**Delft University of Technology**
**Faculty of Electrical Engineering, Mathematics and Computer Science**
**Delft Institute of Applied Mathematics**

# Bayesian Estimation of a Monotone Regression Function

**A method described by Neelon and Dunson applied to climate data**

A thesis submitted to the
Delft Institute of Applied Mathematics
in partial fulfillment of the requirements

for the degree of

**BACHELOR OF SCIENCE**
**in**
**APPLIED MATHEMATICS**

**by**

**Damiaan Bonnet**

**Delft, The Netherlands**
**August 2021**

**BSc thesis Applied Mathematics**

**Delft University of Technology**

**Supervisors**

Prof. Dr. ir. G. Jongbloed

**Thesis committee**

Dr. ir. R. van der Toorn

August, 2021 Delft

# Contents

# Chapter 1

# Introduction

Regression analysis is an area in applied statistics that deals with finding a functional relationship between a response variable and one or more explanatory variables (Chatterjee & Hadi, 2015). One of the primary purposes of regression analysis is predicting or forecasting the response variable based on knowledge concerning the explanatory variables. In order to make accurate predictions, we need to estimate a curve that describes the expected value of the response variable in terms of the explanatory variable as well as possible. There are numerous ways of estimating a regression curve, because which regression technique is convenient depends on the kind of data there is at hand.

We can distinguish between parametric and non-parametric models. In parametric models it is assumed that the true regression curve belongs to a pre-chosen parametric class of functions. Non-parametric regression techniques do not assume a specified form for the function. Non-parametric regression techniques are therefore helpful when there is a lot of data, and little is known about the underlying relationship. An example of a non-parametric method is isotonic regression. It fits a free-form curve such that it is as close as possible to the observations. The only constraint for the shape of the regression curve is that it should be non-decreasing. Adding constraints on the form of the function can provide better estimates if we already know that the relationship satisfies this specific constraint. Especially, when the data sets are small and we know the relationship between response and explanatory variable is non-decreasing, isotonic regression can improve the estimates (Groeneboom & Jongbloed, 2015). The result of isotonic regression is a piece-wise constant curve, but there are also many studies about non-parametric methods that produce smoother monotone curve estimates. In the literature there are frequentist as well as Bayesian methods to produce a smooth monotone curve.

In this thesis, we study a Bayesian way of estimating a smooth monotone function using a method described by Neelon and Dunson in their article "Bayesian isotonic regression and trend analysis" (Neelon & Dunson, 2004). This thesis aims to apply the method to two specific climate data sets. We will use a data set of the average year temperature between 1901 and 2020 provided by weather station De Bilt and a data set of the average winter temperature from 1701 to 2014 (KNMI, 2021). Furthermore, it also aims to provide solutions to problems that we come across if we apply the method to climate data.

First of all, we address the Bayesian way of estimating parameters in chapter 2 to become familiar with this Bayesian approach. In chapter 3, we describe the Bayesian technique proposed by Neelon and Dunson. At the start of this chapter, the functional form of a monotone regression curve will be introduced and we continue by how the Bayesian approach proposed by Neelon and Dunson produces a smooth monotone curve. Chapter 4 addresses some difficulties one may experience when the method of Neelon and Dunson is applied to simulated data sets on

particular regression functions chosen and on two climate data sets.

# Chapter 2

# Bayesian Estimation

We start by explaining Bayesian parameter estimation using a simple statistical model example. Suppose we do an experiment of tossing a coin. We flip a coin $n = 10$ times. Let $Y$ be the random variable which denotes the number of times that heads is thrown. Then $Y \sim \text{Bin}(10, \theta)$ where $\theta$ is the probability that the coin lands heads-up, so $\theta \in [0, 1]$. Suppose the outcome of our experiment is $y = 6$. So we have thrown 6 times heads and 4 times tails. We would like to know whether the coin is fair. In other words, we would like to know what $\theta$ is. To estimate $\theta$ we can use a frequentist or Bayesian method. For a better understanding of Bayesian estimation, it is useful to explain frequentist estimation beforehand.

### Frequentist estimation of $\theta$

A popular frequentist way of estimating $\theta$ is the method of maximum likelihood. That method is concerned with finding $\theta$ that maximizes the probability of observing $y$ (the number of times of heads is tossed) given $\theta$. We write this down as $P(Y = y|\theta)$. We call $P(Y = y \mid \theta)$ as a function of $\theta$ the likelihood function. The probability that we observe $y$-times heads during the $n$ throws given parameter $\theta$ is

$$P(Y = y|\theta) = p_Y(y|\theta) = \binom{n}{y} \theta^y (1-\theta)^{n-y} \tag{2.1}$$

which is the probability mass function of a random variable $Y$, which is binomially distributed with parameters $n$ and $\theta$. For our experiment with outcome $y = 6$ and $n = 10$ we get that the likelihood function is

$$p_Y(y|\theta) = \binom{10}{6} \theta^6 (1-\theta)^4 = 210\, \theta^6 (1-\theta)^4 \tag{2.2}$$

The maximum likelihood estimate for $\theta$ would now be

$$\hat{\theta} = \underset{\theta \in [0,1]}{\arg\max}\, p_Y(y|\theta) \tag{2.3}$$

By differentiating the function of (2.2) with respect to $\theta$ and setting it equal to zero we can obtain the $\theta$ that maximizes the probabiliy that we observe $y = 6$ given $\theta$, which is

$$\frac{d(p_Y(y|\hat{\theta}))}{d\theta} = 0 \implies \hat{\theta} = 0.6 \tag{2.4}$$

### Bayesian estimation of $\theta$

The Bayesian way of estimating $\theta$ is different. The Bayesian way of estimating treats parameters as random variables, so $\theta$ has a distribution. The foundation of the Bayesian way of estimating parameters is Bayes Theorem, which states the following about the probability of event A given that event B has taken place:

$$P(A \mid B) = \frac{P(B \mid A)P(A)}{P(B)} \tag{2.5}$$

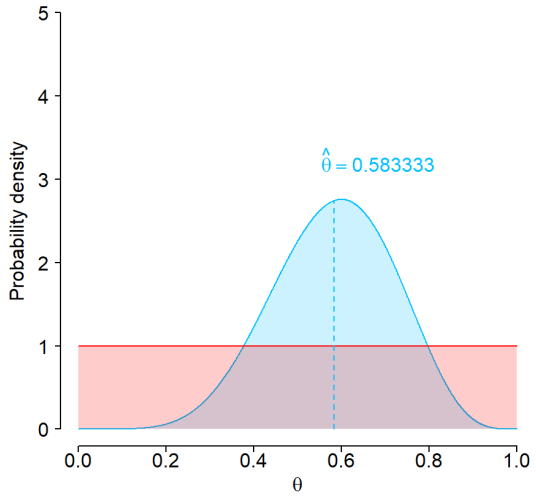
We can compute the probability density for $\theta$ conditional on the outcome $y$ likewise by

$$p(\theta \mid y) = \frac{P(Y = y \mid \theta)p(\theta)}{P(Y = y)} \tag{2.6}$$

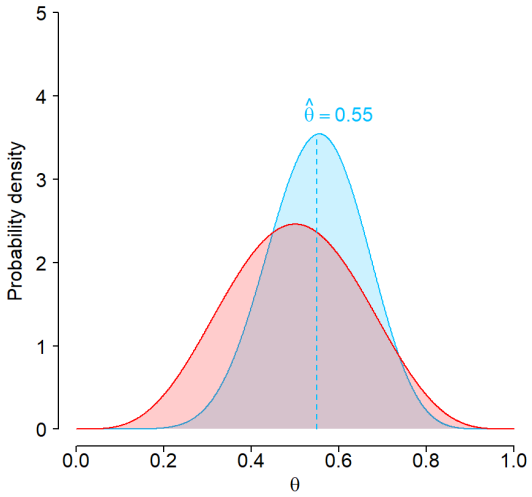
We call $p(\theta \mid y)$ the posterior distribution, $p(\theta)$ the prior distribution, $P(Y = y \mid \theta)$ as a function of $\theta$ the likelihood and $P(Y = y)$ the probability distribution of the data. Note the use of small "$p$" for the prior distribution of $\theta$. The prior distribution can be either discrete or continuous, but for our example we are going to assume $\theta$ has a continuous distribution.Therefore, we use small "$p$" to indicate a probability density function. If the prior has a continuous distribution, then also our posterior is continuous. In Bayesian estimation we aim to compute the posterior distribution of $\theta$, given data $y$. So we try to compute the probability distribution of $\theta$ when we have observed the outcome $y$. The prior distribution can be used to reflect prior beliefs about the prior distribution of $\theta$. For example, if we think that the coin is a priori unfair and biased towards heads we can specify a prior with probability mass concentrated on larger values of $\theta$. If one has no prior beliefs, one can specify a uniform prior distribution in this example. So $\theta \sim U(0, 1)$ and therefore $p(\theta) = 1_{[0,1]}(\theta)$. The posterior distribution for theta becomes:

$$p(\theta \mid y) = \frac{210 \cdot \theta^6(1-\theta)^4 p(\theta)}{P(y)} \propto 210 \cdot \theta^6(1-\theta)^4 \tag{2.7}$$

So the posterior distribution of $\theta$ given that the outcome is 6 is described above. This posterior distribution is a Beta$(7, 5)$ distribution. If you nonetheless want to have a point-estimate for $\theta$, you can summarize the posterior distribution in a single number. A common Bayes estimate is the expectation of the posterior distribution. So the Bayes estimate in this case is the expectation of the Beta$(7, 5)$ distribution. The expectation is $\hat{\theta} = 0.583333$. The prior distribution, posterior distribution and Bayes estimate are visualized in figure 2.1a.




(a) Prior: $\theta \sim U(0,1)$
Posterior: $\theta \mid y \sim \text{Beta}(7,5)$




(b) Prior: $\theta \sim \text{Beta}(5,5)$
Posterior: $\theta \mid y \sim \text{Beta}(11,9)$

Figure 2.1: Prior and posterior distributions for $\theta$. The blue curves are the posterior density functions and the red curves are the prior density functions.

If you do have a more specific prior belief that for example the coin is fair, you can also specify another prior that suits your prior beliefs. For example, if you think the coin is fair, but you are not really sure, you can express your prior beliefs by $\theta \sim \text{Beta}(5,5)$. You do as if you had seen 5 heads and 5 tails before, but you have not seen any data yet. Using a beta prior for Bayesian estimation of $\theta$ is visualized in figure 2.1b. The posterior distribution becomes a $\text{Beta}(11,9)$ distribution which has more probability mass concentrated around 0.5 as you can see in figure 2.1b. That is also why also the Bayes estimate is closer to 0.5 with $\hat{\theta} = 0.55$. The two figures in figure 2.1 clearly show the difference in posterior distributions, if different priors are used. So it shows how much bayesian estimates depend on prior beliefs, whereas frequentist estimates only depend on data.

# Chapter 3

# Method Neelon and Dunson

In chapter 2 we described Bayesian estimation using a simple example of a coin tossing experiment. In this chapter we explicate the Bayesian estimation method to estimate a monotone regression curve proposed by Neelon and Dunson.

## 3.1 The Model

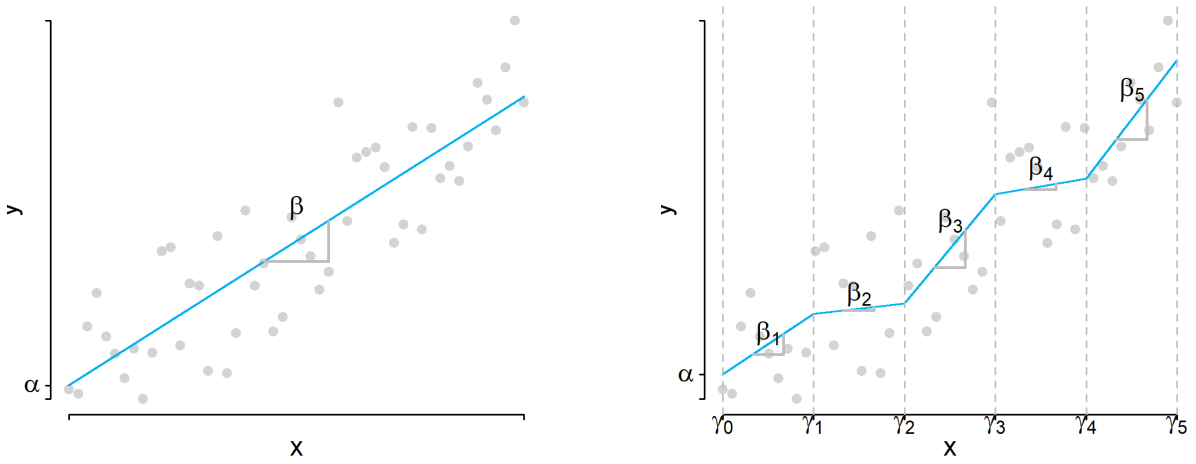
The goal of regression function estimation is to find a function which approximates the relation between two variables: the explanatory variable $x_i$ and the response variable $y_i$. The basis of the regression function that Neelon and Dunson estimate is a univariate normal regression model:

$$y_i = f(x_i) + \epsilon_i, \quad i = 1, \ldots, n \tag{3.1}$$

where the error terms $\varepsilon_i$ are independent and identically normally distributed, $\forall i\ \varepsilon_i \sim N(0, \sigma^2)$. A commonly used form for $f$ is an affine function $f(x) = \alpha + \beta x$. Then, the regression model is given by

$$y_i = \alpha + \beta x_i + \epsilon_i, \quad i = 1, \ldots, n \tag{3.2}$$

where $\alpha$ is the intercept and $\beta$ is the slope of the line. However, the functional form that Neelon and Dunson introduce for the function $f$ is a piece-wise linear isotonic function. In that case the function is not a single straight line but a composition of connected straight-line segments, which are non-decreasing. The endpoints of the straight-line segments lie on the so called knot locations. To indicate the domains for the line-segments, we specify the knot locations $\gamma = (\gamma_0, \ldots, \gamma_k)'$, with $x_i \in [\gamma_0, \gamma_k]\ \forall i = 1, \ldots, n$ and $\gamma_0 < \gamma_1 \ldots < \gamma_k$. So there are $k$ intervals on which they estimate these line-segments. The slope of each line-segment is given by $\beta_j$ which corresponds to interval $(\gamma_{j-1}, \gamma_j]$ (see figure 3.1b)




(a) A linear regression

(b) A piece-wise linear isotonic regression

Figure 3.1: Example of a linear regression function a simulated data set and an example of a piece-wise linear isotonic regression function with 6 knots. The $\beta$'s are the slopes.

