December 2020

Linear Quantile Mixed Modeling: A Study of the 'lqmm' Package in R

Thomas Charles DeMarco
Clemson University, tomcdemarco@yahoo.com

Linear Quantile Mixed Modeling: A Study of the 'lqmm' Package in R

A Thesis
Presented to
the Graduate School of
Clemson University

In Partial Fulfillment
of the Requirements for the Degree
Master of Science
Mathematical Sciences

---

by
Thomas DeMarco
December 2020

Accepted by:
Dr. Colin Gallagher, Committee Co-Chair
Dr. Patrick Gerard, Committee Co-Chair
Dr. Deborah Kunkel

Abstract

Linear quantile mixed modeling is a diverse statistical tool that can replace traditional least squares modeling for analyzing data whose sampling method has some form of clustering and whose response has trends that differ for each quantile level. In this article, we will evaluate the effectiveness of this modeling method through the use of the `lqmm` package in R [2]. Simulations and an applied data analysis will be performed to evaluate the performance of the `lqmm()` function on different types of datasets. We will also introduce background on quantile based mixed modeling and give descriptions of the output and main commands of the `lqmm()` function.

Dedication

This work is dedicated to my late mother, Teri DeMarco, whose love, support, and faith has always and continues to propel me to work hard for the better of myself and others, and to my wife, Kelly DeMarco, for her love, patience, and care through the completion of this work.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

In this article, the effectiveness of linear quantile mixed modeling on analyzing clustered data through the use of the `lqmm` package in R will be assessed [2]. In order to better understand the advantages and limitations of quantile based methods for mixed modeling, we will compare results from the `lqmm` package to those from more common analyses that are based on the assumption of normally distributed measurements. We will use simulation to compare the results of `lqmm` fits to those from more standard model fitting, and we will use `lqmm` to analyze lumber strength data. Two simulation studies and a data analysis will help elucidate the potential benefits and drawbacks of `lqmm`.

In industry, output from machines and factories that are made to perform identical tasks can vary from unit to unit for different reasons (e.g. assembly error, geographic location, level of wear and tear). These variations can cause clustering in the data, where the data from each unit constitutes its own cluster. The offsetting and often non-normality of these clusters from one another makes it difficult for companies to set targets and tolerances for its products. They often choose to either look at each unit individually or combine all into one data set and then use least squares or other traditional methods that assume normally distributed measurements to create the targets and tolerances. The `lqmm()` function allows us to create percentile tolerances and targets on both the cluster level and overall level, and it allows for the measurements to follow non-normal distributions. We will discuss the modeling background and the commands in the `lqmm` package in the next sections before going through the results of our simulations and data analysis in the following chapters.

1.1 Modeling Background

In a designed experiment, a random variable Y_{ik} that represents the response for the k^{th} experimental unit to treatment i is commonly modeled using the effects model

$$Y_{ik} = \mu + \alpha_i + \epsilon_{ik}, \tag{1.1}$$

where

$$\mu = \text{the overall mean},$$
$$\alpha_i = \text{the effect of treatment } i, \text{ and}$$
$$\epsilon_{ik} = \text{the random error associated with } Y_{ik}.$$

In order to analyze the significance of the treatment effect on the response variable (i.e. the significance of α_i on Y_{ik}), assumptions on the error term must be made. The model assumptions typically are:

1. ϵ_{ik} are Normally Distributed
2. ϵ_{ik} have constant variance
3. ϵ_{ik} are independent,

which can all be summarized into $\epsilon_{ik} \overset{\text{iid}}{\sim} N(0, \sigma)$, where σ is constant.

1.1.1 Mixed Modeling

There are many cases where all ϵ_{ik} fail to be completely independent, breaking our third assumption. Many examples of this failure of independence are found in longitudinal data where individuals are measured multiple times over a certain time period. The unobserved variables that define each individual effect each of their responses and cause a dependence between responses within each individual. To account for these unobserved variables within each individual a random effect may be added to the effects model.

The addition of a random effect moves the standard effects model to the mixed model. A random effect accounts for unobserved heterogeneity between groups or "clusters" within the data.

These clusters are what break the independence of all Y_{ik}. Within each cluster there is a level of dependence, and adding a random effect accounts for this dependence. While the significance of the fixed effect on the population of interest is of primary concern, random effects are put in place to account for observed clusters in our sample data that may effect our response. An example model is

$$Y_{ijk} = \mu + \alpha_i + \gamma_j + \epsilon_{ijk}, \tag{1.2}$$

where

$$\begin{aligned} \mu &= \text{the overall mean,} \\ \alpha_i &= \text{the fixed effect of treatment } i, \\ \gamma_j &= \text{the random effect of individual } j, \text{ and} \\ \epsilon_{ijk} &= \text{the random error associated with } Y_{ijk}. \end{aligned}$$

The model assumptions are now

$$\gamma_j \overset{\text{iid}}{\sim} N(0, \sigma_\gamma) \quad \text{and} \quad \epsilon_{ijk} \overset{\text{iid}}{\sim} N(0, \sigma),$$

where σ_γ^2 and σ^2 are constants and are often called variance components since the variance of Y_{ijk} is now broken up into the within cluster variation (σ_γ^2) plus the between cluster variation (σ^2). The variance of Y_{ijk} is thus

$$Var(Y_{ijk}) = \sigma_\gamma^2 + \sigma^2. \tag{1.3}$$

The level of dependence within each cluster in the sample data can be calculated using interclass correlation (ICC). From our model in equation (1.2), the ICC between two responses Y_{ijk} and $Y_{ijk'}$, where k' represents a different value for k than the first response, is calculated by taking the proportion of variation explained by the within cluster variation. A higher interclass correlation means a higher dependence between responses in the cluster. The ICC is given by

$$\text{ICC}(Y_{ijk}, Y_{ijk'}) = \frac{\sigma_\gamma^2}{\sigma_\gamma^2 + \sigma^2}. \tag{1.4}$$

Just like one adds a fixed intercept and a fixed slope coefficient to an explanatory variable

X in linear modeling, one can add a random intercept and a random slope coefficient to the model for linear mixed modeling. The reasoning for including a random slopes and intercepts in a model is similar to the reasoning for including a random effect in equation (1.2). If it is observed that clusters of observations in the data have different trends or starting points with respect to the relationship between X and Y, then a random slope or a random intercept may be added to the model to account for the heterogeneity. An example model is

$$Y_{ijk} = \mu + \alpha_i + \gamma_j + (\beta_i + \zeta_j)X_{ij} + \epsilon_{ijk}, \tag{1.5}$$

where

$$\begin{aligned}
\mu &= \text{the overall mean,} \\
\alpha_i &= \text{the fixed effect of treatment } i, \\
\gamma_j &= \text{the random effect of individual } j, \\
\beta_i &= \text{the fixed slope of treatment } i, \\
\zeta_j &= \text{the random slope of individual } j, \\
\epsilon_{ijk} &= \text{the random error associated with } Y_{ijk}, \\
\gamma_j &\overset{\text{iid}}{\sim} N(0, \sigma_\gamma), \quad \zeta_j \overset{\text{iid}}{\sim} N(0, \sigma_\zeta), \quad \text{and} \quad \epsilon_{ijk} \overset{\text{iid}}{\sim} N(0, \sigma).
\end{aligned}$$

1.1.2 Quantile Regression

Another model assumption that is often failed to be met is the constant variance of ϵ_{ijk}. This is commonly corrected by transforming the response Y_{ijk}, but there are situations where changing the estimated parameter is more appropriate. In instances where different levels of risk factors (e.g. age, gender, socioeconomic status) in individuals affect their outcomes to response variables (e.g. blood pressure, body mass index, size of a tumor) to a different extent or even in opposite ways, quantile regression is more often appropriate for modeling the changes in variation between the different risk factor levels of the data [1]. Instead of modeling only the mean of the response variable's marginal distribution, quantile regression models the quantiles of the response variable's conditional distribution. For data sets like the ones described above, quantile regression gives a larger picture of the distribution instead of just the measures of center that do not match the variation of

the entire distribution.

