approach we considered the confidence intervals and the p-values of the estimates. These concepts are defined differently for Bayesian parameter estimation methods. The Bayesian equivalent of confidence intervals are called credible intervals. The $100(1-\alpha)$ percent Credible interval can be interpreted as the interval that contains the true parameter value with $100(1-\alpha)$ percent certainty

$$P(l(y) < \beta < u(y) \mid Y = y) = 0.95, \tag{4.14}$$

where

$$P(\beta \leq l(y) \mid Y = y) = P(\beta \geq u(y) \mid Y = y) = 0.025 \text{ [12]}.$$

This view differs from the frequentist interpretation of confidence intervals. Instead of calculating the confidence intervals analytically we will in this thesis focus on the empirical credible intervals. If the number of samples is large enough the empirical credible intervals approach the analytical result.

A Bayesian alternative for the frequentist p-value is the probability of direction. The probability of direction is characterized as the probability that a parameter is strictly positive or negative. If the posterior distribution indicates that negative values are more probable the strictly negative interpretation is used. Otherwise we use the strictly positive interpretation. The probability of direction ranges between 0.5 and 1. To obtain the Bayesian equivalent of the frequentist p-value we can perform the following calculation

$$\text{p-value} = 2 \times (1 - \text{probability of direction})$$

The probability of direction is highly correlated with the frequentist p-value. However the interpretation of the two concepts is completely different. Hence the probability of direction cannot directly be used to say something about the significance [18].

# 5 Model Selection and Comparison

## 5.1 Cross-validation

In present-day statistics resampling methods play an essential role. One of the most utilized resampling methods is cross-validation [13]. The idea behind cross-validation is to split the data into two parts: a training and a validation set. The training set is used to fit a model, which is then used to predict the validation set's response. The measure of the model fit can then be used as an estimation of the model's prediction error. Cross-validation is frequently applied to identify possible model overfitting.

K-fold cross-validation is a popular technique to evaluate and compare models. This method randomly divides the data into k groups or folds of a roughly similar size. One of the folds is chosen as the validation set while the other folds form the training set. The training set is used to create a model that predicts the response of the validation set. After model fitting the mean squared error will be computed as an estimate of the prediction error. This process will be repeated k times until each fold has been used as the validation set. To obtain a better estimate of the prediction error we will use the mean of the mean squared errors

$$CV_k = \frac{1}{k}\sum_{i=1}^{k} MSE_i. \tag{5.1}$$

Two common choices for $k$ are $k = 5$ and $k = 10$. These values are chosen because of their computational advantage (larger values of $k$ are not optimal) and relatively low variability in the CV estimate.

## 5.2 Lasso

In general the least squares estimator has a small bias and a large variance. In some cases, raising the bias in exchange for a smaller variance might improve the model's prediction accuracy. One way of doing this is by fitting a sparse regression model. In a sparse regression model some of the regression coefficients are removed from the model. A possible technique that one could apply to achieve this is the least absolute shrinkage and selection operator also known as the lasso. The lasso, as the name implies, sets a few of the coefficients to zero while shrinking some of the others. This method was proposed by Tibshirani [27] and is defined as the vector $\hat{\boldsymbol{\beta}}_{\boldsymbol{lasso}}$ that minimizes the function

$$SSR(\boldsymbol{\beta};\, \lambda) = \sum_{i=1}^{n}(y_i - \mathbf{x_i}'\boldsymbol{\beta})^2 + \lambda\sum_{\mathbf{j=1}}^{\mathbf{k}}|\beta_j|, \tag{5.2}$$

where $\lambda$ is a tuning parameter [12]. This function may be thought of as the sum of squared residuals with an additional penalty term penalizing big $|\beta_j|$ values. The tuning parameter can be selected using k-fold cross-validation. First of all one needs to define a grid of possible $\lambda$ values. The second step is to calculate the cross-validation error $CV_k$ for every value of $\lambda$. Eventually one chooses the tuning parameter value $\lambda$ with the smallest cross-validation error $CV_k$. The final model will be fitted using all the data and the optimal $\lambda$ value [13]. In general $\hat{\boldsymbol{\beta}}_{\boldsymbol{lasso}}$ does not have a closed form solution because the function that needs to be minimized is not differentiable. To solve equation (5.2) and obtain a solution for $\hat{\boldsymbol{\beta}}_{\boldsymbol{lasso}}$ one could for example use a modification of the Least Angle Regression (LARS) algorithm [6]. A complete explanation of the LARS algorithm and how it works can be found in the paper by Efron et al.

To apply the lasso the data needs to be standardized. If this is not already the case the data can be standardized as follows

$$z_{ij} = \frac{x_{ij} - \bar{x}_j}{s_j},\ i = 1,2,\ldots,n,\ j = 1,2,\ldots,k \tag{5.3}$$

and

$$y_i^* = \frac{y_i - \bar{y}}{s_y},\ i = 1,2,\ldots,n, \tag{5.4}$$

where

$$s_j^2 = \frac{\sum_{i=1}^n (x_{ij} - \bar{x}_j)^2}{n-1}$$

is the sample variance of the regressor $x_j$ and

$$s_y^2 = \frac{\sum_{i=1}^n (y_i - \bar{y})^2}{n-1}$$

is the sample variance of the response [20].

## 5.3 Backward Stepwise Model Selection Using AIC

Instead of using the Lasso to obtain a sparse regression model one can also perform a model selection using the ordinary least squares estimator. The Akaike Information Criterion (AIC) is an estimator of the prediction error and therefore a popular criterion for comparing models. The AIC is defined by

$$AIC = 2k - 2ln(L_{max}), \tag{5.5}$$

where $k$ is the number of estimated parameters and $L_{max}$ the maximum value of the likelihood of the model [1]. Because the AIC is an estimator of the prediction error (which we want to minimize), selecting the best model using AIC will result in the model with the lowest AIC. In this paper we will use the AIC to do backward stepwise model selection. This procedure starts with the full model obtained by the OLS estimator. Thereafter we will calculate the AIC of the $k$ different models where only one variable is deleted. If at least one of these models has a lower AIC value than the full model, the model with the lowest AIC value will be selected as the new model. Otherwise we will keep the old model with all the variables and stop the selection procedure. This process will be repeated until deleting variables does not result in a lower AIC value or until there is no variable left in the model [10].

## 5.4 Measure of Model Fit

We will utilize two different measures to asses the model fit: the Root Mean Square Error (RMSE) and the Mean Absolute Deviation (MAD) [2]. The RMSE is the root of the mean squared error and is given by

$$RMSE = \sqrt{\frac{1}{n-k}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}. \tag{5.6}$$

The MAD is defined as the average absolute difference between fitted values and the observations

$$MAD = \frac{\sum_{i=1}^n |y_i - \hat{y}_i|}{n}. \tag{5.7}$$

Note that the unit of both measures equals the unit of the observed values. The MAD will always be less then or equal to the RMSE. A lower RMSE or MAD indicates a better model fit. Because the RMSE and the MAD non-negative, their optimal value would be 0.

# 6 Data

This section contains a description of the data that was utilized in our research. An explanation of the data as well as an exploratory data analysis will be provided.

## 6.1 Data and Variables

This paper will make use of data for census tracts in the Boston Standard Metropolitan Statistical Area (SMSA) in 1970. The data can be found in the mlbench package [15] from R. This package also contains data from the FBI, the Transportation and Air Shed Simulation Model (TASSIM) and the Massachusetts Tax-payers Foundation , among others. The data set consists of 506 rows, each representing a Boston suburb or city. For every suburb or city the following data is available:

- **medv**: the Median value of owner-occupied homes in $1000
- **rm**: average number of rooms per home
- **age**: proportion of owner-occupied homes built prior to 1940
- **b**: the quadratic transformation of B where B is the Proportion of black people in the population
- **lstat**: proportion of the population that is lower status (i.e. proportion of adults without some high school education and proportion of male workers classified as laborers)
- **crim**: the number of reported crimes per 1000 total population per year by town
- **zn**: The proportion of residential land in a community designated for lots larger than $25,000$ square feet
- **indus**: proportion of non-retail business acres per town
- **tax**: full-value property-tax rate per $ 10000
- **ptratio**: pupil-teacher ratio by town
- **chas**: Charles River dummy variable (= 1 if tract bounds the Carles River; 0 otherwise)
- **dis**: weighted distances to five employment centers in the Boston region
- **rad**: index of accessibility to radial highways
- **nox**: Nitrogen oxide concentrations in parts per 10 million

## 6.2 Exploratory Data Analysis

Before we fit a linear regression model it might be useful to critically pre-analyse the data using descriptive statistics and graphical representations. We will start the exploratory data analysis by plotting histograms for the numerical variables and a barplot for the Charles River dummy. These plots show how the observations are distributed.

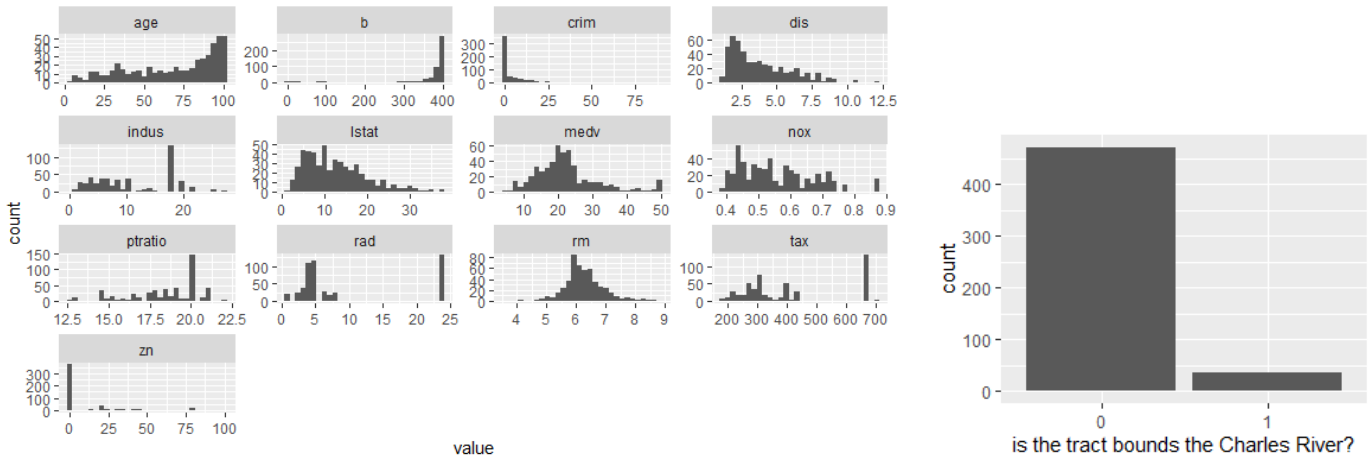


Figure 1: A histogram of each numerical variable (left) and a barplot of the chas variable (right) for the data from the Boston Housing data set

Some noteworthy observations are:

1. Most observations of the variables **zn** and **crim** are rather small (0 or close to 0).
2. For the dummy variable **chas** the value 0 (tract does not bound the Charles River) dominates. More than 90 per cent of the observations equal 0.
3. The majority of the observations of the variable **b** are around 400.

Even though linear regression does not make any assumptions about the distribution of the independent variables these findings suggests a possible nonlinear relationship between some of the variables and the dependent variable. As a result, one might consider to transform the variables that appear to be skewed. One could for example transform the variable **zn** to **log(zn)**. We decide to not transform the variables because the linear regression model does not require them to be non-skewed.

For the exploratory data analysis we will also utilize a correlation matrix to examine the relationship between the variables. Each cell in the correlation matrix represents the correlation coefficient between two variables. The correlation matrix for our data is:

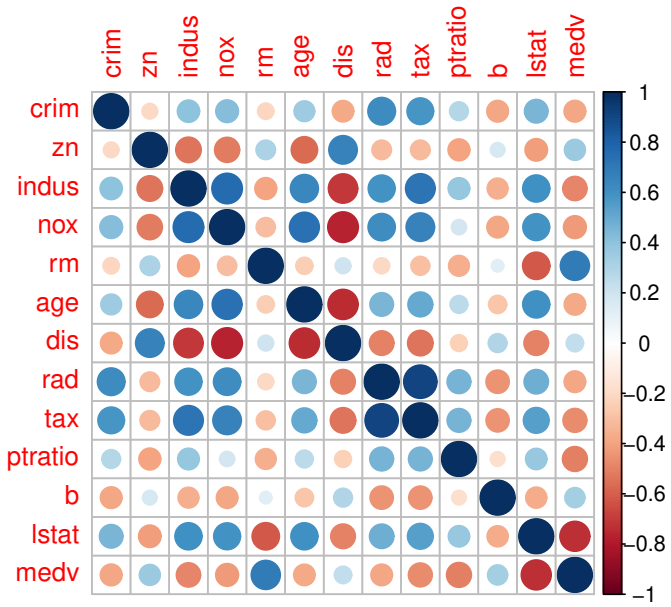


Figure 2: The correlation matrix for the variables of the Boston Housing data set. The (pairwise) correlation between every two variables is shown in each cell of the table.

The following are some important observations:

1. We will use the variable **tax** as the dependent variable in our model. The correlation between **tax** and **rm** appears to be negible (corr(**rm**,**tax**) = $-0.2920478$).

2. The variables **zn** (corr(**zn**,**tax**) = $-0.3145633$), **ptratio** (corr(**ptratio**,**tax**) = $0.4608530$), **b** (corr(**b**,**tax**) = $-0.4418080$) and **medv** (corr(**medv**,**tax**) = $-0.4685359$) are weakly correlated with **tax**.

3. The correlation matrix shows that some of the regressors are strongly correlated. One could for example observe that the variables **medv** and **lstat** have a correlation coefficient of $-0.7376627$. Hence there might be some multicollinearity [19].

Despite the fact that certain variables are stronger correlated with **tax** than others, it is a positive sign that all the variables are correlated with **tax**. This correlation does not necessarily imply a causal relation and hence further analysis is required to determine the causal relationship between the variables. When it comes to the multicollinearity, we have made the decision to neglect it.

We finish the exploratory data analysis with taking a closer look at the variable **medv**. If we plot the variable **medv** against **rm** we namely observe that there appears to be a maximum for the variable **medv**, as seen in the plot below. Gilley and Pace [9] discovered that the Census Bureau censored tracts whose median value was over $\$50,000$ which explains the observation.

