



2010-04-20

Parameter Estimation in Linear-Linear Segmented Regression

Erika Lyn Hernandez

Brigham Young University - Provo

Follow this and additional works at: <https://scholarsarchive.byu.edu/etd>



Part of the [Statistics and Probability Commons](#)

BYU ScholarsArchive Citation

Hernandez, Erika Lyn, "Parameter Estimation in Linear-Linear Segmented Regression" (2010). *All Theses and Dissertations*. 2113.
<https://scholarsarchive.byu.edu/etd/2113>

This Selected Project is brought to you for free and open access by BYU ScholarsArchive. It has been accepted for inclusion in All Theses and Dissertations by an authorized administrator of BYU ScholarsArchive. For more information, please contact scholarsarchive@byu.edu, ellen_amatangelo@byu.edu.

Parameter Estimation in Linear-Linear Segmented Regression

Erika L. Hernandez

A project submitted to the faculty of
Brigham Young University
in partial fulfillment of the requirements for the degree of

Master of Science

C. Shane Reese, Chair
Scott D. Grimshaw
Gilbert W. Fellingham

Department of Statistics
Brigham Young University

August 2010

Copyright © 2010 Erika L. Hernandez

All Rights Reserved

ABSTRACT

Parameter Estimation in Linear-Linear Segmented Regression

Erika L. Hernandez

Department of Statistics

Master of Science

Segmented regression is a type of nonlinear regression that allows differing functional forms to be fit over different ranges of the explanatory variable. This paper considers the simple segmented regression case of two linear segments that are constrained to meet, often called the linear-linear model. Parameter estimation in the case where the joinpoint between the regimes is unknown can be tricky. Using a simulation study, four estimators for the parameters of the linear-linear model are evaluated. The bias and mean squared error of the estimators are considered under differing parameter combinations and sample sizes. Parameters estimated in the model are the location of the change-point, the slope and intercept of the first segment, the change in slope from the first segment to the second, and the variance over both segments.

Keywords: change-point regression, broken-stick model, maximum likelihood, Bayesian estimators, bent-cable regression

ACKNOWLEDGMENTS

I would like to express my gratitude to all those who have supported me and cheered for me throughout this entire process. Dr. Reese, thanks for agreeing to be my advisor and to work with me on a topic that is less familiar to you. I am sure I could have made this triply difficult for myself if it had not been for your encouragement to keep things simple and get the project done. To my parents, I am very grateful for your interest in my project and your prayers for me. Lastly, I would like to thank my wonderful husband, Steven, for being so patient and supportive, not only while I have been trying to complete this project, but also throughout the whole of my graduate studies.

CONTENTS

Contents	iv
1 Introduction	3
1.1 Segmented Regression	3
1.2 Linear-Linear Regression Model	4
2 Literature Review	6
2.1 Maximum Likelihood and Least Squares Estimators	6
2.2 Bent-Cable Methods	8
2.3 Bayesian Change-point Models	8
3 Methods	10
3.1 Maximum Likelihood Estimators	10
3.2 Tishler and Zang Maximum Likelihood Estimator	12
3.3 Bayesian Posterior Mean	14
3.4 Simulation Study	15
4 Simulation Results and Application	23
4.1 Analyses Performed	23
4.2 General Results	25
4.3 Individual Estimator Performance	29
4.4 Simulation Considerations	32
5 Weapon Crack Length Application	34

5.1	Introduction and Data	34
5.2	Maximum Likelihood Estimators	36
5.3	Bayesian Mean	36
5.4	Application Conclusions	37
6	Conclusion and Discussion	39
	Bibliography	41
	Appendices	44
	Appendix A: Estimator Detail	45
	A.1 Maximum Likelihood Estimator-I	45
	A.2 Tishler and Zang Maximum Likelihood Estimator	47
	Appendix B: Simulation Results	48
	Appendix C: Computer Code	56
	C.1 Estimators and Functions Used in Simulation	56
	C.2 Simulation Study	63
	C.3 Plotting Functions	65
	C.4 Simulation Analyses	66
	C.5 Application	79

LIST OF TABLES

3.1	Simulation Setup. Size and parameter factors and levels	16
3.2	Simulation Setup. Orthogonal array design	18
3.3	Simulation Setup. Hand-tuned variances for Metropolis-Hastings candidate distributions in each of the 18 parameter combination cases. Used for the Bayesian mean estimator	21
4.1	Simulation Results. MLE-I Estimator. ANOVA p -values on 10 responses	24
4.2	Simulation Results. MLE-II Estimator. ANOVA p -values on 10 responses	24
4.3	Simulation Results. MLE-TZ Estimator. ANOVA p -values on 10 responses	24
4.4	Simulation Results. Bayesian Estimator. ANOVA p -values on 10 responses	24
4.5	Simulation Results. <i>Nsim</i> for MLE-TZ after those with convergence issues removed	31
5.1	Application. Parameter estimates from each estimator	34
B.1	Simulation Results. Estimated biases and MSEs for MLE-I	48
B.2	Simulation Results. Estimated biases and MSEs for MLE-II	49
B.3	Simulation Results. Estimated biases and MSEs for MLE-TZ	50
B.4	Simulation Results. Estimated biases and MSEs for Bayesian mean	51
B.5	Simulation Results. Estimated biases of all estimators. Combinations 1 to 9 . . .	52
B.6	Simulation Results. Estimated biases of all estimators. Combinations 10 to 18 . .	53
B.7	Simulation Results. Estimated MSEs of all estimators. Combinations 1 to 9 . . .	54
B.8	Simulation Results. Estimated MSEs of all estimators. Combinations 10 to 18 . .	55

LIST OF FIGURES

1.1	Introduction. Sample linear-linear regression model	5
3.1	Simulation Setup. Representative plots of data for the 18 combinations considered in the orthogonal array design	19
4.1	Results. Regressions formed by means of all parameter estimates in each of 18 combinations	26
4.2	Results. Maximum likelihood estimators' bias of $\hat{\sigma}^2$	27
4.3	Results. The effect β_2 has on bias of $\hat{\beta}_0$, $\hat{\beta}_1$, and $\hat{\beta}_2$ across estimators.	28
4.4	Results. Illustration of a data set with large bias in several MLE-I estimates . .	29
4.5	Results. Possible explanation for bias in Bayesian posterior mean change-point estimator	33
5.1	Application. Plot of masked weapon data	35
5.2	Application. Estimated regressions from each of four estimators	35
5.3	Application. Marginal Bayesian posteriors	38

INTRODUCTION

1.1 SEGMENTED REGRESSION

In a world of constant flux, it is important to have statistical models that account for change. Some changes occur abruptly and other changes are more gradual; some are obvious and others are subtle. The statistical tools that model change are as varied as the types of change. Segmented regression is one statistical tool for modeling phenomena that transition and change. In segmented regression different functional forms are fit in different neighborhoods of the explanatory variable.

In segmented regression, the point in the explanatory variable at which the functional form changes is usually of some interest because it represents a shift in the pattern of the data. The location of this shift is called a knot, joinpoint, or change-point, and the neighborhoods delimited by the jointpoints are often referred to as regimes, phases, or stages. This nomenclature gives rise to other names by which segmented regression is known: change-point or joinpoint regression, multistage and multiphase regression, and piecewise regression. Other names, such as splines, hockey-stick models, and bent-cable regression, are descriptive of a specific type or subclass of change-point problems. A survey of these models and change-point problems can be found in [Poirier \(1976\)](#) or in [Seber and Wild \(1989\)](#).

The change-point umbrella covers cases where multiple change-points occur in a single explanatory variable and where a single shift occurs in each of the multiple explanatory variables. In some instances the knot locations are predetermined by the researcher; in other cases the location of the optimal shift is unknown and must be estimated. Some models call for adjacent regression segments to meet at the joinpoint, and others allow for a discontinuity. Additional constraints may impose smoothness conditions at the join,

requiring continuous first and second derivatives. These constraints can be met with one or more nonlinear segments, such as a polynomial or trigonometric function. While there are many types and subclasses of piecewise regression, only a single simple case will be considered herein: the linear-linear model.

1.2 LINEAR-LINEAR REGRESSION MODEL

Linear-linear regression models, sometimes called broken-stick or broken-line models, account for one explanatory variable with a single unknown change-point where the ends of the two linear segments are constrained to meet. There are several ways to parameterize this model, but the following form will be used throughout this paper:

$$y = \begin{cases} \beta_0 + \beta_1 x + \epsilon & x \leq \gamma \\ \beta_0 + \beta_1 x + \beta_2(x - \gamma) + \epsilon & x > \gamma \end{cases}, \quad \text{where } \epsilon \sim N(0, \sigma^2)$$

or equivalently

$$y = \beta_0 + \beta_1 x + \beta_2(x - \gamma)I(x > \gamma) + \epsilon, \quad \text{where } \epsilon \sim N(0, \sigma^2).$$

A generic graphical representation might look like the plot shown in Figure 1.1.

In this parameterization of the model, γ represents the change-point, β_0 is the intercept, β_1 is the slope under the first regime, β_2 represents the change in slope from the first to second regime (thus, $\beta_1 + \beta_2$ is the slope after the change-point), and σ^2 is the common variance for both segments.

In segmented regression, there are usually two objectives: 1) to estimate the model parameters that produced the data and 2) to determine if the data really do come from separate regimes or if the data could be modeled without a change-point. The primary objective of this study is to compare several methods for parameter estimation. The estimators considered are maximum likelihood-I, maximum likelihood-II, the Tishler-Zang maximum likelihood alternative (Tishler and Zang 1981a), and the Bayesian posterior mean. The bias

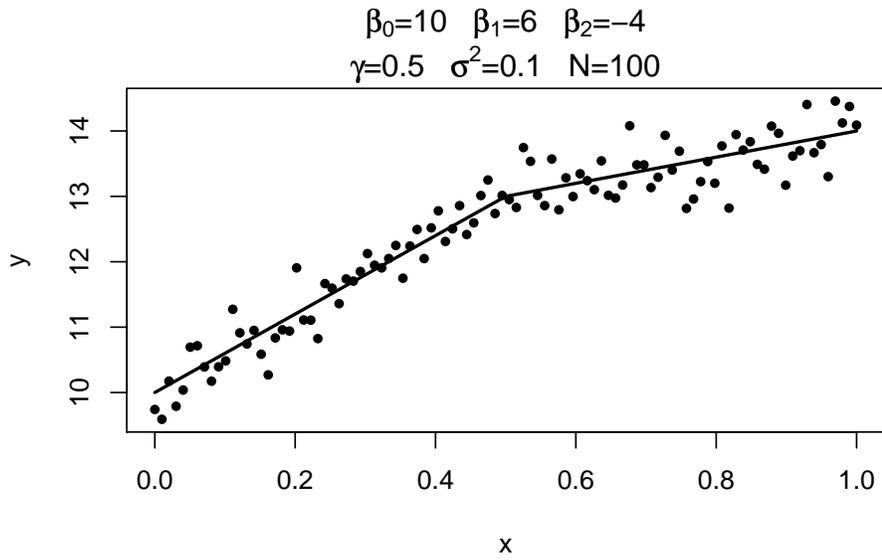


Figure 1.1: A sample linear-linear model

and mean squared error (MSE) of the estimators are found through a simulation study. Additionally, the estimators are applied to data on the aging and deterioration of weapons.

LITERATURE REVIEW

Change-point models first began to receive attention in the 1950s in the quality-control arena, where the objective was to determine if and where a shift in means had occurred (Page 1954). Quandt (1958) is the first to formulate the problem in a regression setting, and the linear-linear regression model is one of the first to be studied. Over the years, the field has become much more diversified, including expansion to other parametric forms besides normal data, inference using a nonparametric approach, estimation through clustering and regression tree methods, comparisons to mixtures of distributions, methods for dealing with correlated data, and computational improvements for processing large data sets. The body of literature covering change-point methods is expansive, but only a small portion of the relevant papers will be reviewed here.

2.1 MAXIMUM LIKELIHOOD AND LEAST SQUARES ESTIMATORS

Much of the early work in change-point models was done in the least squares context; however, when the data are normally distributed the least squares estimates are equivalent to the maximum likelihood estimates (MLE). For purposes of this paper, estimators that were originally derived as least squares estimators may also be referred to as maximum likelihood estimators.

In piecewise models, when the underlying model is discontinuous, the likelihood is discontinuous; when it is continuous, the likelihood is continuous. Similarly, continuity constraints placed on the derivatives at the joinpoints translate into continuity constraints on the likelihood. Without these smoothness constraints, likelihood and least squares based methods that require differentiation are cumbered by discontinuities in the derivatives. In the

case of the linear-linear model, the likelihood is a continuous function, but it is not smooth, meaning it is not continuously differentiable.

Early pioneers of the segmented model (Quandt 1958, Blichschke 1961, Sprent 1961, Robison 1964, Hudson 1966, Hinkley 1969) approach the discontinuity problem by assuming that $x_m < \gamma < x_{m+1}$ for some unknown index m and reposing the problem as maximizing the discrete marginal likelihood $L(m)$. Estimates $\hat{\beta}$, $\hat{\gamma}$, and $\hat{\sigma}^2$ can be found by differentiating over the first m points and the remaining $n - m$ points and adding the desired continuity constraints. The likelihood for each choice of m is then calculated and the final optimal division of data into regimes, or the optimal m , is decided by maximizing $L(m)$.

The approach taken by these early researchers is similar, but they each study slightly different models. Quandt (1958) derives the least squares estimators for normally distributed data with two linear regimes that do not necessarily intersect. Blichschke (1961) addresses the linear-linear model under the continuous joinpoint constraint. Robison (1964) is the first to extend the change-point regression problem to polynomial segments, deriving the MLEs and confidence intervals around the change-point between two polynomials; however, nonreal solutions are possible under his formulation. Not only does Hudson (1966) address this nonreal problem, but he also expands the methodology to more than two polynomial regimes, distinguishes between types of joinpoints, and proposes a method to derive parameter estimators with unknown join types. Hinkley (1969) also considers estimation in the linear-linear model.

The spread of computers has made maximum likelihood methods for piecewise regression somewhat more tractable. A grid search in the change-point parameter space is proposed by Lerman (1980) as a way to ensure that global optimums have been found, as local maxima are often present in piecewise regression. Gallant and Fuller (1973) use numerical optimization routines to directly estimate the joinpoint in the model consisting of smoothly grafted polynomials. While numerical optimization algorithms are convenient to use when the segments join smoothly, regressions that are nonsmooth, such as the linear-

linear model, cannot be estimated by typical nonlinear maximization methods without first smoothing over the corners.