The mathematical expression for the linear regression model is already given in (3.2), but the expression for the piece-wise linear isotonic regression model not yet. For a better understanding of the mathematical expression of a piece-wise linear isotonic model, we initially give the function per interval. The expression of a piece-wise linear function for $x \in (\gamma_0, \gamma_1]$ is

$$f(x) = \alpha + \beta_1(x - \gamma_0) \tag{3.3}$$

If $x \in (\gamma_1, \gamma_2]$ then the function is given by

$$f(x) = \alpha + \beta_1(\gamma_1 - \gamma_0) + \beta_2(x - \gamma_1) \tag{3.4}$$

We can continue like this and we notice that for $x \in [\gamma_0, \gamma_k]$ we can describe a piece-wise linear function given by:

$$f(x) \quad = \alpha + \sum_{j=1}^{k} w_j(x)\ \beta_j \tag{3.5}$$

with $w_j(x) = \min(x, \gamma_i) - \gamma_{i-1}$ if $x_i \geq \gamma_{i-1}$ and otherwise $w_j(x) = 0$. We will write $w_j(x_i)$ as $w_{ij}$. Now that we have an expression for a piece-wise linear function for all the $x$-values. The piece-wise linear isotonic model is given by

$$\begin{aligned} y_i &= \alpha + \sum_{j=1}^{k} w_{ij}\,\beta_j \quad + \varepsilon_i \\ &= w_i'\theta + \varepsilon_i \qquad \text{with } w_i = (1, w_{i1}, \ldots, w_{ik}) \end{aligned} \tag{3.6}$$

Important remarks are that $\beta_j \geq 0 \quad \forall i$ for ensuring the monotonicity and that $\theta$ is not the probability of throwing heads like it was in chapter 2. $\theta$ is defined here as $\theta = (\alpha, \beta)'$ with $\beta = (\beta_1, .., \beta_k)'$. Remember that the goal of Neelon and Dunson is to find a regression curve based on data $\{(x_i, y_i) : 1 \leq i \leq n\}$ . We can estimate a curve by estimating the parameters of (3.6). We can do that using a frequentist way by using the maximum likelihood estimator for example. Or we can use a Bayesian approach. For both methods the likelihood function is needed.

## 3.2 The likelihood function

Now that we have a statistical model, we can describe methods to estimate the parameters. Let us first derive the probability density function of $y$ conditional on $\theta$ . If $x_i, \theta,$ and $\sigma^2$ are fixed, notice from (3.6) that $y_i$ is a linear transformation of a normal random variable $\varepsilon_i$. A linear transformation of normal random variable is also normally distributed. So $y_i$ conditionally on the parameters is normally distributed with mean

$$\begin{aligned} \mathrm{E}(y_i|\theta,\ \sigma^{-2},\ x_i) &= \mathrm{E}(w_i'\theta + \varepsilon_i) \\ &= \mathrm{E}(w_i'\theta) + E(\varepsilon_i) \\ &= w_i'\theta \end{aligned} \tag{3.7}$$

and variance

$$\begin{aligned} \mathrm{Var}(y_i|\theta,\ \sigma^{2},\ x_i) &= \mathrm{Var}(w_i'\theta + \varepsilon_i) \\ &= \mathrm{Var}(\varepsilon_i) \\ &= \sigma^2 \end{aligned} \tag{3.8}$$

Then the probability density function for $y_i|\theta,\ \sigma^{-2},\ x_i$ is

$$p(y_i|\theta, \sigma^2, x_i) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{-\frac{1}{2\sigma^2}(y_i - w_i'\theta)^2\right\} \tag{3.9}$$

By independence of $y_1, , ..., y_n$ the joint probability density function of $(y_1, , ..., y_n)$ is the product of all the densities

$$p(y|\theta, \sigma^2, x) = \prod_{i=1}^{n} p(y_i|\theta) = \left(2\pi\sigma^2\right)^{-n/2} \exp\left\{-\frac{1}{2\sigma^2}\sum_{i=1}^{n}\left(y_i - w_i'\theta\right)^2\right\} \tag{3.10}$$

A frequentist approach would be now to find $\hat{\theta} = (\hat{\alpha}, \hat{\beta})$ and $\hat{\sigma}^2$ that maximize this probability density function in equation (3.10) as a function of $\theta$ and $\sigma^2$. However, Neelon and Dunson propose a Bayesian approach. Note that we can obtain a posterior distribution in the same manner like we did in the example of the coin tossing using Bayes rule (2.5), if we specify a prior distribution for $\alpha, \beta$ and $\sigma^2$. In the Bayesian approach we specifically want that posterior distribution of the parameters.

## 3.3 Prior density

### prior density of $\beta$

The monotone character of the regression curve is established by making sure that under the prior distribution we have $\beta_j \geq 0$ for all $j$ with probability 1. Neelon and Dunson do that by introducing a latent variable $\beta_j^*$ such that

$$\beta_j = 1_{(\beta_j^* \geq \delta)}\beta_j^* \tag{3.11}$$

where $\delta$ is a small positive contant. No matter what distribution you specify for $\beta_j^*$, even a prior with probability mass largely concentrated on negative values, is transformed to a prior which does not allow $\beta_j < 0$. All the probability mass for $\beta_j^* \leq \delta$ ends up as point mass at zero. That point mass at zero allows for flat regions of the corresponding function (3.5). The distribution they specify for $\beta_j^*$ is a normal distribution that depends on the previous slope:

$$\beta_j^*|\beta_{j-1}^* \sim N(\beta_{j-1}^*, \lambda^{-1}) \tag{3.12}$$

where the variance is given by parameter $\lambda^{-1}$; $\lambda$ is the smoothing hyperparameter. How strongly $\beta_j^*$ is correlated to the previous slope $\beta_{j-1}^*$ is specified by this hyperparameter. If $\lambda$ is large then consecutive slopes will be close to each other, which will make the curve smoother. The parameter $\lambda$ is also given a prior distribution as we will show in the next section where the priors for the other parameters will be addressed. For $\beta_1^*$ there is no previous slope, so the distribution is normally distributed with investigator-specified parameters $E_{01}$ as the best guess for the slope of the curve and $V_{01}$ as the uncertainty in this best guess. The joint prior density of $\beta_1^*$ and $\beta_2^*$ is

$$p(\beta_1^*, \beta_2^*) = p(\beta_2^*|\beta_1^*)p(\beta_1^*) \tag{3.13}$$

where $p$ is the probability density function. Small $p$ is used throughout this report as a general notation for a probability density function of what is inside the brackets. In this case (for the latent slope parameter $\beta_j^*$) we have that $p$ is the probability density function of a normal distribution (3.12). We can continue this for the joint distribution of the first three $\beta_j$'s

$$p(\beta_1^*, \beta_2^*, \beta_3^*) = p(\beta_2^*, \beta_3^*|\beta_1^*)p(\beta_1^*) = p(\beta_3^*|\beta_2^*, \beta_1^*)p(\beta_2^*|\beta_1^*)p(\beta_1^*) = p(\beta_3^*|\beta_2^*)p(\beta_2^*|\beta_1^*)p(\beta_1^*) \tag{3.14}$$

using that $\beta_3^*|\beta_2^*, \beta_1^*$ does not depend on $\beta_1^*$. We can do this for $k$ beta's. In this way we obtain a the joint prior distribution of $\beta^*$:

$$\begin{aligned} p(\beta^*) = p(\beta_1^*, ..., \beta_k^*) &= p(\beta_1) \prod_{j=2}^{k} p(\beta_j|\beta_{j-1}) \\ &= (2\pi V_{01})^{-\frac{1}{2}} \exp\left\{-\frac{1}{2V_{01}}(\beta_1^* - E_{01})^2\right\} \prod_{j=2}^{k} \left(2\pi\lambda^{-1}\right)^{-\frac{1}{2}} \exp\left\{-\frac{1}{2}\lambda(\beta_j^* - \beta_{j-1}^*)^2\right\} \end{aligned} \tag{3.15}$$

Neelon and Dunson denote the probability density function of a random variable $X \sim N(\mu, \sigma^2)$ by $N(x; \mu, \sigma^2)$, so that is why Neelon and Dunson denote equation (3.15) as

$$p(\beta^*) = N\left(\beta_1^*; E_{01}, V_{01}\right) \prod_{j=2}^{k} N\left(\beta_j^*; \beta_{j-1}^*, \lambda^{-1}\right) \tag{3.16}$$

If we set $E_{0j} = \beta_{j-1}^*$ and $\mathrm{V}_{0j} = \lambda^{-1}$ for $j = 2, \ldots, k$, then the joint probability density of $\beta$ and $\beta^*$ can be expressed as follows according to Neelon and Dunson

$$\begin{aligned} p(\beta, \beta^*) &= \prod_{j=1}^{k} p\left(\beta_j \mid \beta_j^*\right) p\left(\beta_j^* \mid \beta_{j-1}^*\right) \\ &= \prod_{j=1}^{k} \left\{1_{(\beta_j = 0)} 1_{(\beta_j^* < \delta)} + 1_{(\beta_j = \beta_j^*)} 1_{(\beta_j^* \geq \delta)}\right\} N\left(\beta_j^*; E_{0j}, V_{0j}\right) \end{aligned} \tag{3.17}$$

This joint prior of $\beta$ and $\beta^*$, can be confusing, because if we assume $\delta$ to be fixed and $\beta_j^*$ is conditional on $\beta_j$, $\beta_j$ is just a function of $\beta_j^*$. So $p(\beta_j|\beta_j^*)$ is confusing in the sense that it is not really a probability density. It might be more convenient for the understanding to avoid the joint distribution of $\beta$ and $\beta^*$, but we show how Neelon and Dunson denote it, because the posterior computation of Neelon and Dunson also involves a joint posterior distribution of $\beta$ and $\beta^*$. So important remark to equation (3.17) is that the extended parameter $(\beta, \beta_{*)}$ does not have a probability density with respect to the $2k$-dimensional Lebesgue measure.

How you should interpret this joint distribution will be explained. If we want to draw a sample from this density we first draw a $\beta^*$ in $\mathbb{R}^k$. This single sample is a vector of $k$ real

numbers as elements. Each element is a sample from $N\left(E_{0j}, V_{0j}\right)$. Then we duplicate this vector. For every element in the duplicate which is smaller than $\delta$, the element is set equal to 0. This adjusted duplicate of the vector plus the original vector is a sample from the joint distribution of $\beta$ and $\beta^*$, which has dimension $2k$. This adjustment per element of the duplicate vector is given by the indicator function between braces in the second line of equation (3.17). You should read it as follows

$$\text{if } \beta_j^* < \delta, \text{ then } \beta_j = 0$$
$$\text{if } \beta_j^* \geq \delta, \text{ then } \beta_j = \beta_j^*$$

Then it becomes more comprehensible what this joint prior distribution in (3.17) means.

### Prior density of $\alpha, \lambda, \delta, \sigma^{-2}$

Neelon and Dunson specify conjugate prior distributions for the parameters. Loosely speaking, conjugate priors have the property that the posterior distribution is in the same probability distribution family. So a conjugate normal prior implies that the posterior distribution will be normal, but probably with other parameters. In this case, we mean by "conjugate" not that the posterior is in the same probability distribution family, but the full conditional posterior distribution is in the same probability distribution family. We have not discussed full conditional posterior distributions yet, but they will be important for the computation of the posterior distribution in the section about Gibbs Sampling. The prior distributions that Neelon and Dunson specify for $\alpha, \lambda, \delta, \sigma^{-2}$ are

$$\alpha \sim N(\alpha_0, \sigma_\alpha^2) \quad \lambda \sim \text{Gamma}(c_1, d_1) \quad \delta \sim \text{Gamma}(c_2, d_2) \quad \sigma^{-2} \sim \text{Gamma}(a, b) \tag{3.18}$$

However, in this report we do not use their method to specify a distribution for $\delta$. Instead of that we keep $\delta$ fixed at $\delta = 0.0001$. A prior distribution for $\delta$ is also less natural. It is more a 'tuning parameter' than a parameter which we estimate from data. That we do not have a prior distribution for $\delta$ has implications for the computation of the posterior.