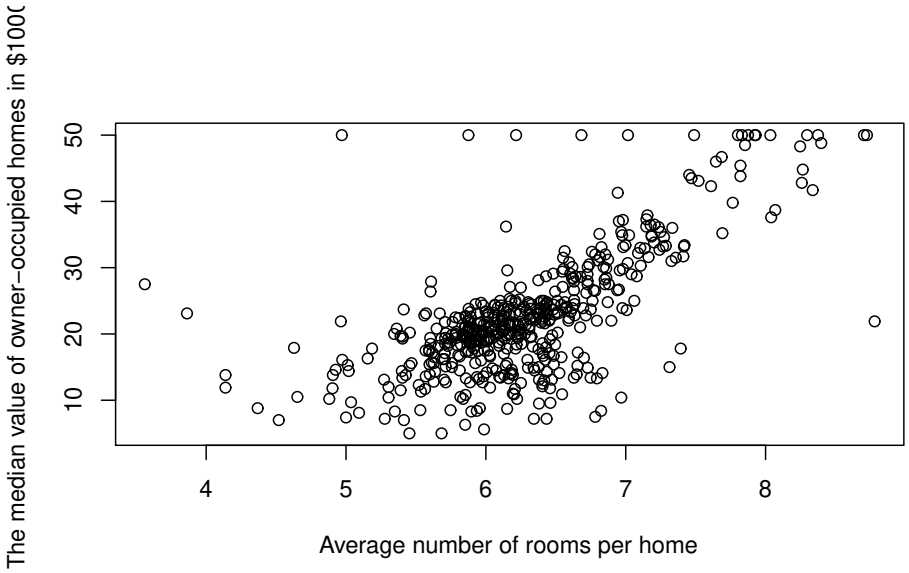


Figure 3: A plot of the data for the median value of owner-occupied homes in $1000 (**medv**) against the average number of rooms per home to illustrate the maximum value of $\$50,000$ for the variable **medv**.

# 7 Results

In this section we will study the linear regression model for the full-value property tax-rate per $1000 based on the Boston housing data set. The section consists of two parts, a model analysis and a model comparison of the linear regression model. In the first subsection we will consider five models based on five parameter estimation techniques, three frequentist and two Bayesian methods. To analyse the models we will check the parameter estimates, credible/confidence intervals and p-values (if available). Furthermore we will study the fitted regression values. In the second subsection we will investigate the model fit of the five models using different sample sizes and the MAD, RMSE and the MSE as the measure of fit statistics.

## 7.1 Analysis of the Models

### 7.1.1 Linear Regression Model with the Ordinary Least Squares Estimator

The first frequentist parameter estimation method that we are going to study is the ordinary least squares (OLS) estimator. The results of the OLS estimate of the linear regression coefficients are shown in Table 1.

| Variable | Estimate | 2.5 %-quantile | 97.5 %-quantile | p-value |
|---|---|---|---|---|
| (Intercept) | 208.508 | 84.945 | 332.070 | $0.001^*$ |
| crim | -0.289 | -1.062 | 0.485 | 0.464 |
| zn | 0.878 | 0.564 | 1.192 | $6.4e-08^*$ |
| indus | 7.044 | 5.754 | 8.334 | $<2e-16^*$ |
| chas1 | -22.512 | -42.687 | -2.336 | $0.029^*$ |
| nox | 43.327 | -47.535 | 134.188 | 0.349 |
| rm | -1.469 | -11.997 | 9.059 | 0.784 |
| age | 0.104 | -0.204 | 0.412 | 0.506 |
| dis | -1.549 | -6.447 | 3.348 | 0.534 |
| rad | 14.135 | 13.173 | 15.097 | $<2e-16^*$ |
| ptratio | 0.916 | -2.291 | 4.124 | 0.575 |
| b | -0.002 | -0.066 | 0.061 | 0.939 |
| lstat | -1.113 | -2.413 | 0.188 | 0.093 |
| medv | -1.735 | -2.774 | -0.696 | $0.001^*$ |

Table 1: The parameter estimates, 95% confidence intervals and p-values for the coefficients of the Boston housing linear regression model with dependent variable **tax** based on the OLS estimator. The asterisk in the final column indicates the significant variables with a significance level of $\alpha = 0.05$.

From the table it can be seen that 7 of the factors are expected to have a negative effect on the full-value property-tax rate, while 5 factors are expected to have a positive effect. Based on the p-values in combination with a significance level of $\alpha = 0.05$ the variables *zn*, *indus*, *chas*1, *rad* and *medv* are predicted to have a significant effect on *tax*. Observe that most of the confidence intervals are relatively wide. This indicates that there is quite some uncertainty about the true parameter values.

Besides the coefficient estimates we are also interested in the fitted values $\hat{\mathbf{y}}$. In figure 4 we have plotted all the 506 predicted responses $\hat{\mathbf{y}}$ against the actual response $\mathbf{y}$.

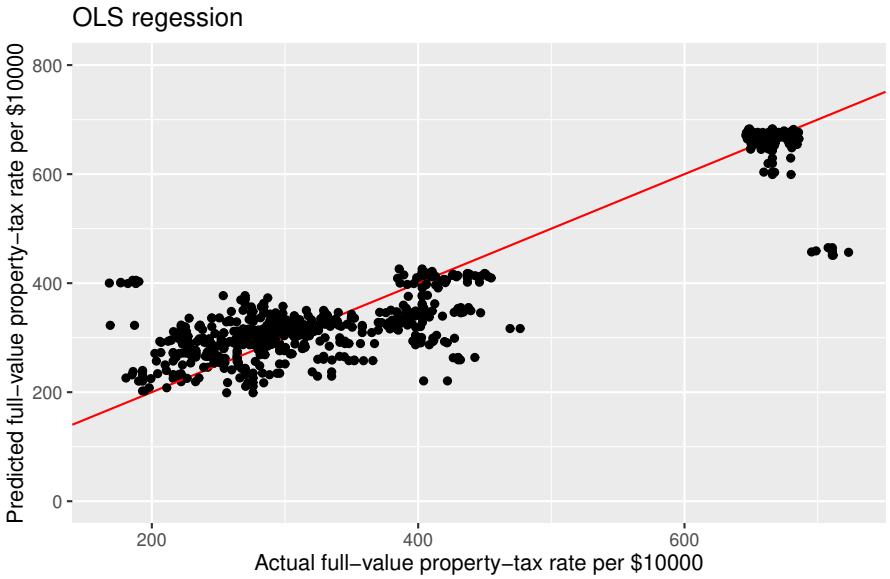


Figure 4: A plot of the predicted response $\hat{\mathbf{y}}$ of the variable **tax** based on a linear regression model with the OLS estimator against the actual response $\mathbf{y}$ from the Boston housing data set. The red line in the plot refers to a perfect fit. Note that a small amount of random noise is added to the results to better visualize overlapping values.

As one can see in the plot most values are between 200 and 400 dollar with a few outliers around 700 dollar. Even after adding a small amount of random noise to the results it is difficult to distinguish all the different points because of the large density. As one can observe most points are near the red line which indicates that the predicted and actual response are relatively close. At the end of this section we will measure and compare the model fit more formal using the root mean square error (RMSE) and the median absolute deviation (MAD).

### 7.1.2 Linear Regression Model with the Lasso

Following our analysis of the OLS estimator, we would like to focus on a more sparse model with fewer parameters. In section 5 we discussed how to achieve this with the lasso. As previously stated, this strategy is similar to the OLS technique but then with an extra penalty term. Therefore one might consider this as an improvement of the OLS estimator rather than a completely different frequentist estimator. Before we can establish the lasso estimate for the coefficients we first need to standardize the data and choose an optimal $\lambda$ value from a grid of possible $\lambda$ values. We will standardize the data as illustrated in section 5 to eventually transform the coefficients back to the original scale. The grid of possible $\lambda$ values is obtained by generating a sequence of 100 values between $-2$ and 5 which are used as exponents in combination with base 10. Ultimately we end up with a grid of 100 values ranging from 100000 to 0.01. For every $\lambda$ we calculate the cross-validation estimate using 10-fold cross-validation. If the value of $\lambda$ with the lowest cross-validation estimate lies close to the boundaries of the grid this suggests that we might need to increase the range of the grid. In figure 5 the cross-validation error is plotted against the log of the $\lambda$ values.

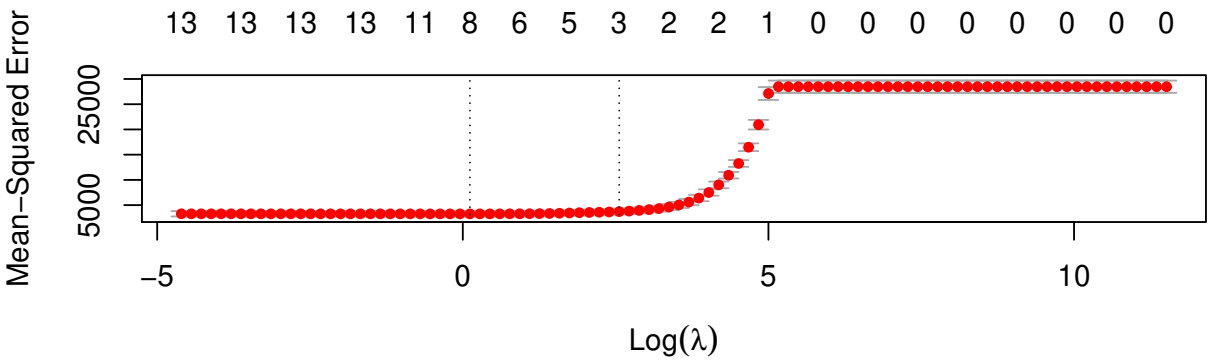


Figure 5: A plot of the cross-validation error against the log of the possible $\lambda$ values. The left dashed line indicates the $\lambda_{min}$ value with the minimum cross-validation estimate, while the right dashed line indicates the largest $\lambda$ value within one standard error of $\lambda_{min}$. The numbers at the top demonstrate the amount of non-zero parameter estimates.

$\lambda = 1.123324$ turns out to be the optimal $\lambda$ value with a cross-validation error of 3263. This value does not lie close to the boundaries of the grid from which we conclude that this is indeed the optimal $\lambda$ value. In figure 5 one can see that this value corresponds to 8 non-zero coefficient values. To really demonstrate that the minimum value of $\lambda$ equals 1.123324 a figure that focuses on a smaller region of $log(\lambda)$ is provided.

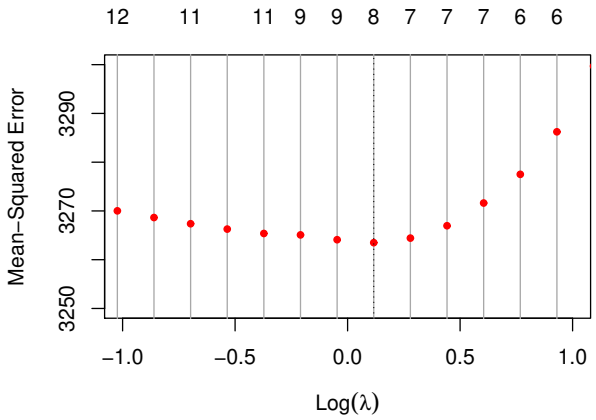


Figure 6: A figure that displays a smaller area of the $log(\lambda)$ values in figure 5 to better demonstrate the $\lambda_{min}$ value. The dashed line indicates the $\lambda_{min}$ value with the minimum cross-validation estimate. The numbers at the top demonstrate the amount of non-zero parameter estimates.

In figure 6 one can see that $\lambda = 1.123324$ really is the minimum $\lambda$ value. Figure 5 might raise the impression that the smaller the $\lambda$ value the lower the MSE but figure 6 shows that for the $\lambda$ values smaller than 1.123324 the MSE increases again. The results of the lasso estimator with all the data and $\lambda_{min}$ can be found in table 2.

| Variable | Estimate |
|---|---|
| (Intercept) | 187.635 |
| crim | 0 |
| zn | 0.670 |
| indus | 6.957 |
| chas1 | −18.499 |
| nox | 39.803 |
| rm | 0 |
| age | 0 |
| dis | 0 |
| rad | 13.941 |
| ptratio | 0.497 |
| b | 0 |
| lstat | −0.052 |
| medv | −1.195 |

Table 2: The parameter estimates of the Boston housing linear regression model with dependent variable **tax** based on the Lasso.

As one can observe from table 2 all the coefficients are shrunken with respect to the OLS estimator. The variables *crim*, *rm*, *age*, *dis* and *b* are even set to zero. It is not a surprise that especially these parameters were shrunken to zero. Recall that those coefficients correspond to variables that did not have a significant effect according to the OLS estimator. The only insignificant variables that remain in this model are *nox*, *ptratio* and *lstat*. Because the lasso estimator is biased it is unfortunately impossible to calculate reliable 95% confidence intervals and p-values as we did for the OLS estimator. We would like to investigate the fitted values $\hat{\mathbf{y}}$ for the lasso estimator as well. Hence we decide to

make a plot that is similar to figure 4 but then for the lasso.


Figure 7: A plot of the predicted response $\hat{\mathbf{y}}$ of the variable **tax** based on a linear regression model with the Lasso against the actual response $\mathbf{y}$ from the Boston housing data set. The red line in the plot refers to a perfect fit. Note that a small amount of random noise is added to the results to better visualize overlapping values.

If we compare this figure to figure 4 we observe a similar outcome. This is not entirely surprising, given that the benefit of employing the Lasso is only apparent after separating the data into a testing and a training set. This is something that we will do in the next section.

### 7.1.3 Linear Regression Model based on Backward Stepwise Model Selection Using AIC

We are currently using OLS in a suboptimal way, which makes it difficult to appropriately compare the OLS and the Lasso. If we want to reasonably compare the model fit of the model with the OLS estimator to the model with the Lasso we should also look at a model with the OLS estimator in combination with model selection. The selection procedure that we will use is the backward stepwise model selection using AIC that was described in section 5. The results of performing backward stepwise model selection for OLS using AIC are shown in Table 3.

| Variable | Estimate | 2.5 %-quantile | 97.5 %-quantile |
|---|---|---|---|
| (Intercept) | 204.193 | 162.905 | 245.481 |
| zn | 0.767 | 0.510 | 1.023 |
| indus | 7.274 | 6.053 | 8.495 |
| chas1 | -22.425 | -42.451 | -2.400 |
| nox | 56.803 | -15.353 | 128.959 |
| rad | 14.070 | 13.324 | 14.816 |
| lstat | -0.930 | -2.089 | 0.229 |
| medv | -1.697 | -2.511 | -0.882 |

Table 3: The parameter estimates and 95% confidence intervals of the Boston housing linear regression model with dependent variable **tax** based on backward stepwise model selection using AIC.

In table 3 one can see that the model selection procedure removed the variables *crim*, *rm*, *age*, *dis*, *ptratio* and *d* from the model. Those are almost exactly the variables that were set to zero by the Lasso except for the variable *ptratio* which was nonzero. The only variables that were predicted to have an insignificant effect on *tax* that remain in this model are *nox* and *lstat*. Removing these 6 variables from the model caused the parameter estimates to slightly differ from the OLS estimator. When we look at the 95% confidence intervals for the variables, we can see that the 95% confidence interval for the intercept has substantially shrunken compared to the interval obtained by the OLS. This model selection was based on the AIC. The AIC values of the original model and the model after model selection can be found in the table below.

| Model | AIC |
|---|---|
| OLS | 5530.425 |
| backward step AIC | 5520.483 |

Table 4: A table containing the AIC values for the model with the OLS estimator and the model based on backward stepwise model selection for OLS using AIC.

