Quantile Regression

Reference range of Thyroid function test in pregnancy

Kevin Brosnan

Final Year Project
Mathematical Sciences

Department Of Mathematics and Statistics
University of Limerick
Limerick
Ireland

March, 2014

A final year project submitted in partial fulfillment of the B.Sc degree in Mathematical Sciences

Supervisor: Dr. Kevin Hayes
Second Reader: Dr. Norma Bargary

Abstract

The aim of this project is to develop a method to provide precise reference ranges for thyroid hormones in women during pregnancy. While other methods have been used in the past, the focus of this research will be to develop these reference ranges using quantile regression. Quantile regression moves away from the general tendency of central regression and is computed at any percentile of interest. Current quantile regression methods do not routinely quantify the precision of the end points of these reference ranges due to a phenomenon called quantile crossing. This results in misclassification of patients to ranges and makes selection for treatment difficult.

Thyroid disease is the most common endocrine condition in women of childbearing age and complicates approximately 1% of pregnancies. Physiological changes of pregnancy mean that thyroid hormone reference ranges for non-pregnant women may not be appropriate in pregnancy. The 2.5% and 97.5% regression quantiles will be of interest in this project, as these are the treatment regions for thyroid disease.

Acknowledgements

I wish to thank the academic staff of the Maths and Statistics Department of University of Limerick for all the knowledge they have instilled on me over the course of my undergraduate programme. I would like to express my deep gratitude to Dr. Kevin Hayes, my supervisor, for his patient guidance, enthusiastic encouragement and useful critiques of my final year project.

Finally, I wish to thank my parents for their support and encouragement throughout my study.

Contents

Chapter 1

Introduction

In statistical regression, the desired estimate of $y|x$ is not always given by a conditional mean, although it is the most commonly used method. While the conditional mean is useful in measuring the relationship between variables around the center of the data, it is often required to understand the relationship between variables at positions other than the center of the data. Quantile regression serves as a method which provides the capabilities to understand the relationship between variables at a given quantile. Throughout this project quantile regression will be used to solely to develop reference ranges for thyroid function in pregnancy, however quantile regression methods have other uses across industry:

- A device manufacturer may wish to know what the 5% and 95% quantiles are for some feature of the production process, so as to keep the process under control for 95% of the devices produced

- A pediatrician requires a growth chart for children given their age and perhaps even medical back-ground, to help determine whether medical interventions are required

Thyroid disorders are the second most common endocrinologic disorders found in pregnancy. Overt hypothyroidism is estimated to occur in $0.3 - 0.5\%$ of pregnancies. Subclinical hypothyroidism tends to occur in $2 - 3\%$, and hyperthyroidism is present in $0.1 - 0.4\%$ (Abalovich et al., 2007). Physiological changes of pregnancy, including 50% increase in plasma volume, increased thyroid binding globulin production and a relative iodine deficiency, means that thyroid hormone reference ranges for non-pregnant women may not be appropriate in pregnancy.

One acceptable approach for establishing legitimate reference ranges requires that a Box-Cox transformation be applied to the data and prediction ranges calculated using classical polynomial regression. Alternatively, non-parametric smoothing such as quantile regression can be used to estimate the 2.5% and 97.5% percentiles. While this approach provides an estimate of the reference range, there are problems with the method. The main problem is the issue of crossing quantiles which makes it difficult to assign each patient to a single range. This is the issue that will be focused on throughout this project.

Initially a background on quantile regression will be explored. This section will look at the theory behind quantile regression, its advantages and disadvantages over more popular regression methods such as ordinary least squares (OLS) and the implementation of quantile regression under a frequentist approach, a Bayesian approach and a linear mixed effects model approach. Current statistical software based in R (R Core Team, 2013) will be detailed and examined using a sample dataset based on the relationship between household income and food expenditure for working class households in Belgium. The output and computational effort will be provided for comparison between each of the three methods.

After introducing the key concepts of quantile regression methods, the thyroid data to be modelled in this project will be examined in detail. Descriptive statistics of each of the variables will be provided, along with selection of the principal variables of interest on which the models will be based. The variables will be tested for normality and power transformations provided if necessary to normalise the data. Simple linear quantile models will initially be applied to model the relationship between each of the thyroid hormone variables and the gestation week of the patient. If the simple model does not fit the data to sufficient standard modifications to the model will be made to increase the model fit.

In the final chapter of this project a solution to the issue of crossing quantile curves will be provided. The theoretical details of the solution will be discussed extensively. The solution to the crossing quantiles will then be applied to model the thyroid hormones once again using the most appropriate model already decided in the exploratory stage. The references ranges for thyroid test function during pregnancy will then be provided using this new approach to non-crossing quantile curve estimation.

Chapter 2

Analytical Background

The objective of regression analysis is to establish a relationship between a response variable, Y, and the predictor variables, $\{x_1, \ldots, x_p\}$. In real world applications, Y cannot be calculated perfectly from the X variables. For modelling purposes we formulate Y, for a fixed value of each x_i as a random variable. We generally proceed by summarising the relationship of the response variable for fixed values of the predictors using measures of centrality, specifically the mean, median and mode.

Quantile regression uses the median as its central tendency and is the method of interest in this project. In this chapter we outline the theory behind quantile regression, focusing attention on the frequentist approach to quantile regression, Bayesian quantile regression and linear quantiles for mixed models. Finally, we will examine the computational implementation required for each of the quantile regression methods outlined above.

2.1 Quantiles

A quantile τ of the dependant variable Y is defined such that $100\tau\%$ of the population have values less than the τ^{th} quantile and $100(1-\tau)\%$ of the population have values greater than the τ^{th} quantile (see Figure 2.1).

The median is defined such that 50% of your population have a value above this value and 50% of your population have a value below this value. The median can therefore be interrupted as the 0.5^{th} quantile. The quantile or percentile refers to the general case of this.

More formally, let Y be a continuous real valued random variable, it may be characterised by its distribution function as

$$F_Y(y) = \mathbb{P}(Y \leq y),$$

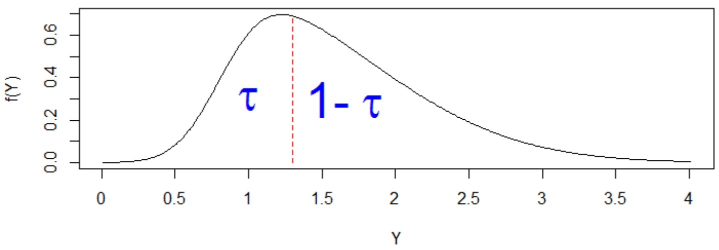

Figure 2.1: Graphical illustration of the τ^{th} quantile.

while for any $0 < \tau < 1$,

$$Q(\tau) = \inf \{ y : F_Y(y) \geq \tau \}$$

is called the τ^{th} quantile of Y. When estimating quantiles, we want to determine the value of y in the sample data corresponding to a given probability τ. The τ^{th} quantile in a sample of data refers to the probability of τ for a value y, such that

$$F_Y(y_\tau) = \tau.$$

Another form of expressing the τ^{th} quantile mathematically is

$$y_\tau = F_Y^{-1}(\tau).$$

y_τ is such that it constitutes the inverse of the function $F_Y(\tau)$ for a probability τ.

If the distribution function $F_Y(y)$ is monotonically increasing, quantiles are well defined for every $\tau \in (0, 1)$. However, if a distribution function $F_Y(y)$ is not strictly monotonically increasing, there are some τ's for which a unique quantile can not be defined. In the latter case one must use the smallest value that y can take on for a given probability τ. In both cases the problem can be defined mathematically as seeking the value of y satisfying

$$y_\tau = F_Y^{-1}(\tau) = \inf \{ y : F_Y(y) \geq \tau \} . \tag{2.1}$$

Therefore, y_τ is equal to the inverse of the function $F_Y(\tau)$ which in turn is equal to the infimum of y such that the distribution function $F_Y(y)$ is greater or equal to a given probability τ which in turn is the τ^{th} quantile.

2.2 Quantile Regression

Quantile regression is a statistical technique used to estimate conditional quantile functions. Classic linear regression methods are based on minimising sum-of-squares residuals and can be used to estimate models for conditional mean functions. Quantile regression methods however offer a way of estimating models for the conditional median function and all other conditional quantile functions. As quantile regression can estimate the entire family of conditional quantile functions it provides a much more powerful statistical analysis of relationships among random variables (Koenker, 2000). The need for something beyond linear regression was first advocated by Mosteller and Tukey (1977):

What the regression curve does is give a grand summary for the averages of the distributions corresponding to the set of x's. We could go further and compute several different regression curves corresponding to the various percentage points of the distributions and thus get a more complete picture of the set. Ordinarily this is not done, and so regression often gives a rather incomplete picture. Just as the mean gives an incomplete picture of a single distribution, so the regression curve gives a correspondingly incomplete picture for a set of distributions. (Mosteller and Tukey, 1977)

Features that characterise quantile regression and distinguish it from other regression methods are the following:

1. quantile regression can characterise the entire conditional distribution of Y through different values of τ;
2. heteroscedasticity can be detected;
3. median regression estimators can be more efficient then mean regression estimators if heteroscedasticity is detected;
4. the minimisation problem as illustrated in equation 2.2 can be solved efficiently by linear programming methods, making estimation easy;
5. quantiles are robust in regards to outliers.

The technical details in the remainder of this section will help to explain how to implement quantile regression contrasting three fitting methods. Quantile regression can be seen as one statistical method which can be used to complete the regression picture.

2.2.1 Frequentist Approach

Quantile regression transforms a conditional distribution function into a conditional quantile function by slicing it in to segments. These segments describe the cumulative distribution of a conditional variable Y given the explanatory variables x_i with the use of quantiles as defined in equation (2.1). For a dependant variable Y given the explanatory variable $X = x$ and fixed τ, $0 < \tau < 1$, the conditional quantile function is defined as the τ^{th} quantile $Q_{Y|X}(\tau|x)$ of the conditional distribution function $F_{Y|X}(y|x)$. For the estimation of the location of the conditional distribution function, the conditional median $Q_{Y|X}(0.5|x)$ can be used as an alternative to the conditional mean.

In ordinary least squares, modelling a conditional distribution function of a random sample $(y_1, \dots, y_n)$ with a parametric function $\mu(x_i, \beta)$ where x_i represents the independent variables, β the corresponding estimates and μ the conditional mean, one addresses the minimisation problem

$$\min_{\beta \in \Re} \sum_{i=1}^{n} (y_i - \mu(x_i, \beta))^2.$$

We therefore obtain the conditional expectation function $\mathbb{E}[Y|x_i]$. The conditional expectation function is the best predictor of Y given x_i in the sense that it solves a minimum mean squared error prediction problem. It can be simply evaluated as

$$\mathbb{E}[Y|x_i] = \underset{m(x_i)}{\operatorname{argmin}} \, \mathbb{E}[(Y - m(x_i))^2]$$

where $m(x_i)$ is any function of x_i. This equation is minimised at $m(x_i) = \mathbb{E}[Y|x_i]$. While the approach is similar for quantile regression the central feature now becomes ρ_τ, which acts as a check function. Define $\rho_\tau(x)$ by

$$\rho_\tau(x) = \begin{cases} \tau * x, & \text{if } x \geq 0 \\ (\tau - 1) * x, & \text{if } x < 0 \end{cases}$$

The check function, $\rho_\tau(x)$, ensures that:

(i) all ρ_τ are positive;

(ii) the scale is according to the probability τ.

In quantile regression, the τ^{th} *sample* quantile may be found by solving:

$$\min_{\xi \in \Re} \sum_{i=1}^{n} \rho_\tau(y_i - \xi) \tag{2.2}$$

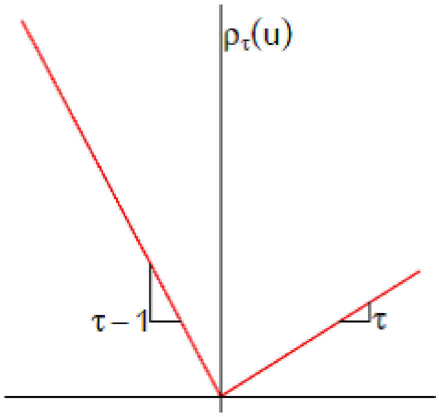

Figure 2.2: Quantile Regression ρ Function

While it is more common to define the sample quantiles in terms of the order statistics, $y_{(1)} \leq y_{(2)} \leq \ldots \leq y_{(n))}$, which results in a sorted arrangement of the original sample. Their formulation as a minimisation problem has the advantage that it yields a natural generalisation of the quantiles to the regression context. The idea of estimating the unconditional mean is simply

$$\hat{\mu} = \operatorname*{argmin}_{\mu \in \Re} \sum (y_i - \mu)^2.$$

This estimation can be extended to the estimation of the linear conditional mean function $E(Y|X = x) = x'\beta$ by solving

$$\hat{\beta} = \operatorname*{argmin}_{\beta \in \Re^p} \sum (y_i - x'\beta)^2.$$

Similarly, the linear conditional quantile function, $Q_Y(\tau|X = x) = x_i'\beta(\tau)$, can be estimated by solving the minimiser

$$\hat{\beta} = \operatorname*{argmin}_{\beta \in \Re^p} \sum \rho_\tau(y_i - x'\beta).$$

In contrast to ordinary least squares, the minimisation in quantile regression is done for each subset defined by ρ_τ. The τ^{th} quantile is estimated with the parametric function $\xi(x_i, \beta)$. The requirement for the square in the unconditional mean case is to ensure each calculated term is positive, the check function ρ_τ ensures this in the quantile regression case and so the square of difference is not required.