To further explain quantile regression, note that we can generalize any model with an explanatory variable denoted X and response denoted Y, where $g(x)$ is some function of x and ϵ is our error in estimating Y with $g(x)$ as follows

$$Y = g(x) + \epsilon. \tag{1.6}$$

In standard least squares regression, we have that $g(x) = E(Y|X = x)$ which results in $E(\epsilon|X = x) = 0$. In quantile regression, $g(x)$ represents the τ quantile of the conditional distribution of Y given $X = x$, so that ϵ has conditional τ quantile equal to 0. In quantile regression, $g(x)$ is found by solving

$$\min_{g(x)} \sum_{i=1}^{n} \rho_\tau \left(y_i - g(x_i) \right), \tag{1.7}$$

where $\rho_\tau(v) = \tau \max(v, 0) + (1 - \tau) \max(-v, 0)$ is the asymmetrically weighted L_1 loss function [3].

As estimation of $g(x)$ changes from standard regression to quantile regression, so does the assumption on ϵ. The assumption of epsilon goes from $\epsilon \sim N(0, \sigma)$ to $\epsilon^{(\tau)} \sim AL(0, \sigma, \tau)$, where the density of the asymmetric Laplace (AL) for ϵ is

$$p(\epsilon|0, \sigma, \tau) = \frac{\tau(1 - \tau)}{\sigma} \exp\left\{ -\frac{1}{\sigma} \rho_\tau(\epsilon - 0) \right\},$$

where ρ_τ is the loss function from equation (1.7). The scale parameter σ can be found by solving

$$\frac{1}{n} \sum_{i=1}^{n} \rho_\tau \left(y_i - g(x_i) \right),$$

which is also the Maximum Likelihood Estimator (MLE) of σ of an AL distribution [4].

1.2 'lqmm' Package

The `lqmm()` function in R fits linear quantile mixed models based on the AL distribution, and we will be using this function to evaluate this modeling method [2].

1.2.1 Inputs

After loading the package

```
R> library("lqmm"),
```

documentation of the package can be found using

```
R> help("lqmm"),
```

and the function arguments can be listed by calling

```
R> args(lqmm),
```

which results in

```
function (fixed, random, group,covariance = "pdDiag", tau = 0.5, nK = 7, type = "normal",
    rule = 1, data = sys.frame(sys.parent()), subset, weights, na.action = na.fail,
    control = list(), contrasts = NULL, fit = TRUE).
```

We will go over the arguments that will be used in our research. All other arguments are left to their default setting. The `fixed` and `random` arguments are formula objects that define, respectively, the fixed and the random parts of the linear predictor $g(x)$, while the clustering or grouping variable is defined in the argument `group`. Each of these three arguments can be data frames themselves or variable names that can be pulled from a data frame inserted into the optional `data` argument. The argument `tau` is the quantile level of interest τ. The default for `tau` is the median ($\tau = 0.5$) but the argument can be any percentile between 0 and 1 and can also be a vector of multiple different percentiles. Note that if `tau` is a vector of multiple percentiles, then a separate model is fit for each level of τ specified. The last argument that we will use is `control`, which specifies the optimization method for finding the parameters of interest. The default is gradient-search optimization, but we will be using derivative-free optimization. More information on the optimization methods and the arguments not previously specified can be found in the article "Linear quantile mixed models: The lqmm package for laplace quantile regression." by M. Geraci [1].

1.2.2 Outputs

The default output to an `lqmm` object are the coefficients for the fixed effects, the covariance matrix for the random effects, the residual scale parameter (i.e. our σ for ϵ in equation (1.2)), the

log-likelihood for the residual scale parameter with respect to the AL distribution, the number of observations in the data, and the number of groups (i.e. clusters) in the data. While this output has its uses in different applications, what we are most interested in is the output that comes from using the `predict()` function on an `lqmm` object.

The `predict()` function uses the best linear predictor (BLP) of the random effects in the `lqmm` model and the known coefficients of the fixed effects in the `lqmm` model to calculate a model prediction for each of our response variable values. The argument `level` specifies whether the predictions are to be made over the marginal distribution (`level` = 0) or over each of the conditional distributions individually (`level` = 1). With regards to the model, level 0 sets the random effects equal to 0 so that the predictions are done across all clusters, while level 1 evaluates the BLP of the random effects so that predictions are different from cluster to cluster [1]. We will use the `predict()` function to compare the lqmm predictions to the true quantiles in our simulation of equation (2.1).

Chapter 2

Simulation of ‘lqmm‘ Effectiveness

In industry, output from machines and factories that are made to perform identical tasks can vary from unit to unit for different reasons (e.g. assembly error, geographic location, level of wear and tear). These variations can cause clustering in the data, where the data from each unit constitutes its own cluster. The offsetting and often non-normality of these clusters from one another makes it difficult for companies to set targets and tolerances for its products. They often choose to either look at each unit individually or combine all into one data set and then use least squares or other traditional methods that assume normally distributed measurements to create the targets and tolerances. Quantile based mixed modeling allows us to create percentile tolerances and targets on both the marginal distribution and the conditional distribution for each unit, `level` = 0 and 1 respectively for the `'predict()'` function, and it allows for the measurements to follow non-normal distributions.

In order to evaluate the effectiveness of the `'lqmm()'` function and quantile based mixed modeling, we will use the function to analyze data through two separate simulations. We start by simulating clustered data and then we will simulate longitudinal, clustered data whose slope depends on the quantile level.

2.1 Simulation 1

We will start by simulating clustered data where the data's i clusters each have mean $\mu+\gamma_i$ with j data points per cluster. After simulating the data, the `'lqmm()'` estimation of the true

parameter quantiles is compared to that of the data itself (i.e. the percentiles of the data). The lqmm function can estimate quantiles over both the marginal distribution and the conditional distributions for each cluster. The lqmm and data quantiles are compared using their MSE values that are calculated using the known, true quantiles for both the marginal and conditional distributions. The model considered is

$$Y_{ij} = \mu + \gamma_i + \epsilon_{ij}, \tag{2.1}$$

where

$$\begin{aligned} \mu &= \text{the overall mean,} \\ \gamma_i &= \text{random intercept for the } i^{th} \text{ cluster,} \\ \epsilon_{ij} &= \text{random error associated with } Y_{ij}, \\ \gamma_i &\overset{\text{iid}}{\sim} N(0, \sigma_\gamma) \quad \text{and} \quad \epsilon_{ij} \overset{\text{iid}}{\sim} N(0, \sigma). \end{aligned}$$

2.1.1 Simulation Run Down

Using R, we simulated 1,000 datasets from equation (2.1) with $i = 100$ clusters that each have $j = 30$ datapoints per cluster. We arbitrarily set $\mu = 10$ and carefully chose values for σ and σ_γ so that we have specific levels of ICC in our datasets. Recall the equation for the interclass correlation coefficient in equation (1.4) for such a model. In order to analyze the effectiveness of `'lqmm()'` across different levels of dependence within each cluster, we will run the simulation 5 times, each with a different ICC level. The different ICC levels will be controlled by setting σ and σ_γ to the values in table (2.1) for each simulation.