2.2 BENT-CABLE METHODS

There are several methods for parameter estimation that revolve around replacing the traditional nonsmooth model with one that transitions smoothly at the change-point. This is usually done for one of two reasons: 1) the belief is that the underlying phenomenon is more accurately represented by a smooth transition than an abrupt change or 2) the estimators are more easily obtained with a smooth model. When used for maximum likelihood or least squares, smoothing allows for differentiability along the entire curve and the use of gradient techniques for estimation. [Bacon and Watts \(1971\)](#) propose the first bent-cable method in which the two-phase linear model is reparameterized using transition functions. Specifically, they use the hyperbolic tangent function to transition from one linear regime to the next. [Griffiths and Miller \(1973\)](#) further clarify the restrictions on transition functions and propose their own hyperbolic form, thereafter performing a least-squares analysis. In two papers published in 1981, [Tishler and Zang](#) propose two smoothed-model, maximum-likelihood based methods. The first ([Tishler and Zang 1981a](#)), presented in more detail later, parameterizes the model as the max (or min) of intersecting lines and uses a smooth approximation to the max function. Their second paper ([Tishler and Zang 1981b](#)) does not replace the whole likelihood function but rather smooths in an arbitrarily small neighborhood around the joinpoint.

2.3 BAYESIAN CHANGE-POINT MODELS

Approaching this problem from a Bayesian perspective obviates the need for smoothing the model since differentiation of the likelihood is not required. Another advantage to the Bayesian approach is that the researcher often has an idea about where the change-point occurs, so it is possible to incorporate this prior expert knowledge into the model. Lastly, the

distributions of the estimates come as a natural by-product of sampling from the posterior distribution of the parameters and readily lend themselves to inference through hypothesis testing and interval estimates. These distributional results, and the inference capabilities they provide, do not come as easily in the classical approach.

Bacon and Watts (1971) are the first to derive Bayesian posterior estimators for the linear-linear model, using an improper joint prior. Ferreira (1975) also uses an improper prior when deriving the posteriors for the case of two linear regimes that are not constrained to meet. Smith and Cook (1980) deal with the linear-linear case and normally distributed data, but they choose to use a multivariate normal prior on the intercept and slope parameters. In the previously mentioned cases, derivation of the posteriors is enabled by conjugacy. The use of MCMC to obtain posterior draws and the computational power to perform the analysis have since facilitated more widespread and varied uses of Bayesian methods. These MCMC methods are demonstrated in Carlin et al. (1992) and in Stephens (1994).

The purpose of this study is to compare parameter estimation methods of the linear-linear model. Two maximum likelihood estimators are considered as well as the Tishler-Zang maximum likelihood alternative and the Bayesian posterior mean. This chapter includes an explanation of each estimator and details regarding how the estimators are evaluated via simulation study.

3.1 MAXIMUM LIKELIHOOD ESTIMATORS

As mentioned in Section 2.1, in any case where errors are normally distributed around the underlying model, the least squares solutions are equivalent to maximum likelihood solutions. Thus, the results derived herein could be thought of as either maximum likelihood or least squares estimators. The goal of maximum likelihood is to find the parameter values that maximize the joint likelihood function given a particular set of data, (\mathbf{x}, \mathbf{y}) .

The likelihood for a single data pair (x_i, y_i) can be written

$$L(\boldsymbol{\beta}, \gamma, \sigma^2 | x_i, y_i) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2\sigma^2} \{y_i - [\beta_0 + \beta_1 x_i + \beta_2(x_i - \gamma)I(x_i \geq \gamma)]\}^2},$$

and because the n observations are independent, their joint likelihood function is the product of the individual likelihoods. Maximization of the likelihood function is equivalent to maximizing the log-likelihood function; however, the log-likelihood is often easier to deal with. The joint log-likelihood function is

$$\mathcal{L}(\boldsymbol{\beta}, \gamma, \sigma^2 | \mathbf{x}, \mathbf{y}) = -\frac{n}{2} \log(2\pi) - \frac{n}{2} \log(\sigma^2) - \frac{1}{2\sigma^2} \sum_{i=1}^n \left\{ y_i - [\beta_0 + \beta_1 x_i + \beta_2(x_i - \gamma)I(x_i \geq \gamma)] \right\}^2.$$

The typical way to solve a nonlinear optimization such as this would be to use a Newton-Raphson numerical optimization algorithm. The one complication is that the likelihood is not smooth, meaning that it is not continuously differentiable, a requirement for

Newton-Raphson implementation. As discussed in Section 2.1, early pioneers of the segmented model approach this problem by assuming that $x_m < \gamma < x_{m+1}$ for some unknown index m and reposing the problem as a maximization of the discrete marginal likelihood $L(m)$. A similar framework is adopted here.

Under the assumption that $x_m \leq \gamma < x_{m+1}$,

$$\begin{aligned} \mathcal{L}(m, \boldsymbol{\beta}, \gamma, \sigma^2 | \mathbf{x}, \mathbf{y}) &= -\frac{n}{2} \log(2\pi) - \frac{n}{2} \log(\sigma^2) - \frac{1}{2\sigma^2} \sum_{i=1}^m \{y_i - [\beta_0 + \beta_1 x_i]\}^2 \\ &\quad - \frac{1}{2\sigma^2} \sum_{i=m+1}^n \{y_i - [\beta_0 + \beta_1 x_i + \beta_2(x_i - \gamma)]\}^2. \end{aligned}$$

The gradient and Hessian of the marginal log-likelihood are derived by differentiating with respect to each parameter. These are included in Appendix A. The proposed method for calculating the maximum likelihood estimates (MLE-I) is to use a Newton-Raphson algorithm to estimate each $L(m)$ while constraining $x_i < \hat{\gamma} < x_{i+1}$ to avoid optimizing across a discontinuity in the gradient. The overall MLEs will be the estimates that maximize the discrete $L(m)$.

Comparison to Traditional Methods

It should be noted that Newton-Raphson is not a necessary requirement for optimizing each $L(m)$. In the 1950s, 60s, and 70s, the optimums were calculated using local least squares, but when the local least squares regression lines did not intersect in (x_m, x_{m+1}) , a series of additional steps had to be taken to show that the marginal optimums occurred at either x_m or x_{m+1} (Hudson 1966, Hinkley 1969). By using Newton-Raphson and constraining $\hat{\gamma}$ to be in $[x_m, x_{m+1}]$, the same routine can be run for each m and convergence occurs on the boundary when the boundary is the appropriate marginal optimum. Also, under the assumption that there is a single joinpoint, the number of optimizations to be performed is no greater than $n - 2$, so the calculations are reasonably fast.

Comparison to Methods in Current Practice

In practice, users often supply their segmented models to nonlinear optimization routines. When continuity of the first derivatives is imposed, most nonlinear algorithms can handle the segmented model, whether they derive the analytic derivatives or use DUD methods to approximate the gradient and Hessian. Also, convergence to the right answer is probable as long as the likelihood is not too flat or the algorithm does not converge to a local optimum. However, when there is no continuity of the first derivatives, as in the case of the linear-linear model, the reliability of the estimate will rely heavily on the accuracy of the initial guess, if the algorithm converges at all. To explore these naïve methods, a second maximum likelihood estimator (MLE-II) will be found by supplying the linear-linear likelihood function to an optimization routine that numerically approximates the gradient and Hessian. By evaluating its performance, MLE-II may shed light on how accurate these estimators are in practice.

3.2 TISHLER AND ZANG MAXIMUM LIKELIHOOD ESTIMATOR

There are many methods in the literature for estimating the parameters of this model. Variants of maximum likelihood or least squares methods typically smooth over the nonsmooth joinpoint to facilitate differentiation over the entire range of change-point values. Tishler and Zang (1981a) propose one such maximum likelihood method for segmented regression, the Tishler and Zang maximum likelihood alternative (MLE-TZ). They represent their model as the maximum of two intersecting lines

$$y_i = \max\{\mathbf{a}'_1 x_{1i}, \mathbf{a}'_2 x_{2i}\} + \epsilon_i, \quad \text{where } \epsilon_i \sim N(0, \sigma^2)$$

and smooth over the discontinuity using the approximation $\max\{\alpha_1, \alpha_2\} = \lim_{p \rightarrow \infty} \{\alpha_1^p + \alpha_2^p\}^{1/p}$, where $p > 0$. When the proposed segmented regression is convex ($\beta_2 > 0$), using \max is appropriate; however, when it is concave ($\beta_2 < 0$), p less than zero models the minimum

between the two intersecting lines. As $p \rightarrow \infty$, this approximation approaches the original nonsmooth model.

An additional requirement of their approximation is that α_1 and α_2 must be positive. To ensure this, they recommend that a constant B be added to each ordinate that is slightly larger (in absolute value) than the all y 's. The model then becomes

$$B + y_i \sim \{(B + \mathbf{a}'_1 x_{1i})^p + (B + \mathbf{a}'_2 x_{2i})^p\}^{1/p} + \epsilon_i.$$

Tishler and Zang note that in some cases adding a large constant, B , and raising it to the power of p may cause computer overflow. They suggest multiplying and dividing by $\delta_i = \max\{|B + \mathbf{a}'_1 x_{1i}|, |B + \mathbf{a}'_2 x_{2i}|\}$ in the following manner

$$B + y_i \sim \delta_i \left\{ \left(\frac{B + \mathbf{a}'_1 x_{1i}}{\delta_i} \right)^p + \left(\frac{B + \mathbf{a}'_2 x_{2i}}{\delta_i} \right)^p \right\}^{1/p} + \epsilon_i.$$

The log-likelihood function can therefore be written

$$\mathcal{L}(\beta, \gamma | \mathbf{x}, \mathbf{y}) = -\frac{n}{2} \log(2\pi) - \frac{n}{2} \log(\sigma^2) - \frac{1}{2\sigma^2} \sum_{i=1}^n \left(B + y_i - \delta_i \left\{ \left(\frac{B + \mathbf{a}'_1 x_{1i}}{\delta_i} \right)^p + \left(\frac{B + \mathbf{a}'_2 x_{2i}}{\delta_i} \right)^p \right\}^{1/p} \right)^2.$$

Maximization of this function is performed by a Newton-Raphson implementation. The first partial derivatives of the gradient are included in Appendix A. The second partial derivatives are unwieldy and were obtained partly using a symbolic derivative calculator. They are included in Appendix C along with the computer code.

The selection of an appropriate B and p must also be considered. Tishler and Zang suggest the following iterative procedure:

1. Let $k = 0$. Choose an even positive integer $p^{(0)}$, a $B^{(0)}$ that is slightly larger in absolute value than all values of y , and starting parameter values $\mathbf{a}_1^{(0)}$ and $\mathbf{a}_2^{(0)}$.
2. Let $k = k + 1$. Estimate the parameters with $p^{(k-1)}$ and $B^{(k-1)}$ as constants. Call the resulting parameter estimates $\mathbf{a}_1^{(k)}$ and $\mathbf{a}_2^{(k)}$.
3. If $B^{(k-1)} + \mathbf{a}_1^{(k)} x > 0$ and $B^{(k-1)} + \mathbf{a}_2^{(k)} x > 0$ for all x , proceed to step 4. If not, go to step 5.

4. If $\{(\mathbf{a}_1^{(k)} - \mathbf{a}_1^{(k-1)})^2 + (\mathbf{a}_2^{(k)} - \mathbf{a}_2^{(k-1)})^2\}^{1/2} < \epsilon$, where ϵ is a small desired tolerance, then stop and use $\mathbf{a}_1^{(k)}$ and $\mathbf{a}_2^{(k)}$ as estimates. If not, go to step 6.
5. With parts of the estimated lines becoming negative after the last iteration, it is necessary to adjust B before performing the next optimization. Set $B^{(k)}$ equal to a value slightly larger than $-\min\{\mathbf{a}_1^{(k)}x, \mathbf{a}_2^{(k)}x, \text{ for all } x\}$. Proceed to step 6.
6. Let $p^{(k+1)} = p^k + \Delta p$, where Δp is a set positive constant. Go back to step 2.

Additionally, because the MLE-TZ is parameterized as two intersecting lines, a reparameterization of the estimates is needed. The change-point, γ , is found at the intersection of both lines, $a_{10} + a_{11}\gamma = a_{20} + a_{21}\gamma \implies \gamma = \frac{a_{20} - a_{10}}{a_{11} - a_{21}}$. The slope is $\beta_2 = a_{21} - a_{11}$. Thus, the following equations will be used to obtain the appropriate parameter estimates:

$$\begin{aligned}
\hat{\beta}_0 &= \hat{a}_{10}, \\
\hat{\beta}_1 &= \hat{a}_{11}, \\
\hat{\beta}_2 &= \hat{a}_{21} - \hat{a}_{11}, \\
\hat{\gamma} &= \frac{\hat{a}_{20} - \hat{a}_{10}}{\hat{a}_{11} - \hat{a}_{21}}, \text{ and} \\
\hat{\sigma}^2 &= \frac{1}{n} \sum_{i=1}^n \left\{ y_i - \left[\hat{\beta}_0 + \hat{\beta}_1 x_i + \hat{\beta}_2 (x_i - \hat{\gamma}) I(x_i > \hat{\gamma}) \right] \right\}^2.
\end{aligned}$$

3.3 BAYESIAN POSTERIOR MEAN

The final estimator considered is the Bayesian posterior mean. The posterior distribution of the parameters is obtained by classical Bayesian methods that can be found in any reliable textbook (Gelman et al. 2004).

The joint sampling distribution is

$$L(\boldsymbol{\beta}, \gamma, \sigma^2 \mid \mathbf{x}, \mathbf{y}) = \prod_{i=1}^n \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2\sigma^2} \left\{ y_i - [\beta_0 + \beta_1 x_i + \beta_2 (x_i - \gamma) I(x_i > \gamma)] \right\}^2}.$$

A joint prior on all five parameters is required. There is a suspicion that the parameters will be correlated; however, it is unclear how that correlation will occur. Therefore, priors are chosen under the *a priori* assumption of parameter independence. They are

$$\begin{aligned}\beta_0 &\sim N(m_{\beta_0}, s_{\beta_0}^2), \\ \beta_1 &\sim N(m_{\beta_1}, s_{\beta_1}^2), \\ \beta_2 &\sim N(m_{\beta_2}, s_{\beta_2}^2), \\ \gamma &\sim N(m_{\gamma}, s_{\gamma}^2), \text{ and} \\ \sigma^2 &\sim IG(a, b).\end{aligned}$$

Draws from the marginal posteriors of each parameter are obtained using Gibb's sampling, a method first introduced by Geman and Geman (1984) and later explored by Gefland and Smith (1990). Because of complications introduced by the indicator function in the likelihood, all but the σ^2 parameter will require draws to be obtained through the Metropolis-Hastings algorithm (Hastings 1970, Gefland and Smith 1990). The (log) complete conditionals required for the Gibb's sampler are

$$\begin{aligned}\ln[\beta_j] &\propto -\frac{1}{2\sigma^2} \sum_{i=1}^n \left\{ y_i - [\beta_0 + \beta_1 x_i + \beta_2(x_i - \gamma)I(x_i > \gamma)] \right\}^2 - \frac{1}{2s_{\beta_j}^2} (\beta_j - m_{\beta_j})^2, \text{ for } j=1,2,3; \\ \ln[\gamma] &\propto -\frac{1}{2\sigma^2} \sum_{i=1}^n \left\{ y_i - [\beta_0 + \beta_1 x_i + \beta_2(x_i - \gamma)I(x_i > \gamma)] \right\}^2 - \frac{1}{2s_{\gamma}^2} (\gamma - m_{\gamma})^2 \quad ; \text{ and} \\ \sigma^2 &\sim IG \left(\frac{n}{2} + a, \left(\frac{1}{b} + \frac{1}{2} \sum_{i=1}^n \left\{ y_i - [\beta_0 + \beta_1 x_i + \beta_2(x_i - \gamma)I(x_i > \gamma)] \right\}^2 \right)^{-1} \right).\end{aligned}$$

The means from each marginal posterior are used as the estimators of the parameters. Reducing the posterior distributions to a one-number summary allows for comparison to the other estimators in the simulation study.