The difference in AIC value for the models is slightly less than 10, as seen in the table. This difference in AIC, according to Burnham and Anderson [3], is strong evidence for picking the model that was obtained by model selection. Hence the AIC values imply that model selection improves the model in this situation. We will now examine if this improvement also becomes clear when plotting the fitted values $\hat{\mathbf{y}}$.

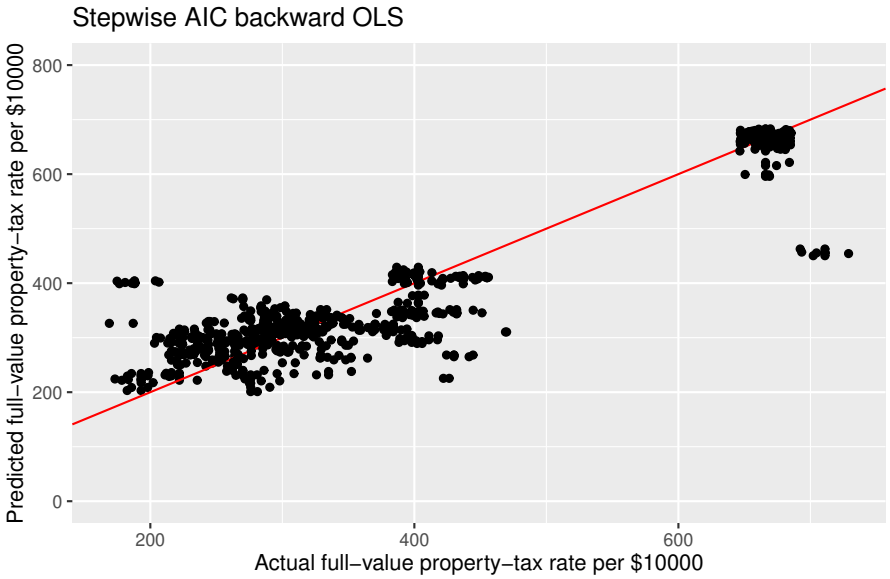


Figure 8: A plot of the predicted response $\hat{\mathbf{y}}$ of the variable **tax** based on a linear regression model with backward stepwise model selection using AIC against the actual response $\mathbf{y}$ from the Boston housing data set. The red line in the plot refers to a perfect fit. Note that a small amount of random noise is added to the results to better visualize overlapping values.

The plot appears to be similar to the plot of fitted values for the OLS and Lasso. Judging from the plot there does not seem to be a substantial improvement in the predicting power of the model. However, drawing conclusions only on the basis of the plot would be a poor choice. Therefore we will come back to this point in the next section.

### 7.1.4 Linear Regression Model based on Gibbs Sampler 1

Gibbs sampling in combination with an uninformative prior distribution is the first Bayesian parameter estimation method that we will look at. We will use the uninformative prior distribution with the corresponding Gibbs sampler that was introduced in section 4. We generate 10000 Gibbs samples to analyse the linear regression model. We will start the analysis with the sample-based posterior densities before moving on to the coefficient estimates and fitted values.

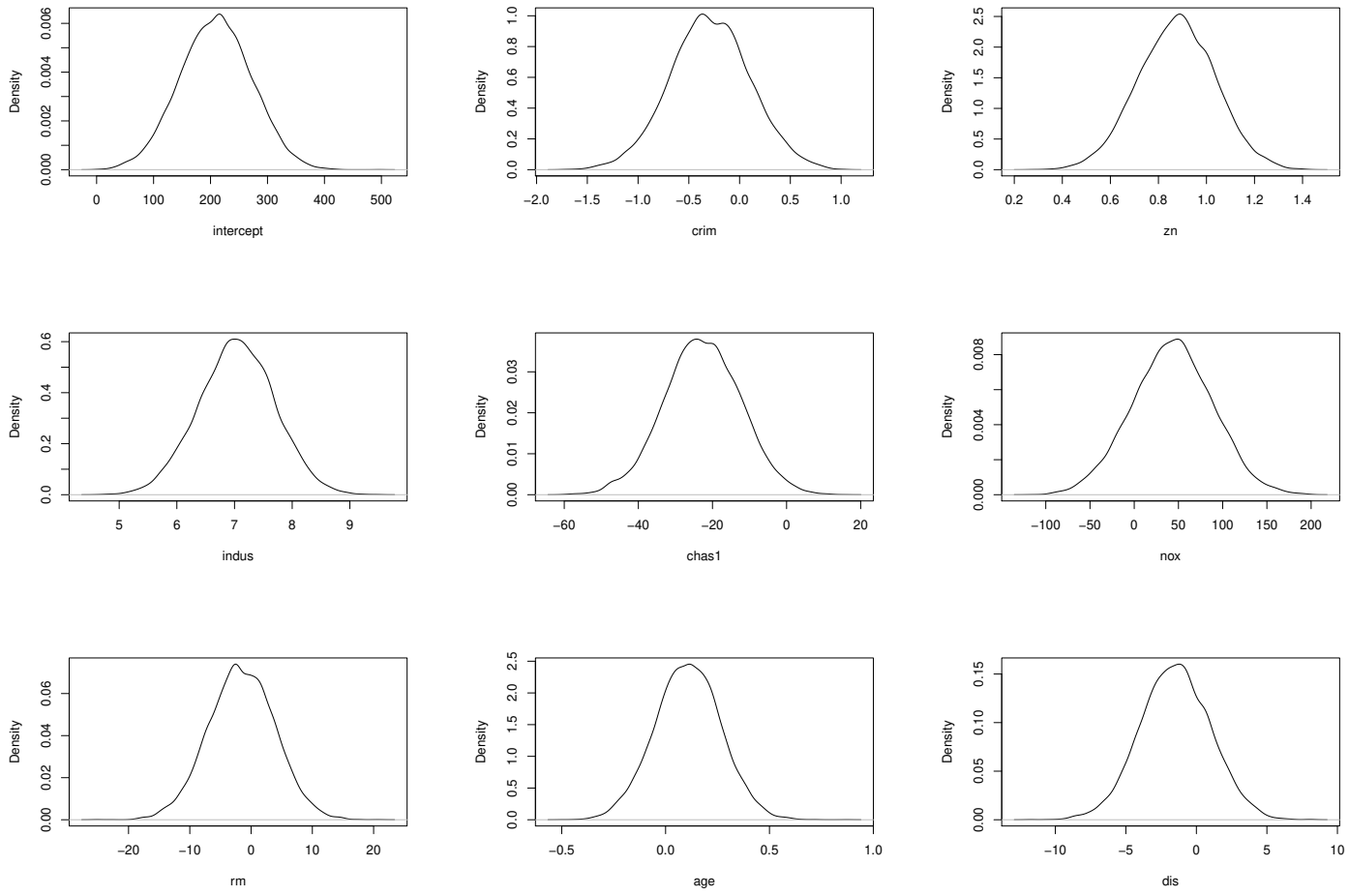


Figure 9: The posterior densities of the coefficients and the error variance (final plot) of the Boston housing linear regression model based on 10000 Gibbs samples.

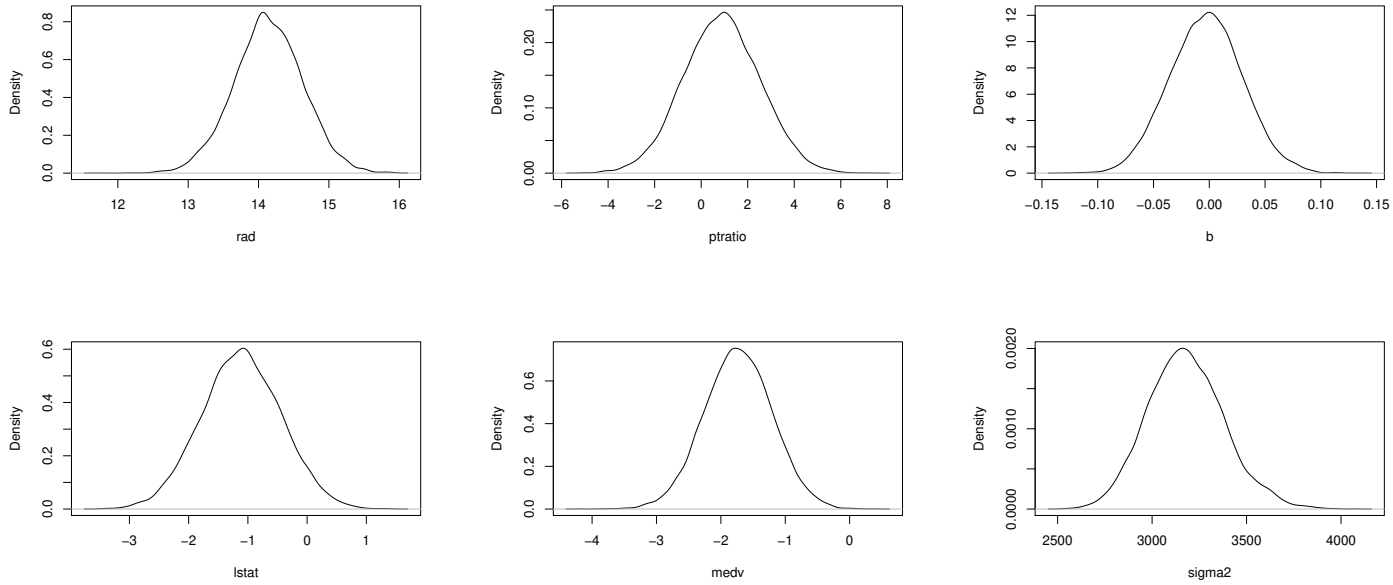


Figure 9 (cont.): The posterior densities of the coefficients and the error variance (final plot) of the Boston housing linear regression model based on 10000 Gibbs samples.

In the density plots we see that the coefficients indeed follow a distribution that looks like a normal distribution and that the error variance indeed follows a distribution that looks like an inverse chi square distribution. Hence we decide to look at the parameter estimates, credible intervals and p-values which are provided in table 5.

| Variable | Estimate | 2.5 %-quantile | 97.5 %-quantile | p-value |
|---|---|---|---|---|
| (Intercept) | 209.364 | 86.634 | 330.869 | 0 |
| crim | −0.290 | −1.075 | 0.483 | 0.453 |
| zn | 0.876 | 0.559 | 1.187 | 0 |
| indus | 7.040 | 5.743 | 8.308 | 0 |
| chas1 | −22.738 | −43.389 | −2.505 | 0.026 |
| nox | 43.180 | −46.946 | 132.014 | 0.344 |
| rm | −1.536 | -12.112 | 8.999 | 0.783 |
| age | 0.107 | −0.208 | 0.415 | 0.494 |
| dis | −1.499 | −6.389 | 3.372 | 0.545 |
| rad | 14.140 | 13.157 | 15.109 | 0 |
| ptratio | 0.883 | −2.349 | 4.101 | 0.593 |
| b | −0.003 | −0.066 | 0.062 | 0.938 |
| lstat | −1.112 | −2.421 | 0.203 | 0.100 |
| medv | −1.732 | −2.764 | −0.704 | 0.001 |

Table 5: The coefficient means, 95% empirical credible intervals and probability of direction based p-values of the Boston housing linear regression model with dependent variable **tax** based on Gibbs sampler 1.

The parameter estimates based on the mean of the Gibbs samples in table 5 look really similar to the parameter estimates in table 1. This result does not come as a surprise since the mean of the

conditional distribution of $\boldsymbol{\beta}$ is given by the OLS estimate of the coefficients. The 95% empirical credible intervals are also comparable to the 95% confidence intervals of the OLS estimator which follows from the conditional distribution too. The empirical confidence intervals of the variables *zn*, *indus* and *rad* do not contain the value 0 which indicates that the value of the parameter is significantly different from 0. This can also be seen in the plots of the posterior densities. For this estimator we would also like to plot the predicted response $\hat{\mathbf{y}}$ against the actual response $\mathbf{y}$. This figure looks almost identical to figure 4. Since the other findings looked a lot like the results of the OLS estimator we could have already expected to see a big similarity between this plot and figure 4.

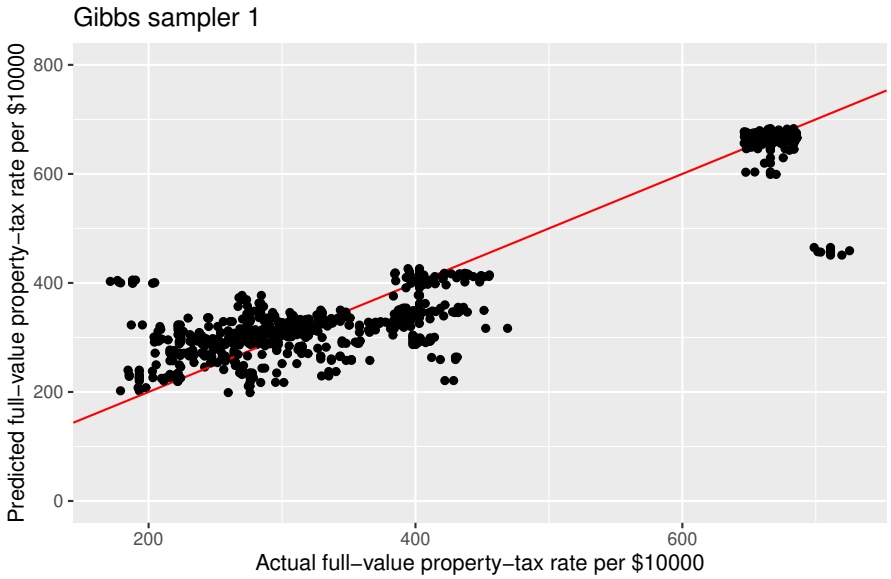


Figure 10: A plot of the predicted response $\hat{\mathbf{y}}$ of the variable **tax** based on a linear regression model with the Gibbs sampler 1 against the actual response $\mathbf{y}$ from the Boston housing data set. The red line in the plot refers to a perfect fit. Note that a small amount of random noise is added to the results to better visualize overlapping values.