σ	σ_γ	ICC
1	10	0.99
1	$\sqrt{3}$	0.75
1	1	0.5
$\sqrt{3}$	1	0.25
10	1	0.01

Table 2.1: Different corresponding ICC levels for values of σ and σ_γ in equation 2.1

For each simulation, we run `'lqmm()'` using the command

```
lqmm(fixed = y ~ 1, random = ~ 1, group = cluster, data = data,
```

```
tau = c(.25,.5,.75), control = lqmmControl(method = "df")),
```

with `data` being our simulated dataset as a data frame that contains `y` as the simulated response variable and `cluster` as an ID indicator for each cluster, and we also have 0.25, 0.5, and 0.75 set as our quantile levels, `tau`. The `'lqmm()'` function fits a separate model for each level of `tau` .

As we simulated the data with normal assumptions on the error terms, we are able to calculate the true quantiles of the individual clusters and of the overall data sets. Each of the i individual clusters are $N(\mu+\gamma_i, \sigma)$ and the overall data set is $N\left(\mu, \sqrt{\sigma^2+\sigma_\gamma^2}\right)$. We calculate the true quantiles using these known distributions. To evaluate the effectiveness of `'lqmm()'` on the data, we calculate the mean squared error (MSE) of the `'lqmm()'` predictions of the quartiles (0.25, 0.5, 0.75) on both the cluster level (`'predict(level = 1)'`) and over the entire dataset (`'predict(level = 0)'`) and compare these values to the MSE of the corresponding sample quantiles calculated from the simulated data.

2.1.2 Visualizing Results

After simulating our data, we have 1,000 datasets for each of the 5 ICC levels specified in table (2.1), and for each dataset we have calculated the MSE for the `'lqmm()'` prediction and the MSE for the sample data. To do an initial analysis of the results, we average all MSE calculations for each ICC level and plot each average as a dot in figure (2.1).

Figure (2.1) shows that for predicting the overall percentiles (predict level 0), the average MSE values for lqmm and data are approximately equal at around 0 for low to mid ICC levels but are large for high ICC levels across all percentile calculations. The only notable difference between the average MSE values is at the ICC level of 99% where lqmm is large compared to the data's average MSE. The figure also shows that for predicting the individual cluster percentiles (predict level 1), the average MSE values for lqmm are notably smaller than that of the data at ICC of 1% and the average MSE values are approximately equal between the two for ICC levels of 25%, 50%, and 75% across all percentile calculations. One notable difference between percentiles is seen in the difference between the average MSE values for the ICC level of 99%. The average MSE for lqmm is higher than that of the data at the 25th and 75th percentiles, but at the 50th percentile the two MSE averages are approximately equal near 0. As it is hard to see which average MSE is larger when they are both approximately 0, we will take the ratio of each pair of averages and display them

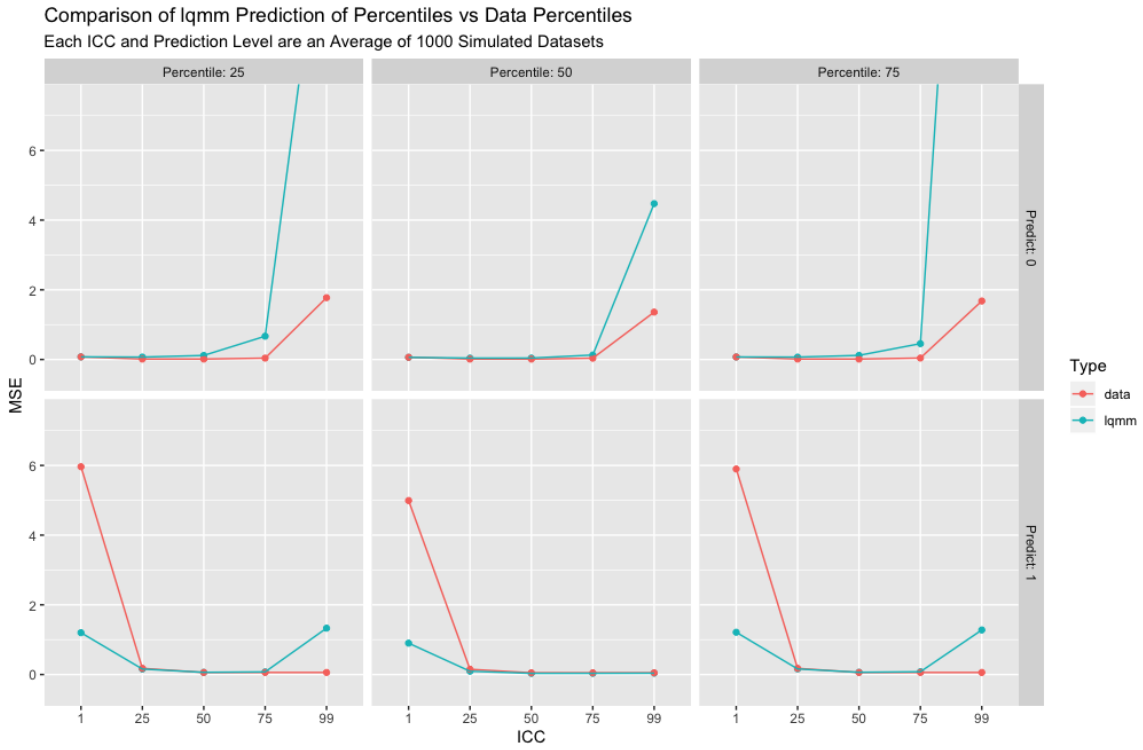


Figure 2.1: Comparison of lqmm Prediction vs Sample Data Percentiles (0.25, 0.5, 0.75). Each dot is an average of MSE values from 1,000 simulated datasets. Columns represent the percentile predicted and rows represent prediction level.

Simulation Set Up			Percentile					
			25%		50%		75%	
Predict Lvl	ICC	Iterations	lqmm	data	lqmm	data	lqmm	data
1	99%	1000	1.3310	0.0591	0.0392	0.0496	1.2784	0.0592
1	75%	1000	0.0761	0.0596	0.0354	0.0499	0.0802	0.0591
1	50%	1000	0.0634	0.0592	0.0344	0.0500	0.0644	0.0588
1	25%	1000	0.1577	0.1781	0.0947	0.1504	0.1606	0.1775
1	1%	1000	1.2004	5.9636	0.9000	4.9927	1.2133	5.8984
0	99%	1000	13.8657	1.7740	4.4726	1.3604	26.6956	1.6784
0	75%	1000	0.6681	0.0395	0.1285	0.0369	0.4554	0.0418
0	50%	1000	0.1164	0.0126	0.0412	0.0114	0.1185	0.0120
0	25%	1000	0.0692	0.0132	0.0403	0.0123	0.0691	0.0128
0	1%	1000	0.0756	0.0743	0.0613	0.0635	0.0729	0.0722

Table 2.2: Average MSE values over 1,000 simulated datasets

in figure (2.2).

In figure (2.2) we are able to identify which average MSE values are higher for each ICC level, predict level, and percentile combination by taking the ratio of the average MSE values in the form of $(lqmm/data)$. Thus if the points on the figure are above the horizontal, dashed, red line at $lqmm/data = 1$, then the average MSE value for lqmm is larger than that of the data. We see immediately that for every ICC level except for 1%, lqmm has a larger average MSE than the data for predict level zero. ICC level of 1% is directly on the line. This holds for every percentile calculated. For predict level 1, we see that the ratios increase as ICC level increases for each percentile level. We do have a number of points below the red line for predict level 1, and to further analyze this, we will look at the percentage of lqmm MSE values that are below that of the data for each simulated dataset in figure (2.3).