2.2.2 Bayesian Approach

Unlike the frequentist approach to statistics, the Bayesian approach provides us with the entire posterior distribution of the parameter of interest. Additionally, it allows for uncertainty of a parameter to be taken into account when making a prediction. Irrespective of the true distribution of the data, Bayesian inference for quantile regression produces the likelihood function based on the asymmetric Laplace distribution.

A random variable ω follows the asymmetric Laplace distribution if its probability density function is given by

$$f_{\tau}(\omega;\mu,\sigma) = \frac{\tau(1-\tau)}{\sigma}\exp\left\{-\rho_{\tau}\left(\frac{\omega-\mu}{\sigma}\right)\right\},$$

where $0<\tau<1$, μ is the location parameter, σ is the scale parameter and $\rho_{\tau}(u)$ is the loss function defined as

$$\rho_{\tau}(x) = \begin{cases} \tau * x, & \text{if } x \geq 0 \\ (\tau-1) * x, & \text{if } x < 0, \end{cases}$$

or in a simpler form,

$$\rho_{\tau}(u) = \frac{|u| + (2\tau-1)u}{2}. \tag{2.3}$$

Using general modelling techniques like ordinary least squares the estimates of the regression parameters β are computed by assuming that

(i) conditional on x, the random variables Y_i, are mutually independent with distributions $f(y;\mu_i)$ specified by the values of $\mu_i = E[Y_i|x_i]$;

(ii) for some known link function g, $g(u_i) = x_i'\beta$.

However, in this project we are interested in the conditional quantile, $q_{\tau}(y_i|x_i)$, in contrast to the conditional mean, $E[Y_i|x_i]$. Simple assumptions can be made so that regardless of the distribution of the data it is possible to solve for the quantiles in the framework of the general linear model. Assuming the following makes this possible,

(i) $f(y;\mu_i)$ is asymmetric Laplace;

(ii) $g(\mu_i) = x_i'\beta(\tau) = q_{\tau}(y_i|x_i)$ for any $0<\tau<1$.

An issue with using Bayesian statistics is the requirement of a conjugate prior distribution for quantile regression formulation. While this is generally not known Markov Chain Monte Carlo (MCMC) methods can extract posterior distributions of the unknown parameters which allows the use of any prior distribution. While giving us the marginal and joint posterior distributions of all the unknown parameters, the Bayesian approach, also provides us with a very practical way of including parameter uncertainty in predictive inferences. Given the observations, $y = (y_1, \ldots, y_n)$, the posterior distribution of β, $\pi(\beta|y)$ is given by

$$\pi(\beta|y) \propto L(y|\beta)p(\beta),$$

where $p(\beta)$ is the prior distribution of β and $L(y|\beta)$ is the likelihood function written as

$$L(y|\beta) = \tau^n(1-\tau)^n \exp\left\{-\sum_i \rho_\tau(y_i - x_i'\beta)\right\} \tag{2.4}$$

which is using equation 2.3 with a location parameter $\mu_i = x_i'\beta$.

The optimum strategy is to choose β such that the resulting joint posterior distribution will be proper. It can be shown that the best choice for the prior of β is for it to be improper uniform.

2.2.3 Linear Mixed Models Approach

In statistics it is sometimes necessary to take into account the correlation of observations which belong to the same unit or cluster of the data being analysed. Mixed effects models represent an efficient, flexible and popular way of analysing this complex data. The modelling technique attempts to model and estimate the variability between clusters by using cluster specific random effects. The fact that mixed models can estimate the between cluster variability is a significant advantage over standard modelling techniques as they can provide conditional inferences. In mixed models, both fixed and random effects are assumed to be location-shift effects.

The general idea of linear mixed models approach for quantile regression came from Marco Geraci and Matteo Bottai's asymmetric Laplace approach. A generalisation of this model was further developed by Geraci and Bottai (2013) and is the foundation of the `lqmm` `R` package which will be discussed later.

A continuous random variable $\omega \in \Re$ is said to follow an asymmetric Laplace distribution with parameters (μ, σ, τ), $\omega \sim AL(\mu, \sigma, \tau)$, if its density can be expressed as

$$p(\omega|\mu, \sigma, \tau) = \frac{\tau(1-\tau)}{\sigma} \exp\left\{-\frac{1}{\sigma}\rho_\tau(\omega - \mu)\right\}$$

where $-\infty < \mu < \infty$ is the location parameter, $\sigma > 0$ is the scale parameter,

$0 < \tau < 1$ is the skew parameter and $\rho_\tau(v)$ is

$$\rho_\tau(x) = \begin{cases} \tau * x, & \text{if } x \geq 0 \\ (\tau - 1) * x, & \text{if } x < 0 \end{cases}$$

This is the general loss function which is used in each method of quantile regression described in this chapter. In our case, the parameter μ is of great interest as this is the τ^{th} quantile of ω, that is that $Pr(\omega \leq \mu) = \tau$.

If the random variable ω is comprised of n independent $\omega_i's$ with common skew (τ) and scale (σ) parameters and different location (μ), then $\omega_i \sim AL(\mu_i, \sigma, \tau)$ for $i = 1, \ldots, n$. This leads to a simplified expression for $\omega's$ density function

$$p(\omega|\mu, \sigma, \tau) = \sigma_n(\tau) \exp\left\{-\frac{1}{\sigma}\rho_\tau(\omega - \mu)\right\}$$

where $\sigma_n(\tau) = \frac{\tau^n(1-\tau)^n}{\sigma^n}$ and $\rho_\tau(y - \mu) = \sum_{i=1}^n \rho_\tau(\omega_i - \mu_i)$.

Geraci and Bottai (2013) proposed a random-intercepts quantile regression model for longitudinal data using the asymmetric Laplace to model the τ^{th} conditional quantile of a continuous response variable. In particular, they assumed the following regression function

$$Q_{y|u}(\tau|x, u) = X\beta^{(\tau)} + u,$$

where (y, X) represents the longitudinal data, u a vector of subject-specific random effects and $Q_{y|u}$ denotes the inverse of the unknown distribution $F_{y|u}$. The τ^{th} regression quantile of $y|u$ was then estimated under the convenient assumption $y|u \sim AL(X\beta^{(\tau)} + u, \sigma^{(\tau)}, \tau)$, where the τ-dependant parameters $\beta^{(\tau)}$ and $\sigma^{(\tau)}$ have a frequentist interpretation. There exists a link between the L_1 norm regression problem and the asymmetric Laplace based estimation of the coefficients $\beta^{(\tau)}$ which I will not get into here.

2.3 Implementation

As R (R Core Team, 2013) will be used as the development environment for the solution to this problem, we will review the current packages that implement the above quantile regression methods. The main package available in R for each of the methods described above are outlined in the list below. We will discuss each of these methods in the following pages. Advantages, disadvantages and inadequacies of each of the approaches will be highlighted. Included in the following is output from R using the specified approach to compute the quantiles. The sample data used is the Engel data set which is available with the `quantreg` package available on CRAN. The data consists of income and food expenditure values for 235 Belgian working class households.

- The frequentist method is available in the `quantreg` package developed by Koenker (2013)
- The Bayesian method is available in the `bayesQR` package developed by Benoit et al. (2013)
- The linear mixed models method is available in the `lqmm` package developed by Geraci (2012)

2.3.1 Frequentist Approach

The frequentist approach to quantile regression is implemented in R through Roger Koenker's extensive `quantreg` package. The package is currently in version 5.05. The package contains an array of functions for calculating regression quantiles, plotting the results, formatting the results table and a multitude of data sets which can be used as test data for the package. The basic fitting routine is used as follows `rq(formula, tau=.5, data, method='br')`. The function can accept more parameters than shown here however these are the parameters of interest here. The `formula` argument specifies the model that is desired. In the Engel data example below I fitted a simple bivariate linear model so the formula was simply `foodexp ~ income`, if we had two explanatory variables it would simply be `foodexp~income + something else`. The parameter `tau` is defaulted to calculate the median regression line, however it will accept a single quantile of interest or a vector of quantiles which is used in the example below. The `data` argument requires the name of the `data.frame` which contains the variables named in the `formula` argument. In the case of our example `data=engel` was passed to the function. The `method` argument specifies the calculation method which the package should use to calculate the regression quantiles of interest.

The `rq` function will automatically use `method='br'` if no method is specified. The `br` method calculates the regression quantiles using exterior point methods. It controls the quantile regression fitting by the simplex approach embodied in the algorithm of Koenker and d'Orey (1987) based on the median regression algorithm of Barrodale and Roberts (1974). If all values of `tau` lie in $(0, 1)$ then the regression values are returned for the single or multiple quantiles requested. On the other hand, if `tau` lies outside $[0, 1]$ parametric programming methods are used to find all the solutions to the quantile regression problem for `tau` in $(0, 1)$. This method is efficient for problems containing up to several thousand observations and has the advantage of being able to calculate the full quantile regression process. It also implements a scheme for computing confidence intervals for the estimated parameters based on an inversion of a rank test described in Koenker (1994).

Two other methods to compute the regression quantiles are `method='fn'` and `method='pfn'`. These methods both use the Frisch-Newton algorithm to compute the regression quantiles. The algorithms full detail's are explained in Koenker and Portnoy (1997). In brief, the approach is the reverse of the simplex method, rather than travelling around the exterior of the constraint set it starts from within the constraint set and moves towards the exterior. Instead of taking steepest decent steps at each intersection of exterior edges, it takes Newton steps based on a log-barrier Lagrangian form of the objective function. For exceptionally large problems `method='pfn'` further adds a pre-processing step to the algorithm which can help to speed up the process considerably.

There are also two methods for penalised quantile regression available in the package are `method='lasso'` and `method='scad'`. These methods implement the lasso penalty and Fan and Li's smoothly clipped absolute deviation penalty, respectively. A parameter `lambda` is passed to both functions and is key to the calculations made. In the `lasso` case, if `lambda` is a scalar quantity the penalty function is the l1 norm of the last coefficients, under the assumption that the first coefficient is an intercept parameter that should not be subject to the penalty. When `lambda` is a vector it defines a coordinatewise specific vector of lasso penalty parameters. Similarly, for the `scad` method, if `lambda` is a scalar quantity the penalty function is the scad modified l1 norm of the last coefficients, under the assumption that the first coefficient is an intercept parameter that should not be subject to the penalty. When `lambda` is a vector it defines a coordinatewise specific vector of scad penalty parameters. It should be noted that while these methods are available, Koenker himself states that "These methods should probably be regarded as experimental".

To provide a somewhat more visual explanation of the `quantreg` package I will illustrate an example of its usage on the Engel data set available with the

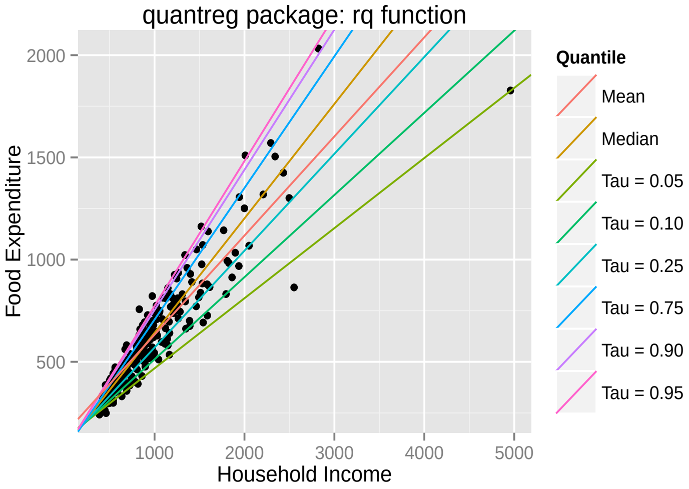

Figure 2.3: Quantreg Estimation of Quantiles on Engel Data

package. The data contains 235 observations of food expenditure and household income for 19^{th} century working class Belgian households. Figure 2.3 shows the $\{0.05_{th}, 0.10_{th}, 0.25_{th}, 0.75_{th}, 0.90_{th}, 0.95_{th}\}$ quantile regression lines, the median fit and the least squares estimate of the conditional mean function. The quantiles were calculated using the `br` method as described above. The least squares estimate was computed using the `lm` function available in `R` for linear regression. It can be seen at the bottom left of the graph the regression lines intersect with one another which is the issue I hope to address in this paper.

Quantile	Intercept	Slope
$\tau = 0.05$	124.880 (98.302,130.517)	0.343 (0.343, 0.390)
$\tau = 0.10$	110.142 (79.888,146.189)	0.402 (0.342, 0.451)
$\tau = 0.25$	95.484 (73.786,120.098)	0.474 (0.420, 0.494)
Median	81.482 (53.259,114.012)	0.560 (0.487, 0.602)
$\tau = 0.75$	62.397 (32.745,107.314)	0.644 (0.580, 0.690)
$\tau = 0.90$	67.351 (37.118,103.174)	0.686 (0.649, 0.742)
$\tau = 0.95$	64.104 (46.265, 83.579)	0.709 (0.674, 0.734)
Mean	147.4754	0.4852

Table 2.1: Regression line coefficients for figure 2.3

The quantiles were also calculated after taking a log transformation of the data. This was used as an attempt to stop the quantiles crossing. It can be seen in figure 2.4 that the quantiles are not crossing at a point close to the origin, however the 0.95^{th} and 0.90^{th} quantile lines are still crossing. A log transformation therefore does not appear to solve the crossing quantiles problem but does improve on the previous example. The R code used to produce this graphic is available in appendix A.1.