### 7.1.5 Linear Regression Model based on Gibbs Sampler 2

The second Bayesian parameter estimation method that we will study is Gibbs sampling in combination with the informative prior distribution from section 4. As explained in section 4 an informative prior distribution is used if there is some prior knowledge available. Unfortunately there is limited prior information available for the Boston housing data. Hence we decide to take the standard multivariate normal distribution for $\boldsymbol{\beta}$ with the zero vector as the mean and the identity matrix times a factor $c$ where $c = 1$ as the variance. Note that if we increase the value of $c$ the informative prior for $\boldsymbol{\beta}$ would eventually converge to the flat prior that we used for the Monte Carlo approximation. For the error variance $\sigma^2$ we decide to choose the default value of the MCMCregress function for the scale and shape parameter of the inverse gamma distribution which equals 0.001 for both parameters. The MCMCregress function in R produces a Gibbs sample using our informative priors. To improve the efficiency of the Gibbs sampler we first standardize the data as described in section 5. After standardizing the data we generate 20000 Monte Carlo sample. The first 10000 samples are used as burn-in period while the other samples are used to analyse the linear regression model. We will start the analysis with the sample-based posterior densities and trace plots to check if the stationary distribution has been reached. In appendix C one can also find the autocorrelation plots.

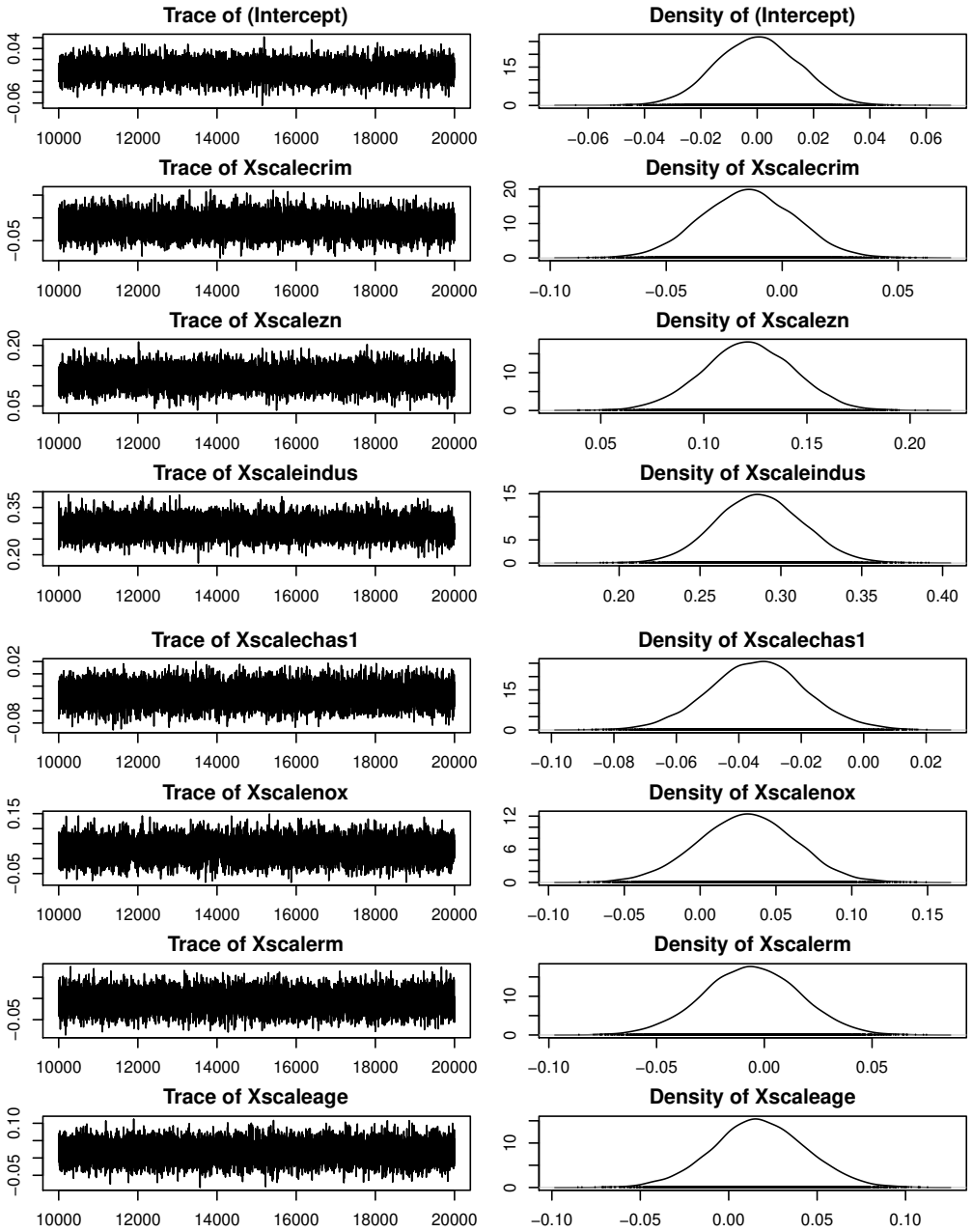


Figure 11: The trace (left) and density (right) plots for the scaled coefficients and error variance of the Boston housing linear regression model obtained by 10000 Gibbs samples with a burn-in of 10000. The y-axis in the trace plot represents the value of the variable while the x-axis represents the iteration number. The y-axis in the density plot represents the density while the x-axis represents the value of the variable.

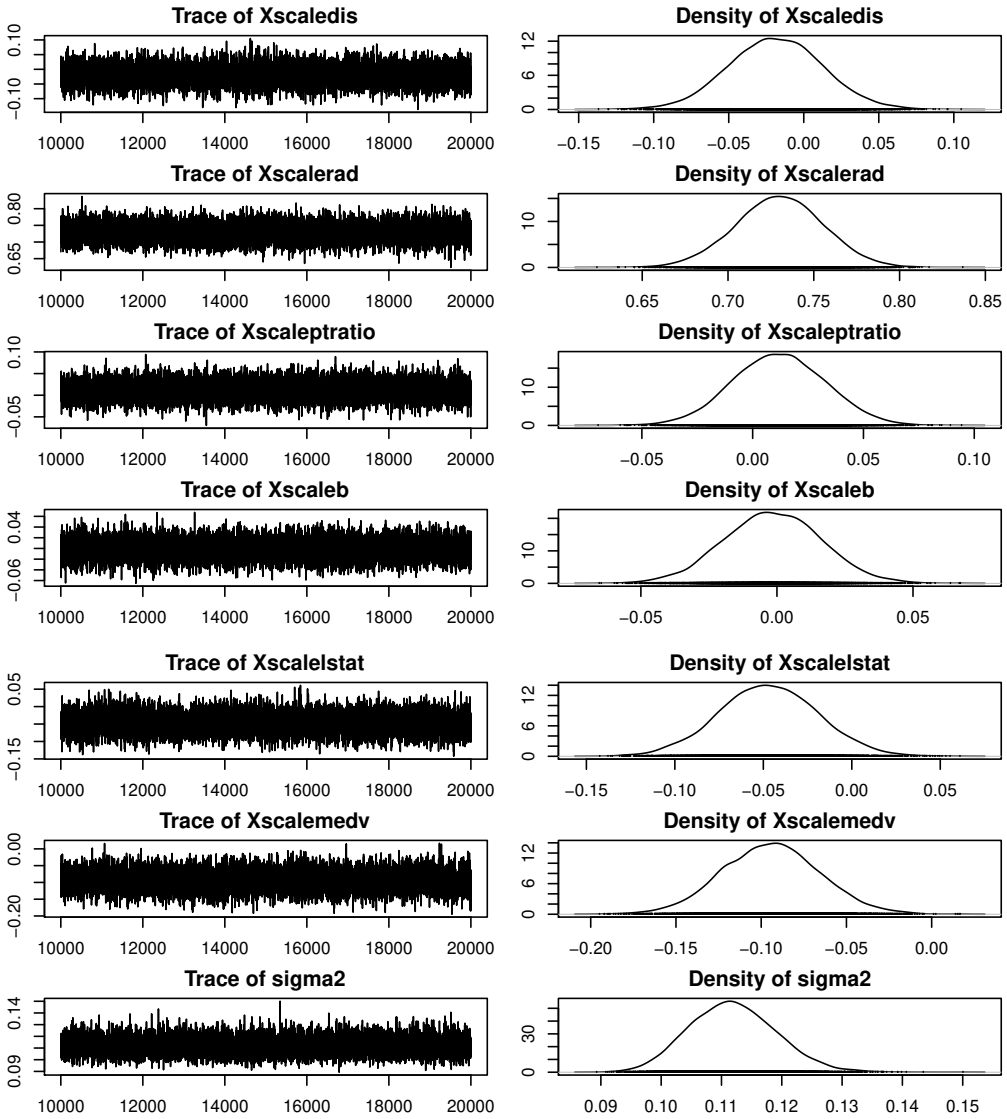


Figure 11 (cont.): The trace (left) and density (right) plots for the scaled coefficients and error variance of the Boston housing linear regression model obtained by 10000 Gibbs samples with a burn-in of 10000. The y-axis in the trace plot represents the value of the variable while the x-axis represents the iteration number. The y-axis in the density plot represents the density while the x-axis represents the value of the variable.

If we study the trace plots of the variables we do not observe any remarkable behaviour. The density plots also suggest that the stationary distribution has been reached. All the variables seem to follow the expected distribution. All in all we conclude that there is no indication against convergence of the Markov Chain. Before we analyse the mean of the samples, the 95% credible intervals and the p-values we will first convert everything back to the original scale. The outcome of the transformed variables can be found in table 6.

| Variable | Estimate | 2.5 %-quantile | 97.5 %-quantile | p-value |
|---|---|---|---|---|
| (Intercept) | 207.246 | 81.552 | 330.127 | 0.001 |
| crim | -0.287 | -1.0554 | 0.485 | 0.472 |
| zn | 0.878 | 0.559 | 1.184 | 0 |
| indus | 7.047 | 5.774 | 8.357 | 0 |
| chas1 | -22.516 | -42.601 | -1.970 | 0.032 |
| nox | 44.548 | -48.369 | 135.003 | 0.337 |
| rm | -1.446 | -11.967 | 8.949 | 0.791 |
| age | 0.104 | -0.208 | 0.416 | 0.498 |
| dis | -1.494 | -6.386 | 3.431 | 0.551 |
| rad | 14.128 | 13.1399 | 15.101 | 0 |
| ptratio | 0.928 | -2.212 | 4.118 | 0.570 |
| b | -0.003 | -0.068 | 0.061 | 0.927 |
| lstat | -1.111 | -2.411 | 0.187 | 0.095 |
| medv | -1.727 | -2.749 | -0.672 | 0.001 |

Table 6: The coefficient means, 95% empirical credible intervals and probability of direction based p-values of the Boston housing linear regression model with dependent variable **tax** based on Gibbs sampler 2.

Just like the outcome of Gibbs sampler 1 the results of the coefficient estimates and 95% empirical credible intervals are really similar to the OLS estimate. The biggest difference between the outcome of Gibbs sampler 1 and the results of the Gibbs sampler 2 is that the confidence intervals obtained by Gibbs sampler 2 suggest the variables *chas*1 and *medv* are significantly different from 0 too. To compare the p-values a table with the p-values of the OLS estimator, Gibbs sampler 1 and Gibbs sampler 2 will be provided.

| Variable | p-value OLS | p-value Gibbs 1 | p-value Gibbs 2 |
|---|---|---|---|
| (Intercept) | 0.001 | 0 | 0.001 |
| crim | 0.464 | 0.453 | 0.472 |
| zn | $6.4e-08$ | 0 | 0 |
| indus | $< 2e-16$ | 0 | 0 |
| chas1 | 0.029 | 0.026 | 0.032 |
| nox | 0.349 | 0.344 | 0.337 |
| rm | 0.784 | 0.783 | 0.791 |
| age | 0.506 | 0.494 | 0.498 |
| dis | 0.534 | 0.545 | 0.551 |
| rad | $< 2e-16$ | 0 | 0 |
| ptratio | 0.575 | 0.593 | 0.570 |
| b | 0.939 | 0.938 | 0.927 |
| lstat | 0.093 | 0.100 | 0.095 |
| medv | 0.001 | 0.001 | 0.001 |

Table 7: The (probability of direction based) p-values for the variables of the Boston housing data set based on the OLS estimator, Gibbs sampler 1 and Gibbs sampler 2.

The p-values of the three approaches are of comparable size, as seen in the table. The only difference is that the probability of direction based p-values are 0 for some of the variables that appear to have a small p-value for the OLS estimator. However, Gibbs sampler 2 assigns a nonzero p-value to the

intercept, whereas Gibbs sampler 1 assigns a 0 p-value to the intercept. If we order the variables by the size of their p-values we obtain a similar order for all three methods. The similarity in the results can also be seen in the plot below of the predicted response $\hat{\mathbf{y}}$ against the actual response $\mathbf{y}$. This plot is really comparable to the plot of the OLS and Monte Carlo method.

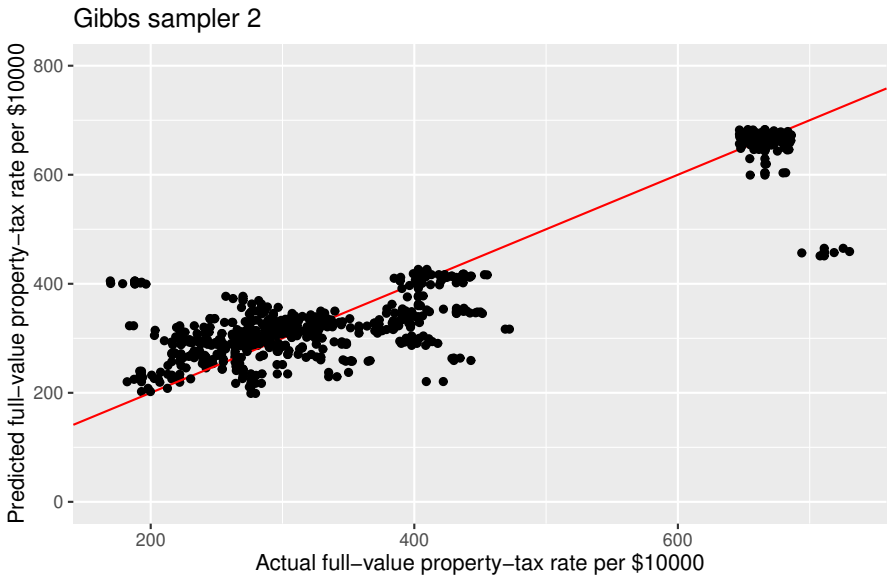


Figure 12: A plot of the predicted response $\hat{\mathbf{y}}$ of the variable **tax** based on a linear regression model with the Gibbs sampler 1 against the actual response $\mathbf{y}$ from the Boston housing data set. The red line in the plot refers to a perfect fit. Note that a small amount of random noise is added to the results to better visualize overlapping values.