3.4 SIMULATION STUDY

The performance of the four estimators is assessed by evaluating the marginal bias and mean squared error of each in a simulation study performed in R (v. 2.9.0 The R Foundation for

Table 3.1: Simulation study factors and levels

Factor	Level		
n	20	100	500
γ	0.25	0.5	0.75
β_0	0		
β_1	10	0.5	
β_2	-15	5	15
σ^2	0.1	1	

Statistical Computing). This section provides details about the setup of the simulation study.

Parameter Combinations

In order to assess the estimators under a variety of potential situations, different parameter combinations and sample sizes should be tested. The parameter values were selected by examining sample plots of simulated data and attempting to cover a wide range of possible modeling scenarios. The parameters and levels at which they will be evaluated are included in Table 3.1, and a justification of the choices follows.

SAMPLE SIZE n is the total number of points between the two regimes. It is tested at three different levels, corresponding to small, medium, and large samples: 20, 100, and 500.

CHANGE-POINT The parameter γ is usually the focal point and most interesting parameter in a change-point model. When γ has little data before or after it, estimating γ and the slopes on either side may be difficult. On the $[0,1]$ interval, γ is tested at 0.25, 0.5, and 0.75, corresponding to a 25/75%, a 50/50%, and a 75/25% split of the data into two phases.

INTERCEPT The intercept is rarely of interest in piecewise regression; hence, β_0 is held constant throughout the study at zero. However, it will still be estimated.

SLOPE 1 The parameter β_1 is selected at a single large magnitude ($\beta_1 = 10$) and a single small magnitude ($\beta_1 = 0.5$). Negative values are not considered because the model can always be modified to have a positive slope in the first regime by modeling $E(-Y|X)$.

CHANGE IN SLOPE β_2 represents the change in slope from the first regime to the second regime. Values are selected to model three scenarios: a positive change in slope ($\beta_2 = 10$), creating a convex function modeled by the maximum of two intersecting lines; a subtle shift in slope ($\beta_2 = 5$), which may be hard to detect with little data or large variance; and a shift in which the slope changes signs between regimes ($\beta_2 = -15$), providing a concave function modeled by the minimum of two intersecting lines.

VARIANCE Two values for σ^2 are used: $\sigma^2 = 0.1$ and $\sigma^2 = 1$. As in the case of all parameters presented, a value can only be labeled “large” or “small” in context of the application or in light of the other parameters. The variance is no exception. When the underlying model produces values for y that have a range of 100, a variance of 0.1 may seem small, but that same variance may overwhelm a model with a range of one. These two values for σ^2 were selected to complement other potential parameter combinations and, thereby, provide some “small,” “medium,” and “large” variance scenarios.

Orthogonal Array Design

A full factorial design would require the evaluation of 108 parameter combinations, but this is prohibitively large. Instead, an orthogonal array design is used with 18 combinations. An appropriate experimental design with the desired number of combinations was produced using SASTM software, Version 9.0 of the SAS System for Windows. Copyright © 2010 SAS Institute Inc. SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc., Cary, North Carolina, USA.

This orthogonal array will allow for analysis of the main effects of size and parameter levels but is not sufficient to obtain interaction effects. The combinations used in the

Table 3.2: Simulation study orthogonal array design

	n	γ	β_0	β_1	β_2	σ^2
1	20	0.25	0	10	15	0.1
2	20	0.25	0	0.5	5	1
3	20	0.50	0	0.5	15	1
4	20	0.50	0	10	-15	1
5	20	0.75	0	10	5	0.1
6	20	0.75	0	0.5	-15	0.1
7	50	0.25	0	10	-15	1
8	50	0.25	0	0.5	-15	0.1
9	50	0.50	0	10	5	0.1
10	50	0.50	0	0.5	5	0.1
11	50	0.75	0	10	15	1
12	50	0.75	0	0.5	15	1
13	100	0.25	0	0.5	15	0.1
14	100	0.25	0	10	5	1
15	100	0.50	0	10	15	0.1
16	100	0.50	0	0.5	-15	1
17	100	0.75	0	0.5	5	1
18	100	0.75	0	10	-15	0.1

simulation are included in Table 3.2, and Figure 3.1 displays a representative data set for each of the combinations.

Generating Realizations

An evenly spaced grid of n points from $[0,1]$ forms the x space. The $[0,1]$ grid is used because it can be generalized into almost any scenario. The y_i are obtained using R's built-in function `rnorm` to generate independent $N(0, \sigma^2)$ errors, which are then added to the underlying model $\beta_0 + \beta_1 x_i + \beta_2(x_i - \gamma)I(x_i \geq \gamma)$.

Simulation Size

For this simulation study, each parameter combination is replicated 1,000 times ($Nsim = 1,000$), and the estimators were applied to each of the 1,000 sets of data.

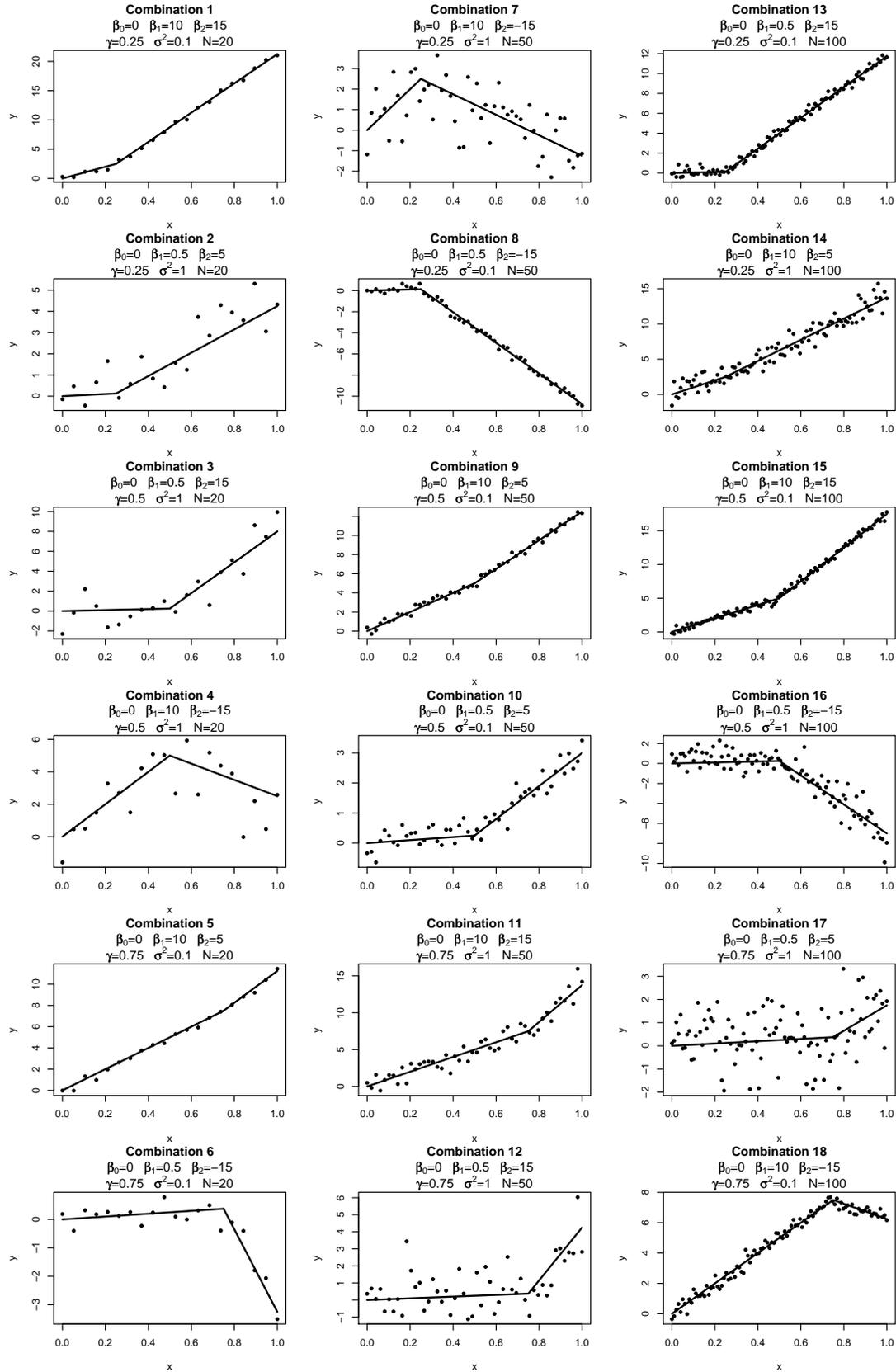


Figure 3.1: Sample data for the 18 combinations used in the orthogonal array design

Additional Setup Considerations

Some additional specification is required for MLE-TZ and the Bayesian posterior mean in order to perform the simulation study.

TISHLER AND ZANG MAXIMUM LIKELIHOOD ALTERNATIVE The initial B will be calculated as 1.1 times the largest absolute value of y_i . Subsequent updates to B (step 5) will be calculated as $-1.1 \times \min\{\mathbf{a}_1^{(k)}x, \mathbf{a}_2^{(k)}x, \text{ for all } x\}$. The initial value for p is set at $p = 20$ if the true regression is convex ($\beta_2 > 0$) and $p = -20$ if the true regression is concave ($\beta_2 < 0$). In step 6, p is incremented or decremented by 20 ($\Delta p = 20$) depending on the convexity or concavity of the desired regression.

BAYESIAN POSTERIOR MEAN Bayesian posterior draws are obtained by Markov chain Monte Carlo methods. Chains are allowed to run for 25,000 iterations, and the first 100 iterations are discarded. Because the chains are started at the true parameter values, a burn-in of 100 draws should be sufficient. Diffuse priors that cover all combinations of the orthogonal array design are used:

$$\begin{aligned}\beta_0 &\sim N(0, 100), \\ \beta_1 &\sim N(0, 100), \\ \beta_2 &\sim N(0, 100), \\ \gamma &\sim UNIF(0, 1), \text{ and} \\ \sigma^2 &\sim IG(2.001, 0.1).\end{aligned}$$

Note that, for purposes of the simulation study, the diffuse, uniform prior is used for γ instead of the previously proposed normal distribution. The inverse gamma prior on σ^2 has mean 9.99 and variance 99,900.

In the Gibb's sampler, the posterior on σ^2 is in closed form, and Metropolis-Hastings (MH) algorithms are used to obtain draws from the posteriors of $\beta_0, \beta_1, \beta_2$, and γ . In

Table 3.3: Hand-tuned variances for Metropolis-Hastings candidate distributions in each of the 18 parameter combination cases. Used for the Bayesian estimator.

	β_0	β_1	β_2	γ
1	0.60	2.0	6.0	0.004
2	0.50	2.0	18.0	0.200
3	0.60	2.0	16.0	0.015
4	0.60	2.0	16.0	0.015
5	0.60	2.0	16.0	0.090
6	0.60	2.0	30.0	0.040
7	0.30	1.0	2.0	0.005
8	0.10	0.5	1.0	0.001
9	0.10	0.5	2.0	0.010
10	0.10	0.5	1.5	0.010
11	0.30	0.7	20.0	0.005
12	0.30	0.5	22.0	0.010
13	0.05	0.1	0.4	0.000
14	0.10	0.5	1.5	0.010
15	0.05	0.1	0.5	0.001
16	0.10	0.4	1.5	0.001
17	0.10	0.4	12.0	0.020
18	0.05	0.1	4.0	0.001

all four MH cases, candidate values are drawn from a normal distribution centered at the previous draw. The draws for γ are further restricted by only accepting draws on the interval $[x_3, x_{n-2}]$. The candidate distributions require further tuning of their variance parameters to ensure that proper mixing takes place in each simulated scenario. A small simulation of 50 data sets for each combination with chains of length 5,000 was run several times, and the candidate distribution variances were adjusted for each size-parameter combination in the orthogonal array design. When the average acceptance rate across the 50 data sets for each parameter was between 25% and 50%, the variance was deemed acceptable. Table 3.3 contains the final candidate variances used in the simulation.

Comparison of Parameters

The parameters are evaluated by comparing their marginal biases and MSEs. The marginal bias and MSE for each parameter estimate are calculated as

$$\begin{aligned} Bias(\hat{\theta}) &= \frac{1}{Nsim} \sum_{i=1}^{Nsim} (\hat{\theta}_i - \theta), \text{ and} \\ MSE(\hat{\theta}) &= \frac{1}{Nsim} \sum_{i=1}^{Nsim} (\hat{\theta}_i - \theta)^2. \end{aligned}$$

The primary analysis of these results is performed using ANOVA on the univariate bias and MSE responses, considering each estimator separately. Additional tests are used when deemed appropriate. Estimators are also evaluated based on their ease of implementation and computational requirements.

SIMULATION RESULTS AND APPLICATION

The data from the simulation study are comprised of eighteen observations for each of four estimation methods, resulting in 72 total observations. Ten response variables were collected for each observation. They consist of five biases and five marginal MSEs, one for each of the five parameters estimated. The raw data from the simulation study can be found in Appendix B organized first by estimator and then by combination, enabling a side-by-side examination of all four estimators.

4.1 ANALYSES PERFORMED

The data are analyzed using ten univariate ANOVA analyses performed on each estimator separately (18 observations at a time). With the orthogonal array setup used in simulation, only the main effects of the size and parameter factors can be appropriately tested. The p -values for each main effect are summarized in Table 4.1, Table 4.2, Table 4.3, and Table 4.4. These ANOVAs alone are not sufficient to determine if the biases are significantly different than zero, but they are useful indications of how the estimators are influenced by their varying size and parameter factor levels. With so many potential effects to explain, after an effect was found to be significant, plots of the raw biases or MSEs were examined to determine the likely relationship. Follow-up tests were only occasionally performed.

As shown in the tables, many effects show up as significant at the $\alpha = 0.05$ level. In a multiple testing scenario such as this, the overall α -error rate is likely to be greater than 0.05. This fact is acknowledged but no adjustments are made for multiple testing, as this can be thought of as an exploratory analysis of these estimators. Additionally, transformations of the bias and MSE responses were considered and decided against: while the data are not perfectly normal (there exist a number of outliers), they are close enough.

Table 4.1: MLE-I Estimator. ANOVA p -values on 10 univariate responses by 5 factors. Note: * indicates significance at the $\alpha = 0.05$ level.