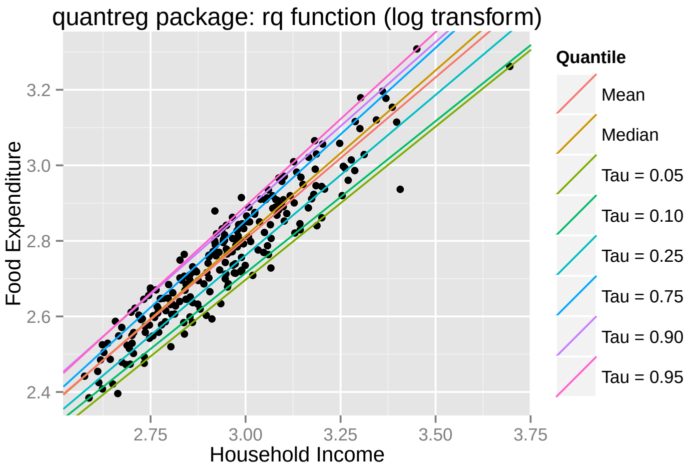

Figure 2.4: Quantreg Estimation of Quantiles on log transformation of Engel Data

Finally, the coefficients of the quantile regression calculations for figure 2.3 are shown in table 2.1 while the log transformation coefficients are shown in table 2.2. The values in parenthesis below the actual value are the confidence bands for that value. Note that the intercept and the slope differ for each of the quantiles and the linear regression model.

Quantile	Intercept	Slope
$\tau = 0.05$	0.2638 (0.2283,0.4607)	0.8113 (0.7398,0.8213)
$\tau = 0.10$	0.3033 (0.1130,0.3601)	0.8041 (0.7866,0.8645)
$\tau = 0.25$	0.2151 (0.1249,0.4095)	0.8495 (0.7822,0.8799)
Median	0.1817 (0.0303,0.3911)	0.8766 (0.8051,0.9302)
$\tau = 0.75$	0.1048 (−0.0057,0.2805)	0.9156 (0.8552,0.9535)
$\tau = 0.90$	0.2075 (0.0335,0.2759)	0.8909 (0.8674,0.9493)
$\tau = 0.95$	0.1258 (0.0804,0.2771)	0.9222 (0.8720,0.9381)
Mean	0.2368	0.8559

Table 2.2: Regression line coefficients for figure 2.4

2.3.2 Bayesian Approach

The implementation of the Bayesian approach to quantile regression is available in the bayesQR package developed by Benoit et al.. The implementation as expected is much less efficient with regard to processing time than the `quantreg` package due to the iteration process required for Bayesian calculations. The package currently resides in version 2.1 and was updated in 2013. The quantile regression function is called by `bayesQR(formula, data, quantile=0.5, ndraw, prior)`.

The `formula` argument specifies the model that is desired. It follows the same format as in the `rq` function discussed earlier. The `data` parameter is an optional parameter to specify the data object from which the dependant and independent variables are taken. The parameter `quantile` is defaulted to calculate the median regression line, however it will accept a single quantile of interest and will also accept a vector of quantiles as was an option in the `quantreg` package. The `ndraw` parameter specifies how many Markov Chain Monte Carlo draws are to be taken when estimating each quantile required. The `prior` argument allows the user to pass a prior distribution to the model if known otherwise the prior distribution is calculated based on the model type being used.

The package can compute Bayesian quantiles for four types of models; continuous dependant variable without adaptive lasso variable selection, continuous dependant variable with adaptive lasso variable selection, binary dependant variable without adaptive lasso variable selection and binary dependant variable with adaptive lasso variable selection. The computational effort required for each of these methods is similar and is in general extremely computationally intense given a relatively large data set.

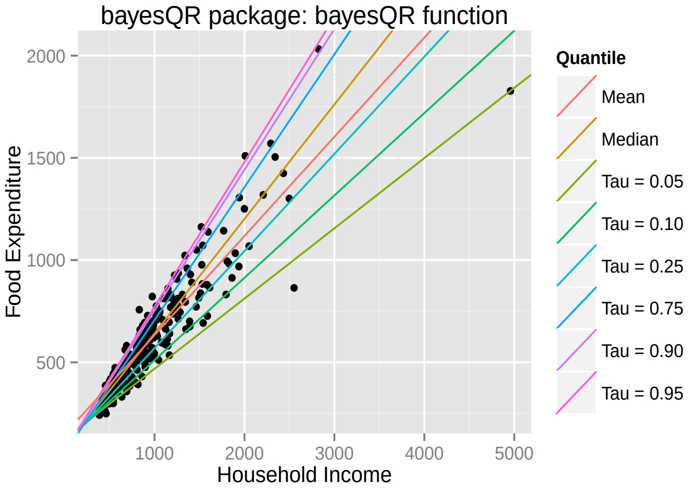

Figure 2.5: bayesQR Estimation of Quantiles on Engel Data

Figure 2.5 shows the regression quantiles produced by the `bayesQR` method on the same data used in the `quantreg` example above. The regression lines are close but not identical to those produced by the frequentist approach in figure 2.3. The problem of crossing quantiles is still present when using the Bayesian approach to quantile regression.

Quantile	Intercept	Slope
Tau = 0.05	124.561	0.344
Tau = 0.10	107.495	0.403
Tau = 0.25	94.078	0.475
Median	81.227	0.560
Tau = 0.75	57.229	0.650
Tau = 0.90	64.690	0.689
Tau = 0.95	64.605	0.708
Mean	147.475	0.485

Table 2.3: Regression line coefficients for figure 2.5

As before the quantiles were also calculated after taking a log transformation of the data. It can be seen in figure 2.6 that the quantiles appear not to cross anywhere in the domain in which it is plotted. The R code used for the two graphics can be found in appendix A.2.

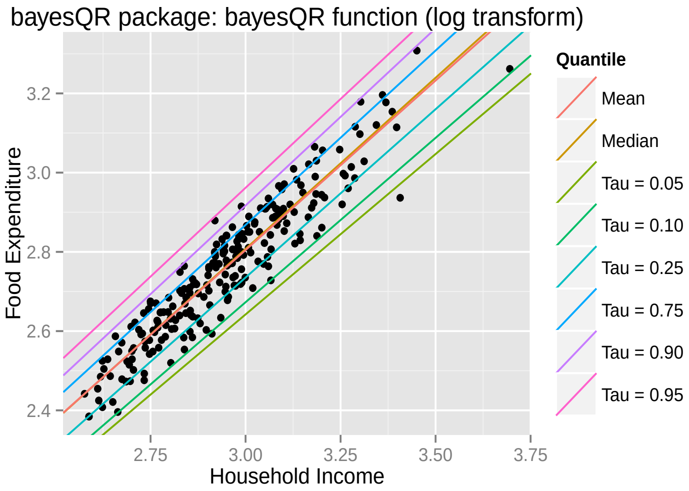

Figure 2.6: bayesQR Estimation of Quantiles on log transformation of Engel Data

Finally, the coefficients of the quantile regression calculations for figure 2.5 are shown in table 2.3 while the log transformation coefficients are shown in table 2.4. The bayesQR package does not provide a simple method for computing the confidence intervals for the quantiles. Note that the intercept and the slope differ for each of the quantiles and the linear regression model. The values are also different to those calculated by the quantreg package.

Quantile	Intercept	Slope
Tau = 0.05	0.2395	0.8084
Tau = 0.10	0.2194	0.8236
Tau = 0.25	0.1976	0.8465
Median	0.2184	0.8653
Tau = 0.75	0.2323	0.8798
Tau = 0.90	0.2312	0.8943
Tau = 0.95	0.2653	0.9031
Mean	0.2368	0.8559

Table 2.4: Regression line coefficients for figure 2.6

2.3.3 Linear Mixed Models Approach

The final package which will be presented here is the `lqmm` package developed by Marco Geraci. The package is the `R` implementation of linear mixed models approach for quantile regression. The package is currently in version 1.03 and was updated in 2013. The package contains a multitude of functions based around the implementation of the asymmetric Laplace solution to the regression quantiles. The functions of most interest are `lqm` and `lqmm` as these are the methods which calculate the regression quantiles.

The `lqm` function is called using `lqm(formula, data, iota=0.5)`. The `formula` and `data` arguments are specified exactly as they were in the `quantreg` packages `rq` function. The `iota` parameter is equivalent to the `tau` parameter in the previous two packages and can accept single or multiple tau values. The function computes an estimate of the τ^{th} quantile function of the response variable, conditional on the covariates, as specified by the `formula` argument. The quantile predictor is assumed to be linear. The function maximises the likelihood of a Laplace regression which is equivalent to the minimisation of the weighted sum of absolute residuals. The optimisation algorithm is based on the gradient of the Laplace log-likelihood.

The `lqmm` function is called using `lqmm(fixed, random, iota=0.5)`. The function is similar to the `lqm` function but allows random effects to be specified as arguments. The `fixed` and `iota` parameters follow directly from the `formula` and `iota` arguments in the `lqm` function. The `random` argument allows the inclusion of random effects and should be specified as `~x1+ ... + xn`, where x_i is a random effect of interest. The function calculates an estimate of the τ^{th} quantile function of the response, conditional on the covariates, as specified by the `fixed` argument and on random effects, as specified by the `random` argument. The quantile predictor is again assumed to be linear and the function maximises the likelihood Laplace regression. The likelihood is numerically integrated via Gaus-

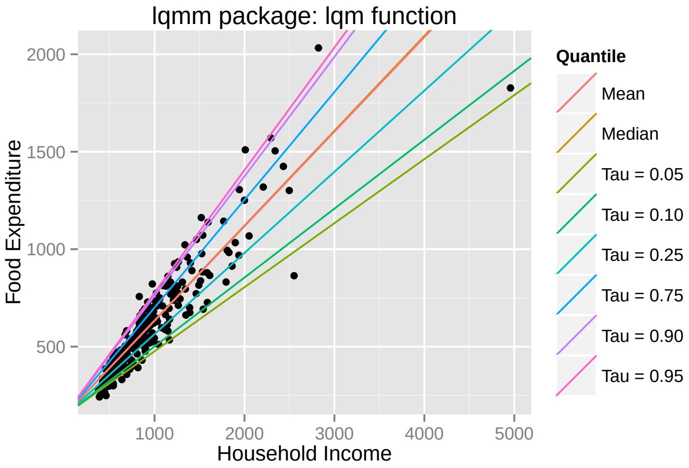

Figure 2.7: lqmm Estimation of Quantiles on Engel Data

sian quadrature techniques. The optimisation algorithm ia based on the gradient of the Laplace log-likelihood.

Quantile	Intercept	Slope
$\tau = 0.05$	147.475	0.329
$\tau = 0.10$	147.475	0.354
$\tau = 0.25$	147.475	0.417
Median	147.475	0.487
$\tau = 0.75$	147.475	0.553
$\tau = 0.90$	147.476	0.613
$\tau = 0.95$	147.476	0.629
Mean	147.475	0.485

Table 2.5: Regression line coefficients for figure 2.7

Again, I have used the Engel data set as the example data set to allow comparison between the three packages. I have used the `lqm` function to calculate the regression quantiles for this example. In figure 2.7 we can see that the regression quantiles appear to meet at a point on the y-axis. However, the median regression line and the mean regression line seem to be identical which is vastly different from what was experienced using the other two packages. The extreme quantile lines, $\tau = \{0.95, 0.05\}$, appear much closer to the median regression fit than they did previously.

When we look at table 2.5, which contains the coefficients for the graphic in figure 2.7, we can see that all the intercepts calculated by `lqm` are exactly equal to the intercept of the least squares estimate for the conditional mean function. This is vastly different from the other two packages where none of the intercepts were equal.

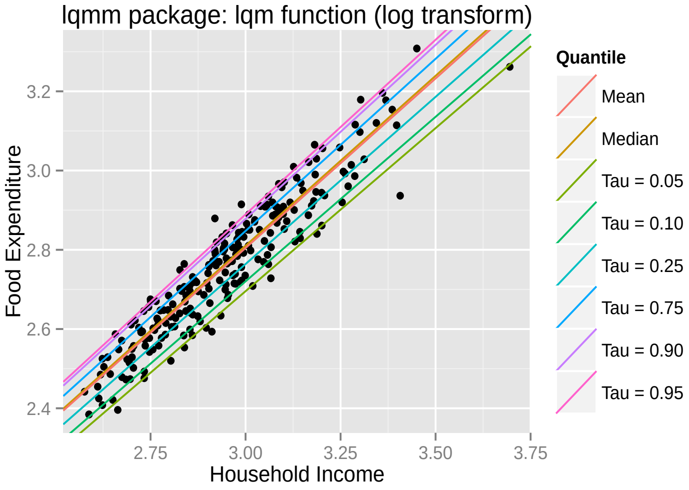

Figure 2.8: lqmm Estimation of Quantiles on log transformation of Engel Data

Finally, a plot of the quantiles calculated by `lqm` for the data after a log transformation is shown in figure 2.8. Again it seems to help in addressing the issue of crossing quantiles, however the upper quantiles, 0.90^{th} and 0.95^{th}, are again very close which could cause a problem. The coefficients are given in table 2.6. The `R` code required to reproduce the graphics and coefficients shown in this section can be found in appendix A.3.