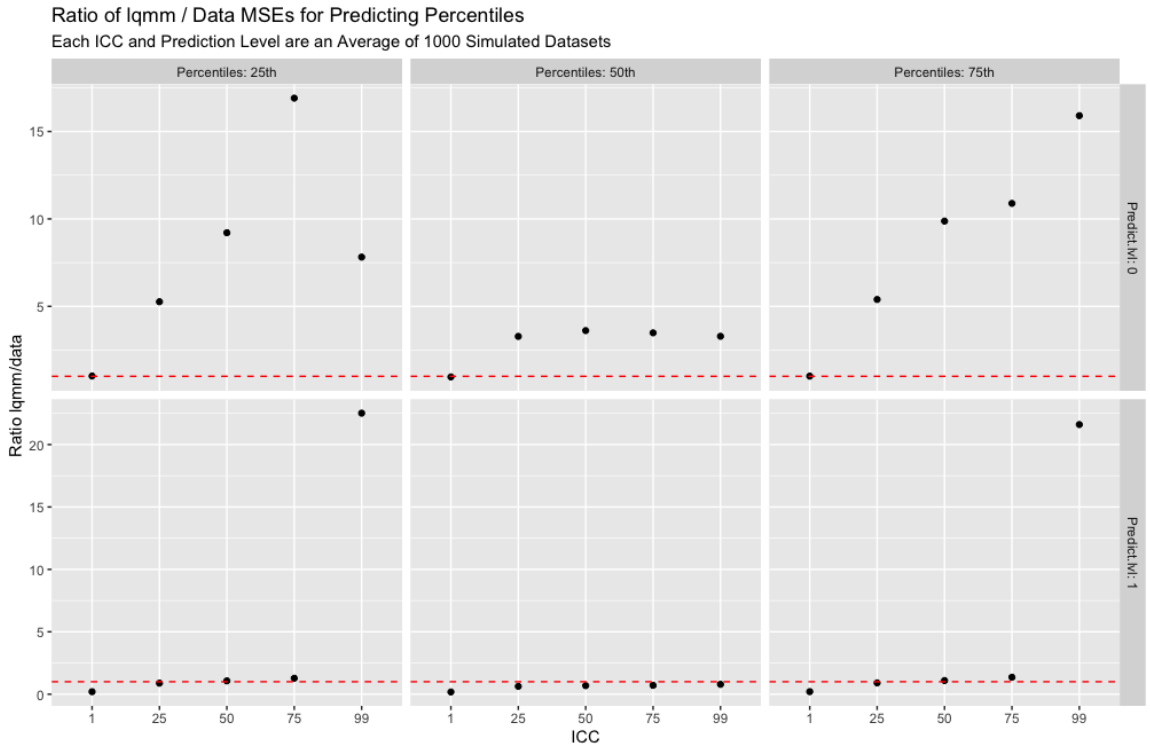


Figure 2.2: Ratio taken by dividing the lqmm MSE average by the data MSE average. Averages taken over a simulation of 1,000 data sets. Ratio less than 1 (below the dotted red line) means lqmm did better at predicting percentile.

Row	Percentiles	ICC	Predict.lvl	Ratio (lqmm/data)
1	25th	99	1	22.5169
2	25th	75	1	1.2772
3	25th	50	1	1.0696
4	25th	25	1	0.8850
5	25th	1	1	0.2013
6	25th	99	0	7.8159
7	25th	75	0	16.9070
8	25th	50	0	9.2100
9	25th	25	0	5.2641
10	25th	1	0	1.0167
11	50th	99	1	0.7897
12	50th	75	1	0.7100
13	50th	50	1	0.6875
14	50th	25	1	0.6301
15	50th	1	1	0.1803
16	50th	99	0	3.2878
17	50th	75	0	3.4851
18	50th	50	0	3.6134
19	50th	25	0	3.2831
20	50th	1	0	0.9644
21	75th	99	1	21.6043
22	75th	75	1	1.3582
23	75th	50	1	1.0950
24	75th	25	1	0.9050
25	75th	1	1	0.2057
26	75th	99	0	15.9050
27	75th	75	0	10.8882
28	75th	50	0	9.8753
29	75th	25	0	5.3932
30	75th	1	0	1.0091

Table 2.3: Ratio taken by dividing the lqmm MSE average by the data MSE average. Averages taken over a simulation of 1,000 data sets.

In figure (2.3) each bar represents the proportion of lqmm MSE values that are smaller than the data MSE values out of the 1,000 simulated datasets for each ICC level, predict level, and percentile combination. We see that predict level 0 has some lqmm predictions that did better than the data percentiles, but none of these bars are notably high as they are all around or below the 0.5 mark. Now for predict level one, we see multiple bars past the 0.5 mark. For the 50th percentile, we see that every bar is approximately 1, meaning that lqmm was more accurate at predicting the cluster medians than the data. It is also clear that as we increase ICC level, the lqmm prediction of the 25th and 75th percentiles becomes less accurate than the data predictions in terms of MSE. We have that the bar is above the 0.5 mark only for ICC of 1% and 25%.

We have found that across every ICC level, lqmm does not perform better than the sample data percentiles for the overall distribution (predict level 0) of balanced clustered data with normal error assumptions, but lqmm almost always has a more accurate prediction of the median on the cluster level (predict level 1) and does better in low ICC level situations for predicting the 25th and 75th percentiles. Predict level 0 seemed to be intended to show if random effects are necessary in fitting the model to the data, but in order to do that, a fixed model must be alone fit to the data and then compared to the mixed model fit. Predict level 0 is based on a mixed model being fit and then the random effects being dropped which leads to different calculations for your fixed effects than fitting the fixed model alone. For this reason, we will ignore predict level 0 for the following simulation and data analysis.

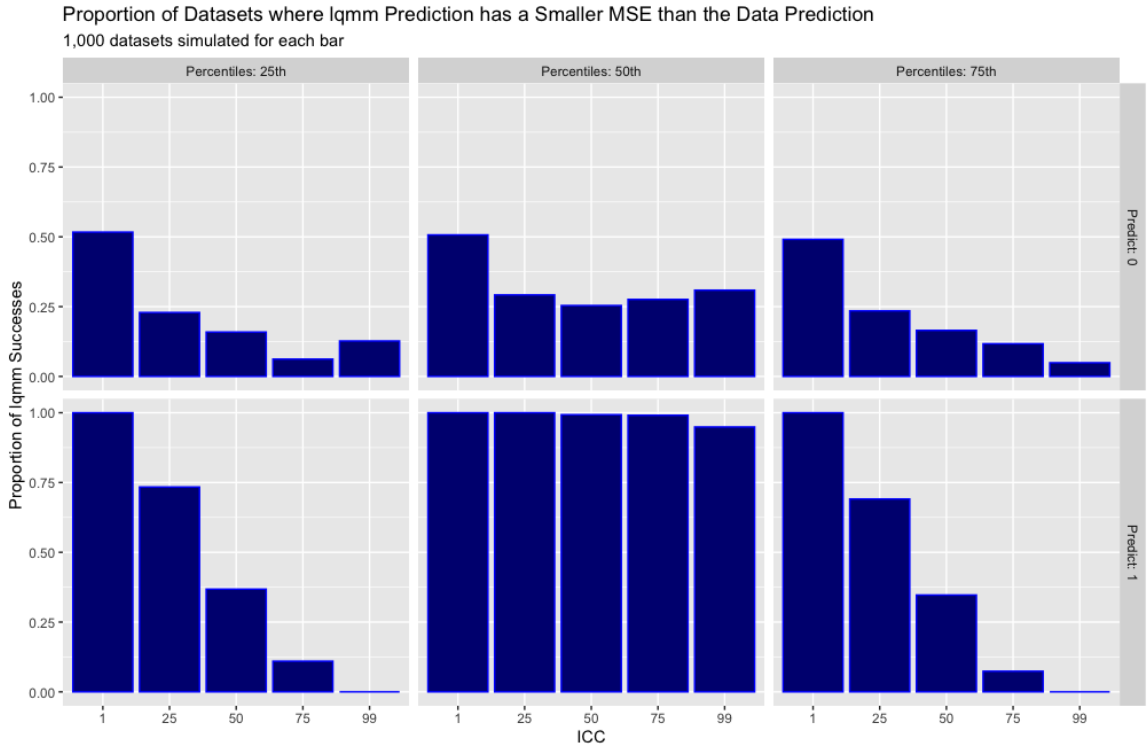


Figure 2.3: Proportion of times lqmm's MSE for predicting the true percentile was less than the data's MSE for predicting the true percentile. 1,000 datasets for each ICC and Prediction Level Combination