	Bias					MSE				
	$\hat{\beta}_0$	$\hat{\beta}_1$	$\hat{\beta}_2$	$\hat{\gamma}$	$\hat{\sigma}^2$	$\hat{\beta}_0$	$\hat{\beta}_1$	$\hat{\beta}_2$	$\hat{\gamma}$	$\hat{\sigma}^2$
n	0.269	0.081	0.060	0.633	0.009 *	0.059	0.066	0.089	0.102	0.024 *
β_1	0.137	0.709	0.293	0.864	0.291	0.432	0.829	0.633	0.209	0.326
β_2	0.008 *	0.018 *	0.008 *	0.627	0.951	0.122	0.047 *	0.064	0.003 *	0.942
γ	0.449	0.328	0.126	0.132	0.420	0.077	0.364	0.350	0.061	0.358
σ^2	0.229	0.081	0.025 *	0.392	0.001 *	< 0.001 *	0.058	0.068	0.007 *	0.001 *

Table 4.2: MLE-II Estimator. ANOVA p -values on 10 univariate responses by 5 factors. Note: * indicates significance at the $\alpha = 0.05$ level.

	Bias					MSE				
	$\hat{\beta}_0$	$\hat{\beta}_1$	$\hat{\beta}_2$	$\hat{\gamma}$	$\hat{\sigma}^2$	$\hat{\beta}_0$	$\hat{\beta}_1$	$\hat{\beta}_2$	$\hat{\gamma}$	$\hat{\sigma}^2$
n	0.257	0.323	0.415	0.566	0.019 *	0.021 *	0.458	0.298	0.147	0.262
β_1	0.106	0.268	0.066	0.766	0.465	0.608	0.889	0.136	0.254	0.407
β_2	0.034 *	0.121	0.011 *	0.503	0.440	0.859	0.091	0.020 *	0.017 *	0.326
γ	0.718	0.401	0.234	0.453	0.410	0.106	0.034 *	0.121	0.302	0.903
σ^2	0.377	0.355	0.064	0.236	< 0.001 *	0.001 *	0.044 *	0.008 *	0.024 *	0.070

Table 4.3: MLE-TZ Estimator. ANOVA p -values on 10 univariate responses by 5 factors. Note: * indicates significance at the $\alpha = 0.05$ level.

	Bias					MSE				
	$\hat{\beta}_0$	$\hat{\beta}_1$	$\hat{\beta}_2$	$\hat{\gamma}$	$\hat{\sigma}^2$	$\hat{\beta}_0$	$\hat{\beta}_1$	$\hat{\beta}_2$	$\hat{\gamma}$	$\hat{\sigma}^2$
n	0.314	0.35	0.351	0.779	0.007 *	0.022 *	0.455	0.309	0.086	0.023 *
β_1	0.193	0.452	0.244	0.823	0.306	0.849	0.969	0.759	0.671	0.329
β_2	0.044 *	0.083	0.016 *	0.402	0.923	0.684	0.069	0.015 *	0.004 *	0.935
γ	0.439	0.194	0.163	0.056	0.362	0.049 *	0.047 *	0.035 *	0.058	0.345
σ^2	0.235	0.195	0.062	0.189	< 0.001 *	< 0.001 *	0.058	0.013 *	0.008 *	0.001 *

Table 4.4: Bayesian Estimator. ANOVA p -values on 10 univariate responses by 5 factors. Note: * indicates significance at the $\alpha = 0.05$ level.

	Bias					MSE				
	$\hat{\beta}_0$	$\hat{\beta}_1$	$\hat{\beta}_2$	$\hat{\gamma}$	$\hat{\sigma}^2$	$\hat{\beta}_0$	$\hat{\beta}_1$	$\hat{\beta}_2$	$\hat{\gamma}$	$\hat{\sigma}^2$
n	0.407	0.192	0.997	0.995	< 0.001 *	0.010 *	0.427	0.077	0.053	< 0.001 *
β_1	0.167	0.14	0.680	0.818	0.426	0.295	0.077	0.862	0.889	0.621
β_2	0.033 *	0.089	0.015 *	0.957	0.209	0.197	0.244	0.299	0.004 *	0.133
γ	0.847	0.961	0.880	0.018 *	0.531	0.046 *	0.006 *	0.236	0.069	0.386
σ^2	0.498	0.386	0.587	0.194	0.127	< 0.001 *	0.007 *	0.013 *	0.152	0.822

4.2 GENERAL RESULTS

The plots in Figure 4.1 are formed by using the mean of all parameter estimates in one simulated combination. They are useful in helping to visualize how the biases of $\beta_0, \beta_1, \beta_2$, and γ interplay with each other. It is immediately apparent from these plots that some of the combinations are hard to estimate, and even after averaging them over many simulations, they were not able to find the true underlying model. Hinkley (1969) calls these cases “ill-defined,” when the variance of $\hat{\gamma}$ is large and there is significant bias in both $\hat{\gamma}$ and $\hat{\beta}_2$. Size-parameter combinations 2, 5, 14, and 17 could reasonably be labeled as ill-defined for each of the estimators. These combinations are characterized by a small, true, underlying change of slope and a change-point that is off center, resulting in little data in one regime or the other. Cases 2, 14, and 17 are additionally complicated by large variance in the data.

The ANOVA reveals that the maximum likelihood estimators (MLE-I, MLE-II, and MLE-TZ) behave similarly in many respects. As was expected, the maximum likelihood estimators for σ^2 were biased, and the bias decreased as the sample size increased (see Figure 4.2). The bias of these estimators, theoretically, may be corrected by multiplying $\hat{\sigma}^2$ by $\frac{n}{n-4}$; however, a simple t -test shows that the bias of the “unbiased” MLE-II $\hat{\sigma}^2$ is significantly different than zero (p -value=0.0057).

From the ANOVA results, it is shown that $Bias(\hat{\beta}_0)$ and $Bias(\hat{\beta}_2)$ are affected by β_2 for all estimators. $Bias(\hat{\beta}_1)$ is additionally influenced by β_2 for MLE-I. Looking to the bias plots in Figure 4.3, it is hypothesized that as the absolute change in slopes increases, the bias of $\hat{\beta}_0$ and $\hat{\beta}_2$ for MLE-I, MLE-II, and MLE-TZ decrease toward zero. While $Bias(\hat{\beta}_0)$ and $Bias(\hat{\beta}_2)$ are also significantly affected by β_2 for the Bayesian estimator, they stand apart from the maximum likelihood estimators. The Bayesian mean estimates seem to tend toward zero, causing overestimates in $\hat{\beta}_0$ and $\hat{\beta}_2$ when the change in slope is negative and underestimates when the change in slope is positive. The plots in Figure 4.3 also demonstrate

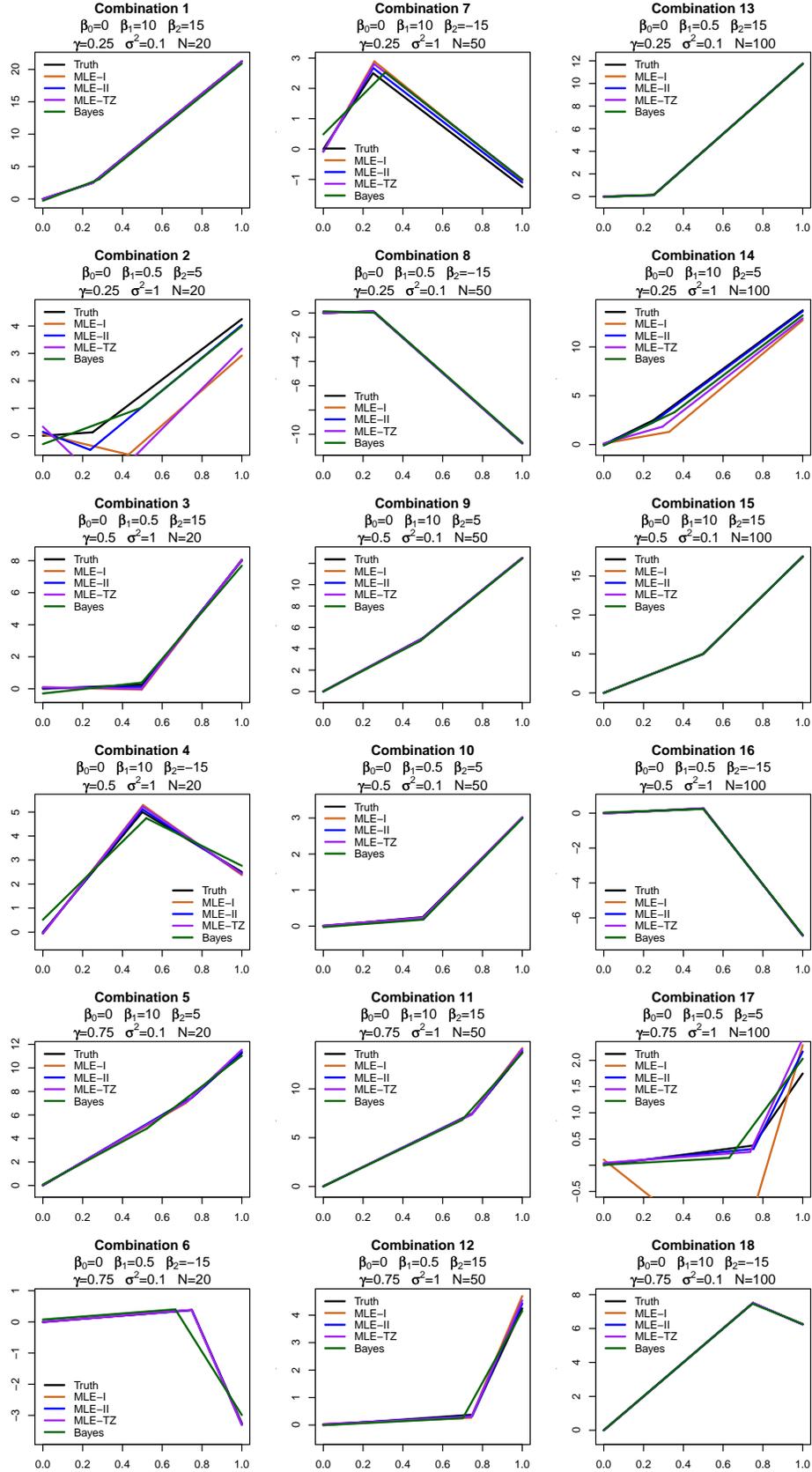


Figure 4.1: Regressions formed by means of all parameter estimates in each combination.

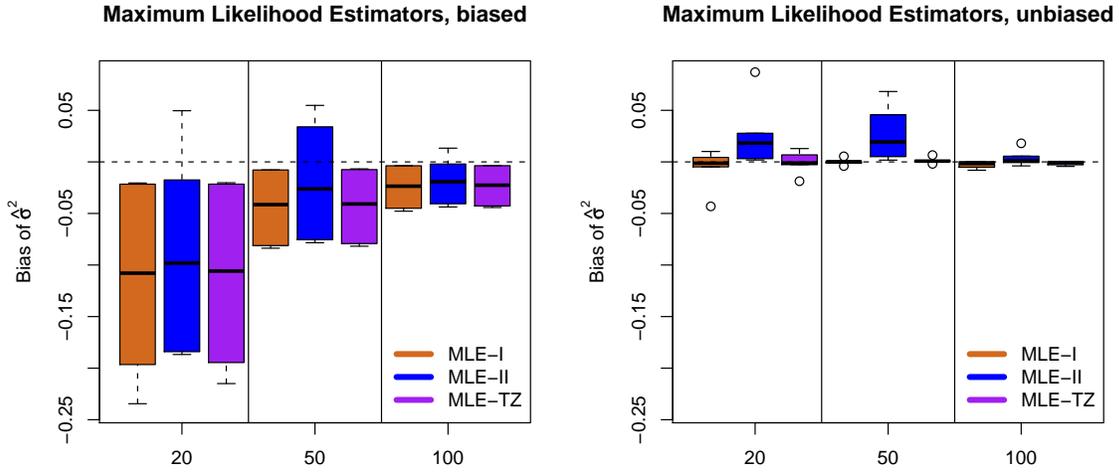


Figure 4.2: Bias of $\hat{\sigma}^2$ for each of the three maximum likelihood estimators. On the left are the original biases. On the right are the biases for the estimators that have been multiplied by $\frac{n}{n-4}$, which should theoretically correct the bias; however MLE-II is still biased.

the positive correlation of $Bias(\hat{\beta}_0)$ and $Bias(\hat{\beta}_2)$ and the negative correlation of $Bias(\hat{\beta}_1)$ with $Bias(\hat{\beta}_0)$ and $Bias(\hat{\beta}_2)$.

Across parameter estimates, MLE-I produced the largest MSEs of all estimators (see tables in Appendix B). The Bayesian means, despite being biased, in many cases, produced the smallest MSEs, especially as the sample size increased. For smaller sample sizes, MLE-II tended to have the smallest MSEs. The ANOVA indicates that σ^2 has an effect on the MSE for all estimators and almost every parameter. It is not surprising that increased variance leads to an increase in MSE.

Additionally, increased sample sizes led to smaller $MSE(\hat{\beta}_0)$ in most of the estimators. This is surprising because the MSE for the other parameter estimates are not, for the most part, being influenced by sample size. But it is not a very useful finding, as the intercept is not usually of interest in a change-point problem.

In the analysis of these data, several interesting features were found. The following section speaks in detail about these interesting features and other findings as they pertain to each estimator.