## 3.4 Posterior computation

Now that we have given the prior distribution and the likelihood function it we would like to obtain the joint posterior distribution of the parameters or even better the marginal distributions of the parameters, because it is not easy to sample from a joint posterior distribution. The joint posterior density can be expressed as follows:

$$\begin{aligned}
p(\alpha, \beta, \beta^*, \lambda, \delta, \sigma^{-2}|y) &= \frac{p(y|\alpha, \beta, \beta^*, \lambda, \delta, \sigma^{-2}) p(\alpha, \beta, \beta^*, \sigma^{-2}, \lambda, \delta)}{p(y)} \\
&= \frac{p(y|\theta, \sigma^2, x) p(\beta, \beta^*) p(\alpha, \sigma^{-2}, \lambda, \delta)}{\int_\Omega p(y|\theta, \sigma^2, x) p(\beta, \beta^*) p(\alpha, \sigma^{-2}, \lambda, \delta) d\eta} \\
&= \frac{p(y|\theta, \sigma^2, x) \left[ \prod_{j=1}^k \{1_{(\beta_j=0)} 1_{(\beta_j^*<\delta)} + 1_{(\beta_j=\beta_j^*)} 1_{(\beta_j^*\geq\delta)}\} \times N(\beta_j^*; E_{0j}, V_{0j}) \right] p(\alpha, \sigma^{-2}, \lambda, \delta)}{\int_\Omega p(y|\theta, \sigma^2, x) \left[ \prod_{j=1}^k \{1_{(\beta_j=0)} 1_{(\beta_j^*<\delta)} + 1_{(\beta_j=\beta_j^*)} 1_{(\beta_j^*\geq\delta)}\} \times N(\beta_j^*; E_{0j}, V_{0j}) \right] p(\alpha, \sigma^{-2}, \lambda, \delta) d\eta}
\end{aligned} \tag{3.19}$$

$\Omega$ is the parameter space of $\eta$, which is a vector notation for all the parameters $\alpha, \beta, \beta^*, \lambda, \delta, \sigma^{-2}$ Now it is very hard analytically find a closed form for the distribution of the data in the denominator and therefore it is is also hard to find an analytical expression for the posterior

distribution (3.19). Neelon and Dunson mention to use the Metropolis Hastings Algorithm to approximate the posterior distribution, since they give $\delta$ a prior distribution. In this thesis we choose a particular value for $\delta$. So we do not specify a prior distribution for $\delta$. In that case, the Gibbs sampling algorithm is sufficient, since we can derive all the full conditional posterior distributions then. The next section will explain what the full conditional posteriors are and how we use them to compute a posterior.

## 3.5 Gibbs sampling algorithm

We explain how Gibbs sampling works for the general case. The general case it that you are given data $y$ and $\theta = (\theta_1, \theta_2, ...., \theta_k)$ is the vector of all parameters in our model, which are all unknown. Then we explain how we obtain the posterior distribution (or actually an approximate posterior distribution) of this $\theta$, which is given by $p(\theta|y)$.

The full conditional posterior distribution of a parameter $\theta_j$ (or shorthand the full conditional) is the conditional distribution of $\theta_j$ given all other parameters. So the full conditional of $\theta_1$, for example, is given by $p(\theta_1 \mid \theta_2, \ldots, \theta_k, y)$.

If one can derive all the full conditional posterior distributions of all the parameters, we can use the Gibbs sampling algorithm to approximate the posterior distribution of $\theta$. So in other words, if we can find $p(\theta_j|, \theta_{(-j)}, y)$ where $\theta_{(-j)} = \theta \backslash \{\theta_j\}$ for all $\theta_j$ with $j = 1, \ldots, k$, then we can approximate $p(\theta \mid y)$. The Gibbs sampling algorithm uses those conditional distributions subsequently to sample from in the following way:

**Algorithm 1** Gibbs Sampling Algorithm

$n$ <- 10000
burnin <- 1500
declare $\theta_1^{(1)}, \theta_2^{(1)}, ..., \theta_k^{(1)}$

**For** $i = 2$ to $n$ do:
$\theta_1^{(i)} \sim p(\theta_1|\theta_2^{(i-1)}, ..., \theta_k^{(i-1)}, y)$
$\theta_2^{(i)} \sim p(\theta_2|\theta_1^{(i)}, \theta_3^{(i-1)}, ..\theta_k^{(i-1)}, y)$
$\vdots$
$\theta_k^{(i)} \sim p(\theta_k|\theta_1^{(i)}, \theta_2^{(i)}, ..\theta_{k-1}^{(i)}, y)$
**end**

In this way for each of parameter $\theta_j$, a sequence $\left(\theta_j^{(i)}\right)_{i=1}^{n}$ is created. For each sequence, the empirical distribution of that sequence converges to its posterior marginal distribution for large $n$. So after a sufficient number of iterations, each draw $\theta_j^{(i)}$ is by approximation a sample from $p(\theta_j \mid y)$. If we put the draw of each parameter per iteration together, each draw $\theta^{(i)} = \left(\theta_1^{(i)}, \ldots, \theta_k^{(i)}\right)$ is by approximation a sample from $p(\theta|y)$.

### 3.5.1 Posterior mean

After the simulation using Gibbs sampling we have a sequence for each parameter. Each sequence represents samples from the marginal posterior distribution of that parameter. As we saw in the coin tossing example in chapter 2 we had an analytical form for the posterior. Then, if we wanted to make a point-wise estimate using the posterior, we computed the expectation

of the (in this case one-dimensional) parameter $\theta$. This same point-wise estimate for the parameters in this case can not be computed exactly. We can not compute the expectation of a parameter without an analytical form for the probability density. So instead we take the mean of each sequence after the burn-in period, which is approximately the expectation of the marginal posterior. Then the point-wise estimate denoted by $\hat{\theta}_j$ for each parameter $\theta_j$ is given by:

$$\hat{\theta}_j = \mathrm{E}[\theta_j] = \int \theta_j p(\theta_j \mid \mathbf{X}) d\theta_j \approx \frac{1}{n-s+1} \sum_{i=s}^{n} \theta_j^{(i)} \tag{3.20}$$

where $s$ is the first iteration after the burn-in. The burn-in period is the first set of values from the sequence which are "unrepresentative" samples. We mean by "unrepresentative" that these samples together do not form a good representation of the posterior distribution.

### 3.5.2 Full conditional posterior distributions

We have seen that using the Gibbs sampling algorithm we can approximate the posterior distribution using the full conditional posterior distributions of the parameters. All full conditionals of this model can be derived from the posterior density kernel, which is the posterior in equation (3.19) without the normalization factor in the denominator:

$$p(\alpha, \beta, \beta^*, \lambda, \delta, \sigma^{-2} | y) \propto p(y|\theta, \sigma^2, x) \left[ \Pi_{j=1}^{k} \{ 1_{(\beta_j = 0)} 1_{(\beta_j^* < \delta)} + 1_{(\beta_j = \beta_j^*)} 1_{(\beta_j^* \geq \delta)} \} \times N(\beta_j^*; E_{0j}, V_{0j}) \right] p(\alpha, \sigma^{-2}, \lambda, \delta) \tag{3.21}$$

As an example, we derive the full conditional distribution of $\lambda$. The derivations for $\alpha$ and $\sigma^2$ can be done likewise, and therefore we just give the full conditionals of $\alpha$ and $\sigma^2$ in Appendix 7.2.

$$\begin{aligned}
p(\lambda|\alpha, \beta, \beta^*, \delta, \sigma^{-2}, y) &\propto (\Pi_{j=1}^{k} N(\beta_j^*; E_{0j}, V_{0j})) p(\lambda) \\
&\propto \left( \Pi_{j=2}^{k} \frac{1}{\sqrt{2\pi\lambda^{-1}}} \exp\left\{ \frac{-(\beta_j^* - \beta_{j-1}^*)^2}{2\lambda^{-1}} \right\} \right) \left( \frac{d_1^{c_1}}{\Gamma(c_1)} \lambda^{c_1 - 1} \exp\{-d_1 \lambda\} \right) \\
&= \left( \frac{(\sqrt{\lambda})^{k-1}}{(\sqrt{2\pi})^{k-1}} \exp\left\{ -\frac{\lambda}{2} \sum_{j=2}^{k} (\beta_j^* - \beta_{j-1}^*)^2 \right\} \right) \left( \frac{d_1^{c_1}}{\Gamma(c_1)} \lambda^{c_1 - 1} \exp\{-d_1 \lambda\} \right) \\
&\propto \left( (\sqrt{\lambda})^{k-1} \exp\left\{ -\frac{\lambda}{2} \sum_{j=2}^{k} (\beta_j^* - \beta_{j-1}^*)^2 \right\} \right) \left( \lambda^{c_1 - 1} \exp\{-d_1 \lambda\} \right) \\
&= \lambda^{c_1 + \frac{k-1}{2} - 1} \exp\left\{ -(d_1 + \frac{1}{2} \sum_{j=2}^{k} (\beta_j^* - \beta_{j-1}^*)^2) \lambda \right\}
\end{aligned} \tag{3.22}$$

We recognize here a gamma probability density. So the full conditional posterior for parameter $\lambda$ is

$$\mathrm{Gamma}\left( c_1 + \frac{k-1}{2}, d_1 + \frac{1}{2} \sum_{j=2}^{k} (\beta_j^* - \beta_{j-1}^*)^2 \right) \tag{3.23}$$

How the full conditional posterior distribution for $\beta$ and $\beta^*$ is derived will be given as well to support the understanding of the posterior distribution of $\beta$ and $\beta^*$. Neelon and Dunson derive the full conditional of $\beta_j$ and $\beta_j^*$ in the following three steps to get a convenient form. By a convenient form we mean a form such from which we know how to sample from. They do not explicitly show how the steps are done. The foundation of the steps are given in Appendix 7.3.

$$p(\alpha,\beta,\beta^*,\lambda,\delta,\sigma^{-2}|y) = \frac{p(y|\alpha,\beta,\beta^*,\lambda,\delta,\sigma^{-2})p(\alpha,\beta,\beta^*,\sigma^{-2},\lambda,\delta)}{p(y)} \tag{3.24}$$

$$\Downarrow \text{ Step A}$$

$$p(\beta_j,\beta_j^*|\alpha,\lambda,\sigma^{-2},y,x) \propto p(y|\theta,\sigma^2,x)\left[\prod_{j=1}^{k}\left\{1_{(\beta_j=0)}1_{(\beta_j^*<\delta)} + 1_{(\beta_j=\beta_j^*)}1_{(\beta_j^*\geq\delta)}\right\} \times N(\beta_j^*;E_{0j},V_{0j})\right] \tag{3.25}$$

$$\Downarrow \text{ Step B}$$

$$\begin{aligned} p(\beta_j,\beta_j^*|\alpha,\lambda,\sigma^{-2},y,x) \propto & \left[\prod_{i=1}^{n} N(y_{ij}^*;w_{ij}\beta_j,\sigma^2)\right]\left\{1_{(\beta_j=0)}1_{(\beta_j^*<\delta)} + 1_{(\beta_j=\beta_j^*)}1_{(\beta_j^*\geq\delta)}\right\} \\ & \times N(\beta_j^*;E_{0j},V_{0j})N(\beta_{j+1}^*;\beta_j^*,\lambda^{-1}) \end{aligned} \tag{3.26}$$

$$\Downarrow \text{ Step C}$$

$$p(\beta_j,\beta_j^*|\alpha,\lambda,\sigma^{-2},y,x) \propto 1_{(\beta_j=0)}1_{(\beta_j^*<\delta)}\left(\frac{N(\beta_j^*;\tilde{E}_{0j},\tilde{V}_{0j})}{N(0;\tilde{E}_{0j},\tilde{V}_{0j})}\right) + 1_{(\beta_j=\beta_j^*)}1_{(\beta_j^*\geq\delta)}\left(\frac{N(\beta_j^*;\hat{E}_j,\hat{V}_j)}{N(0;\hat{E}_j,\hat{V}_j)}\right) \tag{3.27}$$

$$\begin{aligned} \tilde{V}_{0j} &= \left(V_{0j}^{-1} + \lambda\right)^{-1} \\ \tilde{E}_{0j} &= \tilde{V}_{0j}\left(V_{0j}^{-1}E_{0j} + \lambda\beta_{j+1}^*\right) \\ \hat{V}_j &= \left(\tilde{V}_{0j}^{-1} + \sigma^{-2}\sum_{i=1}^{n} w_{ij}^2\right)^{-1} \\ \hat{E}_j &= \hat{V}_j\left(\tilde{V}_{0j}^{-1}\tilde{E}_{j0} + \sigma^{-2}\sum_{i=1}^{n} w_{ij}y_{ij}^*\right) \end{aligned}$$

Equation (3.27) shows that the full conditional posterior distribution of $(\beta_j,\beta_j^*)$ is a mixture distribution of truncated normal distributions. Again, there are some remarks we have to make similar to the remarks about the prior distribution in equation (3.17). First of all, the distribution of $(\beta_j,\beta_j^*)$ is restricted to a subspace of $\mathbb{R}^2$. This subspace is given by $\{0\}\times(-\infty,\delta)\cup\{(u,u) : u \geq \delta\}$. If you would like to sample from equation (3.27), then you draw $\beta_j^*$ from a mixture distribution and $\beta_j$ is constructed using the functional form between them. Neelon and Dunson say that it immediately follows how you should sample $\beta_j$ and $\beta_j^*$ from (3.27). It might not be immediately obvious for the reader who is not familiar with a mixture of distributions. So we continue to discuss how to sample $(\beta_j,\beta_j^*)$ from (3.27) in section 4.1, when we will see that numerical problems occur if we sample from this distribution.