Prediction Lvl	ICC%	25th	50th	75th
1	99	0.000	0.949	0.000
1	75	0.110	0.991	0.074
1	50	0.368	0.993	0.347
1	25	0.734	1.000	0.691
1	1	1.000	1.000	1.000
0	99	0.127	0.309	0.049
0	75	0.062	0.276	0.117
0	50	0.159	0.254	0.165
0	25	0.229	0.292	0.235
0	1	0.517	0.507	0.491

Table 2.4: Proportion of times lqmm's MSE for predicting the true percentile was less than the data's MSE for predicting the true percentile. 1,000 datasets for each ICC and Prediction Level Combination

2.2 Simulation 2

In a second round of simulation, we will give the lqmm function simulated data that would be analyzed more effectively with linear quantile mixed models rather than classical least squares. The simulated data will have $i = 1, 2 \ldots, n$ clusters with $j = 1, 2, \ldots, m$ data points per cluster. There will be a longitudinal explanatory variable x that will take on values j/m for all values of j in each cluster, and a random slope ζ_i will be simulated for each cluster. Our ζ_i allows our slope to change from cluster to cluster. In order to simulate data that have different trends for each quantile level τ within each cluster, we have multiplied ϵ_{ij} by x_{ij} in our model. This creates more spread (i.e. fanning out) in our data as x_{ij} increases within each cluster. The model is

$$Y_{ij} = \mu + \gamma_i + (\zeta_i + \epsilon_{ij})x_{ij}, \tag{2.2}$$

where

$$\begin{aligned}
x_{ij} &= j/m \quad \text{where } j = 1, 2, \ldots, m, \\
\mu &= \text{the overall mean}, \\
\gamma_i &= \text{random intercept for the } i^{th} \text{ cluster}, \quad \zeta_i = \text{random slope for the } i^{th} \text{ cluster}, \\
\epsilon_{ij} &= \text{random error associated with } Y_{ij}, \\
\gamma_i &\overset{\text{iid}}{\sim} N(0, \sigma_\gamma), \quad \zeta_i \overset{\text{iid}}{\sim} N(0, \sigma_\zeta), \quad \text{and} \quad \epsilon_{ij} \overset{\text{iid}}{\sim} N(0, \sigma).
\end{aligned}$$

Note that we now have that the conditional distribution for each of the i clusters is $N(\mu + \gamma_i + \zeta_i * (j/m), \sigma * (j/m))$ and the marginal distribution for the overall data set is $N\left(\mu, \sqrt{\sigma_\gamma^2 + (\sigma_\zeta^2 + \sigma^2) * (1 + j/m)}\right)$ for our model in equation (2.2). This changes our ICC calculation to now be

$$ICC(Y_{ij}, Y_{ij'}) = \frac{\sigma_\zeta * x_{ij} * x_{ij'} + \sigma_\gamma}{\sqrt{((\sigma_\zeta + \sigma) * x_{ij}^2 + \sigma_\gamma) * ((\sigma_\zeta + \sigma) * x_{ij'}^2 + \sigma_\gamma)}}. \tag{2.3}$$

Since our ICC calculation in equation (2.3) depends on x_{ij} and $x_{ij'}$, we will average all ICC

values for consecutive observations. This is calculated as

$$\sum_{j=1}^{m-1} \frac{ICC(Y_{ij}, Y_{i(j+1)})}{m-1}.$$

For $m = 30$ points per cluster, we will calculate different average ICC levels by changing σ and holding σ_γ and σ_ζ constant. We found combinations that create different ICC levels that we want to simulate, and the values of these combinations are in table (2.5).

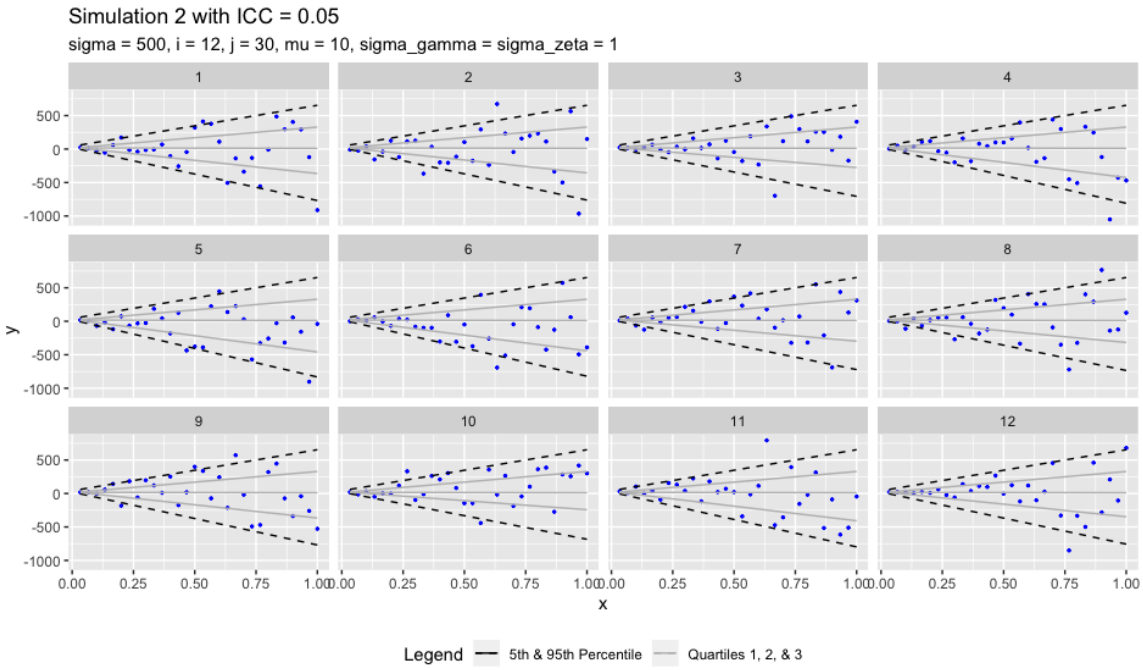
σ	σ_γ	σ_ζ	ICC
500	1	1	0.05
30	1	1	0.25
6.5	1	1	0.5
1.7	1	1	0.75
0.25	1	1	0.95