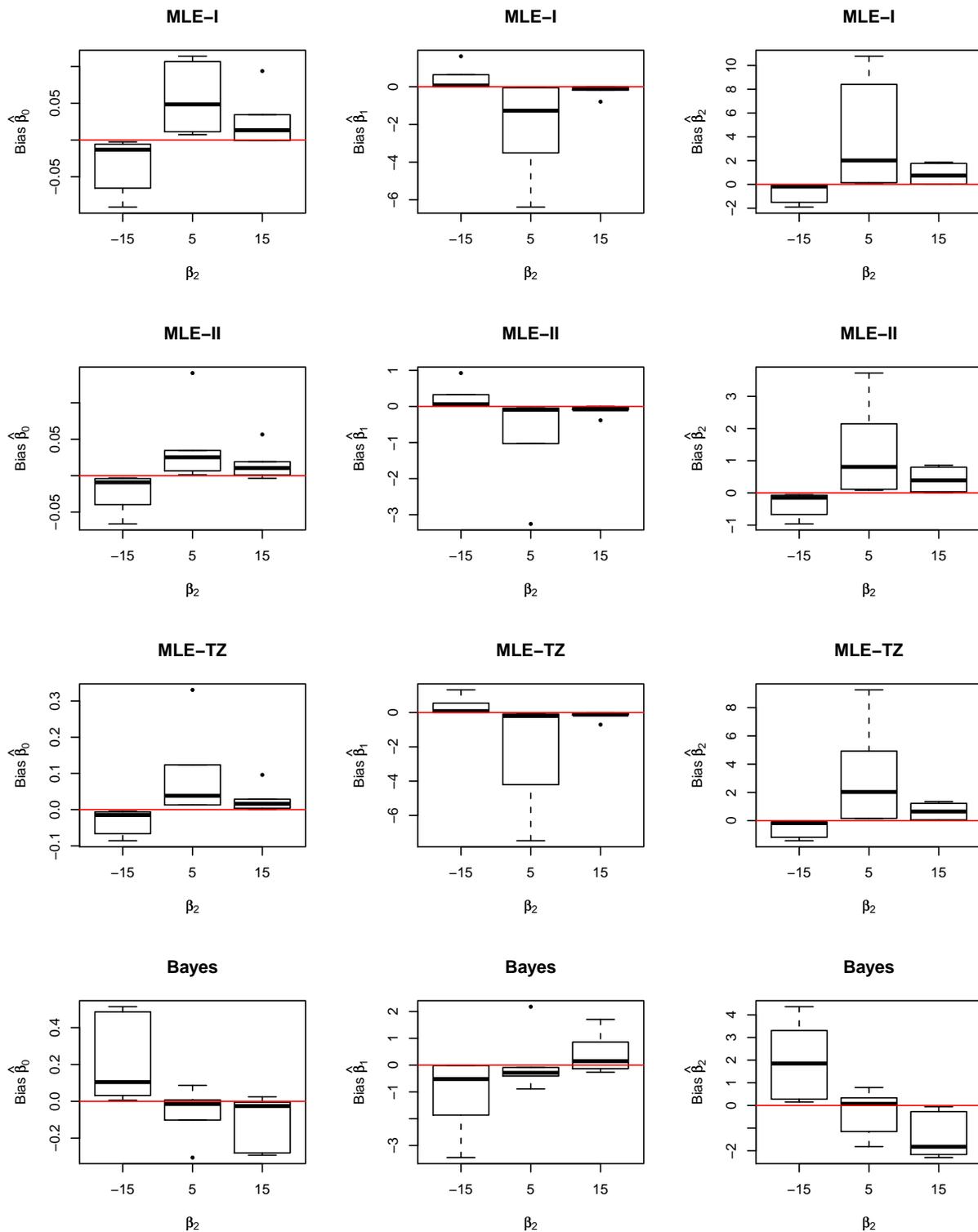


Figure 4.3: The effect β_2 has on bias of $\hat{\beta}_0$, $\hat{\beta}_1$, and $\hat{\beta}_2$ across estimators. Each row in this matrix of plots corresponds to one estimator, and each column corresponds to one response: $Bias(\hat{\beta}_0)$, $Bias(\hat{\beta}_1)$, or $Bias(\hat{\beta}_2)$.

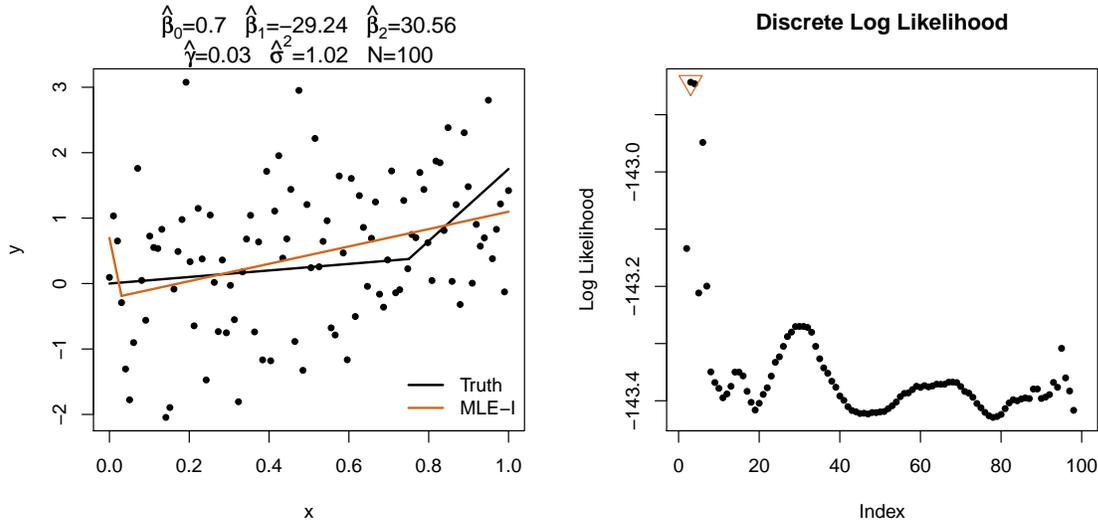


Figure 4.4: An illustration of a data set with large bias in its MLE-I estimates. On the left are the data and estimated regression lines for MLE-I for combination 17, data set 4. On the right is the discrete likelihood $L(m)$ of the data set, showing that the estimated regression lines do in fact maximize the discrete likelihood.

4.3 INDIVIDUAL ESTIMATOR PERFORMANCE

Maximum Likelihood Estimator-I

The raw biases and MSEs for MLE-I are included in Table B.1. It is hard to believe that such bias and variance could exist in this estimator. For those skeptical that such estimates can be correct, an illustration of the regression lines and the corresponding likelihood for one data set with seemingly impossible parameter estimates is shown in Figure 4.4.

Computationally, MLE-I was sufficiently fast for sample sizes 20, 50, and 100 (0.5 to 2 seconds), but the run time does increase linearly with n . No known problems occurred during the simulation with convergence of the constrained nonlinear optimization, regardless of starting values; however, several difficulties were encountered in applying MLE-I to the crack length data (presented in Chapter 5). Therefore, the following caution is issued: those wishing to implement MLE-I should be aware that computational difficulties may arise when parameters exist on vastly different scales, for instance when $0 < \gamma < 100$ and $0 < \sigma^2 < 0.001$. Some optimization algorithms have options to scale these parameters within the

routine. Also, in application the constrained maximization of the marginal likelihood had trouble converging with poorly defined starting values. This can be avoided by setting the starting values for the optimization of $L(m)$ equal to the ending values from the previous marginal likelihood optimization, $L(m - 1)$. See the code in Appendix C for an example of how to cascade these starting values.

Maximum Likelihood Estimator-II

This maximum likelihood estimator performed much better than expected. It had the smallest biases for $\hat{\gamma}$ and the smallest MSEs in general; however, as previously noted, $\hat{\sigma}^2$ is biased.

While accuracy of results is always desirable, sometimes getting a result that is “close enough” is preferable if the setup and computation time are greatly reduced. While MLE-II, which uses the numerically approximated gradient and Hessian, is clearly not “correct,” it does produce, for the most part, unbiased parameter estimates. It is significantly easier to set up and the overall run-time is much shorter than for the other estimators.

Tishler and Zang Maximum Likelihood Estimator

The Tishler and Zang maximum likelihood estimator was, by far, the most computationally expensive estimator of the four considered. Finding optimal B and p required iterative updates, and each update had a nonlinear optimization nested within it (see algorithm in Section 3.2). Additionally, very flat likelihoods made convergence in these optimization routines very slow. In the cases that Tishler and Zang [Tishler and Zang \(1981a\)](#) tested, $4 < p < 200$ was sufficient. However, in these simulations it was much more common to need $40 < p < 800$. With p starting at 20 and incrementing by $\Delta p = 20$ until successive estimates were within tolerance, most estimations took between 1 and 40 updates with a median of 9 required updates.

Some MLE-TZ parameter estimates were excluded from the analysis, either because of convergence problems or because the final $\hat{\gamma}$ s were outside of the $[0,1]$ range of data. Con-

Table 4.5: Number of parameter sets removed from bias and MSE calculations for MLE-TZ because of convergence problems or having $\hat{\gamma}$ outside of $[0,1]$ range.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
conv	0	28	4	1	105	3	0	1	124	90	1	4	6	18	6	4	21	33
$\hat{\gamma}$ range	0	124	0	0	0	0	0	0	0	0	0	0	0	12	0	0	17	0
used	1000	748	996	999	895	997	1000	999	876	910	999	996	994	970	994	967	962	967

vergence problems included cases where a warning or error was detected and the optimization routine aborted. They also include cases where the last iteration was within tolerance of the previous iterate but its validity was in question because the optimization routine reached its maximum number of iterations or possibly found a local minimum. The cases of convergence outside of the γ range occurred only in the hard-to-define cases 2, 14, and 17. In total, 602 parameter estimates were excluded from bias and MSE calculations of the MLE-TZs: 449 for convergence problems and 153 with $\hat{\gamma}$ outside of the $[0,1]$ range. Table 4.5 shows the breakdown of how many data sets were excluded from each size-parameter combination and how many were used to calculate bias and MSE.

Generally speaking, MLE-TZ performed as well as, if not better than, MLE-I but not as well as MLE-II or the Bayesian posterior mean. In most combinations, its estimates were unbiased and had MSEs that were similar to the other estimators. With other adequate estimators available that are easier to implement, this estimator is not recommended for general use.

Bayesian Posterior Mean

Raw biases and MSEs for the Bayesian posterior mean can be found in Table B.4. The most marked bias and MSE is seen in the estimate of σ^2 . The ANOVA shows that n has a significant effect on the bias and MSE of σ^2 (see Table 4.4), and plots of these raw effects support the assertion that bias and MSE decrease as sample size increases. The most likely cause for these undesirably large biases is poor prior specification for σ^2 . Each parameter combination had an underlying variance of either 0.1 or 1.0. Though the inverse gamma prior was very diffuse, its mean of 10 was too large for what was expected in the simulation.

An unexpectedly significant effect found in the ANOVA indicates that differing levels of γ lead to differing $\hat{\gamma}$ biases. To explore this effect, the posteriors of one data set per size-parameter combination were obtained and the marginal posteriors of γ were plotted (see Figure 4.5). The plots demonstrate that some of the posteriors are mildly skewed, which is biasing estimates of γ , specifically in the cases where the true γ is 0.25 or 0.75. Perhaps this biasing might be avoided by using the Bayesian posterior medians instead of means. Some cases, such as combinations 2, 5, 14, and 17, however, are hopelessly biased. For instance, the posterior of γ in combination 2 nearly returns the uniform prior, indicating that the data do not inform the posterior. In cases such as this, the posterior mean will be biased toward the uniform prior mean of 0.5. Another interesting γ posterior is the bimodal distribution of combination 5 with most of its mass at the boundaries, possibly indicating that a change-point is not necessary for the data.

It is theorized that the same effect that is causing bias in the γ posteriors may also be causing the bias in β_0 , β_1 , and β_2 estimators. Hopefully, these biases will be remedied by using the posterior median for these estimators as well.

4.4 SIMULATION CONSIDERATIONS

If this simulation study were to be expanded the following changes would be considered:

- Explore method of choosing B and incrementing p to make the MLE-TZ faster.
- Perform simulations with a design that allows for interactions to be explored.
- Determine how parameters interplay to result in ill-defined cases and define estimator limitations.
- Consider using joint measures of MSE or bias as responses.
- Perform a multivariate analysis on the responses.
- Incorporate robust techniques to down-weight ill defined cases and clarify factor effects.

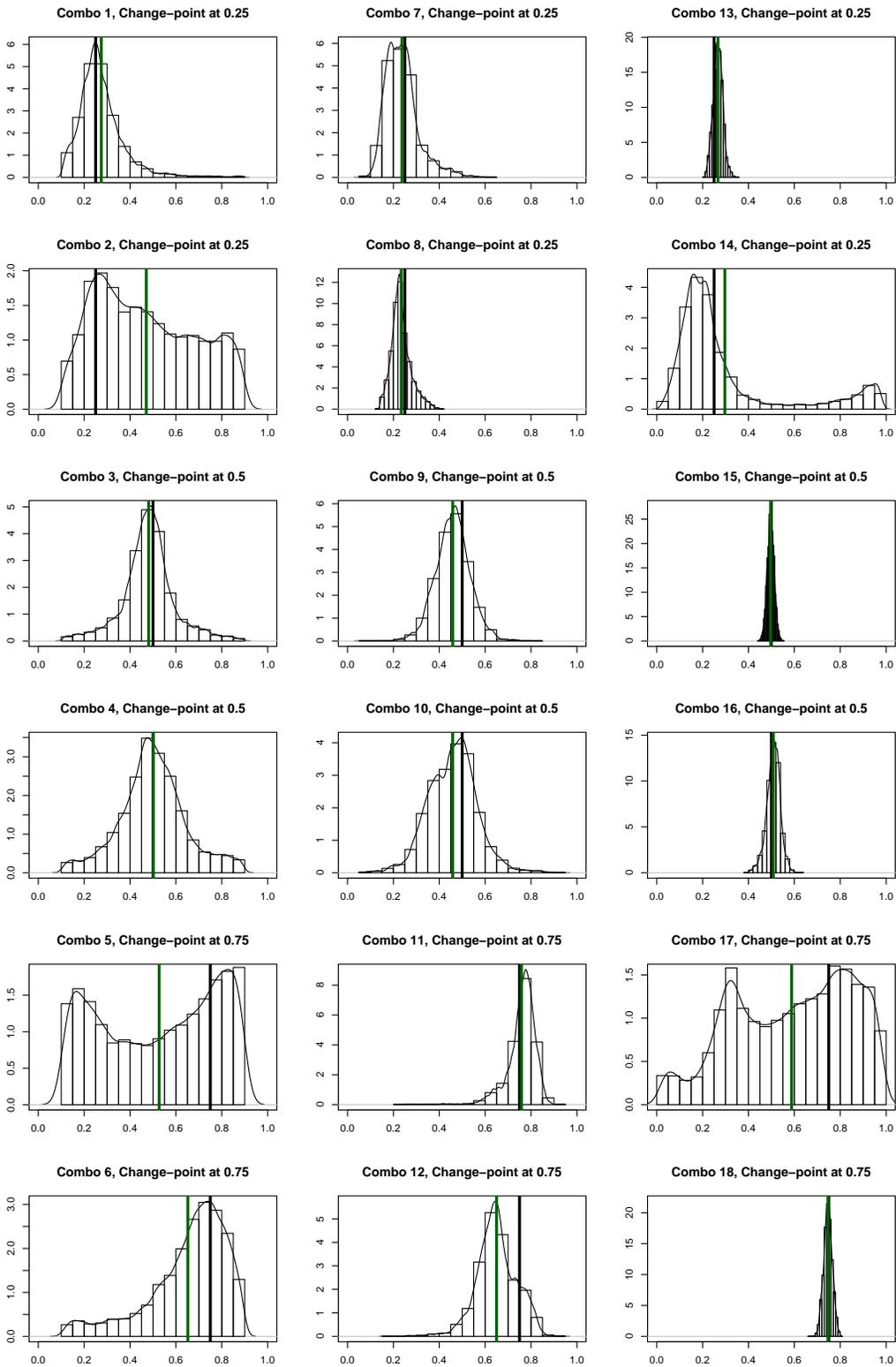


Figure 4.5: Posteriors on the change-point parameter for a single arbitrary data set for each combination. The solid green line indicates the posterior mean, and the solid black line is the true underlying regression change-point.