### 3.5.3 Gibbs sampling for our model

When all the full conditional posteriors are derived we can apply the Gibbs sampling algorithm to obtain approximate marginal posterior distributions for each of the parameters $\alpha,\beta,\beta^*,\lambda,\sigma^{-2}$. The Gibbs sampling algorithm for our model is given in pseudo code.

**Algorithm 2** Gibbs Sampling Neelon and Dunson

iter ← 10000
burnin ← 1500

initialize $\sigma^{(1)}, \alpha^{(1)}, \lambda^{(1)}, \beta^{(1)}, \beta^{*(1)}$ ▷ initial values

**For** i in 2:iter
$\sigma^{-2(i)} \sim \text{Gamma}(a + \frac{n}{2}, b + \frac{1}{2}(y - W\theta^{(i-1)})'(y - W\theta^{(i-1)}))$
$\alpha^{(i)} \sim N(\hat{\alpha}^{(i)}, \hat{\sigma}_{\alpha}^{2(i)})$ ▷ with $\hat{\alpha}^{(i)} \sim \sigma^{-2(i)}$ en $\hat{\sigma}_{\alpha}^{2(i)} \sim \hat{\alpha}^{(i)}, \sigma^{-2(i)}$ [1]
$\lambda^{(i)} \sim \text{Gamma}(c_1 + \frac{k-1}{2}, d_1 + \frac{1}{2}\sum_{j=2}^{k}(\beta_j^{*(i-1)} - \beta_{j-1}^{*}{}^{(i-1)})^2)$
▷ vector $\beta^{(i)}$ and $\beta^{*(i)}$
**For** i in 1:k
$\beta_j^{*(i)} \sim N_{-\infty}^{\delta}\left(\tilde{E}_{0j}\tilde{V}_{0j}\right)$ with probability $\frac{A}{C}$ [2] [3]
$\sim N_{\delta}^{\infty}\left(\hat{E}_j, \hat{V}_j\right)$ with probability $\frac{B}{C}$
$\beta_j^{(i)} = 1_{(\beta_j^{*} \geq \delta)}\beta_j^{*(i)}$
**end**
**end**

[1] $\hat{\alpha}$ and $\hat{\sigma}_{\alpha}^{2}$ are given in Appendix 7.2
[2] $\tilde{E}_{0j}, \tilde{V}_{0j}, \hat{E}_j$ and $\hat{V}_j$ are given in equation (3.27)
[3] How $\beta_j$ and $\beta_j^{*}$ are drawn is also described in section 4.1

# Chapter 4

# Applying the method

In this chapter we apply the method proposed by Neelon and Dunson to two climate data sets and we discuss some problems that occur when one wants to apply the method described in chapter 3 in practice. We also present our own way how to deal with those problems.

## 4.1 Sampling the slope coefficients from a mixture distribution

### 4.1.1 Sampling from the a mixture distribution

During the Gibbs sampling algorithm we sample $\alpha, \lambda$ en $\sigma^2$ from a 1-dimensional posterior distribution. For $\beta_j$ and $\beta_j^*$ derived a joint posterior, which is 2-dimensional (3.27). How we sample from (3.27) is not immediately obvious. What we do is we sample $\beta_j^*$ from a mixture distribution and then set $\beta_j = 1_{(\beta_j^* \geq \delta)}\beta_j^{*(i)}$. From (3.27) we can derive that

$$p(\beta_j^*|\alpha, \lambda, \sigma^{-2}, y, x) \propto \frac{F\left(\delta; \tilde{E}_{0j}, \tilde{V}_{0j}\right)}{N\left(0; \tilde{E}_{0j}, \tilde{V}_{0j}\right)} 1_{(\beta_j^* < \delta)} \left(N(\beta_j^*; \tilde{E}_{0j}, \tilde{V}_{0j})\right) + \frac{1 - F\left(\delta; \hat{E}_j, \hat{V}_j\right)}{N\left(0, \hat{E}_j, \hat{V}_j\right)} 1_{(\beta_j^* \geq \delta)} \left(N(\beta_j^*; \hat{E}_j, \hat{V}_j)\right) \tag{4.1}$$

where $F(x; \mathrm{a}, \mathrm{b})$ is the cumulative distribution function value in the point $x$ of a random variable $X \sim N(\mathrm{a}, \mathrm{b})$. We recognize the mixture density function $f$ of the form

$$f(x) = \sum_{i=1}^{n} w_i p_i(x) \tag{4.2}$$

with $w_i \geq 0$ and $\sum w_i = 1$ and $p_i$ a probability density function. If we sample from a mixture density function $f$ we sample with probability $w_i$. Note that $w_i$ is very different from the $w_i$ in the piece-wise linear isotonic function (3.1b). To make sure that the weights in equation (4.1) count up to 1 we normalize the density by C, which is given by

$$C = \frac{F\left(\delta; \tilde{E}_{0j}, \tilde{V}_{0j}\right)}{N\left(0; \tilde{E}_{0j}, \tilde{V}_{0j}\right)} + \frac{1 - F\left(\delta; \hat{E}_j, \hat{V}_j\right)}{N\left(0, \hat{E}_j, \hat{V}_j\right)} = A + B \tag{4.3}$$

So then the posterior distribution for $\beta_j^*$ that we sample from is

$$p(\beta_j^*|\alpha, \lambda, \sigma^{-2}, y, x) \propto \frac{A}{C} \cdot 1_{(\beta_j^* < \delta)} \left(N(\beta_j^*; \tilde{E}_{0j}, \tilde{V}_{0j})\right) + \frac{B}{C} \cdot 1_{(\beta_j^* \geq \delta)} \left(N(\beta_j^*; \hat{E}_j, \hat{V}_j)\right) \tag{4.4}$$

As we can see from this expression (4.4), we sample from $\beta_j^*$ from a mixture of two truncated normal distributions. We sample $\beta_j^*$ with probability $A/C$ from $N_{-\infty}^{\delta}(\tilde{E}_{0j}, \tilde{V}_{0j})$ and with proba-

bility $B/C$ from $N_\delta^\infty(\hat{E}_j, \hat{V}_j)$. $N_\delta^\infty$ is a notation for a normal distribution, which truncated below by $\delta$. $N_\infty^\delta$ then denotes the normal distribution that is truncated above by $\delta$.

### 4.1.2 Mills ratio for a numerical problem

Numerical problems occur when we compute the weights $A$ and $B$ from equation (4.3). Every iteration $\tilde{E}_{j0}, \hat{E}_j, \hat{V}_j$ en $\tilde{V}_{0j}$ are updated and it often occurs that if $\hat{E}_j$ is very negative and $\hat{V}_j$ is very small. Then the weight $B$ can not be computed numerically in R, because R interprets this as $B = \frac{1-F\left(\delta;\hat{E}_j,\hat{V}_j\right)}{N\left(0;\hat{E}_j,\hat{V}_j\right)} \approx \frac{1-1}{0} = \frac{0}{0}$. Actually the numerator and denominator are very small numbers which R stores as 0. The ratio can nevertheless be well defined. To compute the ratio we rewrite B using the Mills ratio function and then approximate the Mills ratio using an asymptotic expansion from the literature. The Mills ratio of a continuous random variable $X$ is defined as

$$m_X(x) := \frac{\bar{F}(x)}{f(x)} \tag{4.5}$$

where $f$ is the probability density function of X and $\bar{F}(x) = 1 - F(x)$ with $F(x)$ the cumulative distribution function of $X$. (4.5) is well defined for values of $x$ such that $f(x) > 0$. We rewrite $B$ from equation (4.3) as follows if $Y \sim N(\hat{E}_J, \hat{V}_j)$:

$$B = \frac{1 - F\left(\delta; \hat{E}_j, \hat{V}_j\right)}{N\left(0, \hat{E}_j, \hat{V}_j\right)} = \frac{\bar{F}_Y(\delta)}{f_Y(0)} = \frac{\bar{F}_Y(\delta)}{f_Y(\delta)} \cdot \frac{f_Y(\delta)}{f_Y(0)} = m_Y(\delta) \cdot \frac{f_Y(\delta)}{f_Y(0)} \tag{4.6}$$

$m$ is the Mills ratio for random variable $Y$. $f$ is the normal probability density function for $Y$. If we write out the fraction of densities in equation (4.6), then

$$\frac{f_Y(\delta)}{f_Y(0)} = \frac{(2\pi\hat{V}_j)^{-\frac{1}{2}} e^{-\frac{1}{2}(\delta-\hat{E}_j)^2/\hat{V}_j}}{(2\pi\hat{V}_j)^{-\frac{1}{2}} e^{-\frac{1}{2}\hat{E}_j^2/\hat{V}_j}} = e^{-\frac{1}{2}(\delta-\hat{E}_j)^2/\hat{V}_j + \frac{1}{2}\hat{E}_j^2/\hat{V}_j} = e^{-\frac{1}{2}\delta^2/\hat{V}_j + \delta\hat{E}_j/\hat{V}_j} \tag{4.7}$$

If we furthermore transform the Mills ratio for $Y$ in equation (4.6) to the Mills ratio for a standard normally distributed random variable $X$ we get

$$m_Y(\delta) = \frac{P(Y > \delta)}{f_Y(\delta)} = \frac{P\left(\frac{Y-\hat{E}_j}{\sqrt{\hat{V}_j}} > \frac{\delta-\hat{E}_j}{\sqrt{\hat{V}_j}}\right)}{\frac{1}{\sqrt{\hat{V}_j}} f_X\left(\frac{\delta-\hat{E}_j}{\sqrt{\hat{V}_j}}\right)} = \frac{\bar{F}_X\left(\frac{\delta-\hat{E}_j}{\sqrt{\hat{V}_j}}\right)}{\frac{1}{\sqrt{\hat{V}_j}} f_X\left(\frac{\delta-\hat{E}_j}{\sqrt{\hat{V}_j}}\right)} = \sqrt{\hat{V}_j} \cdot m_X\left(\frac{\delta - \hat{E}_j}{\sqrt{\hat{V}_j}}\right) \tag{4.8}$$

Then, using (4.6) and the expression we found for $\frac{f_Y(\delta)}{f_Y(0)}$ in (4.7) and $m_Y(\delta)$ in equation (4.8), we obtain the following expression for $B$

$$B = \sqrt{V} \cdot e^{-\frac{1}{2}\delta^2/V + \delta E/V} \cdot m_X\left(\frac{\delta - E}{\sqrt{V}}\right) \tag{4.9}$$

Calculating this expression using R does not solve our problem, but now we have a convenient expression to approximate $B$. If $\hat{E}_j$ is very negative and $\hat{V}_j$ is small then the argument in Mills ratio function becomes very large. In figure 4.1 we see the output in R. The figure shows that for arguments of approximately 8 the Mills ratio curve starts to oscillate. And for even bigger arguments the Mills ratio function does not produce output anymore, while it theoretically does have values: R can't handle the fraction of such small numbers. That is why we choose to approximate the Mills ratio for large values. For small values we can just use the definition of the Mills ratio. We approximate the Mills ratio using an asymptotic expansion. The Mills ratio

function can be asymptotically expanded as follows (Ruben, 1963) :

$$m_X(x) \sim \frac{1}{x} - \frac{1}{x^3} + \frac{1 \cdot 3}{x^5} - \frac{1 \cdot 3 \cdot 5}{x^7} + \cdots \ (x \to \infty) \tag{4.10}$$

We choose accordingly the following approximation for large arguments ($x > 7$) for the Mills ratio:

$$m_X(x) \approx \frac{1}{x} - \frac{1}{x^3} + \frac{1 \cdot 3}{x^5} - \frac{1 \cdot 3 \cdot 5}{x^7} \tag{4.11}$$

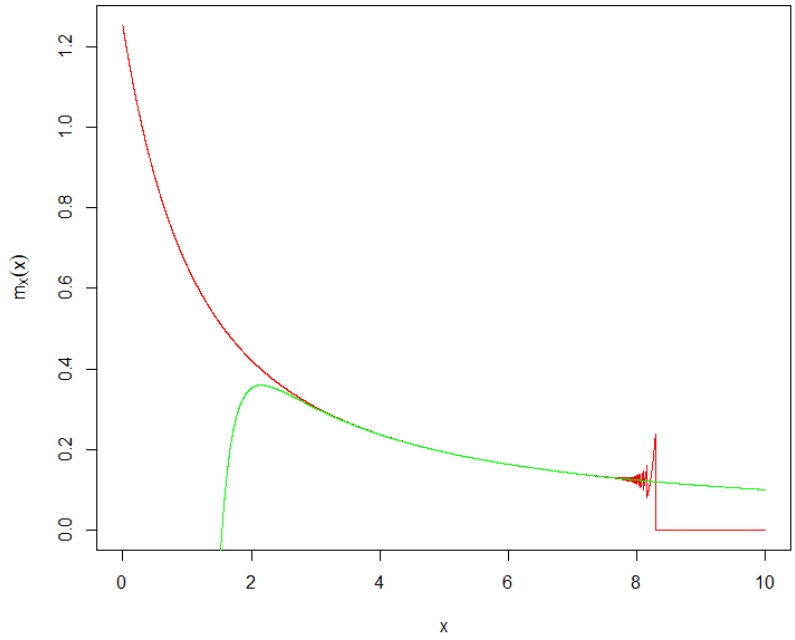



Figure 4.1: The red line is the output of Mills ratio function in R using its definition. The green line is the asymptotic approximation in equation (4.11). For $x$ between 0 and 3 the approximation is not accurate yet. For $x$ between 3 and 7 the approximation is close, but since the Mills ratio can still be computed by R we use the definition of the Mills ratio for arguments up until 7. For arguments larger than 7 we trust the approximation.