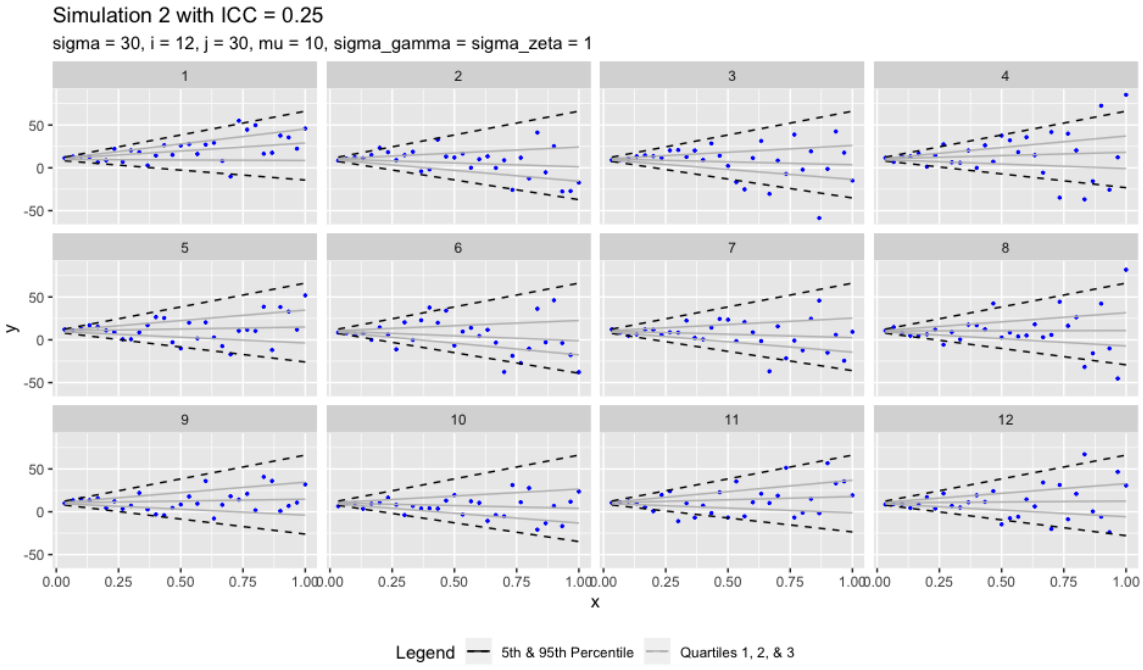
Table 2.5: Average ICC levels for values of σ, σ_γ, and σ_ζ in equation (2.3) for $m = 30$ points per cluster.

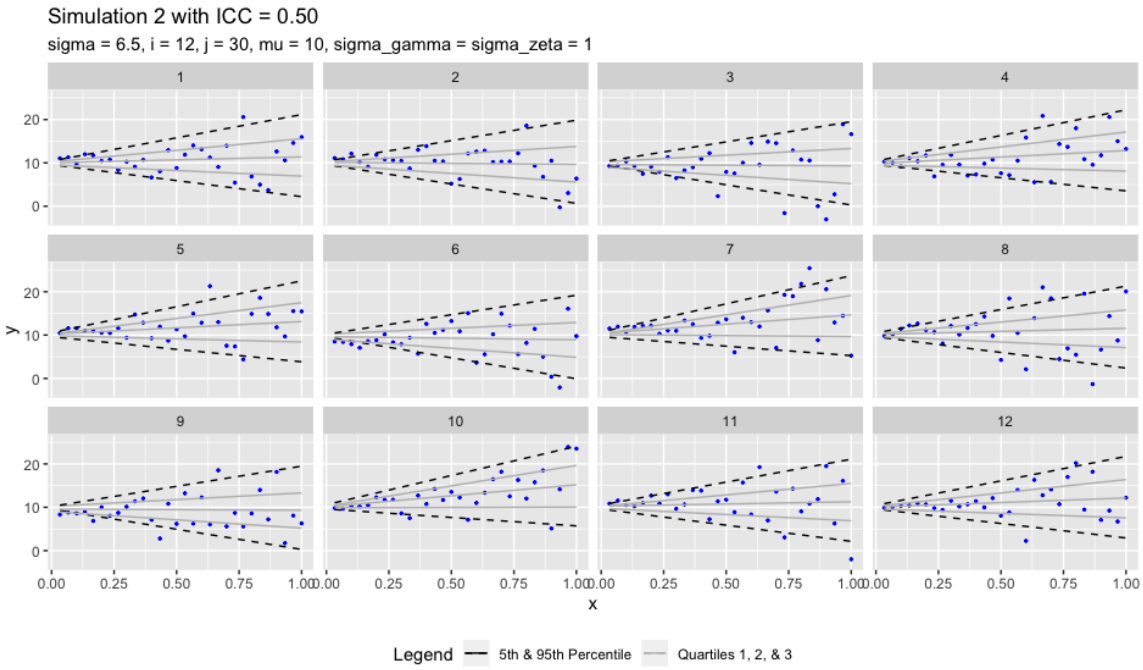
We will not do the same MSE comparison that we did in simulation 1 in this simulation 2. Simply finding the quantiles ignoring the independent variable of this longitudinal data does not make sense in practice. Instead we will perform a smaller scale simulation with $i = 12$ clusters, $j = 30$ points per cluster, $\mu = 10$, and σ values corresponding to table (2.5). As we have 5 ICC levels in table (2.5), we will have 5 trellis plots for each ICC level that will each have a 3x4 grid divided by each cluster. For each simulated dataset, we will run our `lqmm` fit using the command

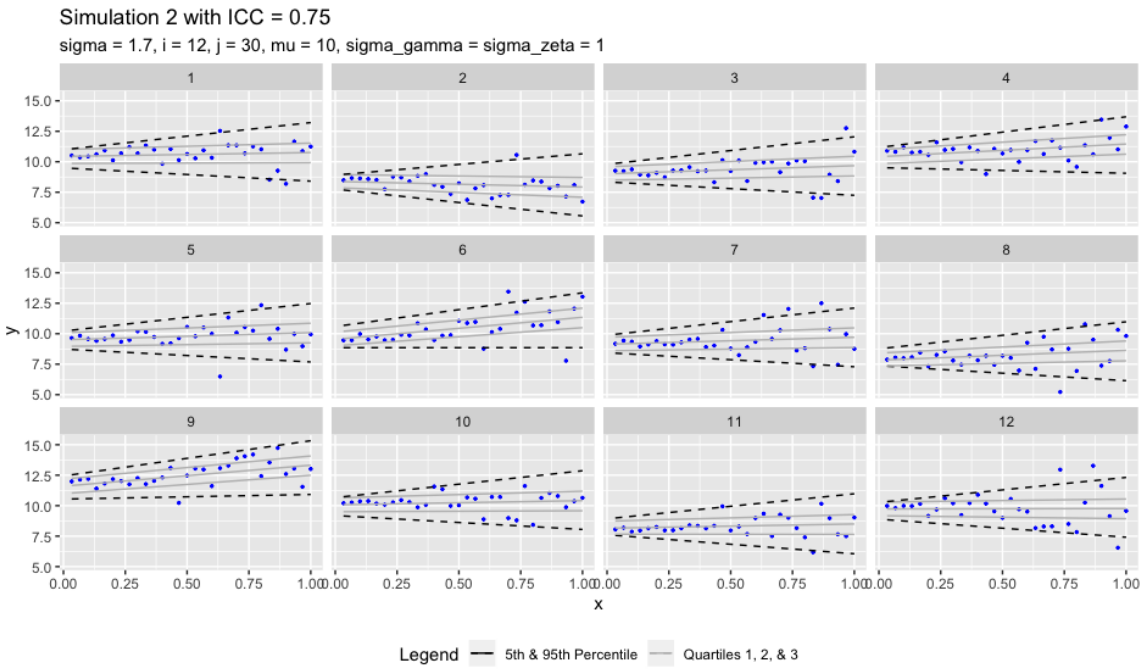
```
lqmm(fixed = y ~ 1, random = ~ x, group = cluster, data = test,
                tau = taus, control = lqmmControl(method = "df"))
```

where `test` is our simulated dataset from equation (2.2), `x` and `y` are our independent and dependent variables respectively, `cluster` is our cluster ID variable, and `taus` is a vector of our desired quantile levels (0.05, 0.25, 0.50, 0.75, 0.95). We choose these quantiles so as to see how the extreme quantiles behave (0.05 and 0.95) and to see how the standard quartiles behave (0.25, 0.50, 0.75) for the different ICC levels. We will also run all of our `lqmm` fits using `predict` level 1. We wish to further investigate the success we saw in simulation 1 with cluster level estimation and ignore the predict level 0 that does not make sense in practice. Our resulting plots and `lqmm` fits are displayed in figure (2.4).