WEAPON CRACK LENGTH APPLICATION

5.1 INTRODUCTION AND DATA

In matters of national defense, having an accurate inventory of the weapons stockpile is imperative. While knowing “what” and “how much” are important, it is also critical to understand the quality and reliability of the weapons on hand. Where it is infeasible to dismantle a weapon and inspect it for signs of deterioration, techniques such as radiography are used to examine the interior. The data presented in this section are of crack length measurements on radiographs taken on weapons of varying ages. The cracks remain small for many years, but after a certain age the cracks begin to widen. By fitting a linear-linear segmented regression as a function of age, the threshold at which the cracks begin to grow may be found.

A plot of the data is included in Figure 5.1. Because of the sensitive nature of the data, the use of this data was contingent upon its being masked. Therefore, the ages represented in the graph are not true years. Additional modifications were made to the crack lengths to obfuscate the true defects. This is the reason there are negative crack lengths.

Table 5.1: Crack length data. Estimates for parameters in the linear-linear model obtained by fitting each of the four estimators.

	$\hat{\beta}_0$	$\hat{\beta}_1$	$\hat{\beta}_2$	$\hat{\gamma}$	$\hat{\sigma}^2$
MLE-I	0.007589	-0.001369	0.023193	17.417	0.001358
MLE-II	0.007844	-0.001400	0.023217	17.400	0.001358
MLE-TZ	0.007000	-0.001350	0.023268	17.467	0.001362
Bayes	0.000755	-0.000660	0.022819	18.078	0.006710



Figure 5.1: Masked crack length data from radiographs of weapons



Figure 5.2: Estimated regressions. Note that the three maximum likelihood estimates are very similar and MLE-TZ overwrites the MLE-I and MLE-II regression lines.

5.2 MAXIMUM LIKELIHOOD ESTIMATORS

While MLE-I and MLE-TZ are fairly robust to starting value specifications, MLE-II can be quite finicky and converge to any number of locations with poor starting specifications. Identical starting values were used for each maximum likelihood estimator and were obtained using yet another segmented regression estimation method (Muggeo 2003), available in R's `segmented` package and developed by Vito M. R. Muggeo.

The estimates for the crack length data are included in Table 5.1 and the estimated regression lines are plotted in Figure 5.2. The three maximum likelihood estimates were very similar, indicating a slope of nearly zero in the first segment, a change-point occurring at about 17.4 years, and a change in slope of about 0.023.

5.3 BAYESIAN MEAN

Diffuse priors were chosen for the β parameters, corresponding in most cases to the null case of zero slope or no change in slope. A normal prior for the change-point parameter, γ , centered at twenty with 99.7% of its probability in the $[0,40]$ interval gave weight to the foreknowledge that a change-point occurs within the midranges of the data. The prior on σ^2 was also chosen to be diffuse with $E(\sigma^2) \approx 0.099$ and $Var(\sigma^2) = 0.98$. The priors were as follows:

$$\begin{aligned}\beta_0 &\sim N(0, 10), \\ \beta_1 &\sim N(0, 10), \\ \beta_2 &\sim N(0, 10), \\ \gamma &\sim N(20, 44), \text{ and} \\ \sigma^2 &\sim IG(2.01, 10).\end{aligned}$$

With 110,000 draws from the posterior and a burn-in of 10,000 draws, the final posteriors were comprised of 100,000 draws. Figure 5.3 contains a plot of the marginal distribution for

each parameter along with 95% credible intervals. The means of the posteriors are included in Table 5.1 along with the maximum likelihood estimates. The Bayesian mean estimates for the intercept and first slope are very close to zero, the change-point estimate is 18.1, and the change in slope was estimated at 0.023. The Bayesian mean estimate for σ^2 was six times larger than the maximum likelihood estimates for σ^2 .

5.4 APPLICATION CONCLUSIONS

The masking of this weapons data, unfortunately, does not allow for accurate interpretation of the parameter estimates obtained. If this were representative of the original data, one could conclude that somewhere between year 17 and 18 the weapon reaches a critical threshold and begins to deteriorate at a faster rate.

Follow-up analyses, which are not within the scope of this paper, might include hypothesis testing, prediction, and calibration. For instance, it might be of interest to test if the first slope is significantly different than zero, to predict the crack length given the weapon's age, or to solve the inverse problem of estimating a particular age at which the cracks reach an unacceptable length.

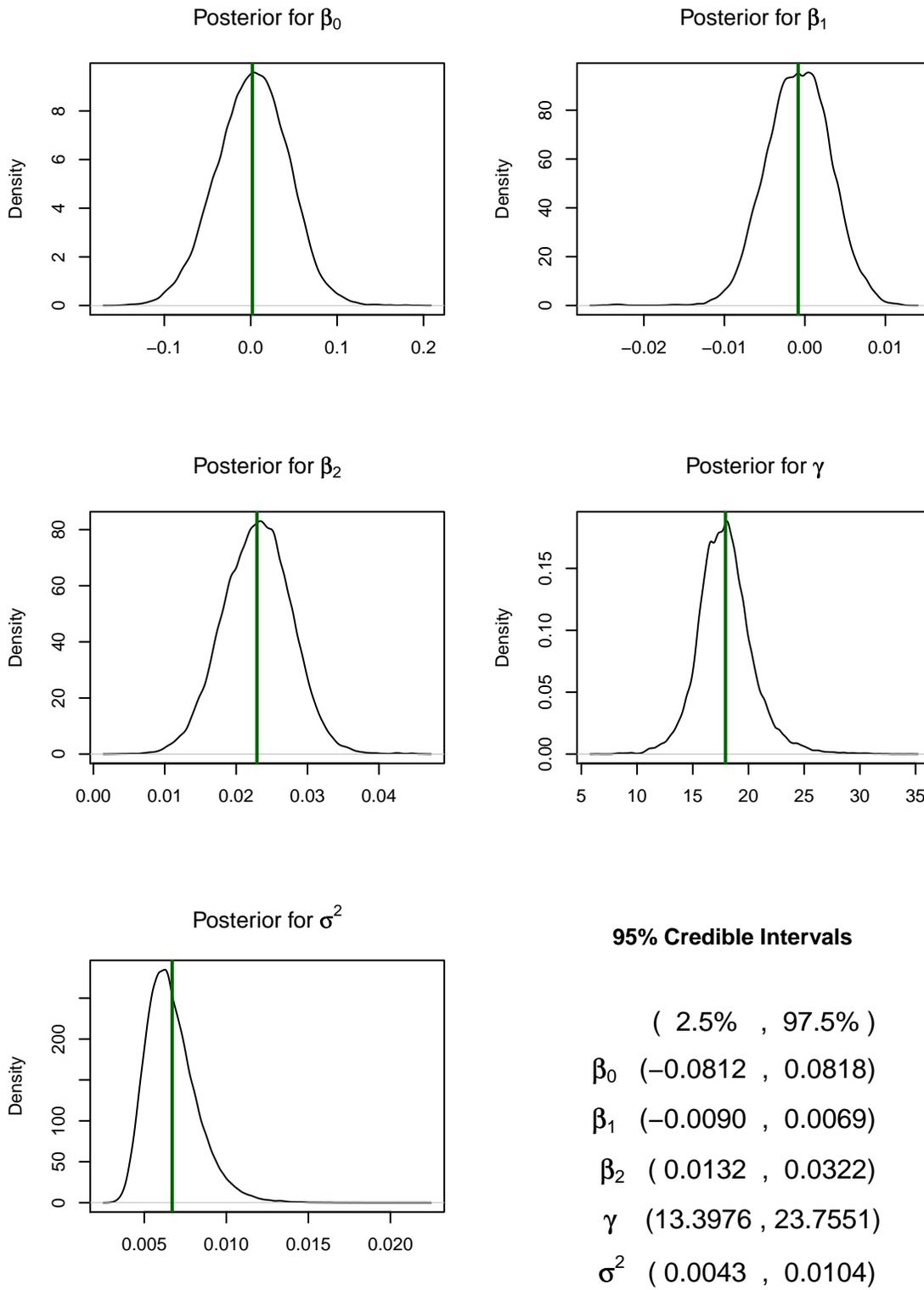


Figure 5.3: Marginal Bayesian posteriors. The bolded green lines are means.

CONCLUSION AND DISCUSSION

Parameter estimation in change-point regression is difficult when the change-point is unknown and also has to be estimated. Maximum likelihood methods are plagued by discontinuities in the gradient and Hessian, turning a seemingly easy nonlinear optimization problem into a problem fraught with complications. Parameter estimation in the segmented model defined by two linear segments constrained to meet at their joinpoint has been considered. Four methods for estimating the parameters of this linear-linear model have been evaluated via simulation. All four methods had difficulty in estimating cases that were poorly defined, that is, cases that had small changes in slope relative to the variance present in the data or they had little data in either regime by which to evaluate the change in slope. These cases produced estimates with high bias and MSE among all estimators.

All four estimators were flawed in some way or another, and so no estimator can be claimed as universally best. MLE-I acknowledges the discontinuities in the derivative and so appeals to a sense of “correctness” for maximizing the likelihood. It performed on-par with the other estimators in the well-defined cases but was the worst of the estimators in terms of bias and MSE for the poorly defined cases. It is not recommended for use if one suspects the change-point is going to be difficult to detect.

MLE-II was easy to implement and had the smallest bias in the change-point parameter estimate. It is definitely a contender for parameter estimation when the change-point is of primary interest. However, as discovered in the application, the algorithm can converge to wildly different values with poor starting values. By starting the algorithm in multiple locations and using the parameter combination corresponding to the maximum likelihood value, MLE-II can still be used to estimate the parameters of the linear-linear model. For

one willing to invest a little time playing around with starting values, this estimator will pay off in the long run since it requires so little setup. It should be noted that estimates of σ^2 in this estimator are biased, and tests performed using this estimate may be adversely affected.

MLE-TZ was the most computationally expensive of the estimators, both in setup time and in run time. While it performed on-par with the other estimators when it did work, its performance was unreliable.

The Bayesian mean estimator was slightly more robust in difficult-to-estimate cases and would likely become even more stable if informative priors were used instead of diffuse priors. Additionally, using posterior medians as point estimates might reduce the bias and thereby the MSE of the Bayesian estimators. As stated previously, the Bayesian estimator completely avoids the challenge inherent in maximum likelihood methods of maximizing a nonsmooth likelihood. Another advantage is that credible intervals on the parameter estimates are easily obtained. Lastly, because entire probability distributions are assigned to the parameter, it is possible to see if the probability that the change-point is near the boundaries is large, possibly indicating that a change-point is not necessary at all.

Overall, the two favored estimators are MLE-II and the Bayesian estimator. MLE-II can be used effectively and simply in well-defined cases where the only objective is parameter estimation. The Bayesian estimator is preferred when interval estimates are desired for the estimates and in hard-to-estimate cases that can be aided by the injection of prior information into the system.

BIBLIOGRAPHY

- Bacon, D. W., and Watts, D. G. (1971), “Estimating the Transition between Two Intersecting Straight Lines,” *Biometrika*, 58, 525–534.
- Blichske, W. (1961), “Least Squares Estimators of Two Intersecting Lines,” *Technical Report No. 7, Contract N. Nonr-401 (39)*.
- Carlin, B. P., Gefland, A. E., and Smith, A. F. M. (1992), “Hierarchical Bayesian Analysis of Change-point Problems,” *Applied Statistics*, 41, 389–405.
- Ferreira, P. E. (1975), “A Bayesian Analysis of a Switching Regression Model: Known Number of Regimes,” *Journal of the American Statistical Association*, 70, 370–374.
- Gallant, A. R., and Fuller, W. A. (1973), “Fitting Segmented Polynomial Regression Models Whose Join Points Have to Be Estimated,” *Journal of the American Statistical Association*, 68, 144–147.
- Gefland, A. E., and Smith, A. F. M. (1990), “Sampling-Based Approaches to Calculating Marginal Densities,” *Journal of the American Statistical Association*, 85, 398–409.
- Gelman, A., Carlin, J., Stern, H., and Rubin, D. (2004), *Bayesian Data Analysis* (2nd ed.), New York: Chapman and Hall, 2nd ed.
- Geman, S., and Geman, D. (1984), “Stochastic Relaxation, Gibbs Distribution and the Bayesian Restoration of Images,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6, 721–741.

- Griffiths, D. A., and Miller, A. J. (1973), “Hyperbolic Regression — A Model Based on Two-Phase Piecewise Linear Regression with a Smooth Transition Between Regimes,” *Communications in Statistics — Simulation and Computation*, 2, 561–569.
- Hastings, W. K. (1970), “Monte Carlo Sampling Methods Using Markov Chains and Their Applications,” *Biometrika*, 57, 97–109.
- Hinkley, D. V. (1969), “Inference about the Intersection in Two-Phase Regression,” *Biometrika*, 56, 495–504.
- Hudson, D. J. (1966), “Fitting Segmented Curves Whose Join Points Have to be Estimated,” *Journal of the American Statistical Association*, 61, 1097–1129.
- Lerman, P. M. (1980), “Fitting Segmented Regression Models by Grid Search,” *Applied Statistics*, 29, 77–84.
- Muggeo, V. M. R. (2003), “Estimating Regression Models with Unknown Break-points,” *Statistics in Medicine*, 22, 3055–3071.
- Page, E. (1954), “Continuous Inspection Schemes,” *Biometrika*, 41, 100–114.
- Poirier, D. J. (1976), *The Econometrics of Structural Change*, Amsterdam: North-Holland Publishing Company.
- Quandt, R. E. (1958), “The Estimation of the Parameters of a Linear Regression System Obeying Two Separate Regimes,” *Journal of the American Statistical Association*, 53, 873–880.
- R Development Core Team (2009), *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Vienna, Austria, ISBN 3-900051-07-0.
- Robison, D. (1964), “Estimates for the Points of Intersection of Two Polynomial Regressions,” *Journal of the American Statistical Association*, 59, 214–224.

- Seber, G. A. G., and Wild, C. J. (1989), *Nonlinear Regression*, New York: John Wiley and Sons, Inc.
- Smith, A. F. M., and Cook, D. G. (1980), “Straight Lines with a Change-point: A Bayesian Analysis of Some Renal Transplant Data,” *Applied Statistics*, 29, 180–189.
- Sprenst, P. (1961), “Some Hypotheses Concerning Two Phase Regression Lines,” *Biometrics*, 17, 634–645.
- Stephens, D. A. (1994), “Bayesian Retrospective Multiple-Change-point Identification,” *Applied Statistics*, 43, 159–178.
- Tishler, A., and Zang, I. (1981a), “A Maximum Likelihood Method for Piecewise Regression Models with a Continuous Dependent Variable,” *Applied Statistics*, 30, 116–124.
- (1981b), “A New Maximum Likelihood Algorithm for Piecewise Regression,” *Journal of the American Statistical Association*, 76, 980–987.

APPENDICES

ESTIMATOR DETAIL

This section includes supplementary detail in the derivation of the estimators.