Quantile	Intercept	Slope
$\tau = 0.05$	0.2259	0.8232
$\tau = 0.10$	0.2285	0.8306
$\tau = 0.25$	0.2327	0.8438
Median	0.2372	0.8577
$\tau = 0.75$	0.2409	0.8689
$\tau = 0.90$	0.2441	0.8780
$\tau = 0.95$	0.2452	0.8814
Mean	0.2368	0.8559

Table 2.6: Regression line coefficients for figure 2.8

Chapter 3

Regression Quantiles for Thyroid Data

The previous chapters in this paper have examined existing quantile regression methods in `R` using sample datasets. However, the central focus of this research paper is to develop thyroid reference ranges in pregnancy using quantile regression. This chapter will include the following:

- introducing the thyroid data which will be used throughout this report;
- fitting linear models to the thyroid data using existing `R` packages;
- outlining the inadequacies of current packages for quantile regression.

3.1 Thyroid Data

The thyroid data used in this research project was collected from pregnant women who attended a large maternity hospital between February 2010 and February 2011. A sample size of 300 or more women were sought for the analysis as results from Cotzias et al. (2008) had shown amble power from a sample size of 300 women.

Healthy women attending the antenatal care clinic with a singleton pregnancy were randomly selected for participation in the study. Women with known thyroid disease, autoimmune diseases, diabetes mellitus, recurrent miscarriage, hyperemesis gravidarum and pre-eclampsia were excluded from the study. The women who took part in the study were between 10 and 42 weeks of gestation, and each woman was sampled on only one occasion. Each individual assessed also had an uncomplicated medical history to avoid discrepancies in the results caused by underlying health conditions.

The final data file consists of 10 variables, each of which is shown in the data extract below, and samples taken from 311 pregnant women. The key concern of this project is to develop gestation-specific reference ranges for each of the following thyroid hormones:

T4 level of thyroxine hormone in each patient

T3 level of triiodothyronine hormone in each patient

TSH level of thyroid stimulating hormone in each patient

TPO level of thyroid peroxidase enzyme in each patient

The sample dataset available contains data for the 1st, 2nd and 3rd trimesters of pregnancy. The 1st trimester data was recorded during the 10th − 14th weeks of gestation, while the 2nd and 3rd trimesters were recorded in the 14th − 26th week period of gestation and the 27th − 42nd week period of gestation respectively. A sample of the combined dataset can be seen in table 3.1. A data dictionary for the variables of the thyroid data is given in table 3.2.

MRN	Age	Ethnicity	Smoking	T4	TSH	T3	TPO	Gestation	Trimester
937395	30	Irish	No	14.6	0.37	4.6	5.3	12.00	T1
1975207	28	Irish	No	13.8	2.07	5.3	0.0	13.00	T1
464433	28	Irish	No	13.8	1.16	4.8	0.0	12.00	T1
1123123	28	Irish	No	12.3	2.71	4.7	5.2	12.00	T1
1855554	35	Irish	No	14.3	0.57	4.8	0.0	13.29	T1
1068450	28	Irish	No	14.2	2.39	4.5	7.0	12.86	T1
2099487	26	Polish	No	11.0	2.82	4.8	22.9	22.00	T2
2162414	34	English	No	14.0	0.62	4.7	6.8	14.43	T2

Table 3.1: Extract from Thyroid data

Variable	**Description**
MRN	unique hospital identifier for each patient
Age	age of patient in years
Ethnicity	country/region of birth
Smoking	binomial response yes or no, if yes the number of cigarettes smoked per day is recorded
T4	level of thyroxine hormone in each patient
TSH	level of thyroid stimulating hormone in each patient
T3	level of triiodothyronine hormone in each patient
TPO	level of thyroid peroxidase enzyme in each patient
Gestation	number of weeks pregnant

Table 3.2: Data dictionary for thyroid data

3.2 Descriptive Statistics

This section provides descriptive statistics which describe the maternal characteristics of the population in this study. A graphical view of the distribution of each of the 4 thyroid hormone variables in our dataset is also provided, along with appropriate statistical tests, is also included. If any of the variables of interest exhibited skewness, the Box-Cox method was used to identify the optimal power transformation to achieve normality of the data (Box and Cox, 1964), a brief explanation of which is provided later.

The final study population, had a median maternal age of 30 years, with an interquartile range of $27-33$ years and a range from $17-40$ years. The population could be divide into four key geographical groupings as follows:

- **White Irish** 77.5% (n=241)
- **European** 16.1% (n=50)
- **Asian** 4.5% (n=14)
- **African** 1% (n=3)

Of the 311 women in the final study population, over three-quarters of the individuals were non-smokers (77.8%), with only 2.6% of the population smoking more than 10 cigarettes per day. Finally, the data was split evenly across the trimesters with 34.4% of women tested being in the 1^{st} trimester, while 32.8% of the population were tested during both their 2^{nd} and 3^{rd} trimesters.

As part of the instal analysis of the data, correlation among each of the thyroid hormone variables were calculated. It was found that a statistically significant relationship existed between TPO and TSH (Pearson's r value 0.230, p-value = 0.001). No other correlations among the thyroid hormone variables existed at a statistically significant level. However, due to the relationship between TSH and TPO it was decided to not model on TPO in our analysis. Therefore, the variables of interest are now free T4, free T3 and TSH.

3.2.1 Distributions of variables

When producing statistical models it is appropriate to model data that is normally distributed. If the initial data is not normally distributed then a power transformation such as a Box-Cox transformation can be applied to the data to correct this. The distributions for each of the four thyroid hormone variables (free T4, free T3, TSH, TPO) will be examined in this section using quantile-quantile plots (Q-Q plots), histograms and box-plots. A Shapiro-Wilk normality test will also be applied to the data and an interpretation of the results will be detailed.

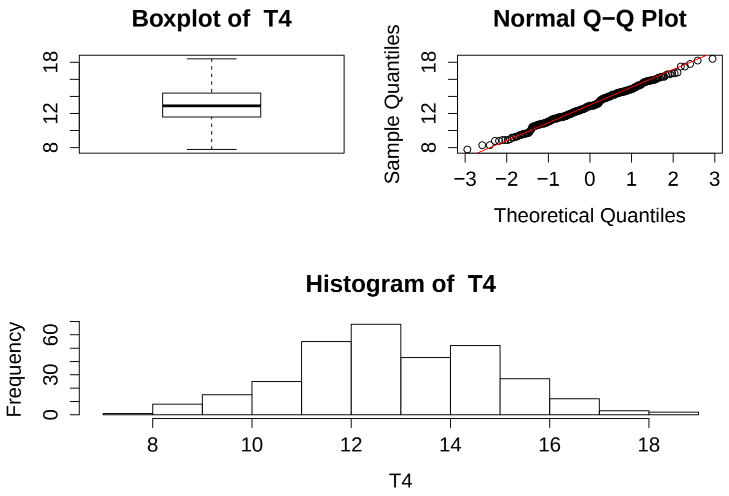

Figure 3.1: Distribution Plots for free T4

The normality plots for free T4 can be seen in figure 3.1. The QQ-plot, boxplot and histogram show slightly heavy tails but otherwise it appears that the free T4 data is normally distributed. A Shapiro-Wilk test carried out on free T4 verified this conclusion further, as a p-value of 0.45 (Shapiro-Wilk W value 0.9952), resulting in a failure to reject the null hypothesis that the data is normally distributed. Thus no transformation of the free T4 data is required.

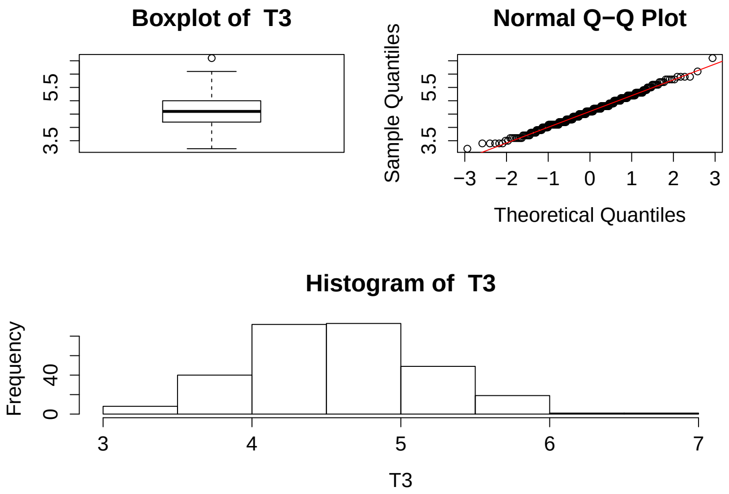

Figure 3.2: Distribution Plots for free T3

Equivalent to the plots produced for free T4 the corresponding plots for free T3 can be seen in figure 3.2. Similarly to free T4 the T3 data looks relatively normally distributed in each of the plots produced, with the exception of one outlier in the upper tail of the distribution. Carrying out the Shapiro-Wilk test for normality results in a p-value of 0.13 (Shapiro-Wilk W value 0.9925) which is not statistically significant at the 5% level and so again we fail to reject the null hypothesis that the T3 data is normally distributed. As before no transformation of the T3 data will be required before modelling.

The normality checks for TSH can be seen in figure 3.3. The boxplot shows two outliers in the upper range of the TSH data. The QQ-plot also shows a departure from normality at each of the tails. From the histogram it can be seen that the data is positively skewed. Carrying out the Shapiro-Wilk normality test for TSH results in a p-value of 0.0001 (Shapiro-Wilk W value 0.9784) which is statistically significant. Therefore, the null hypothesis is rejected in favour of the alternative hypothesis, that the TSH data is not normally distributed. Therefore, a power transform of TSH will be required prior to modelling the regression quantiles for the variable.

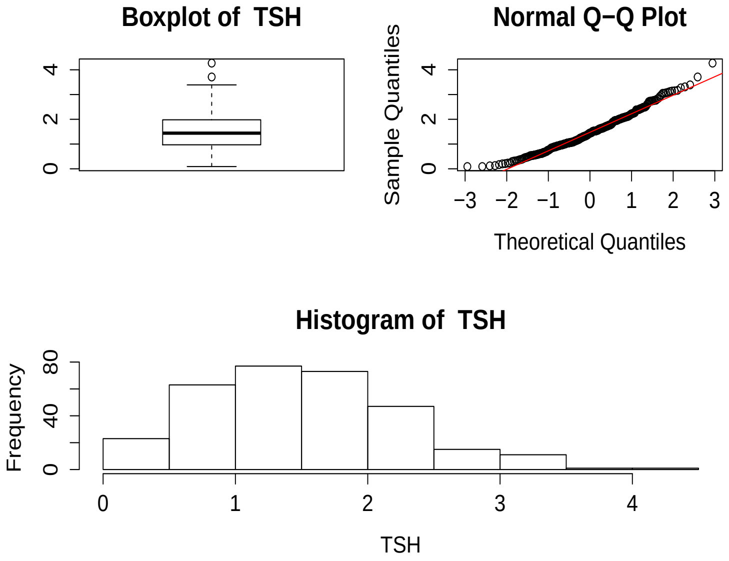

Figure 3.3: Distribution Plots for TSH

All of the R code for the plots and statistics calculated in this section can be found in Appendix B.1.

3.2.2 Box-Cox Transformation

In statistical models many important features follow from the assumption that the data being modelled is normally distributed with a common variance and additive error structure (Sakia, 1992). Once the theoretical assumptions required for the modelling technique are approximately satisfied, then the usual procedures can be applied. In situations where the assumptions are seriously violated several options are available:

1. ignore the violation of the assumptions and carry out the modelling procedure as if all assumptions were satisfied
2. for each of the violated assumptions decide on a more appropriate assumption and use a valid technique that takes account of this new assumption
3. design a new model that has important aspects of the original model and satisfies all the assumptions - apply a power transformation to the data or remove evident outliers in the data
4. use a distribution-free procedure that is valid even if various assumptions are violated

In the vast majority of cases (*iii*) is chosen as the best procedure to follow. In this paper I will use the Box-Cox power transformation for data which is not normally distributed. The Box-Cox transformation is defined as follows

$$y_i = \begin{cases} y_i^\lambda, & \text{if } \lambda \neq 0 \\ \log(y_i), & \text{if } \lambda = 0 \end{cases}$$

From a modelling perspective this results in transforming the data before modelling and then transforming it back to be in line with the original dataset. For a simple linear model of the form

$$y = \alpha + \beta x + \epsilon$$

a power transformation of the y variable results in modelling

$$\begin{cases} y^\lambda = \alpha + \beta x + \epsilon, & \text{if } \lambda \neq 0 \\ \log(y) = \alpha + \beta x + \epsilon, & \text{if } \lambda = 0 \end{cases}$$

However, this model needs to be transformed back to the original variables which results in the following

$$\begin{cases} y = (\alpha + \beta x + \epsilon)^{\frac{1}{\lambda}}, & \text{if } \lambda \neq 0 \\ y = e^{(\alpha + \beta x + \epsilon)}, & \text{if } \lambda = 0 \end{cases}$$

After the instal exploration of the variables of interest in the thyroid data, it was shown that only TSH required a transformation before modelling. Using the `box.cox` method (see Appendix B.2) it was shown that the optimal value of λ was 0.59. However, a square root transformation ($\lambda = 0.5$) is much more practical to use.