Simulation 2 with ICC = 0.05
sigma = 500, i = 12, j = 30, mu = 10, sigma_gamma = sigma_zeta = 1
1
2
3
4
5
6
7
8
9
10
11
12
500
0
-500
-1000
0.00
0.25
0.50
0.75
1.00
y
x
Legend
5th & 95th Percentile
Quartiles 1, 2, & 3


Simulation 2 with ICC = 0.25
sigma = 30, i = 12, j = 30, mu = 10, sigma_gamma = sigma_zeta = 1
1
2
3
4
5
6
7
8
9
10
11
12
50
0
-50
0.00
0.25
0.50
0.75
1.00
y
x
Legend
5th & 95th Percentile
Quartiles 1, 2, & 3


Simulation 2 with ICC = 0.50
sigma = 6.5, i = 12, j = 30, mu = 10, sigma_gamma = sigma_zeta = 1
1
2
3
4
5
6
7
8
9
10
11
12
y
x
Legend
5th & 95th Percentile
Quartiles 1, 2, & 3


Simulation 2 with ICC = 0.75
sigma = 1.7, i = 12, j = 30, mu = 10, sigma_gamma = sigma_zeta = 1
1
2
3
4
5
6
7
8
9
10
11
12
y
x
Legend
5th & 95th Percentile
Quartiles 1, 2, & 3

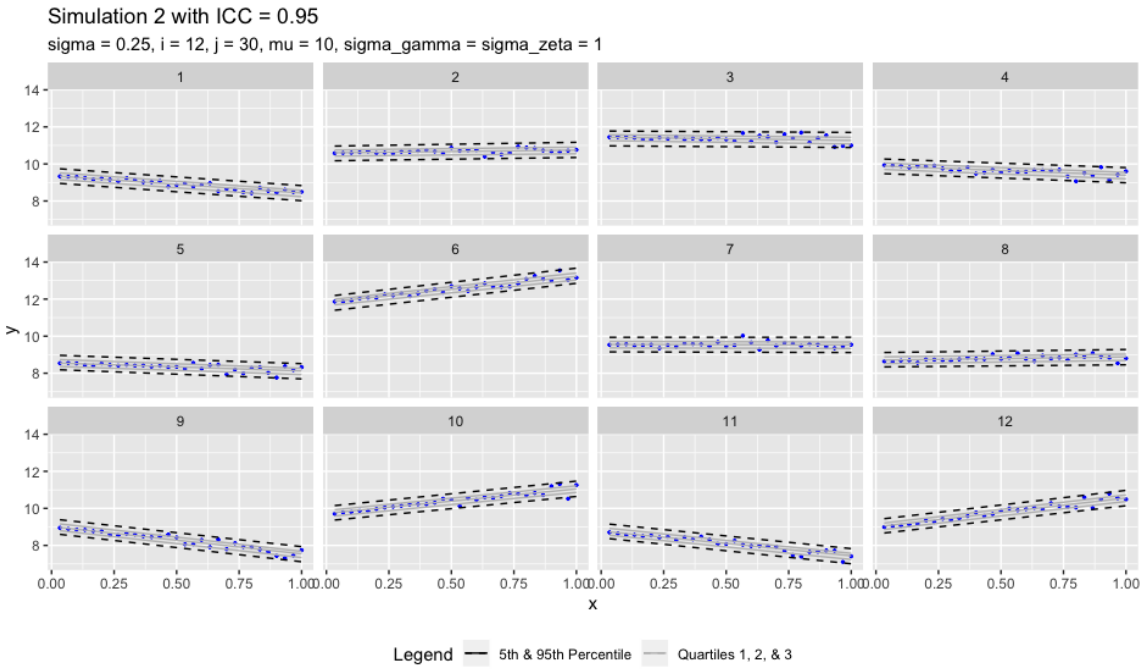


Figure 2.4: Simulated data from equation (2.2) with $i = 12$ clusters, $j = 30$ points per cluster, $\mu = 10$, and σ values corresponding to table (2.5) and each plot's labeled ICC value. Quantile fits found using lqmm at predict level 1.

From our plots in figure (2.4) we see that as our ICC increases for our simulated data from equation (2.2), our data not only become more independent in their trends from cluster to cluster, but the fanning as x_{ij} increases for our low ICC data begins to tighten to almost completely linear trends for our data with high 0.95 ICC. So for low ICC we have data that does not necessarily need random effects for cluster to cluster variation, but the data is effectively modeled by the quantile regression aspect of our `lqmm` fit since each quantile trends differently as our independent variable increases. For high ICC, we have data that is almost linear and does not necessarily need to be modeled using quantile regression, but since the clusters each have completely independent trends in our data, the random slopes and intercepts for each cluster in our `lqmm` fit effectively model our data. For middle ICC levels, we see that both the quantile aspect and the random effects aspect of our `lqmm` fit are equally effective in modeling our data as we have both fanning and independent trends per cluster. This simulation shows the diversity of `lqmm` for all ICC levels in a more applied simulation. We will similarly apply `lqmm's` 5^{th} and 95^{th} quantile estimation to our lumber dataset

in the next chapter and discuss `lqmm's` uses in practice.

Chapter 3

Application of 'lqmm'

We will also apply the lqmm function to a data set that has two strength measurements for multiple cuts of lumber, j, over 41 different mills, i. It is believed that the mills will have different strength levels based on the geographical region they are in. The two strength measurements are adjMOE and adjMOR. A linear quantile mixed model with both random and fixed slopes and intercepts will be fitted to predict the 5^{th} and 95^{th} percentiles of the relationship between the two strength measurements. This will create a 90% confidence interval and allow the individual mills to identify future lumber that do not satisfy a set tolerance in their region. The 90% confidence interval will be set based on the standard strength levels for the geographical region that the mill is in by setting the mills as our cluster variable in our model and setting our `predict` level to 1. The model used is seen in equation (3.1) where the coefficients and error terms are evaluated for each quantile τ. The fitted model is

$$Y_{ij} = \alpha^{(\tau)} + \gamma_i^{(\tau)} + \left(\beta^{(\tau)} + \zeta_i^{(\tau)}\right) X_{ij} + \epsilon_{ij}^{(\tau)}, \tag{3.1}$$

where

$$
\begin{aligned}
Y_{ij} &= \text{ the adjMOE for mill } i \text{ and cut of lumber } j,\\
X_{ij} &= \text{ the adjMOR for mill } i \text{ and cut of lumber } j,\\
\tau &= \text{ the quantile level of interest,}\\
\alpha^{(\tau)} &= \text{the fixed intercept,} \quad \gamma_i^{(\tau)} = \text{random intercept for the } i^{th} \text{ mill,}\\
\beta^{(\tau)} &= \text{the fixed slope,} \quad \zeta_i^{(\tau)} = \text{random slope for the } i^{th} \text{ mill,}\\
\epsilon_{ij}^{(\tau)} &= \text{random error associated with } Y_{ij},\\
\gamma_i^{(\tau)} &\overset{\text{iid}}{\sim} N(0, \sigma_\gamma), \quad \zeta_i^{(\tau)} \overset{\text{iid}}{\sim} N(0, \sigma_\zeta), \quad \text{and} \quad \epsilon_{ij}^{(\tau)} \overset{\text{iid}}{\sim} AL(0, \sigma, \tau).
\end{aligned}
$$

To calculate the model in R, we run the function

```
lqmm(fixed = AdjMOE~adjMOR , random = ~ adjMOR, group = Mill, data = lumber2,
     na.action = na.omit, tau = taus, control = lqmmControl(method = "df")),
```

where `lumber2` is our data set with `Mill` as our ID variable for each mill and `AdjMOE` and `adjMOR` as our lumber strength variables, and where `taus` is a vector of our percentiles (0.05 and 0.95). Our results from the model fit are

```
Fixed effects:
             tau = 0.05  tau = 0.95
(Intercept)  0.3342626   1.0758475
adjMOR       0.0001056   0.0001051

Covariance matrix of the random effects:
tau = 0.05
(Intercept)      adjMOR
    0.01616     4.80048
tau = 0.95
(Intercept)      adjMOR
    0.02547     1.22167
```

```
Residual scale parameter: 0.02063 (tau = 0.05)  0.02024 (tau = 0.95)
Log-likelihood: -146.4 (tau = 0.05)  -142.1 (tau = 0.95)

Number of observations: 403
Number of groups: 41.
```

Now using the `predict` function at `level=1`, we can make a trellis plot of all of the mills and their individually fitted 5^{th} and 95^{th} percentiles. Our resulting trellis plot is shown in figure (3.1).