A.1 MAXIMUM LIKELIHOOD ESTIMATOR-I

Under the assumption that there exists some integer m such that $x_m \leq \gamma < x_{m+1}$, the joint log-likelihood can be written as

$$\begin{aligned} \mathcal{L}(\boldsymbol{\beta}, \gamma, \sigma^2 | \mathbf{x}, \mathbf{y}) &= -\frac{n}{2} \log(2\pi) - \frac{n}{2} \log(\sigma^2) - \frac{1}{2\sigma^2} \sum_{i=1}^m \left\{ y_i - [\beta_0 + \beta_1 x_i] \right\}^2 \\ &\quad - \frac{1}{2\sigma^2} \sum_{i=m+1}^n \left\{ y_i - [\beta_0 + \beta_1 x_i + \beta_2(x_i - \gamma)] \right\}^2. \end{aligned}$$

The first partial derivatives of gradient are then

$$\begin{aligned} \frac{\delta \mathcal{L}}{\delta \beta_0} &= \frac{1}{\sigma^2} \left\{ \sum_{i=1}^m [y_i - (\beta_0 + \beta x_i)] + \sum_{i=m+1}^n [y_i - (\beta_0 + \beta x_i + \beta_2(x_i - \gamma))] \right\} \\ \frac{\delta \mathcal{L}}{\delta \beta_1} &= \frac{1}{\sigma^2} \left\{ \sum_{i=1}^m x_i [y_i - (\beta_0 + \beta x_i)] + \sum_{i=m+1}^n x_i [y_i - (\beta_0 + \beta x_i + \beta_2(x_i - \gamma))] \right\} \\ \frac{\delta \mathcal{L}}{\delta \beta_2} &= \frac{1}{\sigma^2} \sum_{i=m+1}^n x_i [y_i - (\beta_0 + \beta x_i + \beta_2(x_i - \gamma))] \\ \frac{\delta \mathcal{L}}{\delta \gamma} &= -\frac{1}{\sigma^2} \sum_{i=m+1}^n \beta_2 [y_i - (\beta_0 + \beta x_i + \beta_2(x_i - \gamma))] \\ \frac{\delta \mathcal{L}}{\delta \sigma^2} &= -\frac{n}{2\sigma^2} + \frac{1}{2(\sigma^2)^2} \left\{ \sum_{i=1}^m [y_i - (\beta_0 + \beta_1 x_i)]^2 + \sum_{i=m+1}^n [y_i - (\beta_0 + \beta_1 x_i + \beta_2(x_i - \gamma))]^2 \right\}. \end{aligned}$$

The second partial derivatives that comprise the Hessian are

$$\begin{aligned}
\frac{\delta^2 \mathcal{L}}{\delta \beta_0^2} &= -\frac{n}{\sigma^2} \\
\frac{\delta^2 \mathcal{L}}{\delta \beta_0 \delta \beta_1} &= -\frac{1}{\sigma^2} \sum_{i=1}^n x_i \\
\frac{\delta^2 \mathcal{L}}{\delta \beta_0 \delta \beta_2} &= -\frac{1}{\sigma^2} \sum_{i=m+1}^n (x_i - \gamma) \\
\frac{\delta^2 \mathcal{L}}{\delta \beta_0 \delta \gamma} &= -\frac{1}{\sigma^2} \beta_2 (n - m) \\
\frac{\delta^2 \mathcal{L}}{\delta \beta_0 \delta (\sigma^2)} &= -\frac{1}{(\sigma^2)^2} \left\{ \sum_{i=1}^m [y_i - (\beta_0 + \beta_1 x_i)] + \sum_{i=m+1}^n [y_i - (\beta_0 + \beta_1 x_i + \beta_2 (x_i - \gamma))] \right\} \\
\frac{\delta^2 \mathcal{L}}{\delta \beta_1^2} &= -\frac{1}{\sigma^2} \sum_{i=1}^n x_i^2 \\
\frac{\delta^2 \mathcal{L}}{\delta \beta_1 \delta \beta_2} &= -\frac{1}{\sigma^2} \sum_{i=m+1}^n x_i (x_i - \gamma) \\
\frac{\delta^2 \mathcal{L}}{\delta \beta_1 \delta \gamma} &= -\frac{1}{\sigma^2} \sum_{i=m+1}^n \beta_2 x_i \\
\frac{\delta^2 \mathcal{L}}{\delta \beta_1 \delta (\sigma^2)} &= -\frac{1}{(\sigma^2)^2} \left\{ \sum_{i=1}^m x_i [y_i - (\beta_0 + \beta_1 x_i)] + \sum_{i=m+1}^n x_i [y_i - (\beta_0 + \beta_1 x_i + \beta_2 (x_i - \gamma))] \right\} \\
\frac{\delta^2 \mathcal{L}}{\delta \beta_2^2} &= -\frac{1}{\sigma^2} \sum_{i=m+1}^n x_i (x_i - \gamma) \\
\frac{\delta^2 \mathcal{L}}{\delta \beta_2 \delta \gamma} &= -\frac{1}{\sigma^2} \sum_{i=m+1}^n \beta_2 x_i \\
\frac{\delta^2 \mathcal{L}}{\delta \beta_2 \delta (\sigma^2)} &= -\frac{1}{(\sigma^2)^2} \sum_{i=m+1}^n x_i [y_i - (\beta_0 + \beta_1 x_i + \beta_2 (x_i - \gamma))] \\
\frac{\delta^2 \mathcal{L}}{\delta \gamma^2} &= -\frac{1}{\sigma^2} \beta_2^2 (n - m) \\
\frac{\delta^2 \mathcal{L}}{\delta \gamma \delta (\sigma^2)} &= -\frac{1}{(\sigma^2)^2} \sum_{i=m+1}^n \beta_2 [y_i - (\beta_0 + \beta_1 x_i + \beta_2 (x_i - \gamma))] \\
\frac{\delta^2 \mathcal{L}}{\delta (\sigma^2)^2} &= \frac{n}{2(\sigma^2)^2} - \frac{1}{(\sigma^2)^3} \left\{ \sum_{i=1}^m [y_i - (\beta_0 + \beta_1 x_i)]^2 + \sum_{i=m+1}^n [y_i - (\beta_0 + \beta_1 x_i + \beta_2 (x_i - \gamma))]^2 \right\}.
\end{aligned}$$

A.2 TISHLER AND ZANG MAXIMUM LIKELIHOOD ESTIMATOR

First partial derivatives of the gradient with respect to the parameterization of two intersecting lines:

$$\delta_i = \max\{|B + \mathbf{a}'_1 x_{1i}|, |B + \mathbf{a}'_2 x_{2i}|\} \text{ for } i = 1, 2, \dots, n$$

$$\begin{aligned} \frac{\delta \mathcal{L}}{\delta a_{10}} &= -\frac{1}{\sigma^2} \sum_{i=1}^n \left\{ B + y_i - \delta_i \left[\left(\frac{B + a_{10} + a_{11}x_{1i}}{\delta_i} \right)^p + \left(\frac{B + a_{20} + a_{21}x_{2i}}{\delta_i} \right)^p \right]^{1/p} \right\} \\ &\quad \times \left[\left(\frac{B + a_{10} + a_{11}x_{1i}}{\delta_i} \right)^p + \left(\frac{B + a_{20} + a_{21}x_{2i}}{\delta_i} \right)^p \right]^{1/p-1} \times \left(\frac{B + a_{10} + a_{11}x_{1i}}{\delta_i} \right)^p \end{aligned}$$

$$\begin{aligned} \frac{\delta \mathcal{L}}{\delta a_{11}} &= -\frac{1}{\sigma^2} \sum_{i=1}^n \left\{ B + y_i - \delta_i \left[\left(\frac{B + a_{10} + a_{11}x_{1i}}{\delta_i} \right)^p + \left(\frac{B + a_{20} + a_{21}x_{2i}}{\delta_i} \right)^p \right]^{1/p} \right\} \\ &\quad \times \left[\left(\frac{B + a_{10} + a_{11}x_{1i}}{\delta_i} \right)^p + \left(\frac{B + a_{20} + a_{21}x_{2i}}{\delta_i} \right)^p \right]^{1/p-1} \times \left(\frac{B + a_{10} + a_{11}x_{1i}}{\delta_i} \right)^p \times x_{1i} \end{aligned}$$

$$\begin{aligned} \frac{\delta \mathcal{L}}{\delta a_{20}} &= -\frac{1}{\sigma^2} \sum_{i=1}^n \left\{ B + y_i - \delta_i \left[\left(\frac{B + a_{10} + a_{11}x_{1i}}{\delta_i} \right)^p + \left(\frac{B + a_{20} + a_{21}x_{2i}}{\delta_i} \right)^p \right]^{1/p} \right\} \\ &\quad \times \left[\left(\frac{B + a_{10} + a_{11}x_{1i}}{\delta_i} \right)^p + \left(\frac{B + a_{20} + a_{21}x_{2i}}{\delta_i} \right)^p \right]^{1/p-1} \times \left(\frac{B + a_{20} + a_{21}x_{2i}}{\delta_i} \right)^p \end{aligned}$$

$$\begin{aligned} \frac{\delta \mathcal{L}}{\delta a_{21}} &= -\frac{1}{\sigma^2} \sum_{i=1}^n \left\{ B + y_i - \delta_i \left[\left(\frac{B + a_{10} + a_{11}x_{1i}}{\delta_i} \right)^p + \left(\frac{B + a_{20} + a_{21}x_{2i}}{\delta_i} \right)^p \right]^{1/p} \right\} \\ &\quad \times \left[\left(\frac{B + a_{10} + a_{11}x_{1i}}{\delta_i} \right)^p + \left(\frac{B + a_{20} + a_{21}x_{2i}}{\delta_i} \right)^p \right]^{1/p-1} \times \left(\frac{B + a_{20} + a_{21}x_{2i}}{\delta_i} \right)^p \times x_{2i} \end{aligned}$$

$$\frac{\delta \mathcal{L}}{\delta \sigma^2} = -\frac{n}{2\sigma^2} + \frac{1}{(\sigma^2)^2} \sum_{i=1}^n \left\{ B + y_i - \delta_i \left[\left(\frac{B + a_{10} + a_{11}x_{1i}}{\delta_i} \right)^p + \left(\frac{B + a_{20} + a_{21}x_{2i}}{\delta_i} \right)^p \right]^{1/p} \right\}^2$$

APPENDIX B

SIMULATION RESULTS

Table B.1: Estimated biases and MSEs for MLE-I.

	Parameter Combination						Estimated Bias				
	n	β_0	β_1	β_2	γ	σ^2	$\hat{\beta}_0$	$\hat{\beta}_1$	$\hat{\beta}_2$	$\hat{\gamma}$	$\hat{\sigma}^2$
1	20	0	10.0	15	0.25	0.1	-0.00073	-0.02531	0.04986	0.00274	-0.02043
2	20	0	0.5	5	0.25	1.0	0.06301	-2.24271	3.07828	0.18057	-0.23450
3	20	0	0.5	15	0.50	1.0	0.09382	-0.79228	1.45019	-0.00296	-0.19650
4	20	0	10.0	-15	0.50	1.0	-0.06548	0.63807	-1.50922	0.00343	-0.19192
5	20	0	10.0	5	0.75	0.1	0.03389	-0.29063	0.94448	-0.03129	-0.02387
6	20	0	0.5	-15	0.75	0.1	-0.00568	0.01628	-0.17098	-0.00142	-0.02159
7	50	0	10.0	-15	0.25	1.0	-0.09137	1.61860	-1.90660	0.00664	-0.07501
8	50	0	0.5	-15	0.25	0.1	-0.00830	0.04669	-0.06787	0.00132	-0.00774
9	50	0	10.0	5	0.50	0.1	0.00730	-0.04808	0.10376	0.00007	-0.00781
10	50	0	0.5	5	0.50	0.1	0.01125	-0.05518	0.14168	0.00218	-0.00781
11	50	0	10.0	15	0.75	1.0	0.02626	-0.14079	1.76713	-0.00581	-0.08368
12	50	0	0.5	15	0.75	1.0	0.03445	-0.18385	1.86339	-0.00789	-0.08114
13	100	0	0.5	15	0.25	0.1	-0.00081	-0.01759	0.03285	0.00048	-0.00373
14	100	0	10.0	5	0.25	1.0	0.11397	-6.38787	8.40780	0.07902	-0.04332
15	100	0	10.0	15	0.50	0.1	0.00040	0.00757	-0.00080	0.00060	-0.00371
16	100	0	0.5	-15	0.50	1.0	-0.01793	0.08080	-0.18895	0.00091	-0.04496
17	100	0	0.5	5	0.75	1.0	0.10669	-3.50856	10.77104	-0.07926	-0.04777
18	100	0	10.0	-15	0.75	0.1	-0.00264	0.00824	-0.06640	0.00000	-0.00377

	Parameter Combination						Estimated MSE				
	n	β_0	β_1	β_2	γ	σ^2	$\hat{\beta}_0$	$\hat{\beta}_1$	$\hat{\beta}_2$	$\hat{\gamma}$	$\hat{\sigma}^2$
1	20	0	10.0	15	0.25	0.1	0.05609	3.555	3.532	0.00062	0.00120
2	20	0	0.5	5	0.25	1.0	0.83042	145.147	260.463	0.12102	0.13834
3	20	0	0.5	15	0.50	1.0	0.38613	14.296	21.433	0.00903	0.12112
4	20	0	10.0	-15	0.50	1.0	0.41247	13.195	23.746	0.00961	0.11241
5	20	0	10.0	5	0.75	0.1	0.03539	2.619	11.476	0.01936	0.00129
6	20	0	0.5	-15	0.75	0.1	0.02429	0.137	3.969	0.00064	0.00126
7	50	0	10.0	-15	0.25	1.0	0.36225	58.289	82.183	0.00574	0.04465
8	50	0	0.5	-15	0.25	0.1	0.02898	1.627	1.634	0.00025	0.00043
9	50	0	10.0	5	0.50	0.1	0.01707	0.236	0.410	0.00191	0.00043
10	50	0	0.5	5	0.50	0.1	0.01704	0.243	0.393	0.00216	0.00043
11	50	0	10.0	15	0.75	1.0	0.11978	0.851	87.363	0.00503	0.04295
12	50	0	0.5	15	0.75	1.0	0.11994	1.365	93.226	0.00617	0.04321
13	100	0	0.5	15	0.25	0.1	0.01449	0.732	0.734	0.00011	0.00020
14	100	0	10.0	5	0.25	1.0	0.39314	1258.539	1871.910	0.06549	0.02000
15	100	0	10.0	15	0.50	0.1	0.00751	0.096	0.192	0.00007	0.00021
16	100	0	0.5	-15	0.50	1.0	0.07919	1.070	1.822	0.00108	0.01965
17	100	0	0.5	5	0.75	1.0	0.32654	1473.529	2949.487	0.06836	0.02133
18	100	0	10.0	-15	0.75	0.1	0.00539	0.030	0.831	0.00011	0.00020

Table B.2: Estimated biases and MSEs for MLE-II.