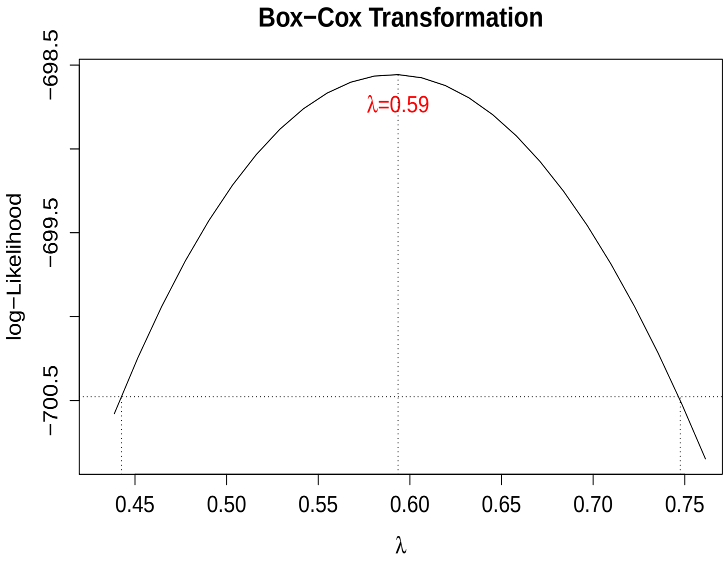

Figure 3.4: Box-Cox transformation parameter for TSH

From the above output, the model that will now be used to calculate the regression quantiles for TSH will be

$$\sqrt{TSH} = \alpha + \beta(Gestation) + \epsilon$$

3.3 Linear Quantile Models

The main objective of this project is to develop a method which can calculate reference ranges for each of the variables with respect to the gestation week. This allows the medical practitioners to gain information on those patients who should be treated for thyroid disease. Quantile regression is the method of choice for calculating the reference ranges, however as has already been discussed in detail the crossing of the quantiles causes problems for classifying patients for treatment or not.

In this section, the quantile regression models will be developed for each of our thyroid hormone variables (T4, T3, TSH). Each will be modelled with respect to gestation week of the patient, in a hope to develop reference ranges for thyroid disease during pregnancy. This issue of crossing quantiles will be addressed in the next chapter and so observations of such behaviour will be mentioned here but not expanded upon.

The R code required for producing the quantile models and the graphics in this section can be found in Appendix B.3.

3.3.1 Free T4

As stated earlier in this chapter the free T4 data was normally distributed and so no power transformation was required on the data before modelling. As a first model I decided to use a simple linear model as follows

$$T4 = \alpha + \beta(Gestation) + \epsilon$$

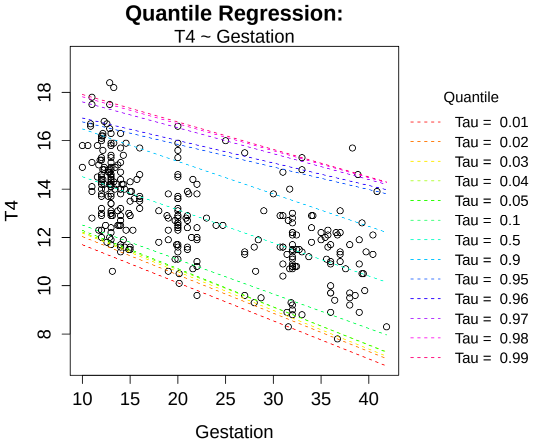

Figure 3.5: Simple Linear Model for free T4

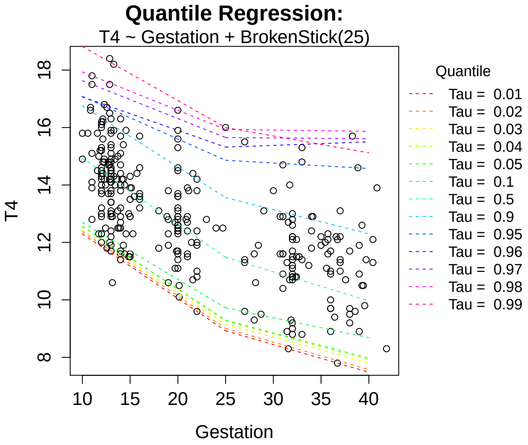

Figure 3.6: Broken Stick Model for free T4

However, from the scatterplot of free T4 v Gestation in figure 3.5, it can be observed that free T4 has a prominent decline until approximately 25 weeks gestation, after which free T4 levels begin to level off. To capture this additional piece of information, a "Broken Stick" model was fitted with a change point at 25 weeks gestation. The model is defined as follows

$$T4 = \alpha + \beta_1(Gestation) + \beta_2(Gestation - 25)^+ + \epsilon$$

While this model appears to capture the change in trend of free T4 and improve the overall fit (figure 3.6), it still has issues with overlapping quantile estimates. This issue will be addressed in the next chapter.

3.3.2 Free T3

Upon analysis of the free T3 variable earlier in this section it was shown that the data approximately followed a normal distribution and that no transformation of the data was required before modelling. Similar, to T4 a simple linear model of the form

$$T3 = \alpha + \beta(Gestation) + \epsilon$$

was fitted to the data and the scatterplot in figure 3.7 was produced.

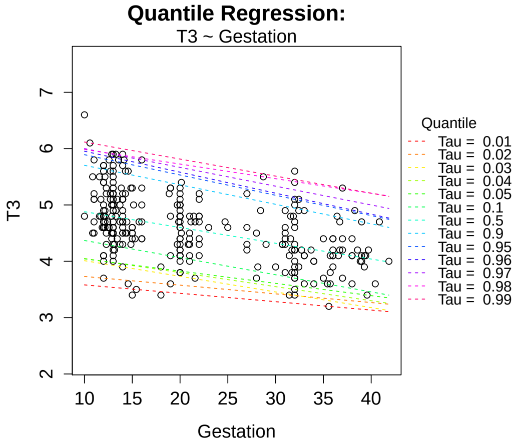

Figure 3.7: Linear Model for free T3

No clear patterns are clear from this scatterplot as was the case when viewing the initial model for T4, however a model of the form

$$T3 = \alpha + \beta_1(Gestation) + \beta 2(Gestation^2) + \epsilon$$

was fitted with the resultant plot shown in figure 3.8.

The quadratic model appears to have improved the fit of the data in comparison with the initial linear model that was fitted. However, even though this fit appears to model the data better the issue of quantile crossing is a dominant feature of the quadratic model plot.

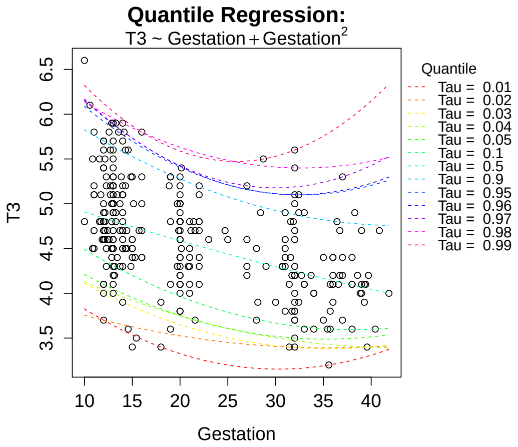

Figure 3.8: Quadratic Model for free T3

3.3.3 TSH

Finally, a model will be fitted to establish the relationship between TSH and the week of gestation of patients. From the distribution analysis of TSH it was seen that the data was not normally distributed and so a square root transformation needed to be applied. The model took the form of

$$\sqrt{TSH} = \alpha + \beta(Gestation)$$

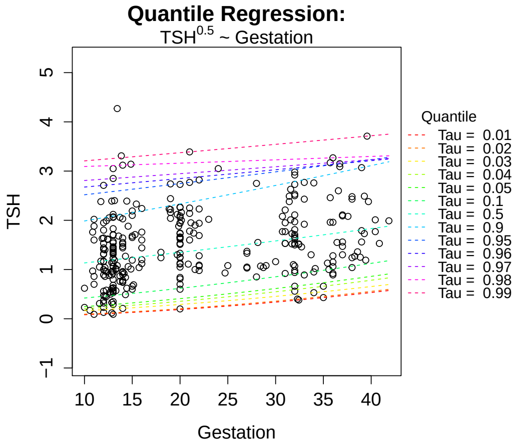

Figure 3.9: Linear model for TSH

The output from the model can be seen in figure 3.9. Again the regression quantile curves cross and while the model seems a good fit for the data this major issue makes it particularly hard to interpret sensible confident reference intervals for thyroid disease in pregnancy.

Chapter 4

Non-crossing Quantile Estimates

As outlined previously, the major problem with quantile estimates for this data is the crossing of the quantile lines. Crossing quantile estimates make it impossible to assign an observation to a single reference range. For example, if we only treat the upper 5% of the population and if the 0.95^{th} and the 0.90^{th} quantile estimates are crossing then how do we decide who to treat? If we assign a patient to be above the 0.95^{th} quantile estimate than they get treated but may not actually need the treatment. The opposite, which is much worse, could also occur, we assign a patient to be in the range between the 0.90^{th} and the 0.95^{th} quantile estimates, they do not receive the treatment and they do contract a thyroid disease. A simple example like this shows just how important non-crossing quantile estimates really are.

To address the problem, the intention is to use non-parametric approaches to calculating the quantile estimates. While some work has been carried out by others on the topic there exists no general off-the-shelf solution to the problem of overlapping quantile estimates. The solution provided uses a combination of linear programming constraints and b-spline functions.

4.1 Theory behind Proposed Solution

The objective function for quantile regression is defined as

$$\hat{\beta} = \underset{\beta \in \Re^p}{\operatorname{argmin}} \sum \rho_\tau(y_i - x'\beta)$$

and thus the β estimates are independent for each quantile estimated. That is to say that the β estimates for $\tau = 0.01$ are entirely independent of the β estimates for $\tau = 0.02$. This independent estimation of the quantiles allows for the possibility of crossing quantile curves, making classification of an individual to an interval impossible.

Crossing quantile occur when, for some values of x,

$$\hat{y}_{\tau_1} = \hat{\beta}_0^{(\tau_1)} + X\hat{\beta}^{(\tau_1)}$$

intersects and crosses

$$\hat{y}_{\tau_2} = \hat{\beta}_0^{(\tau_2)} + X\hat{\beta}^{(\tau_2)}$$

given that $0 \leq \tau_1 \leq \tau_2 \leq 1$. To eliminate the crossing of quantiles we require that

$$\hat{y}_{\tau_1} < \hat{y}_{\tau_2}$$

for all x and all $\tau_1 < \tau_2$. This in turn implies that

$$\hat{\beta}_0^{(\tau_1)} + X_1\hat{\beta}_1^{(\tau_1)} + \ldots + X_p\hat{\beta}_p^{(\tau_1)} < \hat{\beta}_0^{(\tau_2)} + X_1\hat{\beta}_1^{(\tau_2)} + \ldots + X_p\hat{\beta}_p^{(\tau_2)}$$

which when simplified using matrix, notation results in

$$\tilde{X}[\beta_{\tau_2} - \beta_{\tau_1}] > 0$$

where $\tilde{X} = [\mathbf{1}\ \mathbf{X}]$ and $\beta_{\tau_i} = [\beta_0^{\tau_i}\ \beta_1^{\tau_i}\ \ldots\ \beta_p^{\tau_i}]$.

One way to enforce that $\hat{\beta}_{\tau_2} - \hat{\beta}_{\tau_1} > 0$ is to enforce that all $\hat{\beta}_i^{(\tau_2)} > \hat{\beta}_i^{(\tau_1)}$, during the linear programming stage in which the quantiles are calculated. The Frisch-Newton adaption of the simplex method allows such a constraint to be added to the optimisation problem. The Frisch-Newton allows a $\geq$ constraint to be set, as a result the constraint used here will be $\hat{\beta}_i^{(\tau_2)} \geq \hat{\beta}_i^{(\tau_1)} + \epsilon$, where epsilon is of the order of 3.6×10^{-11}. ϵ is included to act as a correction factor after including a $\geq$ constraint instead of a $>$ constraint. In concise notation, the proposed solution is to solve the following optimisation problem

$$\hat{\beta} = \underset{\beta \in \Re^p}{\operatorname{argmin}} \sum \rho_\tau(y_i - x'\beta)$$

subject to the constraint $\hat{\beta}_{(\tau_i)} \geq \hat{\beta}_{(\tau_{i-1})} + \epsilon$ for $0 < \tau < 1$.

The approach thus far attempts to estimate the current quantile by constraining the coefficients to be greater than the coefficients of the previous estimated quantile. While carrying out quantile regression analysis in this fashion will remove crossing in quantiles it will not produce optimal solutions. The asymptotic variance of a quantile estimator is proportional to

$$\frac{\tau(1-\tau)}{f(F^{-1}(\tau))}$$

where $f(\cdot)$ is the pdf of the error distribution and $F(\cdot)$ is the cdf of the error distribution.