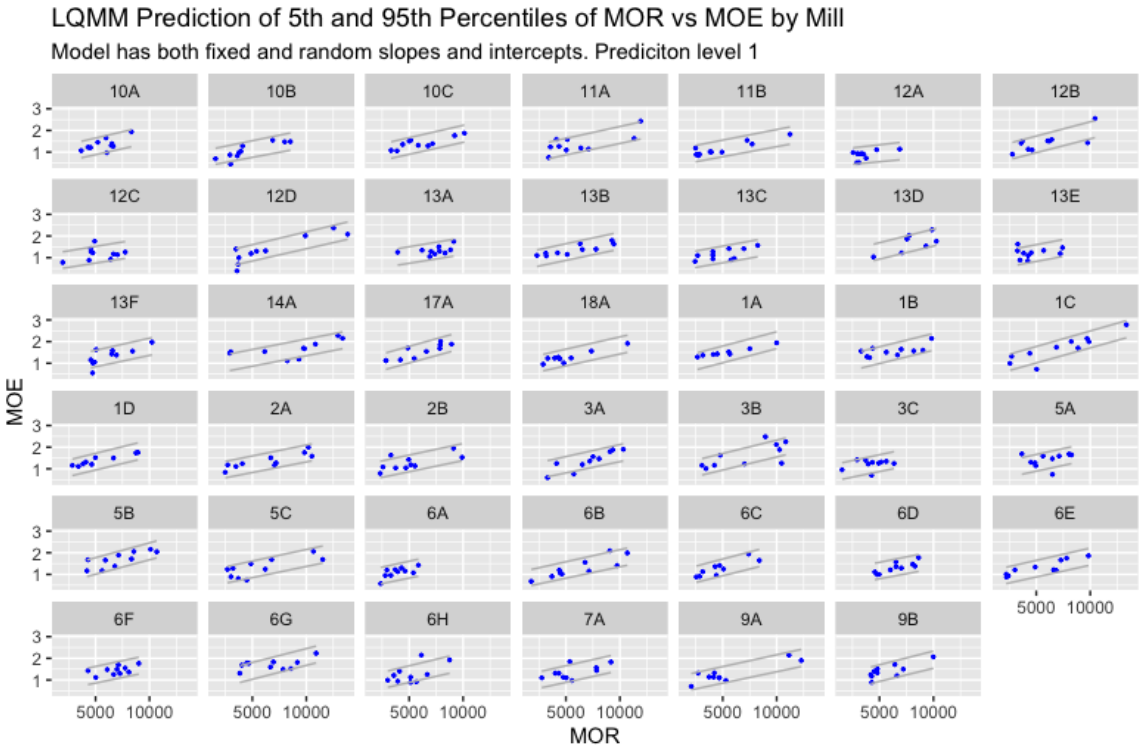


Figure 3.1: Trellis plot of lqmm fit for 5th and 95th percentiles of MOR vs MOE by Mill.

In statistics, the more predictions you make, the more likely you are at making a mistake. With 41 different models for every single mill, it may make sense to the company to broaden their clusters in the model. In order to do this, we will change our cluster ID from the 41 mills to the 14 regions that the mills can be divided into. In fitting our model that uses the region as our cluster,

i, we will run the function

```
lqmm(fixed = AdjMOE~adjMOR , random = ~ adjMOR, group = Region, data = lumber2,
     na.action = na.omit, tau = taus, control = lqmmControl(method = "df")),
```

where we have simply changed `group` from `Mill` to `Region`. The resulting output is

```
Fixed effects:
             tau = 0.05  tau = 0.95
(Intercept)  0.3408945   1.0655192
adjMOR       0.0001057   0.0001181

Covariance matrix of the random effects:
tau = 0.05
(Intercept)      adjMOR
    0.01468     2.94519
tau = 0.95
(Intercept)      adjMOR
    0.00914     2.12966

Residual scale parameter: 0.02377 (tau = 0.05)  0.02605 (tau = 0.95)
Log-likelihood: -151.1 (tau = 0.05)  -182.6 (tau = 0.95)

Number of observations: 403
Number of groups: 14
```

and the trellis plot of our resulting predictions is shown in figure (3.2).

Figure 3.2: Trellis plot of lqmm fit for 5th and 95th percentiles of MOR vs MOE by Region.

These two plots can be used to identify mills or regions that stand out with respect to the others as well as give tolerance limits for new lumber in each location. This 90% confidence interval is just one of the many applications of `lqmm` on this data. `lqmm` can be used to compare the median strength values between the mills to see which mills produce the strongest lumber, it can be used to predict MOE strength from a given MOR at a specific mill or region, or it can be used to see if for certain regions the lower percentiles of MOE strength in lumber trend differently in MOR strength than the higher percentiles of MOE strength. These are just some of `lqmm's` many applications to this data set.

Chapter 4

Conclusions and Discussion

We have tested the effectiveness of linear quantile mixed modeling through the use of the `lqmm` package in R. We found through our first simulation that predict level 0 of `lqmm` prediction does not have any practical use for quantile based mixed modeling. We also found that for one way clustered data following equation (2.1), predict level 1 estimates are more accurate than the data at predicting the median for all levels of ICC, and they are more accurate at predicting the first and third quartiles for low ICC levels. We furthered our investigation by performing a second simulation from equation (2.2) where we included a longitudinal independent variable in each cluster and simulated random error that depended on the independent variable. From this simulation we saw the diversity of the `lqmm` function. For low ICC levels, the predict level 1 estimates of our `lqmm` fit effectively modeled the fanning trends of each quantile as our independent variable increased. For high ICC levels, the estimates effectively modeled the heterogeneic linear trends in each cluster. For medium ICC levels, we saw a balance between the effective aspects of predictions that we saw in the low and high ICC level data. Our analysis of `lqmm` was finished with a data analysis application of lumber strength data. Here we successfully fit our model and plotted trellis plots to analyze our mills and regions. The 90% confidence interval created for each mill and each region can aid in prediction of future strength measurements, identification of out of tolerance lumber cuts, and identification of mills or regions that have lumber with different overall strength levels.

We have also identified some opportunities for further investigation of the `lqmm` function. The error that occurred when predicting the 95^{th} percentile in the first plot of figure (2.4) came from an issue of our covariance matrix estimation. A further investigation of the `control` and `covariance`

commands in the `lqmm` function will be necessary to identify and remedy the error. We can also further our investigation of `lqmm` by simulating higher level multivariate datasets and observing our prediction performance again for different ICC levels.

Bibliography

[1] M. Geraci. Linear quantile mixed models: The lqmm package for laplace quantile regression. *Journal of Statistical Software*, 57(13):1–29, 2014.

[2] M. Gerci. *lqmm: Linear Quantile Mixed Models*, 2019. R package version 1.5.5.

[3] R. Koenker and G. Basset. Regression quantiles. *Econometrica*, 46(1):33–50, 1978.

[4] KM. Yu and J. Zhang. A three-parameter asymmetric laplace distribution and its extension. *Communications in Statistics - Theory and Methods*, 34(9-10):1867–1879, 2005.