## 7.2 Model Comparison

### 7.2.1 Model Comparison Based On All the Data

In the previous subsection we have only examined the predicting power of the various models graphically. Looking at the different plots there does not seem to be a major difference between the five estimators. We would also like to compare the models more formally using the Root mean squared error (RMSE) and the mean absolute deviation (MAD). In table 8 one can find the MAD and the RMSE for the five different models.

| Estimation Method | RMSE | MAD |
|---|---|---|
| OLS | 56.277 | 34.439 |
| Stepwise AIC backward OLS | 56.051 | 34.804 |
| Lasso | 56.567 | 35.177 |
| Gibbs sampler 1 | 56.278 | 34.459 |
| Gibbs sampler 2 | 56.277 | 34.443 |

Table 8: The root mean squared error and the mean absolute deviation for the models with the OLS, OLS in combination with backward stepwise model selection using AIC, Lasso, Gibbs sampler 1 and Gibbs sampler 2 based on all the data from the Boston housing data set as training and testing set.

Recall that a lower MAD and RMSE value imply a better model fit. As one can observe in the table the linear regression model with OLS in combination with backward stepwise model selection using AIC has the lowest RMSE while the linear regression model with the Lasso has the highest value. If we look at the MAD we observe that the OLS has the lowest value and the Lasso again the highest. This outcome is not unexpected since the OLS was defined to minimize the sum of squared residuals while Lasso penalizes the sizes of the regression coefficients. However these criteria do not indicate a significant difference between the model fit of the the five models.

### 7.2.2 Model Comparison Based on Cross-Validation

So far we have used the same data for training and testing the model. A possible risk of doing this is that the analysis of the fitted values only tells you something about the predicting power for this specific data set and not for new observations. OLS for example is likely to overfit the data which means that it might perform worse for new observations. Therefore it might be a good idea to evaluate the model fit of the five models for data that it has not seen before. We will use 10-fold cross-validation to split the data in a training and a testing set. In this situation 90% of the data is used to train the model and 10% is used to evaluate the model fit. As described in section 5 we will repeat this procedure 10 times to eventually obtain the CV estimate. Because of the reasons stated above the CV estimate can be regarded as a better measure of the model's predicting power. The outcome of the 10-fold cross-validation can be found in table 9.

| Estimation Method | $CV_{(10)}$ |
|---|---|
| OLS | 3,233.074 |
| Stepwise AIC backward OLS | 3,229.758 |
| Lasso | 3,230.064 |
| Gibbs sampler 1 | 3,233.285 |
| Gibbs sampler 2 | 3,232.976 |

Table 9: The 10-fold cross-validation estimate for the models with the OLS, OLS in combination with backward stepwise model selection using AIC, Lasso, Gibbs sampler 1 and Gibbs sampler 2. 90% of the data is used to train the model and 10% of the data is used to test the model

Notice that the cross-validation error is the lowest for the linear regression model with the OLS in combination with backward stepwise model selection and the highest for the linear regression model with Gibbs sampler 1. Hence 10-fold cross-validation suggests that the linear regression model with the OLS in combination with backward stepwise model selection is the best model for predicting the full-value property-tax rate per $1000. Furthermore observe that the Lasso has a lower cross-validation error than the OLS, Gibbs sampler 1 and Gibbs sampler 2. This differs from the previous result where the Lasso was the estimator that performed the worst. This outcome indicates that the Lasso is effective to prevent overfitting. Nevertheless the difference between the different models is again rather small so we need to do further analysis to strengthen the conclusion.

### 7.2.3 Model comparison for a small number observations

A possible explanation for the similarity in the results is the reasonably large size of the data set. As a result, we might wish to see how the models perform when the size of the data set to train the model is reduced. Instead of only looking at the case where 90% of the data is used to train the model and 10% to evaluate the model we will consider many different sample sizes. In total we will repeat the experiment 10 times for a percentage of the data set used as the training set ranging from 10% to 100% with steps of 10%. For every percentage we will calculate the mean squared error of the five estimators. The results of the MSE values for every percentage and every estimator can be found in figure 13.

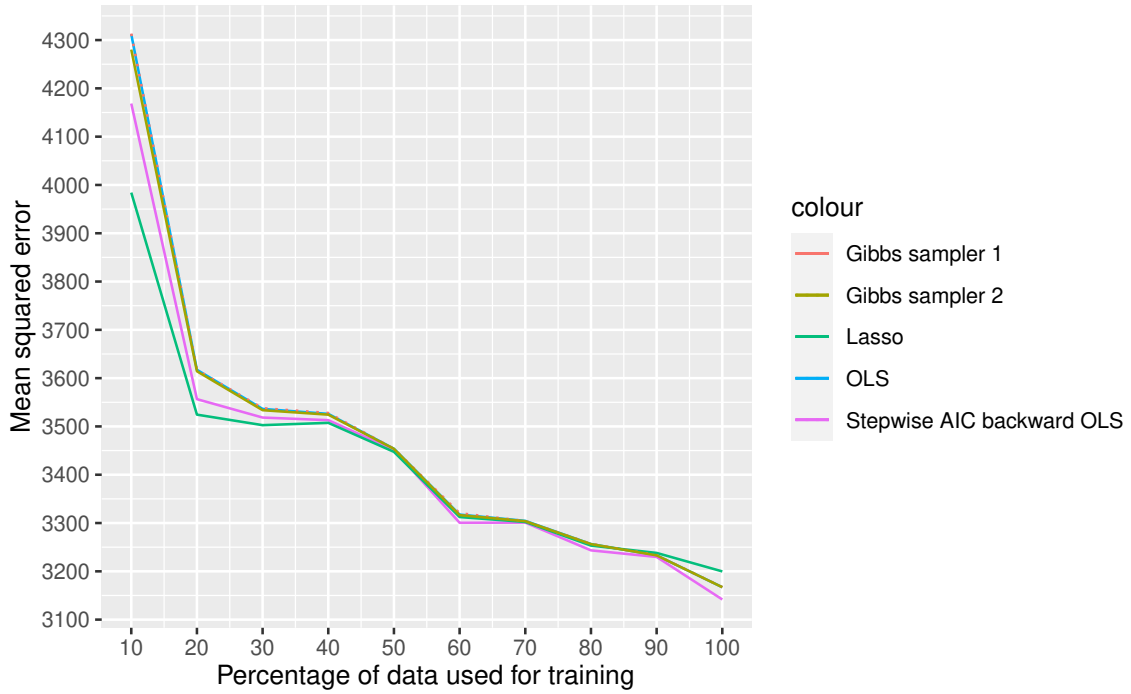


Figure 13: A plot of mean squared error against the percentage of the Boston housing data set that was used to train the model. Every line represents another method to estimate the coefficients. In the plot the following methods are considered: OLS, OLS in combination with backward stepwise model selection using AIC, Lasso, Gibbs sampler 1 and Gibbs sampler 2. The line of Gibbs sampler 1 is dotted to make the line of the OLS better visible.

In appendix C a table with the exact values of the MSE for every method and every percentage can be found. We start the analysis of the plot with some general remarks on the effect of the sample size on the MSE value. To all the methods applies the smaller the training set, the higher the MSE value. In general the decrease in the MSE values becomes smaller if the size of the data set increases. One could have expected this result since the common principle is that increasing the amount training data will lead to a better model fit (this is not always true). Moreover it is not surprising that the MSE decreases faster if the size of the data set is smaller because adding 10% of the data to the training data has a relatively bigger impact on the smaller data sets.

When comparing the results for the different estimators one can observe that there does not seem to be a substantial difference in the model fit if the training set is bigger than 50% of the total data set. Only for the full data set there is a noticeable difference between the Lasso, the OLS in combination with model selection and the other three estimators. If the training set is smaller than 50% of the total data set one can observe that the smaller the training set, the bigger the advantage of choosing an estimator that will result in a sparse regression model. Hence we conclude that if the amount of data is limited, variable selection substantially enhances the model's predicting power (in this case). Comparing the Lasso and the OLS in combination with model selection we can see that the model with the Lasso has the lower MSE values if the percentage of data used for training is lower than or equal to 50%. For the smallest sample size the Lasso even performs substantially better. For training sets with more than 50% of the data the model with the OLS estimator in combination with model selection leads to the lowest MSE values of all estimators. In this situation, we conclude that for small data sets the Lasso would be the best method for estimating the regression coefficients. Moreover the

results suggests that the Lasso indeed helps to prevent/reduce overfitting. For a bigger data set (bigger than 50% of the Boston housing data set) the OLS in combination with model selection would be the preferred method based on the MSE.

Another important aspect to focus on is the relationship between the findings of the frequentist and the Bayesian estimators. Since we did not perform any kind of model selection for the Bayesian approach we will only consider the OLS, Gibbs sampler 1 and Gibbs sampler 2. The curves of these estimators in figure 13 look really comparable which makes it difficult to distinguish the different curves. Especially the curves of Gibbs sampler 1 and the OLS appear to be almost identical. Hence the difference in predicting power between these two models appears to be limited (based on the MSE). For the smaller sample sizes one can observe that the curve of Gibbs sampler 2 lies slightly below the other two curves. As a result, we carefully infer that if the amount of data available is restricted, a Bayesian estimating technique with an informative prior performs slightly better. A disadvantage of this Bayesian estimation technique is that it is computationally much more expensive than the frequentist approach. Therefore one could question if choosing the Bayesian estimator to obtain a slightly better fit is worth the computational costs.

# 8 Conclusion

The purpose of this thesis was to provide a more comprehensive examination of the two fundamentally different approaches to estimate the parameters of a linear regression model. In this thesis we have compared three frequentist and two Bayesian estimators based on a simulation study with the Boston housing data set. When using the entire data set as the training set we obtain similar variable estimates, confidence/credible intervals and p-values for the OLS, Gibbs sampler with an uninformative prior and the Gibbs sampler with an informative prior. With the Lasso and the OLS in combination with model selection we obtain a sparse model with different parameter estimates. Jugding from a model where all the data is used to train the model the RMSE and the MAD do not suggest a substantial difference in model fit of the five estimators. Using 90% of the data to train the model again results in a minimal difference between the models. However this way to split the data already gives a small indication of the advantage of doing model selection. If we look at the MSE of the five models for ten different sample sizes we come to the conclusion that the smaller the training set, the higher the MSE value. This decrease in the MSE values becomes smaller if the size of the data set increases. If we compare the models using the MSE for the different sample sizes we can conclude that for small sample sizes the Lasso would be the best method for estimating the regression coefficients. Moreover the results suggests that the Lasso indeed helps to prevent/reduce overfitting. The OLS in combination with model selection would be the preferable technique for larger sample sizes based on the MSE. If we would only consider the estimators that do not perform any variable selection a Bayesian estimator with an informative prior distribution would achieve a slightly better model fit than the frequentist approach.

# 9 Discussion

Of course the analysis in this paper was not perfect. Therefore we should address some of the limitations and possible suggestions for future research. I would like to start the discussion with a remark about the model assumptions. In our research we assumed the model assumptions to hold and did not extensively check them. If one of the model assumptions turns out to be violated this might affect the reliability of our results. The correlation matrix for example suggested that there might be some multicollinearity. In our research we have made the decision to neglect it, because multicollinearity often does not cause any problems if the model is used for predicting. However multicollinearity might affect our results if we want to use it for causal inference. For this reason we might want to check the Variance Inflation Factor (VIF) [5] which measures the multicollinearity. If there appears to be multicollinearity we could for example use Ridge regression to correct for this.

Another questionable decision in our analysis is that we used an informative prior distribution for the Boston housing data set without any prior knowledge. Since it was difficult to find studies that we could use to determine the hyperparameters we have chosen to just guess some values for the hyperparameters. Hence one could argue that the title informative prior distribution that we used for the prior of Gibbs sampler 2 is not completely correct. The whole idea behind using an informative prior distribution disappears if there is no prior information used to specify the prior. It would be interesting to see how the Gibbs sampler with a real informative prior would compare to the other estimation methods so this is something that we can study in future research.

In our simulation study we have considered two different frequentist estimators that perform some sort of variable selection. We did not consider a Bayesian estimator in combination with variable selection. One could for example use a backwards elimination procedure using the deviance information criterion (DIC) [5] for the two Bayesian estimators that were studied in this paper. Due to computational constraints, this estimator was unfortunately left out of the analysis. Since we apply the Bayesian estimators in a suboptimal way it is difficult to draw a strong conclusion from the simulation study about which method (Bayesian or frequentist) results in the best model fit. A suggestion for future research would therefore be to also include a Bayesian estimator in combination with a backwards elimination procedure using DIC or another selection procedure to make the simulation study more complete.

In the model analysis section we have seen that there was a big similarity in the results for the different estimators. This similarity in the results made it more difficult to graphically compare the different methods. If we would for example plot the predicted observations for multiple estimators in one plot it would be extremely hard to distinguish the different estimators. It might have been a good idea to also plot the results for the smallest sample size that we analysed. Because the findings were so diverse for this sample size, it would have been much easier to graphically compare the different techniques.

Other recommendations for future research are also using different data sets and estimators. This study focuses on the estimation methods and not necessarily on the specific data set. For this reason we could also use different data sets to check if we end up with similar results and conclusions. For the priors that we studied in this paper it was relatively easy to obtain the posterior distribution. We can use different or more complex priors to produce different estimators.

# Bibliography

[1] H. Akaike. A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, 19(6):716–723, 1974.

[2] Ali Ali and Tasnim Kadhim. Linear regression model using bayesian approach for iraqi unemployment rate. pages 2279–0888, 02 2021.

[3] Kenneth P. Burnham and David R. Anderson. Multimodel inference: understanding AIC and BIC in model selection. *Sociological Methods & Research*, 33(2):261–304, November 2004.

[4] Angus Deaton. *Understanding Consumption*. Oxford University Press, 1992.

[5] Annette J. Dobson. *An introduction to generalized linear models / Annette J. Dobson*. Chapman Hall/CRC Boca Raton, 3rd ed. edition, 2008.

[6] Bradley Efron, Trevor Hastie, Iain Johnstone, and Rob Tibshirani. Least angle regression" (with discussions). *The Annals of Statistics*, 32, 01 2004.

[7] Francis Galton. Regression Towards Mediocrity in Hereditary Stature., January 1886.

[8] Andrew Gelman, John B. Carlin, Hal S. Stern, and Donald B. Rubin. *Bayesian Data Analysis*. Chapman and Hall/CRC, 2nd ed. edition, 2004.

[9] Otis W. Gilley and R.Kelley Pace. On the harrison and rubinfeld data. *Journal of Environmental Economics and Management*, 31(3):403–405, 1996.

[10] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning*. Springer Series in Statistics. Springer New York Inc., New York, NY, USA, 2001.

[11] Fumio Hayashi. *Econometrics*. Princeton Univ. Press, Princeton, NJ [u.a.], 2000.

[12] Peter D. Hoff. *A First Course in Bayesian Statistical Methods*. Springer Publishing Company, Incorporated, 1st edition, 2009.

[13] Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani. *An Introduction to Statistical Learning: with Applications in R*. Springer, 2013.

[14] A. M. Legendre. *Nouvelles methodes pour la determination des orbites des cometes*. F. Didot Paris, 1805.