For the normal distribution, it can be shown that the asymptotic variance is minimised at $\tau = 0.5$. Thus the estimated quantile at $\tau = 0.5$ is relatively more accurate than other estimated quantiles. From this result it makes sense to estimate the quantiles by starting at the median regression line and working out to the smaller quantiles, in both directions. The solution proposed here incorporates this feature and estimates the 0.5^{th} quantile even if it is not required as a quantile of interest. For quantiles greater than the median the original constraint remains valid, while no constraint is no applied to the median quantile. However, the constraint needs to be changed for quantiles less than the median quantile. The constraint required is

$$\hat{\beta}_{(\tau_i)} \leq \hat{\beta}_{(\tau_{i+1})} - \epsilon$$

however, the Frisch-Newton method does not allow $\leq$ constraints to be set and so the constraint which is used is

$$-\hat{\beta}_{(\tau_i)} \geq -\hat{\beta}_{(\tau_{i+1})} + \epsilon$$

In short the solution proposed here applies constraints to the original optimisation problem to eradicate the issue of quantiles crossing, thus allowing simpler classification of individuals to a range. The proposed solution results in the optimisation problem with constraints taking the form of,

$$\hat{\beta} = \underset{\beta \in \Re^p}{\operatorname{argmin}} \sum \rho_\tau (y_i - x'\beta)$$

$$\text{subject to} \begin{cases} -\hat{\beta}_{(\tau_i)} \geq -\hat{\beta}_{(\tau_{i+1})} + \epsilon, & \text{if } 0 < \tau < 0.5, \\ \text{no constraints}, & \text{if } \tau = 0.5, \\ \hat{\beta}_{(\tau_i)} \geq \hat{\beta}_{(\tau_{i-1})} + \epsilon, & \text{if } 0.5 < \tau < 1 \end{cases}$$

4.2 Implementation of Proposed Solution in R

The solution proposed here has been implemented in `R`, the code for which is provided in Appendix C.1. Roger Koenker's `quantreg` package provided a basis for the implementation of the method and the Frisch-Newton simplex code is adapted from this package.

The code provided as a solution to the issue of crossing quantiles is broken into 7 key sections which are outlined in the following list. The following also details the tasks carried out in each step and how they contribute to the overall solution of the problem.

1) **Read Model Arguments**

   The code `QR(formula, tau, data)` calls the function for non-crossing quantiles. The `formula` argument takes the form `Y ~ X`, just as it did in the examples provided in Chapter 2. The `tau` argument accepts single tau values to be estimated or a vector of tau to be estimated. The `data` argument takes the name of the dataset from which the variables of interest come from.

   The model arguments are read into a `model.frame` object, this object will contain the dependent and independent variables and will act as the data frame for the entire analysis. In this section, checks are also carried out to ensure that the design matrix is not singular, if the design matrix is singular computation is stopped and a warning returned to the user.

2) **Load required packages, constants and functions**

   This section of the code installs and loads the required `quantreg` package, so that the `rq.fit` fortran code is available for the calculations of the quantiles. The check function denoted by $\rho_\tau(u)$ is also defined here, more information on the check function can be found in Chapter 2.2.1. Finally, ϵ is also defined in this section to be a value of `.Machine$double.eps^(2/3)` which is equivalent to 3.666853×10^{-11}.

3) **Check validity of inputted taus**

   As discussed earlier the tau values required for quantile regression must be in the range $0 < \tau < 1$, this section of code ensures this constraint is satisfied and stops computation if it is not. As users specifing $\tau = 0$ or $\tau = 1$ is likely, these requests do not throw an error but rather $\tau = 0$ becomes $\tau = \epsilon$ and $\tau = 1$ becomes $\tau = 1 - \epsilon$.

4) **Define required data objects**

   All data objects to hold the relevant output is defined here based on the number of tau values which we are interested in. A matrix, `coef`, of size $n \times p$ is defined for the coefficients, where n is the number of tau values of interest and p is the number of model parameters plus one to include the intercept. A vector, `rho`, is defined to hold the value of the check function for each of the specified quantiles. Finally, two matrices, `fitted` and `resid`, are defined to hold the fitted values and the residuals respectively, each of size $n \times d$, where d is the number of observations in the data.

5) **Calculate regression quantiles**

   The calculation of the regression quantiles is carried out in full in this section.

There is two different approaches depending on the number of quantiles required.

If only one tau is specified in the model arguments then the function will return the relevant output for that single quantile model. In this case no constraints are applied to the coefficients as non-crossing is gaurenteed when only one quantile is of interest.

If more than one tau is of interest in the analysis then the first step is to calculate the median regression quantile, weather it is of interest or not. The tau values are then split into two subsets, one set will contain the taus with values greater than 0.5 while the other will contain the taus with values less than 0.5. The quantiles will then be estimated with the correct constraints applied. A call to the `rq.fit.fnc` function in the `quantreg` package with the correct constraints results in the calculation of each τ^{th} quantile required by the user.

6) **Tidy up the output**

   The output is formatted into tables to make output from the function concise and clear.

7) **Return the required information**

   The required output is returned as a single variable which has multiple components, including the coefficients, fitted values, residuals, check function values and more, for each tau specified. Each individaul component can be accessed using the $ symbol in `R`, for example if `model` is an object produced by `QR` then `model$coef` will return the coefficients for that object.

The entire code for the function, which is commented, is provided in Appendix C.1.

4.3 Application to thyroid data

In this section, the thyroid data, which this project is based upon, will be revisited and in particular the models specified for T3, T4 and TSH will be reviewed. The models will then be fitted using the proposed solution to non-crossing quantiles provided in this chapter, to provide a real-life example of this method in action.

The T3 hormone variable appeared to be curve linear from the analysis in Chapter 3 and so a model of the form

$$T3 = \beta_0 + \beta_1(Gestation) + \beta_2(Gestation^2) + \epsilon$$

was fitted. However, crossing quantiles were still evident in figure 3.8 and so reference ranges were still impossible to specify exactly. Using the same model and calculating the same quantile curves, using the new method outlined above results in figure 4.1. The quantiles never cross in this graphic and as a result it

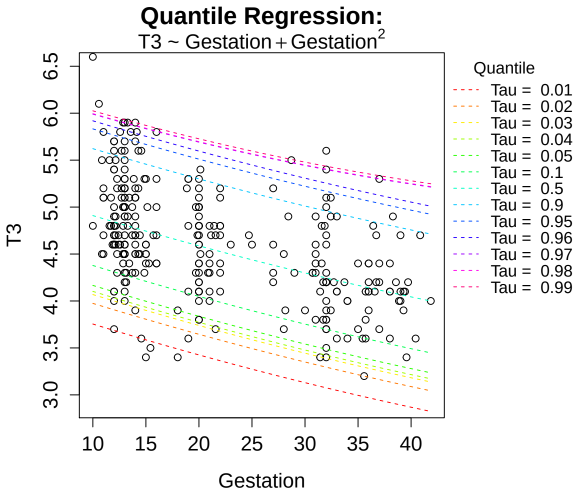

Figure 4.1: Non-crossing quantile curves for T3

is possible to differentiate between individuals which belong to different reference ranges. This allows doctors to treat those who need to be treated and also to not treat those who do not require treatment.

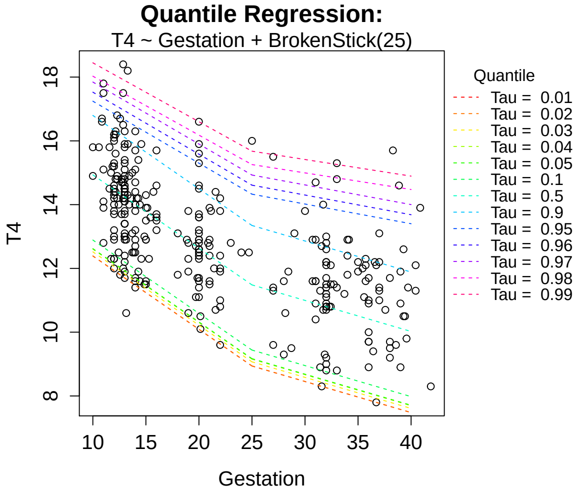

Figure 4.2: Non-crossing quantile curves for T4

It was found that the T4 hormone variable was best modelled using a ”Broken Stick” model as due to a significant change in slope of the data at 25 weeks gestation. The model fitted was

$$T4 = \beta_0 + \beta_1(Gestation) + \beta_2(Gestation - 25)^+ + \epsilon$$

where $(\cdot)^+$ is the $\max(0,\cdot)$. Calculating the quantile curves for this model using the non-crossing quantile approach results in figure 4.2. As expected, the quantile curves never cross in this graphic and as before the reference ranges are now easier to specify. In the case of thyroid disease it is easier to identify those patients who lie in the upper and lower 2% ranges, who require treatment.

Finally, the TSH hormone variable required a box-cox transformation before modelling and the best fitting model was

$$\sqrt{TSH} = \alpha + \beta(Gestation)$$

As was the issue with all the model fittings in Chapter 3 crossing quantiles did not allow classification of some individuals to a single reference range. Applying the algorithm described in this chapter to the TSH hormone variable using the model specified resulted in figure 4.3, with no crossing quantiles evident, the algorithm has again performed to requirements.

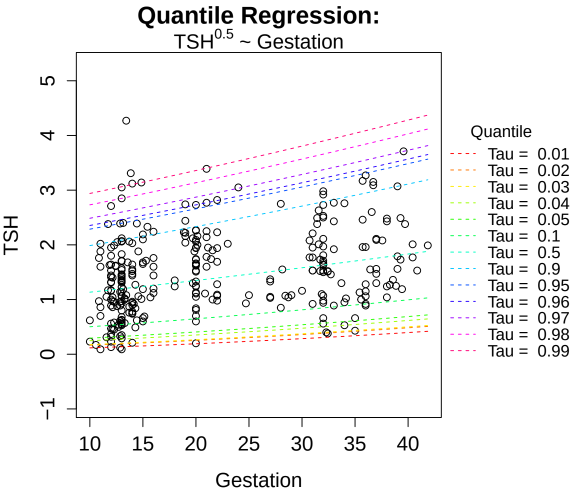

Figure 4.3: Non-crossing quantile curves for TSH

The code used to produce all of the above graphics can be found in Appendix C.2.

References

Abalovich, M., N. Amino, L. Barbour, R. Cobin, L. De Groot, and D. Glinoer (2007, August). Management of thyroid dysfunction during pregnancy and postpartum: an endocrine society clinical practice guideline. *The Journal of Clinical Endocrinology and Metabolism.*

Barrodale, I. and F. Roberts (1974). Solution of an overdetermined system of equations in the $\ell 1$ norm. *Communications of the ACM* (17), 319–320.

Benoit, D. F., R. Al-Hamzawi, K. Yu, and D. Van den Poel (2013). *bayesQR: Bayesian quantile regression.* R package version 2.1.

Box, G. and D. Cox (1964). An analysis of transformations. *Journal of the Royal Statistical Society 26*, 211–252.

Cotzias, C., S. Wong, E. Taylor, P. Seed, and J. Girling (2008). A study to establish gestation-specific reference intervals for thyroid function tests in normal singleton pregnancy. *Eur J Obstet Gynecol Reprod Biol.*, 137:61–6.

Geraci, M. (2012). *lqmm: Linear Quantile Mixed Models.* R package version 1.02.

Geraci, M. and M. Bottai (2013). Linear quantile mixed models. *Statistics and Computing.*

Koenker, R. (1994). Confidence intervals for regression quantiles. *Asymptotic Statistics*, 349–359.

Koenker, R. (2000, October). Quantile regression. *International Encyclopedia of the Social Sciences.*

Koenker, R. (2013). *quantreg: Quantile Regression.* R package version 5.05.

Koenker, R. and V. d'Orey (1987). Computing regression quantiles. *Applied Statistics* (36), 383–393.

Koenker, R. and S. Portnoy (1997). The gaussian hare and the laplacian tortoise. *Statistical Science* (12), 279–300.

Mosteller, F. and J. Tukey (1977). *Data Analysis and Regression: a second course in statistics.* Addison-Wesley.

R Core Team (2013). *R: A Language and Environment for Statistical Computing.* Vienna, Austria: R Foundation for Statistical Computing. ISBN 3-900051-07-0.

Sakia, R. (1992). The box-cox transformation technique: a review. *The Statistician 41*, 169–178.