### 4.1.3 Infinite weight

Another numerical problem occurs when $\tilde{E}_{0j}$ becomes large and $\tilde{V}_{0j}$ becomes small. We get that the numerator for $A$ in equation (4.3) is close to 1 and that the denominator is close to 0. R returns a value stating that $A$ is infinitely large. If $B$ is equally big, then it is interesting find the ratio between the two and normalize them. In practice, we find that $B$ is very small. That has to do with the fact that $\hat{E}_j$ and $\hat{V}_j$ depend on $\tilde{E}_{0j}$ and $\tilde{V}_{0j}$ respectively. So if $\tilde{E}_{0j}$ becomes large and $\tilde{V}_{0j}$ is small such that R returns infinity we set $A$ equal to 1 and $B$ equal to 0. In that case, we always sample from the normal distribution bounded above by $\delta$.

## 4.2 Boundary problem

When we apply our Gibbs sampling algorithm we encounter some more problems with estimating the regression curve when we plot them. The estimated regression curve is piece-wise linear isotonic function, with as estimated parameters the posterior means (see equation (3.20)). So the estimated regression curve is

$$\hat{f}(x) \quad = \hat{\alpha} + \sum_{j=1}^{k} w_j(x)\ \hat{\beta}_j \tag{4.12}$$

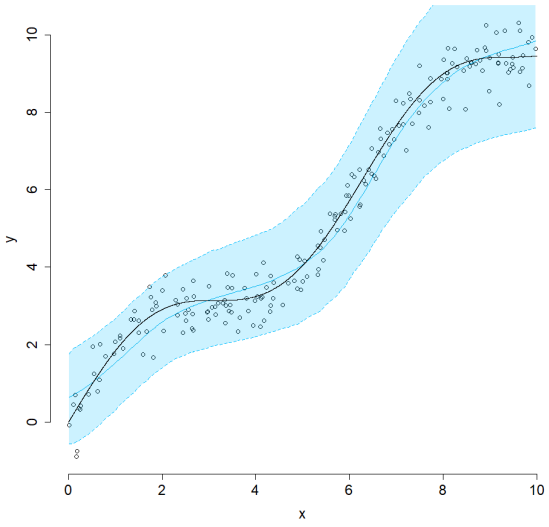
The problem shows up already when we use simulated data to perform our regression on. We use the following test functions for the simulated data:

1. $f_1(x) = x + \sin(x)$
2. $f_2(x) = 3 + 0.5 \cdot 1_{(8,\infty)}(x)$

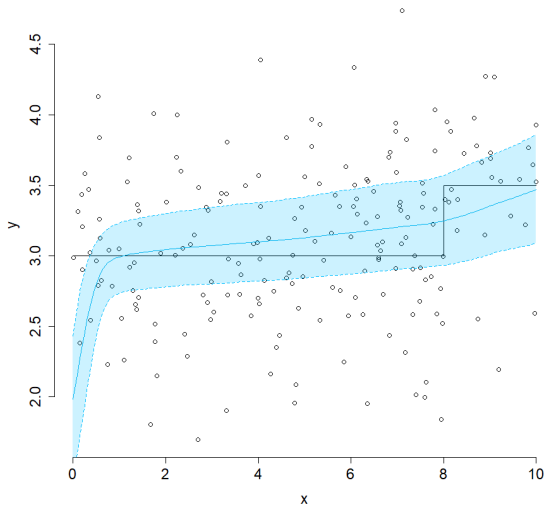
For both functions we simulate a data set of 200 points to with an error which is normally distributed with mean 0 and variance $0.5^2$. We used the same investigator- specified parameters as Neelon and Dunson do in their article, because they also use them for testing on these functions, but also because they give vague prior distributions. By vague we mean that the variance of the prior distribution is large. A prior with large variance is chosen when we have little prior beliefs about the parameters. They specified

$$\alpha_0 = 0.0,\ \ \sigma_\alpha^2 = 10,\ \ a = b = 0.1, E_{01} = 0.0,\ \ \ V_{01} = 10.$$

From we this relatively large $\sigma_\alpha$ and $V_{01}$ we can deduce that the prior distribution of $\alpha$ and $\beta_1$ are relatively vague. They also specified parameters for the prior distribution of $\delta$, but remember we let $\delta$ be fixed at 0.0001. The results are given in figure 4.2




(a) Data were generated as $X \sim U(0,10), Y = X + \sin(X) + \varepsilon, \varepsilon \sim N(0, 0.5^2)$




(b) Data were generated as $X \sim U(0,10), Y = 3 + 0.5 \cdot 1_{(8,\infty)}(x) + \varepsilon, \varepsilon \sim N(0, 0.5^2)$

Figure 4.2: Blue line is the curve estimate and the blue shadow is the 95% credible interval. The black line is the true curve.

The curve estimate of function $f_1$ is rather close to the true curve (figure 4.2a). The curve estimate of function $f_2$ in figure 4.2b displays a problem we encounter more generally: at the boundary we encounter a steeply upward sloping curve. For $x \in [0, 1]$ the estimated regression curve is definitely not close to the true curve. For the average temperature and the average winter temperature we observe the same phenomenon (figure 4.3). Even though we do not know the true curve for the climate data, we know by common sense that the overall average temperature and the average winter temperature do not increase so rapidly in time interval $[1900, 1905]$ and $[1700, 1710]$ respectively. After the steep slope interval, the estimated curve appears to be a good fit for both of the climate data sets.


(a) Average temperature 1901-2020

(b) Average winter temperature 1701-2014

Figure 4.3: Blue line is the curve estimate and the blue shadow is the 95% credible interval

All the curve estimates that have this steep slope (figure 4.2b, 4.3a, 4.3b) have a posterior mean for intercept parameter $\alpha$ that is too small and a posterior mean for the first few $\beta_j$'s that is too large. As an example we have a look at the approximate posterior distributions of the parameters that are at the basis of the curve estimate of $f_2$ in figure 4.2b. In figure 4.4 the histograms of the posterior samples for $\alpha$ and for the first four $\beta_j$'s are given. One can see that the mean for alpha is around 2, while we would hope to see the posterior probability mass of alpha concentrated around three, because $f_2(0) = 3$ . The posterior probability mass of the $\beta_j$'s is concentrated too much on large values, while posterior distribution around 0 would give a better Bayes estimate of the true function, since the true function is constant.

To show this "steep-slope" does not only occur by accident for this particular simulated data set we simulated three more data sets for $f_2$. When we estimate three curves for these three simulated sets, the same "steep slope" appears, because these data sets give approximately the same distributions for the posterior distributions of the parameters in the beginning. For each of the three simulated data sets the approximate posterior distributions for $\alpha$ and and for the first four $\beta_j$'s are given in the Appendix 7.1. If you compare it to the histograms in figure 4.4 you see similar distributions. So also for the three other simulations the boundary effect occurs. So this "steep slope"- effect seems to be a structural problem. Can we adjust the investigator-specified parameters such that we remove this effect at the boundary?

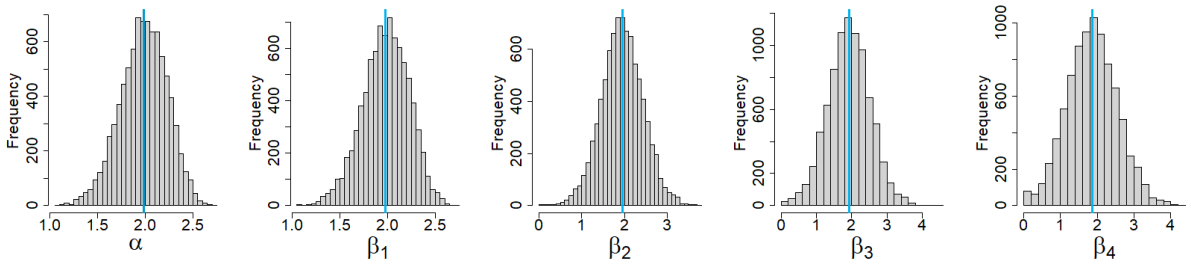


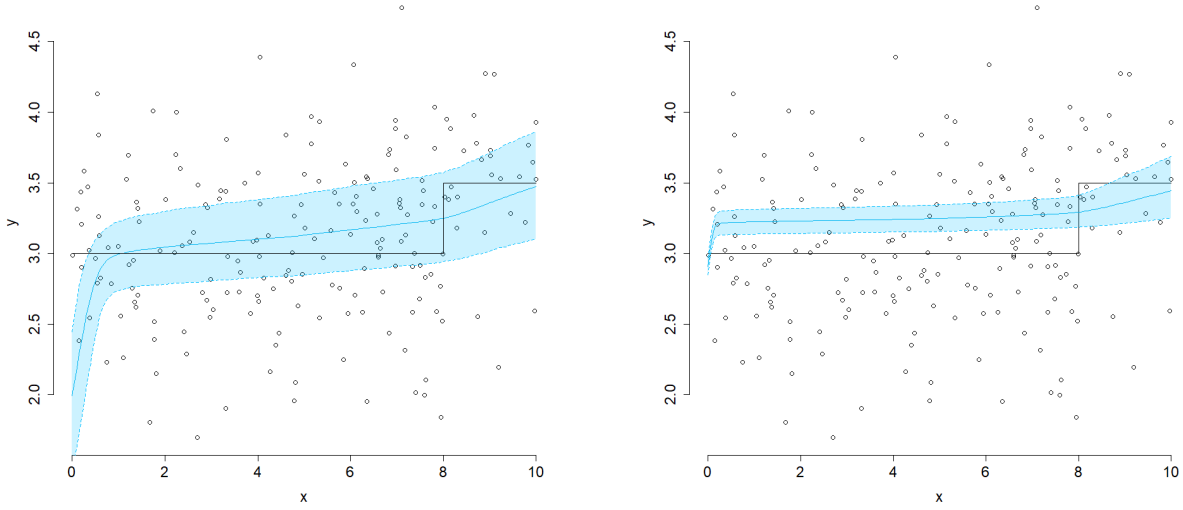

Figure 4.4: Histograms of the posterior samples for $\alpha$ and for the first four $\beta_j$'s

### 4.2.1 Adjusting the investigator specified parameters

It is not the prior specification of the intercept $\alpha$ that causes this steeply upward-sloping part. If we choose a more suitable prior mean of the intercept for estimating $f_2$, the steep slope does not vanish as we can see in figure 4.5. However, the result becomes better if we set a more specific prior for $\alpha$. Setting the prior mean of alpha equal to 3 and and prior variance equal to 2 does not give a visible better result in figure 4.5a than $\alpha_0 = 0$ and $\sigma_\alpha^2 = 10$ as in figure 4.2b. A very specific prior with $\alpha_0 = 3$ and $\sigma_\alpha^2 = 0.001$ does give a better result (see figure 4.5b). We can be this specific for the simulated data set, because we know the true value of $\alpha$ is 3.

We cannot be so specific for the climate data sets, because we do not know the true curve. However, we can be more specific based on prior beliefs we have than a normal distribution around 0 with variance 10. For example, it is reasonable to assume that the intercept is the winter temperature data set is normally distributed around 2 with variance 2. The same applies to the average year temperature: it is reasonable to think that the prior distribution for $\alpha$ is normal around 9 with variance 2. This is reasonable, because it may be someone's prior beliefs that the average year temperature is 9 degrees celsius.

The problem with the slope in the beginning appears to be a problem with the first few distributions of the $\beta_j$'s. The investigator specified best guess for the distribution of $\beta_1$ is also not the problem, because $E_{01}$ is specified as $E_{01} = 0$ with a large variance. Only when $E_{01}$ would have been very large then that may explain why only the first few estimated $\beta_j$'s are so large, because of the strong auto regressive structure between the $\beta_j$'s and the fact that in each iteration $\beta_j$ is sampled from a normal distribution with mean $E_{01}$.