[15] Friedrich Leisch and Evgenia Dimitriadou. *mlbench: Machine Learning Benchmark Problems*, 2021. R package version 2.1-3.

[16] Steven J. Leon. *Linear algebra with applications*. Pearson, 9th edition, 2015.

[17] E. Lesaffre and A.B. Lawson. *Bayesian Biostatistics*. Statistics in Practice. Wiley, 2012.

[18] Dominique Makowski, Mattan S. Ben-Shachar, S. H. Annabel Chen, and Daniel Lüdecke. Indices of effect existence and significance in the bayesian framework. *Frontiers in Psychology*, 10:2767, 2019.

[19] Mukaka MM. Statistics corner: A guide to appropriate use of correlation coefficient in medical research. *Malawi Med J.*, pages 69–71, 2012.

[20] Douglas C. Montgomery, Elizabeth A. Peck, and Geoffrey G. Vining. *Introduction to Linear Regression Analysis (5th ed.)*. Wiley & Sons, 2006.

[21] Syarifah Diana Permai and Heruna Tanty. Linear regression model using bayesian approach for energy performance of residential building. *Procedia Computer Science*, 135:671–677, 2018. The 3rd International Conference on Computer Science and Computational Intelligence (ICCSCI 2018) : Empowering Smart Technology in Digital Era for a Better Life.

[22] Jie. Mo Ling-Yun. Zeng Hong-Hu. Liang Yan-Peng Qin, Li-Tang. Wu. Linear regression model for predicting interactive mixture toxicity of pesticide and ionic liquid. *Environmental Science and Pollution Research*, 2015.

[23] Svetlozar T. Rachev and S. T. Rachev. *Bayesian methods in finance /*. Wiley,, Hoboken, N.J. :, 2008.

[24] MM Rodríguez del Águila and N Benítez-Parejo. Simple linear and multivariate regression models. *Allergologia et immunopathologia*, 39(3):159—173, 2011.

[25] Stephen M. Stigler. Gauss and the Invention of Least Squares. *The Annals of Statistics*, 9(3):465 – 474, 1981.

[26] IRA B. TAGER, SCOTT T. WEISS, BERNARD ROSNER, and FRANK E. SPEIZER. EFFECT OF PARENTAL CIGARETTE SMOKING ON THE PULMONARY FUNCTION OF CHILDREN. *American Journal of Epidemiology*, 110(1):15–26, 07 1979.

[27] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society (Series B)*, 58:267–288, 1996.

[28] Don van Ravenzwaaij, Peter Cassey, and Scott Brown. A simple introduction to markov chain monte–carlo sampling. *Psychonomic Bulletin Review*, 25, 03 2016.

[29] J. Wakefield. *Bayesian and frequentist regression methods*. 2013.

[30] Xin Yan and Xiao Gang Su. *Linear Regression Analysis: Theory and Computing*. World Scientific Publishing Co., Inc., USA, 2009.

# Appendices

## A Proof

**Theorem .1.** *Under the null hypothesis $H_0 : \beta_j = \beta_{j0}$ , $t_j$ follows a Student's t-distribution with $n-k$ degrees of freedom.*

*Proof.* Following the definition by Dobson [5] proving that $t_j$ follows a Student's t-distribution with $n-k$ degrees of freedom is equivalent to proving that $Z_j \sim N(0,1)$, $\frac{s^2(n-k)}{\sigma^2} \sim \chi^2(n-k)$ and that $Z_j$ and $\frac{s^2(n-k)}{\sigma^2}$ are independent.

To prove that $Z_j \sim N(0,1)$, first note that we can rewrite the OLS estimator as

$$\begin{aligned}\hat{\boldsymbol{\beta}} &= (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y} \\ &= (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'(\mathbf{X}\boldsymbol{\beta}+\boldsymbol{\varepsilon}) \\ &= (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{X}\boldsymbol{\beta}+(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\boldsymbol{\varepsilon} \\ &= \boldsymbol{\beta}+(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\boldsymbol{\varepsilon}.\end{aligned}$$

Using this and the model assumptions we can conclude that

$$\hat{\boldsymbol{\beta}} - \boldsymbol{\beta} \sim N(0, \sigma^2(\mathbf{X}'\mathbf{X})^{-1}).$$

From which directly follows that

$$\hat{\beta}_j - \beta_{j0} \sim N(0, \sigma^2((\mathbf{X}'\mathbf{X})^{-1})_{jj}).$$

Hence

$$Z_j = \frac{\hat{\beta}_j - \beta_{j0}}{\sqrt{\sigma^2((\mathbf{X}'\mathbf{X})^{-1})_{jj}}} \sim N(0,1).$$

Before we prove $\frac{s^2(n-k)}{\sigma^2} \sim \chi^2(n-k)$, note that we can rewrite $\frac{s^2(n-k)}{\sigma^2}$ as

$$\frac{s^2(n-k)}{\sigma^2} = \frac{\mathbf{e}'\mathbf{e}}{\sigma^2} = \frac{\boldsymbol{\varepsilon}}{\sigma}' \mathbf{M} \frac{\boldsymbol{\varepsilon}}{\sigma}.$$

To prove that $\frac{s^2(n-k)}{\sigma^2} \sim \chi^2(n-k)$ we will use a theorem from Dobson [5] that states that if $\frac{\boldsymbol{\varepsilon}}{\sigma} \sim N(0,I)$ and $\mathbf{M}$ is a symmetric and idempotent matrix then

$$\frac{\boldsymbol{\varepsilon}}{\sigma}' \mathbf{M} \frac{\boldsymbol{\varepsilon}}{\sigma} \sim \chi^2(\mathbf{n}),$$

where $n$ is $rank(\mathbf{M})$. The fact that $\frac{\boldsymbol{\varepsilon}}{\sigma} \sim N(0,I)$ directly follows from the model assumptions. It is straightforward to prove that $\mathbf{M}$ is symmetric and idempotent. For idempotent matrices the rank of matrix equals the trace. Recall that we calculated $trace(\mathbf{M}) = n-k$ which implies that $rank(\mathbf{M}) = n-k$. Therefore $\frac{s^2(n-k)}{\sigma^2} \sim \chi^2(n-k)$

To finalise the proof we need to show that $Z_j$ and $\frac{s^2(n-k)}{\sigma^2}$ are independent. $\hat{\boldsymbol{\beta}}$ and $\mathbf{e}$ are independent because we assumed $\boldsymbol{\varepsilon}$ to be normally distributed. The independence of $\hat{\boldsymbol{\beta}}$ and $s^2$ follows from the fact that $s^2$ is a function of $\mathbf{e}$. From this we conclude that $Z_j$ and $\frac{s^2(n-k)}{\sigma^2}$ are independent.