Appendix A

R code for Engel Data Example

A.1 Quantreg Code

```r
# Quantreg Package: Simulation on Sample Data

 # log transformation of Data: TRUE/FALSE

  logTrans <- FALSE

 # Loading data and packages

 library(quantreg)
 library(ggplot2)
 data(engel)
 attach(engel)

 if(logTrans==TRUE){
    x <- log10(engel$income)
    y <- log10(engel$foodexp)
 }else{
    x <- engel$income
    y <- engel$foodexp
 }

 # Creating a data frame of quantile regression lines

 mdl.quant <- rq(y ~ x,tau=c(0.05,0.1,0.25,0.5,0.75,0.9,0.95))
 quant.results <- data.frame(t(coef(mdl.quant)))
 colnames(quant.results) <- c("Intercept", "Slope")

 # Creating a linear model to compare to quantiles (median)
```

```r
linear.quant <- lm(y ~ x)

# Creating a data set to output to a latex table

quant.results[8,1] <- linear.quant$coefficients[1]
quant.results[8,2] <- linear.quant$coefficients[2]

quant.results[,"Names"] <- c("Tau = 0.05", "Tau = 0.10",
                         "Tau = 0.25", "Median", "Tau = 0.75",
                           "Tau = 0.90", "Tau = 0.95", "Mean")

quant.results <- data.frame(quant.results[,"Names"],
                         round(quant.results[,"Intercept"],3),
                             round(quant.results[,"Slope"],3))
colnames(quant.results) <- c("Quantile", "Intercept", "Slope")

# Plotting points and the quantile regression lines

if(logTrans==TRUE){
   title <-"quantreg package: rq function (log transform)"
}else{
   title <-"quantreg package: rq function"
}

plt.quant <- qplot(x=x, y=y, xlab="Household Income",
                   ylab="Food Expenditure", main=title) +
           geom_abline(aes(intercept=Intercept, slope=Slope,
           color=Quantile),show_guide=TRUE, data=quant.results)
```

A.2 BayesQR Code

```r
# BayesQR Package: Simulation on Sample Data

  # log transformation of Data: TRUE/FALSE

  logTrans <- FALSE

  # Loading data and packages

  library(bayesQR)
  library(ggplot2)
  library(quantreg)
  data(engel)
  attach(engel)

  if(logTrans==TRUE){
    x <- log10(engel$income)
    y <- log10(engel$foodexp)
  }else{
    x <- engel$income
    y <- engel$foodexp
  }
  # Creating a data frame of quantile regression lines

  taus <- c(0.05,0.1,0.25,0.5,0.75,0.9,0.95)
  intercept <- c(rep(0,7))
  slope <- c(rep(0,7))

  mdl.bays <- bayesQR(y ~ x, quantile=taus, ndraw=5000)

  sum <- summary(mdl.bays, burnin=500)

  for(i in 1:length(sum)){
    intercept[i] <- sum[[i]]$betadraw[1,1]
    slope[i] <- sum[[i]]$betadraw[2,1]
  }

  bays.results <- data.frame(cbind(intercept,slope))
  colnames(bays.results) <- c("Intercept", "Slope")

  # Creating a linear model to compare to quantiles (median)

  linear.bays <- lm(y ~ x)
  bays.results[8,1] <- linear.bays$coefficients[1]
  bays.results[8,2] <- linear.bays$coefficients[2]

  bays.results[,"Names"] <- c("Tau = 0.05","Tau = 0.10",
                        "Tau = 0.25","Median","Tau = 0.75",
                            "Tau = 0.90","Tau = 0.95","Mean")

  bays.results <- data.frame(bays.results[,"Names"],
                        round(bays.results[,"Intercept"],4),
                            round(bays.results[,"Slope"],4))
  colnames(bays.results) <- c("Quantile", "Intercept", "Slope")
```

```
  # Plotting points and the quantile regression lines

  if(logTrans==TRUE){
    title <-"bayesQR package: bayesQR function (log transform)"
  }else{
    title <-"bayesQR package: bayesQR function"
  }

  plt.bays <- qplot(x=x, y=y, xlab="Household Income",
                     ylab="Food Expenditure",main=title) +
    geom_abline(aes(intercept=Intercept, slope=Slope,
            color=Quantile),show_guide=TRUE, data=bays.results)
```

A.3 Lqmm Code

```
# lqmm Package: Simulation on Sample Data

 # log transformation of Data: TRUE/FALSE

 logTrans <- FALSE

 # Loading data and packages

 library(lqmm)
 library(ggplot2)
 library(quantreg)
 data(engel)
 attach(engel)

 if(logTrans==TRUE){
    x <- log10(engel$income)
    y <- log10(engel$foodexp)
 }else{
    x <- engel$income
    y <- engel$foodexp
 }
 # Creating a data frame of quantile regression lines

 mdl.mix <- lqm(y ~ x,iota=c(0.05,0.1,0.25,0.5,0.75,0.9,0.95))
 mix.results <- data.frame(t(coef(mdl.mix)),row.names=NULL)
 colnames(mix.results) <- c("Intercept", "Slope")
```

```

# Creating a linear model to compare to quantiles (median)

linear.mix <- lm(y ~ x)
mix.results[8,1] <- linear.mix$coefficients[1]
mix.results[8,2] <- linear.mix$coefficients[2]

mix.results[,"Names"] <- c("Tau = 0.05", "Tau = 0.10",
                        "Tau = 0.25", "Median", "Tau = 0.75",
                         "Tau = 0.90", "Tau = 0.95", "Mean")

mix.results <- data.frame(mix.results[,"Names"],
                       round(mix.results[,"Intercept"],4),
                        round(mix.results[,"Slope"],4))
colnames(mix.results) <- c("Quantile", "Intercept", "Slope")

# Plotting points and the quantile regression lines

if(logTrans==TRUE){
   title <- "lqmm package: lqm function (log transform)"
}else{
   title <- "lqmm package: lqm function"
}

plt.mix <- qplot(x=x, y=y, xlab="Household Income",
               ylab="Food Expenditure", main=title) +
           geom_abline(aes(intercept=Intercept, slope=Slope,
           color=Quantile), show_guide=TRUE, data=mix.results)
```

Appendix B

R Code for Chapter 3

B.1 Distributions

```
##### Distribution Analysis: Graph + Normality Test #####
#                                                       #
# Author: Kevin Brosnan                                 #
# Date: 24/02/2014                                      #
#                                                       #
# Description:                                          #
#  1) Histogram                                         #
#  2) Boxplot                                           #
#  3) QQ-plots                                          #
#  4) Shapiro-Wilk Test                                 #
#                                                       #
# # # # # # # # # # # # # # # # # # # # # # # # # # # # #

distribution <- function(data){

  name <- deparse(substitute(data))
  split.screen(figs=c(2,1))
  split.screen(figs=c(1,2), screen=1)

  screen(3)
    # Box-Plot of Data
    titleBox <- paste("Boxplot of ", name)
    boxplot(data, main=titleBox)
  screen(4)
    # QQ-plot of data
    qqnorm(data)
    qqline(data, col="red")
  screen(2)
```

```
    # Histogram of Data
    titleHist <- paste("Histogram of ", name)
    hist(data, xlab=name, main=titleHist)

  close.screen(all.screens=TRUE)

  normtest <- shapiro.test(data)
  return(normtest)
}

distribution(T4)
distribution(T3)
distribution(TSH)
```

B.2 Box-Cox

```
#####   Box-Cox Power Transformation Calculation    #####
#                                                       #
# Author: Kevin Brosnan                                 #
# Date: 19/02/2014                                      #
#                                                       #
# Description:                                          #
#  1) Produce a plot of lambda v's log-likelihood       #
#  2) Return the exact value of lambda to the user      #
#  3) Return suggested values of lambda to user         #
#                                                       #
# # # # # # # # # # # # # # # # # # # # # # # # # # # #

box.cox <- function(formula, plotit=TRUE){

  # Checking the class of the formula argument passed
  # to the function

    if(class(formula)=="lm"){
      formula <- update(object, y=TRUE, qr=TRUE)
    }else if(class(formula)=="formula"){
      formula <- lm(formula, y=TRUE, qr=TRUE)
    }else{
      stop("Formula must be of class formula or of class lm")
    }

    y <- formula$y
    xqr <- formula$qr

  # Make sure response variable y is positive

    if(any(y<=0)){
      stop("Response variable must be positive")
    }

  # Box-Cox Algorithm

    lambda <- seq(-2,2,length=length(y))
    eps <- 0.02

    n <- length(y)
    y <- y/exp(mean(log(y)))
    logy <- log(y)

    xl <- loglik <- as.vector(lambda)
    m <- length(xl)

    for(i in 1:m){
      if(abs(la <- xl[i]) > eps){
        yt <- (y^la - 1)/la
      }else{
        yt <- logy*(1 + (la*logy)/2*(1 + (la*logy)/3*
                                   (1 + (la*logy)/4)))
      }
      loglik[i] <- -n/2 * log(sum(qr.resid(xqr, yt)^2))
    }

  # Calculating maximum values of lambda and log-likelihood

    max_index <- which(loglik==max(loglik))
    Loglik_max <- loglik[max_index]
    lambda_max <- xl[max_index]

  # Plotting log-likelihood vs Lambda

    if(plotit){

      lim <- Loglik_max - qchisq(19/20, 1)/2

      ind <- range((1:m)[loglik>lim])
      xlim <- c(0,0)

      if(loglik[1] < lim){
        i <- ind[1]
        xlim[1] <- xl[i-1] + ((lim - loglik[i-1])*
           (xl[i] - xl[i-1]))/(loglik[i] - loglik[i-1])
      }

      if(loglik[m] < lim){
        i <- ind[2]
        xlim[2] <- xl[i-1] + ((lim - loglik[i-1])*
           (xl[i] - xl[i-1]))/(loglik[i] - loglik[i-1])
      }

      xlim <- sort(xlim)

      xlower <- which(xl >= xlim[1]-0.01)
      xupper <- which(xl >= xlim[2]+0.01)

      xlower <- xlower[1]
      xupper <- xupper[1]

      dev.hold()
      on.exit(dev.flush())

      plot(x=xl[xlower:xupper], y=loglik[xlower:xupper],
               xlab=expression(lambda),ylab="log-Likelihood",
                type="l",  xlim=c(xlim[1]-0.01,xlim[2]+0.01))

      title(main="Box-Cox Transformation")

      plims <- par("usr")
      scal <- (1/4 * (plims[4]-plims[3]))/par("pin")[2]

      abline(h=lim, lty=3)
      text <- bquote(lambda*"="*.(round(lambda_max,2)))
      text(lambda_max, Loglik_max - scal, text, col="red")

      y0 <- plims[3]

      if(max_index>1 && max_index<m){
        segments(lambda_max, y0, lambda_max, Loglik_max,lty=3)
      }
```

```
      segments(xlim[1], y0, xlim[1], lim, lty=3)
      segments(xlim[2], y0, xlim[2], lim, lty=3)
    }

  # Outputting best suggestion for lambda

    lambda_suggestion <- FALSE

    if(abs(lambda_max)>0.4 && abs(lambda_max)<0.6){
      lambda_suggestion <- TRUE
      if(lambda_max<0){
        lambda_sug <- "1/sqrt(y)"
      }else{
        lambda_sug <- "sqrt(y)"
      }
    }else if(abs(lambda_max)>0.9 && abs(lambda_max)<1.1){
      lambda_suggestion <- TRUE
      if(lambda_max<0){
        lambda_sug <- "1/y"
      }else{
        lambda_sug <- "y"
      }
    }else if(abs(lambda_max)<0.1){
      lambda_suggestion <- TRUE
      lambda_sug <- "ln(y)"
    }

    if(lambda_suggestion){
      out <- sprintf("The optimal value of lambda to use in
           transforming the supplied data is lambda = %.2f.
           However a %s transformation would be easier to
           interpret and apply", lambda_max, lambda_sug)
    } else{
      out <- sprintf("The optimal value of lambda to use in
           transforming the supplied data is lambda = %.2f",
                                              lambda_max)
    }

  # Returning final values
    print(out)
    invisible(list(x=xl, y=loglik, lambda=lambda_max))
}
```

```
box.cox(TSH, plotit=TRUE)
```

B.3 Linear Quantile Models

free T4

```
#####               T4 Model Fitting                #####
#                                                       #
# Author: Kevin Brosnan                                 #
# Date: 21/02/2014                                      #
#                                                       #
# Description:                                          #
#  Modelling T4 v Gestation initially with a linear     #
#  model and then including a "Broken Stick" element    #
#                                                       #
# # # # # # # # # # # # # # # # # # # # # # # # # # # # #

  taus <- c(0.01,0.02,0.03,0.04,0.05,0.1,
                0.5,0.9,0.95,0.96,0.97,0.98,0.99)

  # T4 Model 1

  modT4_1 <- rq(T4 ~ Gestation, data=thyroid, tau=taus)
  plot.QR(modT4_1,x=Gestation, y=T4)

  # T4 Model 2
  temp <- Gestation
  for(i in 1:length(Gestation)){
    if(Gestation[i]>=25){
      temp[i] <- Gestation[i] - 25
    }else{
      temp[i] <- 0
    }
  }

  modT4_2 <- rq(T4 ~ Gestation + temp,
                        data=thyroid, tau=taus)
  plot(y=T4, x=Gestation)

  col <- rainbow(length(taus))
  for(i in 1:length(taus)){
    curve(from=10,to=25,modT4_2$coef[1,i]+x*modT4_2$coef[2,i],
                                add=TRUE,lty=2, col=col[i])
    curve(from=25, to=40,modT4_2$coef[1,i]+x*modT4_2$coef[2,i]
            +(x-25)*modT4_2$coef[3], lty=2,col=col[i],add=TRUE)
  }
  title(main="Quantile Regression:", cex=1.5)
  mtext("T4 ~ Gestation + BrokenStick(25)",cex=1)
  labels <- rep("", length(modT4_2$tau))

  for(i in 1:length(modT4_2$tau)){
    labels[i] <- paste("Tau = ", modT4_2$tau[i])
  }

  par(xpd=TRUE)
  legend((max(Gestation)+1),max(T4, na.rm=TRUE),
    labels, cex=0.8, col=col, lty=2, title="Quantile",bty="n")
```

free T3

```
#####              T3 Model Fitting              #####
#                                                    #
# Author: Kevin Brosnan                              #
# Date: 21/02/2014                                   #
#                                                    #
# Description:                                       #
#  Modelling T3 v Gestation initially with a linear   #
#  model and then a quadratic model                  #
#                                                    #
# # # # # # # # # # # # # # # # # # # # # # # # # # # #

taus <- c(0.01,0.02,0.03,0.04,0.05,0.1,
                        0.5,0.9,0.95,0.96,0.97,0.98,0.99)

  # T3 Model 1

  modT3_1 <- rq(T3 ~ Gestation, data=thyroid, tau=taus)
  plot.QR(modT3_1,x=Gestation, y=T3)

  # T3 Model 2

  modT3_2 <- rq(T3 ~ Gestation + I(Gestation^2),
                        data=thyroid, tau=taus)
```

```
plot(x=Gestation, y=T3)

col <- rainbow(length(taus))
for(i in 1:length(taus)){
  curve(modT3_2$coef[1,i]+x*modT3_2$coef[2,i]+(x^2)
          *modT3_2$coef[3,i],add=TRUE,lty=2, col=col[i])
}

title(main="Quantile Regression:", cex=1.5)
subtitle <- bquote(T3*" ~ "*Gestation + Gestation^2)
mtext(subtitle,cex=1)

labels <- rep("", length(modT4_2$tau))

for(i in 1:length(modT4_2$tau)){
  labels[i] <- paste("Tau = ", modT4_2$tau[i])
}

par(xpd=TRUE)
legend((max(Gestation)+1),max(T3, na.rm=TRUE),labels,
           cex=0.8, col=col, lty=2, title="Quantile",bty="n")
```