(a) $\alpha_0 = 3, \sigma_\alpha^2 = 2$ (b) $\alpha_0 = 3, \sigma_\alpha^2 = 0.001$

Figure 4.5: Blue line is the curve estimate and the blue shadow is the 95% credible interval

### 4.2.2 Pseudo data

In the previous subsection, we have seen that we could improve the curve estimate in the beginning, if we specified a more specific prior for $\alpha$. But we could not completely vanish the steeply upward sloping part of the curve. For several different prior specifications of $\alpha$ the unusual behaviour kept showing up. In the literature about non-parametric curve estimation, estimation near the boundary is a well-known problem (Müller, 1993). For kernel methods as well as smoothing methods it is a problem. For kernel estimators many techniques have been proposed to remove problems at the boundary. One of those techniques is to generate pseudo data at the other end of the boundary and estimate the regression curve on an extended interval. Although our way of estimating the curve is not a kernel estimation method, the concept of pseudo data for solving boundary problems of kernel estimators provides us with a pragmatic way of dealing with our issue.

What we do is, we generate pseudo-data left from the boundary by reflecting the data points at the boundary like they do in several articles (Silverman, 1986) (Schuster, 1985) (Hall & Wehrly, 1991). The idea is that if we extend the data set at the left boundary we shift the steep slope to the left and only consider the curve estimate on the original domain of the data set. If $\{(x_i, y_i) : 1 \leq i \leq n\}$ is our original data set, then we specify $P_b = \{i : x_i \in (a, b]\}$. This is the set of indices for the $x$-values which fall in the interval for which we will create a "mirror-image". So we define for $i \in P_b$

$$\tilde{x}_i = a - (x_i - a) = 2a - x_i \tag{4.13}$$

Then we define the new data set as the union of the original data set and the pseudo data set:

$$\{(x_i, y_i) : 1 \leq i \leq n\} \cup \{(\tilde{x}_i, y_i) : i \in P_b\} \tag{4.14}$$

the method is visualized in figure 4.6 for the average temperature data set with a "mirror-image" of all $x_i \in (1901, 1920]$. The choice for the width of the reflection interval is based on the width of the steep slope. $b$ is chosen where the slope does not appear to suffer from boundary effects anymore.

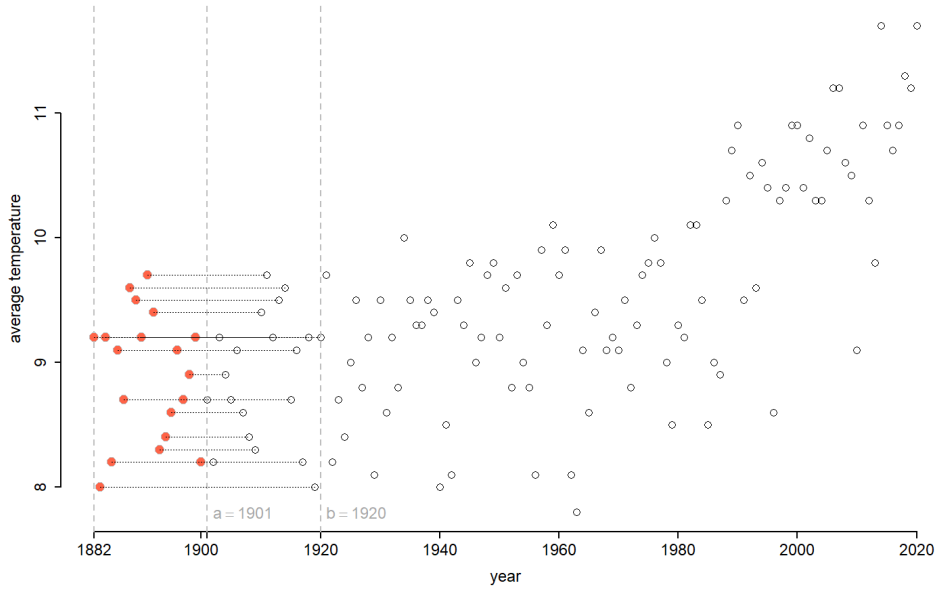



Figure 4.6: Average temperature data set with pseudo data set. The orange set of points is the pseudo data set. The middle and right dashed line indicate boundaries of the interval, which is reflected.

If we apply our estimation method to the extended data set of the average temperature in figure 4.6 we get the red curve estimate in figure 4.7.

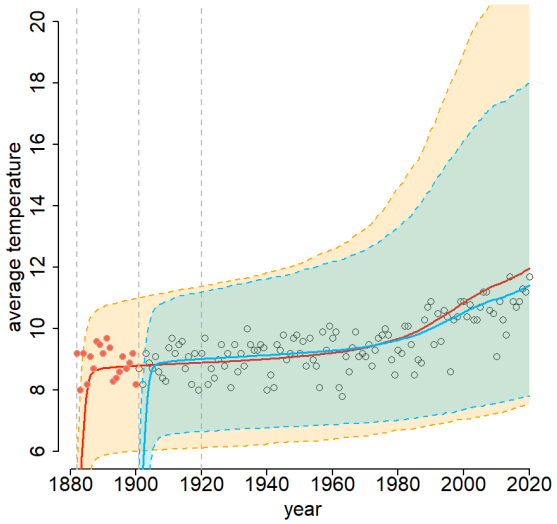
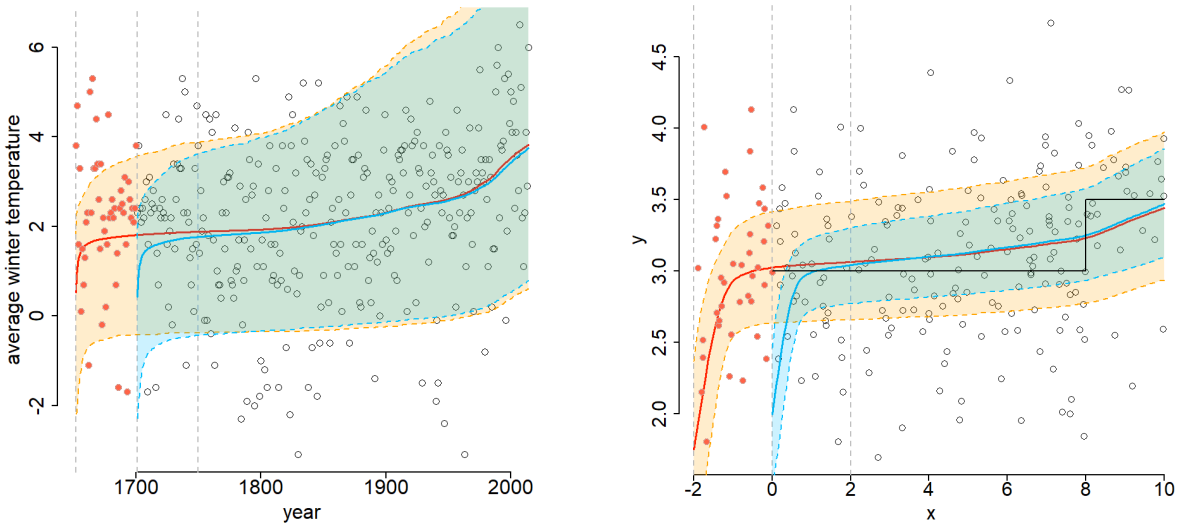



Figure 4.7: These are two curve estimates. The red/orange features correspond to the estimation on the extended data set and the blue parts correspond to the estimation of the original data set. The solid lines are the curve estimates and the dashed lines with the shadow represent the 95% credible interval.

As expected the steep slope problem is limited to the domain of the pseudo data, as we can see in figure 4.7. For $x_i \in (1901, 1920]$ we now have a more realistic curve estimate compared to the

original estimate. On the other hand, the curve estimate for the extended data set has a wider credible interval. In addition, the original estimate appears to fit the data between 1980-2020 better, since the other curve digresses a little from the concentration of points.

We also create this pseudo data extension for the winter temperature and the threshold function $f_2$, because they also suffered from this boundary effect. For the winter temperature data set, we reflected all $x_i \in (1701, 1750]$ (see figure 4.8a). For the threshold function $f_2$ we reflected all $x_i \in (0, 2]$. The result is given in figure 4.8b. Similarly, for the average temperature data set in figure 4.7 we also see in these figures 4.8a and 4.8b that using the pseudo data gives a better estimate of the "true"-function. This becomes explicitly clear at the $f_2$-curve estimate, because the true function is plotted in the same figure. The extension estimate for $x \in (0, 2]$ is much closer to the true function (black line) than the original estimate curve in blue. At $x = 0$ the estimate by the extension estimate is $f_2(0) \approx 3.02$ and $f_2(0) \approx 1.99$ for the original estimate, while the true value is $f_2(0) = 3$. For the winter temperature data we do not know the "true"-function, but the curve estimate is much more realistic for $x_i \in (1701, 1750]$.




(a) Winter temperature 1701-2020

(b) simulation from $f_2$

Figure 4.8: In both figures the orange points are the extended data points. The red solid line and the orange credible interval corresponds to the estimation on the extended data set and the blue features correspond to the estimation on the original data set.

This pseudo data method is a pragmatic way to solve this problem at the boundary, but it also has a disadvantage. A disadvantage of this method is that this symmetric reflection may cause the curve estimate in the reflection point to be more horizontal than it should be. This is also the case with density kernel estimation. If all data is reflected around the left boundary, which would be 0, then $\hat{f}'(0) = 0$ where $\hat{f}$ is the density estimate (Silverman, 1986). In this case, in which we estimate a monotone curve we will not have that the curve estimate is exactly horizontal, even if the bandwidth covers all data points. We have to nuance the statement that symmetric reflection is a disadvantage. For data with a clear trend it is a disadvantage actually. However, for estimating $f_2$ this symmetric reflection is very convenient, because $f_2$ is constant on $[0, 2]$. So a rather horizontal curve estimate in the reflection point gives an accurate estimate of $f_2$.

# Chapter 5

# Conclusion

The goal of this thesis was to implement and experiment Neelon and Dunsons method by applying it to climate data. In addition, this thesis also aimed to solve specific problems that came across when we try to apply this method. We managed to implement the method using R and applied it to two climate data sets. We encountered mainly two problems, which we described in chapter 4.

The first problem concerned the numerical issues with the computation of the weights of the mixture distribution. If the weight for the truncated normal distribution was approximately 0/0 R could not evaluate the weight. We solved this by rewriting the weight using the Mills ratio and approximate it such that we got a value for the weight. For other numerical issues with the weights we set the chance of drawing from one equal to one and the other to zero and vice versa.

The second problem was a boundary problem. For the simulated data set of $f_2$, which did have a monotone trend at the first glance, the curve estimate did not suffer noticeably from boundary effects. The other three sets did suffer from boundary effects. These three sets got a curve estimate at the left boundary, which was steeply upward sloping. A more specific prior specification of $\alpha$ can improve this "steep slope"- effect, but does not solve it. To get a better estimate at the boundary, we used a pragmatic way, which is inspired by the literature about similar problems with kernel estimators. Using the extended data to estimate a new curve improved the estimates at the original boundary.

# Chapter 6

# Discussion

The way we tackle the boundary problem is effective, but there is room for improvement. The reflection method reduces possible positive trend in the beginning. So the regression curve might be a little biased towards a more horizontal trend. However, for the application on climate data in this thesis this effect may not be so significant, because in the beginning there seems to be no monotone trend in the data. To prevent bias towards a horizontal curve in the boundary one can select a smaller "reflection"-interval. For obvious monotone data sets it might be more useful to inspect methods for generating pseudo data that incorporates the monotonicity of the data, like an asymmetric reflection.

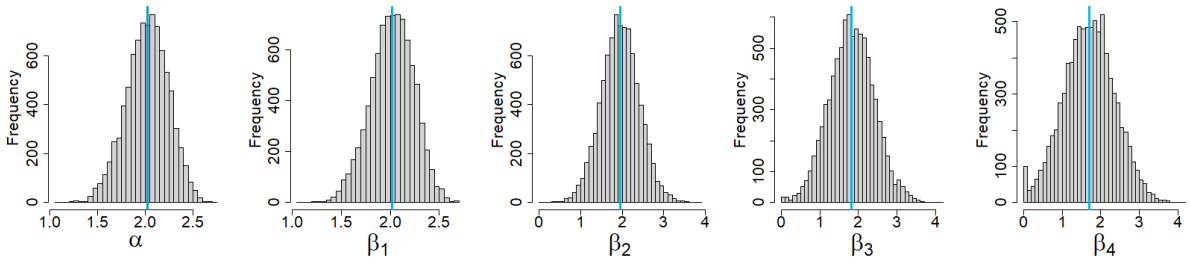
Furthermore, the problems we discussed for applying the method described by Neelon and Dunson are problems that we encountered Neelon and Dunson might not have encountered them. The way we applied this method differs slightly from how they did it. They also gave $\delta$ a distribution to make sure that it also allows for flatter regions. If $\delta$ also takes on values larger than zero, the point mass probability in zero for the posterior distribution of the $\beta_j$'s will be larger causing the slope of the curve estimate to be smaller. That might also reduce the boundary effects and might therefore be worth trying. We could also just take a larger fixed value for $\delta$ and check whether that reduces the boundary effects.

Last point for improvement is measuring the goodness of fit. We could use the root mean squared error to quantify the goodness of fit of the estimate and then compare it to the root mean squared error of other estimates made by other estimation methods.