□

# B R code

```
library(ISLR)
library(caret)
library(arm)
#library(Ecdat)
library(gridExtra)
library(corrplot)
library(purrr)
library(tidyr)
library(ggplot2)
library(glmnet)
library(invgamma)
library(bayestestR)
library(see)
library(R1magic)
library(mlbench)
library(stargazer)

data(BostonHousing)

dat <- BostonHousing
#dat <- Hedonic
#analyse the structure
str(dat)
#summarize data
summary(dat)

#create correlation plot
M <-cor(dat[sapply(dat, is.numeric)])
corrplot(M, method="circle")

# Plot max value medv
plot(dat$rm, dat$medv, xlab="Average number of rooms per home",
ylab="The median value of owner-occupied homes in $1000")

# Barplot of chas:
```

```
ggplot(dat, aes(x = chas)) +
  geom_bar() +
  labs(x = "is the tract bounds the Charles River?", title = "")

# Histogram of all the numerical values.
dat %>%
  keep(is.numeric) %>%
  gather() %>%
  ggplot(aes(value)) +
  facet_wrap(~ key, scales = "free") +
  geom_histogram()

# Perform frequentist regression

X <- model.matrix(tax~. , dat)[,-1]
y <- dat$tax

freq.reg<-lm(y ~. , data=dat[,-10])
sumfreq <- summary(freq.reg)

# Find confidence intervals
ci <- confint(freq.reg, level=0.95)

# Make a table with the coefficient estimates,
the confidence intervals and the p-values
stargazer(cbind(sumfreq$coefficients[,1], ci, sumfreq$coefficients[,4]))

# Fit the linear regression
freq.regfit <- predict.lm(freq.reg,dat,interval = 'prediction',se.fit = T)

# Plot the regression fit
p.freq.reg <- ggplot(data = as.data.frame(cbind(y,freq.regfit$fit)),
aes(x = y, y = freq.regfit$fit[,1])) + geom_point() + ggtitle("OLS regession")
+ labs(x = "Actual full-value property-tax rate per $10000",
y = "Predicted full-value property-tax rate per $10000")
p.freq.reg + geom_abline(slope=1, intercept = 0, col="red")
+ expand_limits(y = c(0, 800)) + geom_jitter(width=20)

#+ geom_errorbar(ymin = freq.regfit$fit[,2], ymax = freq.regfit$fit[,3])

# Calculate the fitted value
y_hat <- cbind(1,X)%*%coef(freq.reg)

# Calculate the MAD and the RMSE
MAD_lm <- sum(abs(y - y_hat))/length(y)
RMSE_lm <- sqrt(t(y - y_hat)%*%(y - y_hat)/freq.reg$df.residual)


# Do a backward AIC selection
freqselect.reg <- stepAIC(freq.reg, direction = "backward")
```

```
# Find confidence intervals
ci <- confint(freqselect.reg, level=0.95)

# Make a table with the coefficient estimates,
the confidence intervals and the p-values
stargazer(cbind(freqselect.reg$coefficients, ci))

# Fit the linear regression
freq.regselectfit <- predict.lm(freqselect.reg,dat,interval = 'prediction',
se.fit = T)

# Plot the regression fit
p.freq.reg <- ggplot(data = as.data.frame(cbind(y,freq.regselectfit$fit)),
aes(x = y, y = freq.regselectfit$fit[,1])) + geom_point()
+ ggtitle("Stepwise AIC backward OLS")
+ labs(x = "Actual full-value property-tax rate per $10000",
y = "Predicted full-value property-tax rate per $10000")
p.freq.reg + geom_abline(slope=1, intercept = 0, col="red")
+ expand_limits(y = c(0, 800)) + geom_jitter(width=20)

#+ geom_errorbar(ymin = freq.regfit$fit[,2], ymax = freq.regfit$fit[,3])

# Calculate the fitted value
y_hat <- freqselect.reg$fitted.values

# Calculate the MAD and the RMSE
MAD_lmselect <- sum(abs(y - y_hat))/length(y)
RMSE_lmselect <- sqrt(t(y - y_hat)%*%(y - y_hat)/freqselect.reg$df.residual)



# Lasso regression
# Create a grid with possible lamdba values
grid=10^seq(5,-2, length =100)

set.seed(1)
#cv.glmnet also standardizes the data
cv_output <- cv.glmnet(X, y, alpha = 1, lambda=grid, family="gaussian")

#plot the mean-squared error against the log of lambda
plot(cv_output)

#Choose best lambda
best_lambda <- cv_output$lambda.min

# Rebuilding the model with best lamda value identified
lasso.reg <- glmnet(X, y, alpha = 1, lambda = best_lambda)

# Extract the coefficients
```

```
coeflasso.reg <- coef(lasso.reg)

# Make a table with the coefficient estimates
stargazer(t(coef(lasso.reg)[,1]))

# Fit the lasso regression
lasso.regfit <- predict(lasso.reg, s = best_lambda, newx = X)

# Calculate the fitted value
y_hat <- lasso.regfit

# Calculate the MAD and the RMSE
MAD_lasso <- sum(abs(y - y_hat))/length(y)
RMSE_lasso <- sqrt(t(y - y_hat)%*%(y - y_hat)/freq.reg$df.residual)

# Plot the regression fit
p.lasso.reg <- ggplot(data = as.data.frame(cbind(y,lasso.regfit)),
aes(x = y, y = lasso.regfit)) + geom_point()
+ ggtitle("Lasso Regression")
+ labs(x = "Actual full-value property-tax rate per $10000",
y = "Predicted full-value property-tax rate per $10000")
p.lasso.reg + geom_abline(slope=1, intercept = 0, col="red")
+ expand_limits(y = c(0, 800)) + geom_jitter(width=20)

#Bayesian regression
#Monte Carlo
#define some constants and vectors
X = cbind(1,model.matrix(tax~. , dat)[,-1])
beta.hat = solve(t(X)%*%X)%*%t(X)%*%y
s2 = t(y-X%*%beta.hat) %*% (y-X%*%beta.hat) / freq.reg$df.residual
M = 10000
set.seed(1)
sigma2_sample=beta_sample=NULL
# Generate M Monte Carlo samples
for(i in 1:M) {
  sigma2_sample = c(sigma2_sample,rinvgamma(1,shape =
  freq.reg$df.residual/2, rate = freq.reg$df.residual*s2/2))
  beta_sample = rbind(beta_sample,mvrnorm(1, beta.hat,
  sigma2_sample[i]*solve(t(X)%*%X)))
}

# Function to summarize Monte Carlo
Bayes.sum<-function(x)
{
  c("mean"=mean(x),
    "se"=sd(x),
    "t"=mean(x)/sd(x),
    "median"=median(x),
    "CrI"=quantile(x,prob=0.025),
    "CrI"=quantile(x,prob=0.975)
```

```
  )
}

# Combine the samples in a matrix
mcsample = cbind(intercept = c(beta_sample[,1]), beta_sample[,2:14],
sigma2_sample)

# Summarize Monte Carlo
mc_estimate <- NULL
for(i in 1:15) {
  mc_estimate <- rbind(mc_estimate, Bayes.sum(mcsample[,i]))
}

# Fit the linear regression
montecarlomean.regfit <- X%*%mc_estimate[1:14,1]

# Calculate the fitted value
y_hat <- montecarlomean.regfit

# Calculate the MAD and the RMSE
MAD_montecarlo <- sum(abs(y - y_hat))/length(y)
RMSE_montecarlo <- sqrt(t(y - y_hat)%*%(y - y_hat)/freq.reg$df.residual)
#montecarlo.regfit <- X%*%t(beta_sample)

#mc_lwr <- NULL
#for(i in 1:nrow(montecarlo.regfit)){
#mc_lwr <- rbind(mc_lwr, quantile(montecarlo.regfit[i,],prob=0.025))
#}
#mc_upr <- NULL
#for(i in 1:nrow(montecarlo.regfit)){
#  mc_upr <- rbind(mc_upr, quantile(montecarlo.regfit[i,],prob=0.975))
#}

# Plot the regression fit
p.montecarlo.reg <- ggplot(data = as.data.frame(cbind(y,montecarlomean.regfit)),
aes(x = y, y = montecarlomean.regfit)) + geom_point()
+ ggtitle("Gibbs sampler 1")
+ labs(x = "Actual full-value property-tax rate per $10000",
y = "Predicted full-value property-tax rate per $10000")
p.montecarlo.reg + geom_abline(slope=1, intercept = 0, col="red")
+ expand_limits(y = c(0, 800)) + geom_jitter(width=20)

#+ geom_errorbar(ymin = mc_lwr[,1], ymax = mc_upr[,1])

# Compute the probability of direction
pd <- NULL
for(i in 1:ncol(beta_sample)){
pd <- cbind(pd, p_direction(beta_sample[,i]))

}
```

```
pval <- pd_to_p(pd)

# Make a table with the coefficient estimates, the confidence intervals and
the p-values
stargazer(t(rbind(t(mc_estimate[1:14,1]), t(mc_estimate[1:14,5]),
t(mc_estimate[1:14,6]), pval)))

plot1 <- plot(p_direction(beta_sample[,1]))

## Marginal posterior distribution of the first few parameters
s2plot = plot(density(mcsample[,15]), main="", xlab = "sigma2")
b0plot = plot(density(mcsample[,1]), main="", xlab = "intercept")
b1plot = plot(density(mcsample[,2]), main="", xlab = "crim")
b2plot = plot(density(mcsample[,3]), main="", xlab = "zn")
b3plot = plot(density(mcsample[,4]), main="", xlab = "indus")
b4plot = plot(density(mcsample[,5]), main="", xlab = "chas1")
b5plot = plot(density(mcsample[,6]), main="", xlab = "nox")
b6plot = plot(density(mcsample[,7]), main="", xlab = "rm")
b7plot = plot(density(mcsample[,8]), main="", xlab = "age")
b8plot = plot(density(mcsample[,9]), main="", xlab = "dis")
b9plot = plot(density(mcsample[,10]), main="", xlab = "rad")
b10plot = plot(density(mcsample[,11]), main="", xlab = "ptratio")
b11plot = plot(density(mcsample[,12]), main="", xlab = "b")
b12plot = plot(density(mcsample[,13]), main="", xlab = "lstat")
b13plot = plot(density(mcsample[,14]), main="", xlab = "medv")

library(MCMCpack)

Xscale <- scale(model.matrix(tax~. , dat)[,-1])
yscale <- scale(dat$tax)

X <- model.matrix(tax~. , dat)[,-1]
y <- dat$tax

# Gibbs sampler with MCMCregress

# Generate multiple chains
Chain1scale <- MCMCregress(
  yscale~Xscale,
  data = dat,
  burnin = 10000,
  mcmc = 10000,
  thin = 1,
  verbose = 0,
  seed = 1,
  beta.start = 0,
  b0 = 0,
  B0 = 1,
  c0 = 0.001,
  d0 = 0.001)
```

```
# Summarize the chains
summary(Chain1scale)

# If traceplot does not work
par("mar")
par(mar=c(1,1,1,1))
par(mar = c(2, 2, 2, 1.5))
# Make a traceplot and density plot of the parameters
plot(Chain1scale)


library(plotMCMC)

# Make an autocorrelation plot
plotAuto(Chain1scale, thin=1, log=FALSE, base=10, main=NULL, xlab="Lag",
         ylab="Autocorrelation", lty=1, lwd=1, col="black")

# Rescale chain 1 to original scale and obtain the estimates
Chain1 <- NULL
for(i in 1:13) {
  Chain1new <- (Chain1scale[,i+1]*sd(y))/sd(X[,i])
  Chain1 <- cbind(Chain1, Chain1new)
}

coeff <- NULL
for(i in 1:13) {
  coeffnew <- mean(Chain1[,i])
  coeff <- cbind(coeff, coeffnew)
}

sum <- 0
for(i in 1:13) {
  sum <- sum + (Chain1scale[,i+1]*mean(X[,i])/sd(X[,i]))
}
intercept <- -sum*sd(y) + mean(y)

Chain1 <- cbind(intercept, Chain1)

coeffnew <- mean(intercept)
coeff <- cbind(coeffnew, coeff)

# Fit the linear regression
X = cbind(1,model.matrix(tax~., dat)[,-1])

gibbsmean.regfit <- X%*%t(coeff)

# Calculate the fitted value
y_hat <- gibbsmean.regfit
```

```
# Calculate the MAD and the RMSE
MAD_gibbs <- sum(abs(y - y_hat))/length(y)
RMSE_gibbs <- sqrt(t(y - y_hat)%*%(y - y_hat)/freq.reg$df.residual)

#gibbs.regfit <- X%*%t(Chain1)

#gibbs_lwr <- NULL
#for(i in 1:nrow(gibbs.regfit)){
#  gibbs_lwr <- rbind(gibbs_lwr, quantile(gibbs.regfit[i,],prob=0.025))
#}
#gibbs_upr <- NULL
#for(i in 1:nrow(gibbs.regfit)){
#  gibbs_upr <- rbind(gibbs_upr, quantile(gibbs.regfit[i,],prob=0.975))
#}

# Plot the regression fit
p.gibbs.reg <- ggplot(data = as.data.frame(cbind(y,gibbsmean.regfit)),
aes(x = y, y = gibbsmean.regfit)) + geom_point()
+ ggtitle("Gibbs sampler 2")
+ labs(x = "Actual full-value property-tax rate per $10000",
y = "Predicted full-value property-tax rate per $10000")
p.gibbs.reg + geom_abline(slope=1, intercept = 0, col="red")
+ expand_limits(y = c(0, 800)) + geom_jitter(width=20)

#+ geom_errorbar(ymin = gibbs_lwr[,1], ymax = gibbs_upr[,1])
# Compute the probability of direction
pd_gibbs <- NULL
for(i in 1:ncol(Chain1)){
  pd_gibbs <- cbind(pd_gibbs, p_direction(Chain1[,i]))

}
pval_gibbs <- pd_to_p(pd_gibbs)

# Determine the emperical credible interval
lwr <- Null
upr <- Null
for(i in 1:ncol(Chain1)){
  lwr <- cbind(lwr, quantile(Chain1[,i],prob=0.025))
  upr <- cbind(upr, quantile(Chain1[,i],prob=0.975))
}

# Make a table with the coefficient estimates, the confidence intervals and
the p-values
stargazer(cbind(t(coeff), lwr[,2:15], upr[,2:15], t(pval_gibbs)))

# Make a table with the MAD and the RMSE
stargazer(cbind(rbind(RMSE_lm, MAD_lm),rbind(RMSE_lasso, MAD_lasso),
rbind(RMSE_montecarlo, MAD_montecarlo), rbind(RMSE_gibbs, MAD_gibbs)))

#Randomly shuffle the data
```

```
dat<-dat[sample(nrow(dat)),]

#Create equally size folds
folds <- cut(seq(1,nrow(dat)),breaks=10,labels=FALSE)
#folds <- cut(seq(1,nrow(dat)),breaks=5,labels=FALSE)
#folds <- cut(seq(1,nrow(dat)),breaks=2,labels=FALSE)

#Perform n fold cross validation for linear regression
MSE_lm_vec <- NULL
MSE_lmselect_vec <- NULL

#for(i in 1:5){
#for(i in 1:2){
for(i in 1:10){
  #Segement your data by fold using the which() function
  testIndexes <- which(folds==i,arr.ind=TRUE)
  testdat <- dat[testIndexes, ]
  traindat <- dat[-testIndexes, ]

  y_test <- testdat$tax

  freq.reg<-lm(traindat$tax ~., data = traindat)
  y_hat <- predict(freq.reg, newdata=testdat)

  freqselect.reg <- stepAIC(freq.reg, direction = "backward")

  yAIC_hat <- predict(freqselect.reg,  newdata=testdat)

  MSE_lm <- t((y_test - y_hat))%*%(y_test - y_hat)/length(y_test)
  MSE_lm_vec <- rbind(MSE_lm_vec, MSE_lm)

  MSE_lmselect <- t((y_test - yAIC_hat))%*%(y_test - yAIC_hat)/length(y_test)
  MSE_lmselect_vec <- rbind(MSE_lmselect_vec, MSE_lmselect)
}

MSE_lm <- sum(MSE_lm_vec)/length(MSE_lm_vec)
MSE_lmselect <- sum(MSE_lmselect_vec)/length(MSE_lmselect_vec)

#Perform n-fold cross validation for lasso regression

MSE_lasso_vec <- NULL
coefmat <- NULL
set.seed(1)

#for(i in 1:5){
#for(i in 1:2){
for(i in 1:10){
  #Segement your data by fold using the which() function
  testIndexes <- which(folds==i,arr.ind=TRUE)
  testdat <- dat[testIndexes, ]
```

```
  traindat <- dat[-testIndexes, ]

  X_train <- model.matrix(tax~., traindat)[,-1]
  y_train <- traindat$tax

  X_test <- model.matrix(tax~., testdat)[,-1]
  y_test <- testdat$tax

  grid=10^seq(5,-2, length =100)

  set.seed(1)
  #cv.glmnet also standardizes the data
  cv_output <- cv.glmnet(X_train, y_train, alpha = 1, lambda=grid,
  family="gaussian")

  #Choose best lambda
  best_lambda <- cv_output$lambda.min

  # Rebuilding the model with best lamda value identified
  lasso.reg <- glmnet(X_train, y_train, alpha = 1, lambda = best_lambda)

  # Extract the coefficients
  coef(lasso.reg)

  # put the coefficients for all the runs in a matrix
  coefmat <- rbind(coefmat, coef(lasso.reg)[,1])

  y_hat <- cbind(1,X_test)%*%coef(lasso.reg)

  MSE_lasso <- t((y_test - y_hat))%*%(y_test - y_hat)/length(y_test)
  MSE_lasso_vec <- rbind(MSE_lasso_vec, MSE_lasso)
}

MSE_lasso <- sum(MSE_lasso_vec)/length(MSE_lasso_vec)

#Perform n-fold cross validation for Monte Carlo

MSE_montecarlo_vec <- NULL

#for(i in 1:5){
#for(i in 1:2){
for(i in 1:10){
  #Segement your data by fold using the which() function
  testIndexes <- which(folds==i,arr.ind=TRUE)
  testdat <- dat[testIndexes, ]
  traindat <- dat[-testIndexes, ]

  X_train <- model.matrix(tax~., traindat)[,-1]
  y_train <- traindat$tax
```

```
  X_test <- model.matrix(tax~. , testdat)[,-1]
  y_test <- testdat$tax

  #Monte Carlo
  #define some constants and vectors
  X = cbind(1,X_train)
  beta.hat = solve(t(X)%*%X)%*%t(X)%*%y_train
  s2 = t(y_train-X%*%beta.hat) %*% (y_train-X%*%beta.hat) /
  (length(y_train)-length(coef(freq.reg)))
  M = 10000
  set.seed(1)
  sigma2_sample=beta_sample=NULL
  # Generate M Monte Carlo samples
  for(i in 1:M) {
    sigma2_sample = c(sigma2_sample,rinvgamma(1,shape =
    (length(y_train)-length(coef(freq.reg)))/2, rate =
    (length(y_train)-length(coef(freq.reg)))*s2/2))
    beta_sample = rbind(beta_sample,mvrnorm(1, beta.hat,
    sigma2_sample[i]*solve(t(X)%*%X)))
  }

  # Combine the samples in a matrix
  mcsample = cbind(beta_sample, sigma2_sample)

  # Summarize Monte Carlo
  mc_estimate <- NULL
  for(i in 1:15) {
    mc_estimate <- rbind(mc_estimate, Bayes.sum(mcsample[,i]))
  }

  # Fit the linear regression
  y_hat <- cbind(1,X_test)%*%mc_estimate[1:14,1]

  MSE_montecarlo <- t((y_test - y_hat))%*%(y_test - y_hat)/length(y_test)
  MSE_montecarlo_vec <- rbind(MSE_montecarlo_vec, MSE_montecarlo)
}

MSE_montecarlo <- sum(MSE_montecarlo_vec)/length(MSE_montecarlo_vec)

#Perform n-fold cross validation for Gibbs sampling
library(MCMCpack)
MSE_gibbs_vec <- NULL

#for(i in 1:5){
#for(i in 1:2){
for(i in 1:10){
  #Segement your data by fold using the which() function
  testIndexes <- which(folds==i,arr.ind=TRUE)
  testdat <- dat[testIndexes, ]
  traindat <- dat[-testIndexes, ]
```

```
X_train <- model.matrix(tax~. , traindat)[,-1]
y_train <- traindat$tax

X_trainscale <- scale(model.matrix(tax~. , traindat)[,-1])
y_trainscale <- scale(traindat$tax)

X_test <- model.matrix(tax~. , testdat)[,-1]
y_test <- testdat$tax

# Generate chain
Chain1scale <- MCMCregress(
  y_trainscale~X_trainscale,
  data = traindat,
  burnin = 10000,
  mcmc = 10000,
  thin = 1,
  verbose = 0,
  seed = 1,
  beta.start = 0,
  b0 = 0,
  B0 = 1,
  c0 = 0.001,
  d0 = 0.001)

# Rescale chain 1 to original scale and obtain the estimates
Chain1 <- NULL
for(i in 1:13) {
  Chain1new <- (Chain1scale[,i+1]*sd(y_train))/sd(X_train[,i])
  Chain1 <- cbind(Chain1, Chain1new)
}

coeff <- NULL
for(i in 1:13) {
  coeffnew <- mean(Chain1[,i])
  coeff <- cbind(coeff, coeffnew)
}

sum <- 0
for(i in 1:13) {
  sum <- sum + (Chain1scale[,i+1]*mean(X_train[,i])/sd(X_train[,i]))
}
intercept <- -sum*sd(y_train) + mean(y_train)

Chain1 <- cbind(intercept, Chain1)

coeffnew <- mean(intercept)
coeff <- cbind(coeffnew, coeff)

# Fit the linear regression
```

```
  y_hat <- cbind(1,X_test)%*%t(coeff)

  MSE_gibbs <- t((y_test - y_hat))%*%(y_test - y_hat)/length(y_test)
  MSE_gibbs_vec <- rbind(MSE_gibbs_vec, MSE_gibbs)
}

MSE_gibbs <- sum(MSE_gibbs_vec)/length(MSE_gibbs_vec)

# Make a table with the MSE for the n-fold cross-validation
stargazer(cbind(MSE_lm, MSE_lmselect, MSE_lasso, MSE_montecarlo, MSE_gibbs))

# inverse n-fold Cross-Validation

#Create equally size folds
folds <- cut(seq(1,nrow(dat)),breaks=10,labels=FALSE)
#folds <- cut(seq(1,nrow(dat)),breaks=5,labels=FALSE)
#folds <- cut(seq(1,nrow(dat)),breaks=2,labels=FALSE)

#Perform n fold cross validation for linear regression
MSE_lm_vec <- NULL
MSE_lmselect_vec <- NULL

#for(i in 1:5){
#for(i in 1:2){
for(i in 1:10){
  #Segement your data by fold using the which() function
  testIndexes <- which(folds==i:i+2,arr.ind=TRUE)
  testdat <- dat[-testIndexes, ]
  traindat <- dat[testIndexes, ]

  y_test <- testdat$tax

  freq.reg<-lm(traindat$tax ~., data = traindat)
  y_hat <- predict(freq.reg, newdata=testdat)

  freqselect.reg <- stepAIC(freq.reg, direction = "backward")

  yAIC_hat <- predict(freqselect.reg,  newdata=testdat)

  MSE_lm <- t((y_test - y_hat))%*%(y_test - y_hat)/length(y_test)
  MSE_lm_vec <- rbind(MSE_lm_vec, MSE_lm)

  MSE_lmselect <- t((y_test - yAIC_hat))%*%(y_test - yAIC_hat)/length(y_test)
  MSE_lmselect_vec <- rbind(MSE_lmselect_vec, MSE_lmselect)
}

MSE_lm <- sum(MSE_lm_vec)/length(MSE_lm_vec)
MSE_lmselect <- sum(MSE_lmselect_vec)/length(MSE_lmselect_vec)
```

```
#Perform n-fold cross validation for lasso regression

MSE_lasso_vec <- NULL
coefmat <- NULL
set.seed(1)

#for(i in 1:5){
#for(i in 1:2){
for(i in 1:10){
  #Segement your data by fold using the which() function
  testIndexes <- which(folds==i,arr.ind=TRUE)
  testdat <- dat[-testIndexes, ]
  traindat <- dat[testIndexes, ]

  X_train <- model.matrix(tax~. , traindat)[,-1]
  y_train <- traindat$tax

  X_test <- model.matrix(tax~. , testdat)[,-1]
  y_test <- testdat$tax

  grid=10^seq(5,-2, length =100)

  set.seed(1)
  #cv.glmnet also standardizes the data
  cv_output <- cv.glmnet(X_train, y_train, alpha = 1, lambda=grid,
  family="gaussian")

  #Choose best lambda
  best_lambda <- cv_output$lambda.min

  # Rebuilding the model with best lamda value identified
  lasso.reg <- glmnet(X_train, y_train, alpha = 1, lambda = best_lambda)

  # Extract the coefficients
  coef(lasso.reg)

  # put the coefficients for all the runs in a matrix
  coefmat <- rbind(coefmat, coef(lasso.reg)[,1])

  y_hat <- cbind(1,X_test)%*%coef(lasso.reg)

  MSE_lasso <- t((y_test - y_hat))%*%(y_test - y_hat)/length(y_test)
  MSE_lasso_vec <- rbind(MSE_lasso_vec, MSE_lasso)
}

MSE_lasso <- sum(MSE_lasso_vec)/length(MSE_lasso_vec)

#Perform n-fold cross validation for Monte Carlo

MSE_montecarlo_vec <- NULL
```

```
#for(i in 1:5){
#for(i in 1:2){
for(i in 1:10){
  #Segement your data by fold using the which() function
  testIndexes <- which(folds==i,arr.ind=TRUE)
  testdat <- dat[-testIndexes, ]
  traindat <- dat[testIndexes, ]

  X_train <- model.matrix(tax~., traindat)[,-1]
  y_train <- traindat$tax

  X_test <- model.matrix(tax~., testdat)[,-1]
  y_test <- testdat$tax

  #Monte Carlo
  #define some constants and vectors
  X = cbind(1,X_train)
  beta.hat = solve(t(X)%*%X)%*%t(X)%*%y_train
  s2 = t(y_train-X%*%beta.hat) %*% (y_train-X%*%beta.hat) /
  (length(y_train)-length(coef(freq.reg)))
  M = 10000
  set.seed(1)
  sigma2_sample=beta_sample=NULL
  # Generate M Monte Carlo samples
  for(i in 1:M) {
    sigma2_sample = c(sigma2_sample,rinvgamma(1,shape =
    (length(y_train)-length(coef(freq.reg)))/2, rate =
    (length(y_train)-length(coef(freq.reg)))*s2/2))
    beta_sample = rbind(beta_sample,mvrnorm(1, beta.hat,
    sigma2_sample[i]*solve(t(X)%*%X)))
  }

  # Combine the samples in a matrix
  mcsample = cbind(beta_sample, sigma2_sample)

  # Summarize Monte Carlo
  mc_estimate <- NULL
  for(i in 1:15) {
    mc_estimate <- rbind(mc_estimate, Bayes.sum(mcsample[,i]))
  }

  # Fit the linear regression
  y_hat <- cbind(1,X_test)%*%mc_estimate[1:14,1]

  MSE_montecarlo <- t((y_test - y_hat))%*%(y_test - y_hat)/length(y_test)
  MSE_montecarlo_vec <- rbind(MSE_montecarlo_vec, MSE_montecarlo)
}

MSE_montecarlo <- sum(MSE_montecarlo_vec)/length(MSE_montecarlo_vec)
```

```
#Perform n-fold cross validation for Gibbs sampling
library(MCMCpack)
MSE_gibbs_vec <- NULL

#for(i in 1:5){
#for(i in 1:2){
for(i in 1:10){
  #Segement your data by fold using the which() function
  testIndexes <- which(folds==i,arr.ind=TRUE)
  testdat <- dat[-testIndexes, ]
  traindat <- dat[testIndexes, ]

  X_train <- model.matrix(tax~. , traindat)[,-1]
  y_train <- traindat$tax

  X_trainscale <- scale(model.matrix(tax~. , traindat)[,-1])
  y_trainscale <- scale(traindat$tax)

  X_test <- model.matrix(tax~. , testdat)[,-1]
  y_test <- testdat$tax

  # Generate chain
  Chain1scale <- MCMCregress(
    y_trainscale~X_trainscale,
    data = traindat,
    burnin = 10000,
    mcmc = 10000,
    thin = 1,
    verbose = 0,
    seed = 1,
    beta.start = 0,
    b0 = 0,
    B0 = 1,
    c0 = 0.001,
    d0 = 0.001)

  # Rescale chain 1 to original scale and obtain the estimates
  Chain1 <- NULL
  for(i in 1:13) {
    Chain1new <- (Chain1scale[,i+1]*sd(y_train))/sd(X_train[,i])
    Chain1 <- cbind(Chain1, Chain1new)
  }

  coeff <- NULL
  for(i in 1:13) {
    coeffnew <- mean(Chain1[,i])
    coeff <- cbind(coeff, coeffnew)
  }
```

```
  sum <- 0
  for(i in 1:13) {
    sum <- sum + (Chain1scale[,i+1]*mean(X_train[,i])/sd(X_train[,i]))
  }
  intercept <- -sum*sd(y_train) + mean(y_train)

  Chain1 <- cbind(intercept, Chain1)

  coeffnew <- mean(intercept)
  coeff <- cbind(coeffnew, coeff)

  # Fit the linear regression

  y_hat <- cbind(1,X_test)%*%t(coeff)

  MSE_gibbs <- t((y_test - y_hat))%*%(y_test - y_hat)/length(y_test)
  MSE_gibbs_vec <- rbind(MSE_gibbs_vec, MSE_gibbs)
}

MSE_gibbs <- sum(MSE_gibbs_vec)/length(MSE_gibbs_vec)

# Make a table with the MSE for the 10-fold cross-validation
stargazer(cbind(MSE_lm, MSE_lmselect, MSE_lasso, MSE_montecarlo, MSE_gibbs))

#Perform n fold cross validation for linear regression
MSE_lm_vec <- NULL
MSE_lmselect_vec <- NULL

#Segement your data by fold using the which() function
n <- nrow(dat)

#train_i <- 1:round(0.7*n)
#test_i <- round(0.7*n+1):n

#train_i <- 1:round(0.6*n)
#test_i <- round(0.6*n+1):n

#train_i <- 1:round(0.4*n)
#test_i <- round(0.4*n+1):n

train_i <- 1:round(0.3*n)
test_i <- round(0.3*n+1):n

traindat <- dat[train_i,]
testdat <- dat[test_i,]

x <- seq(10,100,10)
y_lm <- c(4311.984, 3617.236, 3536.271, 3526.033, 3453.793,
3317.459, 3304.491, 3256.505, 3233.074, 3167.101)
y_lmAIC <- c(4168.550, 3556.483, 3518.413, 3512.955,
```

```
3451.511, 3300.542, 3300.818, 3243.422, 3229.758, 3141.715)
y_lasso <- c(3984.132, 3524.443, 3502.638, 3507.443,
3447.345, 3312.400, 3302.202, 3253.269, 3230.064, 3199.825)
y_gibbs1 <- c(4313.377, 3616.828, 3537.715, 3527.906,
3453.788, 3320.405, 3304.000, 3256.893, 3233.285, 3167.213)
y_gibbs2 <- c(4280.245, 3614.249, 3533.635, 3524.364,
3453.619, 3316.309, 3303.759, 3256.347, 3232.976, 3167.101)
test_data <-
  data.frame(x, y_lm, y_lmAIC, y_lasso, y_gibbs1, y_gibbs2)

ggplot(test_data, aes(x)) +
  geom_line(aes(y = y_lm, colour = "OLS")) +
  geom_line(aes(y = y_lmAIC, colour = "Stepwise AIC backward OLS")) +
  geom_line(aes(y = y_lasso, colour = "Lasso")) +
  geom_line(aes(y = y_gibbs1, colour = "Gibbs sampler 1")) +
  geom_line(aes(y = y_gibbs2, colour = "Gibbs sampler 2")) +
  labs(x = "Percentage of data used for training", y = "Mean squared error") +
  theme(legend.position="right") +
  scale_x_continuous(n.breaks = 10) +
  scale_y_continuous(n.breaks = 10)
```