TSH

```
#####              TSH Model Fitting                   #####
#                                                          #
# Author: Kevin Brosnan                                    #
# Date: 21/02/2014                                         #
#                                                          #
# Description:                                             #
#  Modelling TSH v Gestation                               #
#                                                          #
# # # # # # # # # # # # # # # # # # # # # # # # # # # # # #

  taus <- c(0.01,0.02,0.03,0.04,0.05,0.1,
              0.5,0.9,0.95,0.96,0.97,0.98,0.99)


  # TSH Model

  modTSH <- rq(TSH^0.5 ~ Gestation,
                          data=thyroid, tau=taus)
  plot.QR(modTSH,x=Gestation, y=TSH, lambda=0.5)
```

Appendix C

Non-Crossing Quantiles

C.1 QR Function Code

```
#####  Quantile Regression Function - Non-Crossing  #####
#                                                       #
# Author: Kevin Brosnan                                 #
# Adapted from: quantreg package by Roger Koenker       #
#                                                       #
# Date: 09/03/2014                                      #
#                                                       #
# Description:                                          #
#  1) Reading model arguments                           #
#  2) Required constants, functions and packages        #
#  3) Checking validity of inputted Tau values          #
#  4) Defining required data objects                    #
#  5) Calculating the regression quantiles              #
#  6) Tidying up the output                             #
#  7) Returning the required list                       #
#                                                       #
# # # # # # # # # # # # # # # # # # # # # # # # # # # # #


QR <- function(formula, tau=0.5, data, ...){

#### 1) Reading in model arguments to a data frame ####

  # Accessing data if not supplied
    if(missing(data)){
      data <- environment(formula)
    }
```

```
 # Returns the call with all specified arguments named in full
  call <- match.call()

 # Returns the call with all specified arguments named in full
 # excluding the additional arguments passed to the function

  model_frame <- match.call(expand.dots=FALSE)

  m <- match(c("formula", "data"), names(model_frame), 0L)
  model_frame <- model_frame[c(1,m)]
  model_frame$drop.unused.levels <- TRUE
  model_frame[[1L]] <- as.name("model.frame")
  model_frame <- eval.parent(model_frame)

 # Defining x and y data which will be used in modelling

  model_terms <- attr(model_frame, "terms")
  Y <- model.response(model_frame)

  if(!is.empty.model(model_terms)){
   X <- model.matrix(model_terms, model_frame)
  }else{
   stop("Error in the Design Matrix")
  }

  B <- X
  p <- ncol(B)
  Ident <- diag(p)

#### 2) Required constants and functions ####

 # Tolerance level required
  eps <- .Machine$double.eps^(2/3)

 # Check Function required in for calculation of QR
  Rho <- function(u, tau){
   u * (tau - (u<0))
  }

 # Loading quantreg package
  if(!require("quantreg")){
   print("Trying to install quantreg")
   install.packages("quantreg",quiet=TRUE)
```

```
      if(!require("quantreg")){
        stop("Could not install quantreg")
      }
    }

#### 3) Checking validity of inputted Tau values ####

  # Checking value of Tau's that have been input

    if(length(tau)>0){

      # Make sure tau is in the range 0<tau<1
      if(any(tau<0) || any(tau>1)){
        stop("Invalid values of tau input: 0<tau<1")
      }

      # Make sure tau value is not equal to 0
      if(any(tau==0)){
        tau[tau==0] <- eps
      }

      # Make sure tau value is not equal to 1
      if(any(tau==1)){
        tau[tau==1] <- 1 - eps
      }

      # Only keep unique values of tau
      tau <- unique(tau)
      tau <- sort(tau)

    }else{

      # Make sure tau is in the range 0<tau<1
      if(tau<0 || tau>1){
        # Temporarily function will not accept a negative value
        # Future will hopefully calculate full quantile range
        stop("Invalid values of tau input: 0<tau<1")
      }

      # Make sure tau is not equal to 0
      if(tau==0){
        tau <- eps
      }
      # Make sure tau is not equal to 1
      if(tau==1){
        tau <- 1-eps
      }
    }

#### 4) Defining required data objects ####

  # Coefficient Matrix to hold model coefficients
    coef <- matrix(0, ncol(B), length(tau))

  # Vector of rho values
    rho <- rep(0, length(tau))

  # Fitted and residual values
    fitted <- matrix(0, nrow(B), length(tau))
    resid <- matrix(0, nrow(B), length(tau))

#### 5) Calculating regression quantiles ####

  # If length(tau)=1 calculate quantile and return
    if(length(tau)==1){
      z <- rq.fit(x=X, y=Y, tau=tau)

      fit <- z
      fit$coefficients <- z$coefficients
      fit$residuals <- z$residuals
      fit$fitted.values <- z$fitted.values
      fit$formula <- formula
      fit$terms <- model_terms
      fit$xlevels <- .getXlevels(model_terms, model_frame)
      fit$call <- call
      fit$tau <- tau
      fit$residuals <- drop(fit$residuals)
      fit$rho <- rho
      fit$fitted.values <- drop(fit$fitted.values)
      fit$model <- model_frame
      fit$data <- data
      return(fit)
    }

  # Splitting tau values required into two groups
    start.tau <- 0.5
    pos.taus <- tau[(tau-start.tau)>0]
    n.pos.taus <- length(pos.taus)
    neg.taus <- tau[(tau-start.tau)<0]
    n.neg.taus <- length(neg.taus)

  # Calculate tau for 0.5
    z.start <- rq.fit(x=B, y=Y, tau=start.tau)

    if(any(tau==0.5)){
      k <- which(tau==0.5)
      coef[,k] <- z.start$coefficients
      resid[,k] <- z.start$residuals
      rho[k] <- sum(Rho(z.start$residuals, tau=start.tau))
      fitted[,k] <- Y - z.start$residuals
    }

  # Calculate remaining quantiles

    if(n.pos.taus>0){

      b.start <- z.start$coef
      RR <- Ident
      rr <- b.start + eps

      for(i in 1:n.pos.taus){
        k <- which(tau==pos.taus[i])
        z <- rq.fit(x=B, y=Y, tau=pos.taus[i],
                            method="fnc", R=RR, r=rr)

        coef[,k] <- z$coefficients
        resid[,k] <- z$residuals
        rho[k] <- sum(Rho(z$residuals, tau[i]))
        fitted[,k] <- Y - z$residuals

        rr <- coef[,k] + eps
      }
    }

    if(n.neg.taus>0){

      b.start <- z.start$coef
      neg.taus <- sort(neg.taus,TRUE)
```

```
      RR <- -Ident
      rr <- -b.start + eps

      for(i in 1:n.neg.taus){
        k <- which(tau==neg.taus[i])
        z <- rq.fit(x=B, y=Y, tau=neg.taus[i],
                            method="fnc", R=RR, r=rr)

        coef[,k] <- z$coefficients
        resid[,k] <- z$residuals
        rho[k] <- sum(Rho(z$residuals, tau[i]))
        fitted[,k] <- Y - z$residuals

        rr <- -coef[,k] + eps
      }
    }

#### 6) Tidying up the output ####

  taulabs <- paste("Tau=", format(round(tau, 3)))
  dimnames(coef) <- list(dimnames(B)[[2]], taulabs)
  dimnames(resid) <- list(dimnames(B)[[1]], taulabs)

  fit <- z
  fit$coefficients <- coef
  fit$residuals <- resid
  fit$fitted.values <- fitted
  fit$formula <- formula
  fit$terms <- model_terms
  fit$xlevels <- .getXlevels(model_terms, model_frame)
  fit$call <- call
  fit$tau <- tau
  fit$residuals <- drop(fit$residuals)
  fit$rho <- rho
  fit$fitted.values <- drop(fit$fitted.values)
  fit$model <- model_frame
  fit$data <- data

#### 7) Returning the required list ####
  fit
}
```

C.2 Quantile Models

T3

```
#####        T3 Model Fitting - Non-Crossing        #####
#                                                       #
# Author: Kevin Brosnan                                 #
# Date: 12/03/2014                                      #
#                                                       #
# Description:                                          #
#  Modelling T3 v Gestation with a quadratic model with #
#  non-crossing constraints applied                     #
#                                                       #
# # # # # # # # # # # # # # # # # # # # # # # # # # # # #


  taus <- c(0.01,0.02,0.03,0.04,0.05,0.1,
            0.5,0.9,0.95,0.96,0.97,0.98,0.99)

  model <- QR(T3 ~ Gestation + I(Gestation^2), data=thyroid,
                                                    tau=taus)

  # Plotting the regression quantile curves

    par(pty="s")
    plot(x=Gestation, y=T3, ylim=c(2.9,6.5))

    col <- rainbow(length(taus))
    for(i in 1:length(taus)){
      curve(model$coef[1,i]+x*model$coef[2,i]
                            +(x^2)*model$coef[3,i],
                    add=TRUE, lty=2, col=col[i])
    }

    title(main="Quantile Regression:", cex=1.5)
    subtitle <- bquote(T3*" ~ "*Gestation + Gestation^2)
    mtext(subtitle,cex=1)

    labels <- rep("", length(model$tau))

    for(i in 1:length(model$tau)){
      labels[i] <- paste("Tau = ", model$tau[i])
    }

    par(xpd=TRUE)
    legend((max(Gestation)+1),max(T3, na.rm=TRUE),labels,
             cex=0.8, col=col, lty=2, title="Quantile",bty="n")
```

T4

```
#####        T4 Model Fitting - Non-Crossing        #####
#                                                       #
# Author: Kevin Brosnan                                 #
# Date: 12/03/2014                                      #
#                                                       #
# Description:                                          #
#  Modelling T4 v Gestation using a broken stick model  #
#                                                       #
# # # # # # # # # # # # # # # # # # # # # # # # # # # # #

  taus <- c(0.01,0.02,0.03,0.04,0.05,0.1,
              0.5,0.9,0.95,0.96,0.97,0.98,0.99)

  # Setting up the Broken Stick element of the model

    temp <- Gestation
    for(i in 1:length(Gestation)){
      if(Gestation[i]>=25){
        temp[i] <- Gestation[i] - 25
      }else{
        temp[i] <- 0
      }
    }

  # Modelling T4 ~ Gestation + max(0, Gestation-25)

    model <- QR(T4 ~ Gestation + temp, data=thyroid, tau=taus)

  # Plotting Quantile Regression lines

    par(pty="s")
    plot(y=T4, x=Gestation)

    col <- rainbow(length(taus))
    for(i in 1:length(taus)){
```

```
    curve(from=10,to=25,model$coef[1,i]+x*model$coef[2,i],
                         add=TRUE, lty=2, col=col[i])
    curve(from=25, to=40,model$coef[1,i]+x*model$coef[2,i]+
                (x-25)*model$coef[3], lty=2,col=col[i],add=TRUE)
  }

  title(main="Quantile Regression:", cex=1.5)
  mtext("T4 ~ Gestation + BrokenStick(25)",cex=1)
  labels <- rep("", length(model$tau))

  for(i in 1:length(model$tau)){
    labels[i] <- paste("Tau = ", model$tau[i])
  }

  par(xpd=TRUE)
  legend((max(Gestation)+1),max(T4, na.rm=TRUE), labels,
         cex=0.8, col=col, lty=2, title="Quantile",bty="n")
```

TSH

```
#####        TSH Model Fitting - Non-Crossing        #####
#                                                        #
# Author: Kevin Brosnan                                  #
# Date: 12/03/2014                                       #
#                                                        #
# Description:                                           #
#  Modelling TSH v Gestation - sqrt(TSH)~Gestation       #
#                                                        #
# # # # # # # # # # # # # # # # # # # # # # # # # # # # #

 taus <- c(0.01,0.02,0.03,0.04,0.05,0.1,
           0.5,0.9,0.95,0.96,0.97,0.98,0.99)

 # TSH Model

  model <- QR(TSH^0.5 ~ Gestation, data=thyroid, tau=taus)

 # Plotting regression quantile lines

  plot.QR(model, x=Gestation, y=TSH, lambda=0.5)
```