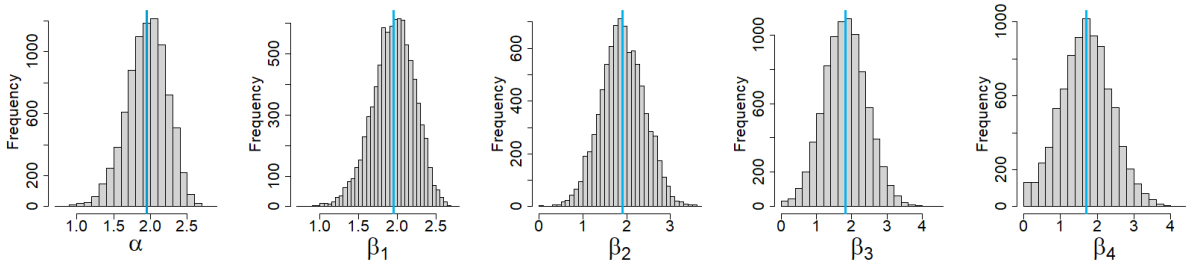
# Chapter 7

# Appendix

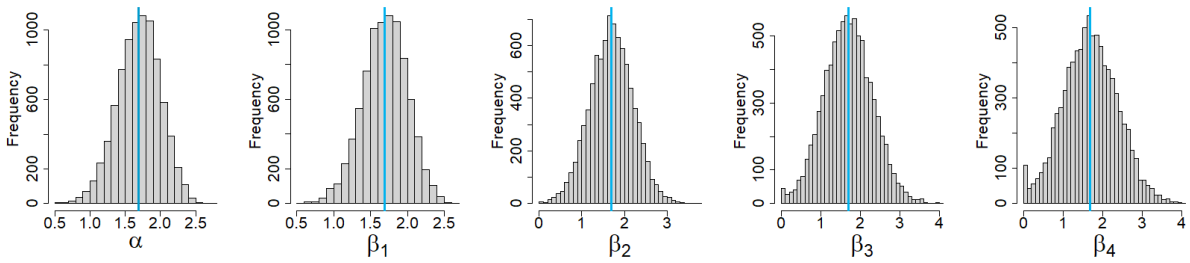
## 7.1 Approximate posterior distributions in histograms




(a) simulated data set 2




(b) simulated data set 3




(c) simulated data set 4

Figure 7.1: Approximate posterior distributions of $\alpha$ and first four $\beta_j$'s for three different data sets. Each data set is simulated from function $f_2$.

## 7.2 Full conditional distributions

The full conditionals of $\alpha$ and $\sigma^{-2}$ are the proportional to:

$$p(\sigma^{-2}|\theta, y) \sim \text{Gamma}(\sigma^{-2}; a + \frac{n}{2}, b + \frac{1}{2}(y - W\theta)'(y - W\theta))$$

$$W' = (w_1, \ldots, w_n)$$

$$p(\alpha|\beta, \sigma^{-2}) \sim N(\hat{\alpha}, \hat{\sigma}_\alpha^2)$$

$$\hat{\alpha} = \hat{\sigma}_\alpha^2 \left\{ \sigma_\alpha^{-2}\alpha_0 + \sigma^{-2} \sum_{i=1}^{n} \left( y_i - \mathbf{w}'_{i(-1)}\boldsymbol{\beta} \right) \right\}$$

$$\hat{\sigma}_\alpha^2 = \left( \sigma_\alpha^{-2} + n\sigma^{-2} \right)^{-1}$$

## 7.3 Steps in deriving the full conditional posterior distribution of $\beta_j$ and $\beta_j^*$

### Step A

If our posterior density kernel is

$$p(\alpha, \beta, \beta^*, \lambda, \delta, \sigma^{-2}|y) \propto p(y|\alpha, \beta, \beta^*, \lambda, \delta, \sigma^{-2}) p(\alpha, \beta, \beta^*, \sigma^{-2}, \lambda, \delta) \tag{7.1}$$

then we the posterior for $\beta_j$ and $\beta_j^*$ is just picking out terms from the posterior density kernel which involve $\beta_j$ and $\beta_j^*$, so then

$$p(\beta_j, \beta_j^*|\alpha, \lambda, \sigma^{-2}, y, x) \propto p(y|\theta, \sigma^2, x) \left[ \prod_{j=1}^{k} \{1_{(\beta_j=0)} 1_{(\beta_j^*<\delta)} + 1_{(\beta_j=\beta_j^*)} 1_{(\beta_j^*\geq\delta)}\} \times N(\beta_j^*; E_{0j}, V_{0j}) \right] \tag{7.2}$$

### Step B

$$p(\beta_j, \beta_j^*|\alpha, \lambda, \sigma^{-2}, y, x) \propto p(y|\theta, \sigma^2, x) \left[ \prod_{j=1}^{k} \{1_{(\beta_j=0)} 1_{(\beta_j^*<\delta)} + 1_{(\beta_j=\beta_j^*)} 1_{(\beta_j^*\geq\delta)}\} \times N(\beta_j^*; E_{0j}, V_{0j}) \right] \tag{7.3}$$

Step B consists of two parts: We rewrite the likelihood function

$$\begin{aligned}
p(y|\theta,\sigma^2,x) &= \left(2\pi\sigma^2\right)^{-n/2}\exp\left\{-\frac{1}{2\sigma^2}\sum_{i=1}^{n}\left(y_i-\boldsymbol{w}_i'\boldsymbol{\theta}\right)^2\right\}\\
&= \left(2\pi\sigma^2\right)^{-n/2}\exp\left\{-\frac{1}{2\sigma^2}\sum_{i=1}^{n}\left(y_i-\left(\alpha+\sum_{\tilde{j}=1}^{k}w_{i\tilde{j}}\beta_{\tilde{j}}\quad\right)\right)^2\right\}\\
&= \left(2\pi\sigma^2\right)^{-n/2}\exp\left\{-\frac{1}{2\sigma^2}\sum_{i=1}^{n}\left(y_i-\left(\alpha+\sum_{\tilde{j}=1,\tilde{j}\neq j}^{k}w_{i\tilde{j}}\beta_{\tilde{j}}\quad+w_{ij}\beta_j\right)\right)^2\right\}\\
&= \left(2\pi\sigma^2\right)^{-n/2}\exp\left\{-\frac{1}{2\sigma^2}\sum_{i=1}^{n}\left(y_i-\left(\mathbf{w}_{i(-j)}'\boldsymbol{\theta}_{(-j)}+w_{ij}\beta_j\right)\right)^2\right\}\\
&= \left(2\pi\sigma^2\right)^{-n/2}\exp\left\{-\frac{1}{2\sigma^2}\sum_{i=1}^{n}\left(y_{ij}^*-w_{ij}\beta_j\right)^2\right\}\quad\text{with}\quad y_{ij}^*=y_i-\mathbf{w}_{i(-j)}'\boldsymbol{\theta}_{(-j)}\\
&= \prod_{i=1}^{n}N(y_{ij}^*;w_{ij}\beta_j,\sigma^2)
\end{aligned}\tag{7.4}$$

and we note that

$$\begin{aligned}
p(\beta_j,\beta_j^*|\alpha,\lambda,\sigma^{-2},y,x) &\propto p(y|\theta,\sigma^2,x)\left[\prod_{j=1}^{k}\{1_{(\beta_j=0)}1_{(\beta_j^*<\delta)}+1_{(\beta_j=\beta_j^*)}1_{(\beta_j^*\geq\delta)}\}\times N(\beta_j^*;E_{0j},V_{0j})\right]\\
&\propto p(y|\theta,\sigma^2,x)\left\{1_{(\beta_j=0)}1_{(\beta_j^*<\delta)}+1_{(\beta_j=\beta_j^*)}1_{(\beta_j^*\geq\delta)}\right\}\times N(\beta_j^*;E_{0j},V_{0j})N(\beta_{j+1}^*;\beta_j^*,\lambda^{-1})
\end{aligned}\tag{7.5}$$

**Step C**

$$
\begin{aligned}
N(\beta_j^*; E_{0j}, V_{0j}) N(\beta_{j+1}^*; \beta_j^*, \lambda^{-1}) &\propto \exp\left(-\frac{1}{2V_{0j}}\left(\beta_j^* - E_{0j}\right)^2\right)\exp\left(-\frac{\lambda}{2}\left(\beta_{j+1}^* - \beta_j^*\right)^2\right) \\
&= \exp\left\{-\frac{1}{2V_{0j}}\left(\beta_j^{*^2} - 2E_{0j}\beta_j^* + E_{0j}^2\right) - \frac{\lambda}{2}\left(\beta_{j+1}^{*^2} - 2\beta_j^*\beta_j^* + \beta_j^{*^2}\right)\right\} \\
&= \exp\left(-\frac{1}{2}\left(V_{0j}^{-1} + \lambda\right)\beta_j^{*2} + \left(V_{0j}^{-1}E_{0j} + \lambda\beta_{j+1}^*\right)\beta_j^*\right) \\
&= \exp\left(-\frac{1}{2\underbrace{\left(V_{0j}^{-1} + \lambda\right)^{-1}}_{=\tilde{\mathbf{V}}_{\mathbf{0j}}}}\left(\beta_j^{*^2} - 2\beta_j^*\underbrace{\left(V_{0j}^{-1} + \lambda\right)^{-1}\left(V_{0j}^{-1}E_{0j} + \lambda\beta_{j+1}^*\right)}_{=\tilde{\mathbf{E}}_{\mathbf{0j}}}\right)\right) \\
&= \exp\left(-\frac{1}{2\tilde{V}_{0j}}\left(\beta_j^{*^2} - 2\beta_j^*\tilde{E}_{0j}\right)\right) \\
&= \exp\left(-\frac{1}{2\tilde{V}_{0j}}\left(\beta_j^{*^2} - 2\beta_j^*\tilde{E}_{0j} + \tilde{E}_{0j}^2\right) - \left(-\frac{1}{2\tilde{V}_{0j}}\tilde{E}_{0j}^2\right)\right) \\
&= \frac{\exp\left(-\frac{1}{2\tilde{V}_{0j}}\left(\beta_j^* - \tilde{E}_{0j}\right)^2\right)}{\exp\left(-\frac{1}{2\tilde{V}_{0j}}\left(\tilde{E}_{0j}\right)^2\right)} \\
&\propto \frac{N\left(\beta_j^*; \tilde{E}_{0j}, \tilde{V}_{0j}\right)}{N\left(0; \tilde{E}_{0j}, \tilde{V}_{0j}\right)} \quad \text{with} \tilde{V}_{0j} = \left(V_{0j}^{-1} + \lambda\right)^{-1}\tilde{E}_{0j} = \tilde{V}_{0j}\left(V_{0j}^{-1}E_{0j} + \lambda\beta_{j+1}^*\right)
\end{aligned}
\tag{7.6}
$$

So by

$$
\begin{aligned}
p(\beta_j, \beta_j^* | \alpha, \lambda, \sigma^{-2}, y, x) &\propto \left[\prod_{i=1}^{n} N(y_{ij}^*; w_{ij}\beta_j, \sigma^2)\right] \left\{1_{(\beta_j=0)} 1_{(\beta_j^*<\delta)} + 1_{(\beta_j=\beta_j^*)} 1_{(\beta_j^*\geq\delta)}\right\} \\
&\quad \times N(\beta_j^*; E_{0j}, V_{0j}) N(\beta_{j+1}^*; \beta_j^*, \lambda^{-1}) \\
&\propto \left[\prod_{i=1}^{n} N(y_{ij}^*; w_{ij}\beta_j, \sigma^2)\right] \left\{1_{(\beta_j=0)} 1_{(\beta_j^*<\delta)} + 1_{(\beta_j=\beta_j^*)} 1_{(\beta_j^*\geq\delta)}\right\} \\
&\quad \times \frac{N\left(\beta_j^*; \tilde{E}_{0j}, \tilde{V}_{0j}\right)}{N\left(0; \tilde{E}_{0j}, \tilde{V}_{0j}\right)} \\
&= 1_{(\beta_j=0)} 1_{(\beta_j^*<\delta)} \left(\frac{N(\beta_j^*; \tilde{E}_{0j}, \tilde{V}_{0j})}{N(0; \tilde{E}_{0j}, \tilde{V}_{0j})}\right) + 1_{(\beta_j=\beta_j^*)} 1_{(\beta_j^*\geq\delta)} \left(\frac{N(\beta_j^*; \hat{E}_j, \hat{V}_j)}{N(0; \hat{E}_j, \hat{V}_j)}\right) \qquad (7.7)
\end{aligned}
$$

$$
\begin{aligned}
\tilde{V}_{0j} &= \left(V_{0j}^{-1} + \lambda\right)^{-1} \\
\tilde{E}_{0j} &= \tilde{V}_{0j}\left(V_{0j}^{-1} E_{0j} + \lambda\beta_{j+1}^*\right) \\
\hat{V}_j &= \left(\tilde{V}_{0j}^{-1} + \sigma^{-2}\sum_{i=1}^{n} w_{ij}^2\right)^{-1} \\
\hat{E}_j &= \hat{V}_j\left(\tilde{V}_{0j}^{-1}\tilde{E}_{j0} + \sigma^{-2}\sum_{i=1}^{n} w_{ij} y_{ij}^*\right)
\end{aligned}
$$