# C Plots Tables

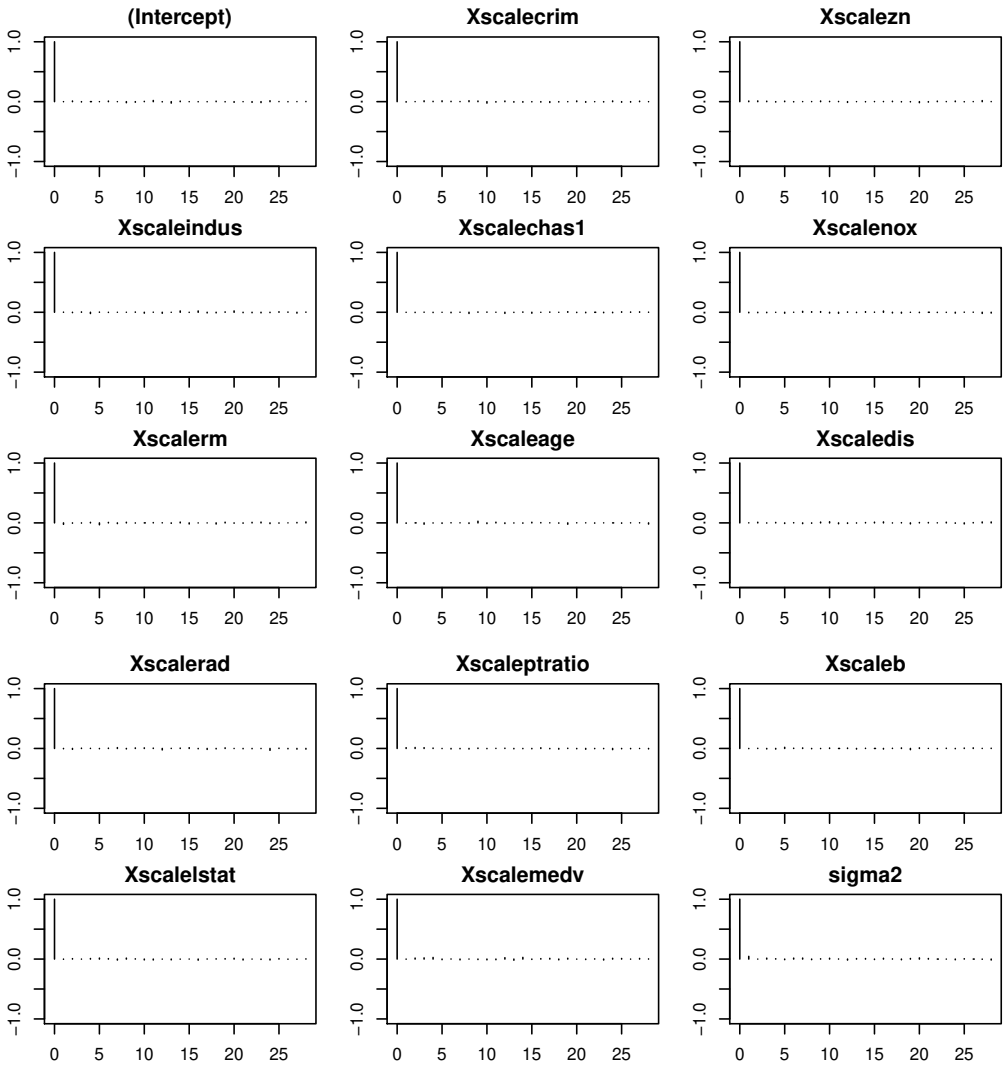


Figure 14: The autocorrelation plots for the scaled coefficients and error variance obtained by 10000 Gibbs samples with a burn-in of 10000. The y-axis represents the autocorrelation while the x-axis represents the lag. The plot suggests a high mixing rate.

| Percentage | MSE OLS | MSE Stepwise AIC backward OLS | MSE Lasso | MSE Gibbs sampler 1 | MSE Gibbs sampler 2 |
|---|---|---|---|---|---|
| 10 | 4311.984 | 4168.550 | 3984.132 | 4313.377 | 4280.245 |
| 20 | 3617.236 | 3556.483 | 3524.443 | 3616.828 | 3614.249 |
| 30 | 3536.271 | 3518.413 | 3502.638 | 3537.715 | 3533.635 |
| 40 | 3526.033 | 3512.955 | 3507.443 | 3527.906 | 3524.364 |
| 50 | 3453.793 | 3451.511 | 3447.345 | 3453.788 | 3453.619 |
| 60 | 3317.459 | 3300.542 | 3312.400 | 3320.405 | 3316.309 |
| 70 | 3304.491 | 3300.818 | 3302.202 | 3304.000 | 3303.759 |
| 80 | 3256.505 | 3243.422 | 3253.269 | 3256.893 | 3256.347 |
| 90 | 3233.074 | 3229.758 | 3230.064 | 3233.285 | 3232.976 |
| 100 | 3167.101 | 3141.715 | 3199.825 | 3167.213 | 3167.101 |

Table 10: A table containing the mean squared error for the different percentages of the data that was used to train the model for five different estimators. The estimators are: the OLS, OLS in combination with backward stepwise model selection using AIC, Lasso, Gibbs sampler 1 and Gibbs sampler 2.