## 7.4 R code

```
library(readxl)
library("dplyr")
library(truncnorm)
library(stats)

rm(list = ls())

#############################################################################################################
#  IMPORT DATA  #
#############################################################################################################

data <- read_excel("C:/Users/damia/Documents/BEP/wintertemps.xlsx")
#data <- read_excel("C:/Users/damia/Documents/BEP/debilt1901_2021.xlsx")

#############################################################################################################
#   FUNCTIONS    #
#############################################################################################################

w.fun <- function(i,j,x,knots){
  if(x[i]>=knots[j]){
    return(min(x[i],knots[j+1])-knots[j])
  }else{
    return(0)
  }
}

v0j.tilde.compute <-function(v0j,lambda){
  return(1/((1/v0j) +lambda))
}

e0j.tilde.compute <- function(v0j.tilde,v0j,e0j,lambda,betastarjplus1){
  return(v0j.tilde*((1/v0j)*e0j + lambda*betastarjplus1))
}

vj.hat.compute <-function(v0j.tilde,sigma,wj){
  return(1/((1/v0j.tilde) + sigma*sum(wj^2)))
}

ej.hat.compute <- function(vj.hat,v0j.tilde,e0j.tilde,sigma,wj,yj){
  return(vj.hat * ((1/v0j.tilde) * e0j.tilde + sigma * sum(wj*yj)))
}

# Returns a sample from the mixture distribution for beta*
sample.beta.star <- function(A,C,delta,e0j.tilde,v0j.tilde,ej.hat,vj.hat){
  u <- runif(1)
  if(u < (A/C)){
    res <- rtruncnorm(1, a = -Inf,  b = delta, mean = e0j.tilde , sd = sqrt(v0j.tilde))
  }else{
    res <- rtruncnorm(1, a = delta, b = Inf,   mean = ej.hat,     sd = sqrt(vj.hat)   )
  }
  return(res)
}


# Returns value of the mills ratio for x <7. For x>7 it returns an approximation.
asym.mill <- function(x){
  if(x <7){
    return((1 - pnorm(x)) / dnorm(x))
```

```
  }else{
    return((1/x)-(1/(x^3))+(3/(x^5))+(15/(x^7)))
  }
}

# Returns A and B (weights for sampling from mixture)
f_AB <- function(delta,ej.hat,e0j.tilde,v0j,vj.hat,v0j.tilde){
  A <- pnorm(delta,e0j.tilde,sqrt(v0j.tilde) )/dnorm(0,e0j.tilde,sqrt(v0j.tilde))
  macht    <- (-1/2)*((delta^2)/(vj.hat))+delta *((ej.hat)/(vj.hat))
  B        <- sqrt(vj.hat)*exp(macht)*asym.mill((delta-ej.hat)/(sqrt(vj.hat)))
  if(is.infinite(A)){
    A <- 1
    B <- 0
  }
  if(is.infinite(B) | is.na(A)){
    A <- 0
    B <- 1
  }
  return(c(A,B))
}

# returns the union of the original data and -
# pseudo data set
ext.spiegel <-function(x,y,lb,rb){
  indices <-which((x >= lb) & (x <= rb))
  snip.x <- x[indices]
  snip.y <- y[indices]
  spiegel.x <- rev(rep(2*lb, length(snip.x))-snip.x)
  spiegel.y <- rev(snip.y)
  res <- cbind(matrix(c(spiegel.x,x)),matrix(c(spiegel.y,y)))
  colnames(res)<-c("x","y")
  return(res)
}


##############################################################################################################

# load("22_BILT.RData")

#
y <- pull(data[1:314,],'average winter temperature') # pull is function of library dplyr and makes vector fr
x <- pull(data[1:314,],year)

# creating pseudo data
xy <-ext.spiegel(x,y,1701,1750)
x <-xy[,"x"]
y <-xy[,"y"]

# create vector of knot locations
knots <- seq(min(x) ,max(x),length = 501)

n <- length(y)                          # number of datapoints
k <- length(knots)-1                    # number of beta's slopes we need to estimate



# create matrix W
W <- matrix(rep(0,n*(k+1)),nrow = n, ncol = k+1 )
W[,1] <- rep(1,n)
```

```
for(j in 2:(k+1)){
  for(i in 1:n){
    W[i,j]<- w.fun(i,j-1,x,knots)
  }
}

# Hyperparameters: alpha_0, sigma_alpha^{2}, a, and b are investigator-specified hyperparameters.
a <- 0.1                # shape parameter
b <- 0.1                # rate parameter <=> inverse scale parameter = 1/theta
sigma.alpha <- 10       # uncertainty in the investigator specified parameter a0
a0 <- 0                 # prior mean for alpha
c1 <- k/25              # shape parameter
d1 <- 1                 # rate parameter <=> inverse scale parameter = 1/theta

#############################################################################################################
#   GIBBS SAMPLING ALGORITHM    #
#############################################################################################################

# number of iterations and the burnin
iter    <- 10000
burnin  <- 1500

# create empty vectors/matrices
sigma    <- rep(NA,iter)
alpha    <- rep(NA,iter)
lambda   <- rep(NA,iter)
beta     <- matrix(data =NA,nrow =iter,ncol =k)
beta.star<- matrix(data =NA,nrow =iter,ncol =k)

# intitial values
sigma[1]   <- 2
alpha[1]   <- 1
lambda[1]  <- 1
beta[1,]       <- rep(0.1,k)
beta.star[1,]  <- rep(0.1,k)

v01 <- 10
e01 <- 0

delta <- 0.0001

for(i in 2:iter){

  #sigma ==========================================================================================

  theta           <- c(alpha[i-1],beta[i-1,])
  sigma[i]        <- rgamma(1,shape = a + n/2  , rate = b + (1/2)*t(y-W %*% theta) %*% (y-W %*% theta))

  #alpha ==========================================================================================

  sigma.alpha.hat <- 1/((1/sigma.alpha) +n*sigma[i])
  alpha.hat       <- sigma.alpha.hat*((1/sigma.alpha)*a0 +sigma[i]*sum(y - W[,-1] %*% beta[i-1,]))
  alpha[i]        <- rnorm(1,mean = alpha.hat, sd = sqrt(sigma.alpha.hat))

  #lambda ==========================================================================================
  sh              <- c1 + (k-1)/2
  ra              <- d1 +(1/2)*sum((beta.star[i-1,2:k]-beta.star[i-1,1:(k-1)])^2)
  lambda[i]       <- rgamma(1,shape = sh, rate = ra)
```

```
  #beta and beta.star ================================================================================

  for(j in 1:k){

    #computing e0j, v0j, ej, and vj ------------------------------------------------------

    if(j>1){
      v0j.tilde <- v0j.tilde.compute(lambda[i],lambda[i])

      #if statement to prevent that for j=k e0j.tilde.compute uses as beta_k+1 as argument (which doesn't exi
      if(j <k){
        e0j.tilde <- e0j.tilde.compute(v0j.tilde,lambda[i],beta.star[i-1,j-1],lambda[i], betastarjplus1 =beta
      }else{
        e0j.tilde <- e0j.tilde.compute(v0j.tilde,lambda[i],beta.star[i-1,j-1],lambda[i], betastarjplus1 =beta
      }

    }else{
      v0j.tilde   <- v0j.tilde.compute(v01,lambda[i])
      e0j.tilde   <- e0j.tilde.compute(v0j.tilde,v01,e01,lambda[i],betastarjplus1 =beta.star[i-1,j+1] )
    }
    vj.hat     <- vj.hat.compute(v0j.tilde,sigma[i],W[,j])
    theta.minj <- matrix(data = c(alpha[i],beta[i-1,-j]), nrow = k, ncol = 1)
    yj.star    <- y - W[,-j] %*% theta.minj
    ej.hat     <- ej.hat.compute(vj.hat, v0j.tilde, e0j.tilde,sigma[i],W[,j],yj.star)

    # computing A,B, and C ---------------------------------------------------------------

    AB <- f_AB(delta,ej.hat,e0j.tilde,v0j,vj.hat,v0j.tilde)
    C  <- sum(AB)

    # computing in the i'th iteration the j'th slope for beta ------------------------------

    beta.star[i,j] <- sample.beta.star(AB[1],C,delta,e0j.tilde,v0j.tilde,ej.hat,vj.hat)
    beta[i,j]      <- ifelse(beta.star[i,j]>delta,beta.star[i,j],0)
  }
}

beta.star.postmean <- apply(beta.star[burnin:iter,], 2, mean)
beta.postmean      <- apply(beta[burnin:iter,], 2, mean)


save.image("betamat_file.RData")

#############################################################################################################
# Load data for PLOTS    #
#############################################################################################################

rm(list = ls())
load(file = "betamat_file.RData")


#############################################################################################################
#   FUNCTIES VOOR HET PLOTTEN   #
#############################################################################################################

#Returns regression curve points
create.nd <- function(alpha,knots,beta.postmean,lb){

    # create points for the nd regression line
    punta <- rep(0,k+1)
```

```
    punta[1] <- mean(alpha)
    for(i in 2:(k+1)){
      punta[i] <- punta[i-1] + beta.postmean[i-1]*(knots[i]-knots[i-1])
    }
    # create specific lines
    begin        <- length(which(knots< lb))
    if (begin >0){
      value.lb     <- punta[begin]+beta.postmean[begin]*(lb -knots[begin])
    }else{
      value.lb     <- punta[1]
    }
    res <- data.frame( x= c(lb,knots[(begin+1):(k+1)]), y = c(value.lb,punta[(begin+1):(k+1)]))
    return(res)
}

# Returns the estimated points on the knot location for an alpha and vector beta
extract.joint <- function(alpha,beta,knots){
  res      <-  rep(0,length(knots))
  res[1]   <-  alpha
  for(j in 2:(length(knots))){
    res[j] <- res[j-1]+ beta[j-1]*(knots[j]-knots[j-1])
  }
  return(res)
}


# Returns two vectors with points of the credible interval
create.cred<- function(alpha,beta,knots,burnin){
    M <- matrix(extract.joint(alpha[burnin],beta[burnin,],knots), ncol =1, nrow =length(knots))

    for( u in (burnin+1):(iter)){
      M <- cbind(M,extract.joint(alpha[u],beta[u,],knots))
    }

    intervall <- apply(M,1,quantile, prob =c(0.025,0.975))
    res <- data.frame( x= knots, q25= intervall[1,],q975 = intervall[2,])
    return(res)
}

#########################################################################################################
#   PLOT    #
#########################################################################################################

windows(width =10, height =10)
par(mar = c(5.1, 4.1, 4.1, 2.1))
plot(x,y, frame.plot = FALSE, axes = FALSE,xlab = "x",ylab = "y",cex.lab =2 ,cex=1.5)
axis(1,cex.axis =2,lwd =2)
axis(2,cex.axis =2,lwd =2)


# 2------- {POSTERIOR MEAN LINE } ------------------------------------------------

df.post <- create.nd(alpha,knots,beta.postmean,knots[1])
lines(df.post$x,df.post$y, col ="red",lwd =3)

# 3-------- {CREDIBLE INTERVAL PLOT} ---------------------------------------------

# get line coordinates
df.cred <-create.cred(alpha,beta,knots, burnin)
```

```r
# plot the credible interval lines
lines(df.cred$x, df.cred$q25, lty =2, lwd =2, col = "orange")
lines(df.cred$x, df.cred$q975,lty =2, lwd =2, col ="orange")

# plot the shadow

#get rgb values for named colors
rgb.value <- col2rgb("orange")
transparant.color <- rgb(rgb.value[1],rgb.value[2],rgb.value[3],
                         max =255,
                         alpha = (100 - 80) * 255 / 100)
polygon(x =c(knots, rev(knots)),
        y =c(df.cred$q25,rev(df.cred$q975)),
        col = transparant.color,
        border = NA)

abline(v =-2, lty =2,lwd =2, col = "grey")
abline(v =0, lty =2,lwd =2, col = "grey")
abline(v =2, lty =2,lwd =2, col = "grey")
points(x[which(x <0)],y[which(x <0)], pch = 21, bg = "tomato", col = "grey",cex =1.5)
```

# References


Chatterjee, S., & Hadi, A. S. (2015). *Regression analysis by example*. John Wiley & Sons.

Groeneboom, P., & Jongbloed, G. (2015). Statistiek met vormrestricties. *Nieuw Archief voor Wiskunde*, *5*(4), 279–283.

Hall, P., & Wehrly, T. E. (1991). A geometrical method for removing edge effects from kernel-type nonparametric regression estimators. *Journal of the American Statistical Association*, *86*(415), 665–672.

KNMI. (2021). *Monthly and yearly mean temperatures per weather station in the netherlands.* Royal Dutch Meteorological Institute De Bilt, The Netherlands. Retrieved from `https://cdn.knmi.nl/knmi/map/page/klimatologie/gegevens/maandgegevens/mndgeg_260_tg.txt`

Müller, H.-G. (1993). On the boundary kernel method for non-parametric curve estimation near endpoints. *Scandinavian Journal of Statistics*, 313–328.

Neelon, B., & Dunson, D. B. (2004). Bayesian isotonic regression and trend analysis. *Biometrics*, *60*(2), 398–406.

Ruben, H. (1963). A convergent asymptotic expansion for mill's ratio and the normal probability integral in terms of rational functions. *Mathematische Annalen*, *151*(4), 355–364.

Schuster, E. F. (1985). Incorporating support constraints into nonparametric estimators of densities. *Communications in Statistics-Theory and methods*, *14*(5), 1123–1136.

Silverman, B. W. (1986). Density estimation for statistics and data analysis. *London: Chapman and Hall*.