	Parameter Combination						Estimated Bias				
	n	β_0	β_1	β_2	γ	σ^2	$\hat{\beta}_0$	$\hat{\beta}_1$	$\hat{\beta}_2$	$\hat{\gamma}$	$\hat{\sigma}^2$
1	20	0	10.0	15	0.25	0.1	0.00308	-0.07672	0.08697	0.00111	-0.01850
2	20	0	0.5	5	0.25	1.0	0.14079	-3.25518	3.72340	-0.01222	-0.18675
3	20	0	0.5	15	0.50	1.0	0.05658	-0.38289	0.69311	-0.00116	-0.18402
4	20	0	10.0	-15	0.50	1.0	-0.03979	0.32563	-0.67018	-0.00049	-0.17778
5	20	0	10.0	5	0.75	0.1	0.02195	-0.05614	0.48894	0.00023	0.04962
6	20	0	0.5	-15	0.75	0.1	-0.00461	0.01233	-0.14537	-0.00114	-0.01748
7	50	0	10.0	-15	0.25	1.0	-0.06626	0.92469	-0.96351	0.00074	-0.06945
8	50	0	0.5	-15	0.25	0.1	-0.00402	0.05942	-0.07358	0.00044	0.05479
9	50	0	10.0	5	0.50	0.1	0.00118	-0.03583	0.08167	-0.00097	0.01699
10	50	0	0.5	5	0.50	0.1	0.00666	-0.04620	0.11319	0.00082	0.03402
11	50	0	10.0	15	0.75	1.0	0.01798	-0.09281	0.79860	-0.00269	-0.07839
12	50	0	0.5	15	0.75	1.0	0.01920	-0.08430	0.85774	-0.00159	-0.07528
13	100	0	0.5	15	0.25	0.1	-0.00364	-0.02564	0.03617	-0.00028	0.01332
14	100	0	10.0	5	0.25	1.0	0.03465	-1.02776	1.12896	-0.00498	-0.03467
15	100	0	10.0	15	0.50	0.1	0.00077	0.00397	-0.00065	0.00026	-0.00207
16	100	0	0.5	-15	0.50	1.0	-0.01372	0.06274	-0.15363	0.00073	-0.04372
17	100	0	0.5	5	0.75	1.0	0.02857	-0.12458	2.14725	0.00297	-0.04061
18	100	0	10.0	-15	0.75	0.1	-0.00300	0.00974	-0.05836	-0.00015	-0.00374

	Parameter Combination						Estimated MSE				
	n	β_0	β_1	β_2	γ	σ^2	$\hat{\beta}_0$	$\hat{\beta}_1$	$\hat{\beta}_2$	$\hat{\gamma}$	$\hat{\sigma}^2$
1	20	0	10.0	15	0.25	0.1	0.05294	3.116	3.164	0.00054	0.00249
2	20	0	0.5	5	0.25	1.0	0.49394	97.581	102.222	0.02651	0.12962
3	20	0	0.5	15	0.50	1.0	0.31431	4.883	8.728	0.00495	0.12065
4	20	0	10.0	-15	0.50	1.0	0.33235	5.063	9.013	0.00477	0.11201
5	20	0	10.0	5	0.75	0.1	0.03478	0.156	4.996	0.00539	0.61712
6	20	0	0.5	-15	0.75	0.1	0.02394	0.132	3.517	0.00052	0.01612
7	50	0	10.0	-15	0.25	1.0	0.27011	21.029	20.754	0.00256	0.04450
8	50	0	0.5	-15	0.25	0.1	0.04176	1.474	1.495	0.00035	0.63030
9	50	0	10.0	5	0.50	0.1	0.02367	0.202	0.393	0.00261	0.16755
10	50	0	0.5	5	0.50	0.1	0.02320	0.204	0.375	0.00290	0.55289
11	50	0	10.0	15	0.75	1.0	0.11106	0.636	16.649	0.00237	0.04258
12	50	0	0.5	15	0.75	1.0	0.10523	0.685	31.369	0.00228	0.04263
13	100	0	0.5	15	0.25	0.1	0.02136	0.669	0.673	0.00019	0.20961
14	100	0	10.0	5	0.25	1.0	0.12494	64.899	65.067	0.00830	0.01968
15	100	0	10.0	15	0.50	0.1	0.00741	0.093	0.188	0.00009	0.00151
16	100	0	0.5	-15	0.50	1.0	0.07265	0.911	1.652	0.00096	0.01962
17	100	0	0.5	5	0.75	1.0	0.05115	0.346	152.462	0.00942	0.02105
18	100	0	10.0	-15	0.75	0.1	0.00535	0.030	0.782	0.00010	0.00020

Table B.3: Estimated biases and MSEs for MLE-TZ.

	Parameter Combination						Estimated Bias				
	n	β_0	β_1	β_2	γ	σ^2	$\hat{\beta}_0$	$\hat{\beta}_1$	$\hat{\beta}_2$	$\hat{\gamma}$	$\hat{\sigma}^2$
1	20	0	10.0	15	0.25	0.1	0.00465	-0.11130	0.13906	0.00195	-0.02022
2	20	0	0.5	5	0.25	1.0	0.33057	-7.47655	9.26016	0.06161	-0.21496
3	20	0	0.5	15	0.50	1.0	0.09604	-0.70296	1.15343	-0.00432	-0.19454
4	20	0	10.0	-15	0.50	1.0	-0.06632	0.55026	-1.17713	0.00183	-0.18966
5	20	0	10.0	5	0.75	0.1	0.03146	-0.18360	1.17538	-0.02448	-0.02212
6	20	0	0.5	-15	0.75	0.1	-0.00632	0.02023	-0.15480	-0.00166	-0.02156
7	50	0	10.0	-15	0.25	1.0	-0.08583	1.32084	-1.42364	0.00526	-0.07392
8	50	0	0.5	-15	0.25	0.1	-0.01060	0.07568	-0.09978	0.00111	-0.00767
9	50	0	10.0	5	0.50	0.1	0.01302	-0.08285	0.16184	-0.00075	-0.00677
10	50	0	0.5	5	0.50	0.1	0.01278	-0.06256	0.15643	0.00207	-0.00745
11	50	0	10.0	15	0.75	1.0	0.02734	-0.13888	1.34547	-0.00576	-0.08175
12	50	0	0.5	15	0.75	1.0	0.02886	-0.13457	1.22569	-0.00537	-0.07914
13	100	0	0.5	15	0.25	0.1	0.00080	-0.03755	0.05691	0.00040	-0.00363
14	100	0	10.0	5	0.25	1.0	0.12367	-4.20577	4.92756	0.04506	-0.04152
15	100	0	10.0	15	0.50	0.1	0.00384	-0.01420	0.03894	0.00050	-0.00366
16	100	0	0.5	-15	0.50	1.0	-0.01912	0.08764	-0.20254	0.00090	-0.04420
17	100	0	0.5	5	0.75	1.0	0.04501	-0.21718	2.89042	-0.01425	-0.04269
18	100	0	10.0	-15	0.75	0.1	-0.00472	0.01620	-0.09972	-0.00004	-0.00364

	Parameter Combination						Estimated MSE				
	n	β_0	β_1	β_2	γ	σ^2	$\hat{\beta}_0$	$\hat{\beta}_1$	$\hat{\beta}_2$	$\hat{\gamma}$	$\hat{\sigma}^2$
1	20	0	10.0	15	0.25	0.1	0.05791	3.872	3.822	0.00064	0.00120
2	20	0	0.5	5	0.25	1.0	0.58020	349.478	382.963	0.05016	0.13558
3	20	0	0.5	15	0.50	1.0	0.37035	9.300	12.422	0.00729	0.12150
4	20	0	10.0	-15	0.50	1.0	0.38566	9.525	13.477	0.00753	0.11275
5	20	0	10.0	5	0.75	0.1	0.03139	0.495	25.816	0.01503	0.00123
6	20	0	0.5	-15	0.75	0.1	0.02418	0.135	3.816	0.00059	0.00126
7	50	0	10.0	-15	0.25	1.0	0.32977	36.457	34.751	0.00431	0.04454
8	50	0	0.5	-15	0.25	0.1	0.02916	1.653	1.659	0.00025	0.00043
9	50	0	10.0	5	0.50	0.1	0.01704	0.229	0.421	0.00169	0.00043
10	50	0	0.5	5	0.50	0.1	0.01713	0.243	0.401	0.00208	0.00044
11	50	0	10.0	15	0.75	1.0	0.11583	0.738	31.405	0.00420	0.04271
12	50	0	0.5	15	0.75	1.0	0.11025	0.849	32.320	0.00404	0.04316
13	100	0	0.5	15	0.25	0.1	0.01457	0.741	0.742	0.00011	0.00020
14	100	0	10.0	5	0.25	1.0	0.25377	326.147	338.816	0.03334	0.01999
15	100	0	10.0	15	0.50	0.1	0.00754	0.096	0.193	0.00007	0.00021
16	100	0	0.5	-15	0.50	1.0	0.07924	1.072	1.826	0.00108	0.01965
17	100	0	0.5	5	0.75	1.0	0.05545	0.614	107.840	0.01494	0.02107
18	100	0	10.0	-15	0.75	0.1	0.00542	0.030	0.842	0.00011	0.00020

Table B.4: Estimated biases and MSEs for Bayesian mean.

	Parameter Combination						Estimated Bias				
	n	β_0	β_1	β_2	γ	σ^2	$\hat{\beta}_0$	$\hat{\beta}_1$	$\hat{\beta}_2$	$\hat{\gamma}$	$\hat{\sigma}^2$
1	20	0	10.0	15	0.25	0.1	-0.28016	1.70615	-1.79616	0.03278	1.08633
2	20	0	0.5	5	0.25	1.0	-0.30536	2.18130	-1.81818	0.24127	0.92707
3	20	0	0.5	15	0.50	1.0	-0.29261	0.86097	-1.84956	-0.00304	1.05538
4	20	0	10.0	-15	0.50	1.0	0.51389	-1.86483	2.73620	0.01986	1.10083
5	20	0	10.0	5	0.75	0.1	0.08661	-0.88781	-1.14913	-0.22622	1.04860
6	20	0	0.5	-15	0.75	0.1	0.07428	-0.00864	4.36061	-0.08413	1.13510
7	50	0	10.0	-15	0.25	1.0	0.48582	-3.44919	3.30999	0.06212	0.41163
8	50	0	0.5	-15	0.25	0.1	0.13505	-0.94863	0.96820	0.01168	0.41400
9	50	0	10.0	5	0.50	0.1	-0.00099	-0.28489	0.33447	-0.01210	0.41754
10	50	0	0.5	5	0.50	0.1	-0.02750	-0.08704	0.22876	0.00259	0.41705
11	50	0	10.0	15	0.75	1.0	0.02431	-0.26565	-2.16651	-0.05399	0.39131
12	50	0	0.5	15	0.75	1.0	-0.00722	-0.13278	-2.30117	-0.04903	0.39353
13	100	0	0.5	15	0.25	0.1	-0.04400	0.27359	-0.27299	0.00329	0.20242
14	100	0	10.0	5	0.25	1.0	-0.10186	-0.40770	0.79177	0.10490	0.18735
15	100	0	10.0	15	0.50	0.1	-0.00462	0.01959	-0.05727	0.00002	0.20245
16	100	0	0.5	-15	0.50	1.0	0.03191	-0.09360	0.15093	0.00109	0.18039
17	100	0	0.5	5	0.75	1.0	0.00730	-0.29058	-0.08600	-0.11955	0.18522
18	100	0	10.0	-15	0.75	0.1	0.00607	-0.01299	0.27730	-0.00305	0.20255

	Parameter Combination						Estimated MSE				
	n	β_0	β_1	β_2	γ	σ^2	$\hat{\beta}_0$	$\hat{\beta}_1$	$\hat{\beta}_2$	$\hat{\gamma}$	$\hat{\sigma}^2$
1	20	0	10.0	15	0.25	0.1	0.12383	4.752	5.090	0.00164	1.18131
2	20	0	0.5	5	0.25	1.0	0.33650	7.475	10.468	0.06043	0.95982
3	20	0	0.5	15	0.50	1.0	0.35019	3.194	9.306	0.00241	1.22405
4	20	0	10.0	-15	0.50	1.0	0.52835	5.975	14.104	0.00258	1.32126
5	20	0	10.0	5	0.75	0.1	0.03462	1.214	2.246	0.05228	1.10068
6	20	0	0.5	-15	0.75	0.1	0.03003	0.159	20.234	0.00759	1.28976
7	50	0	10.0	-15	0.25	1.0	0.43978	17.831	17.299	0.00697	0.21481
8	50	0	0.5	-15	0.25	0.1	0.04622	2.200	2.272	0.00037	0.17181
9	50	0	10.0	5	0.50	0.1	0.01932	0.581	0.587	0.00235	0.17478
10	50	0	0.5	5	0.50	0.1	0.01841	0.325	0.406	0.00209	0.17436
11	50	0	10.0	15	0.75	1.0	0.11239	0.818	12.885	0.00646	0.19452
12	50	0	0.5	15	0.75	1.0	0.10518	0.789	13.706	0.00568	0.19642
13	100	0	0.5	15	0.25	0.1	0.01686	0.807	0.812	0.00012	0.04117
14	100	0	10.0	5	0.25	1.0	0.14030	8.489	8.677	0.02641	0.05411
15	100	0	10.0	15	0.50	0.1	0.00764	0.099	0.195	0.00007	0.04119
16	100	0	0.5	-15	0.50	1.0	0.07516	0.953	1.704	0.00092	0.05099
17	100	0	0.5	5	0.75	1.0	0.05693	1.411	6.605	0.02577	0.05470
18	100	0	10.0	-15	0.75	0.1	0.00541	0.029	0.816	0.00011	0.04122

Table B.5: Estimated biases of all estimators. Combinations 1 to 9

		Parameter Combination						Estimated Bias				
		n	β_0	β_1	β_2	γ	σ^2	$\hat{\beta}_0$	$\hat{\beta}_1$	$\hat{\beta}_2$	$\hat{\gamma}$	$\hat{\sigma}^2$
1	MLE-I	20	0	10.0	15	0.25	0.1	-0.00073	-0.02531	0.04986	0.00274	-0.02043
1	MLE-II	20	0	10.0	15	0.25	0.1	0.00308	-0.07672	0.08697	0.00111	-0.01850
1	MLE-TZ	20	0	10.0	15	0.25	0.1	0.00465	-0.11130	0.13906	0.00195	-0.02022
1	Bayes	20	0	10.0	15	0.25	0.1	-0.28016	1.70615	-1.79616	0.03278	1.08633
2	MLE-I	20	0	0.5	5	0.25	1.0	0.06301	-2.24271	3.07828	0.18057	-0.23450
2	MLE-II	20	0	0.5	5	0.25	1.0	0.14079	-3.25518	3.72340	-0.01222	-0.18675
2	MLE-TZ	20	0	0.5	5	0.25	1.0	0.33057	-7.47655	9.26016	0.06161	-0.21496
2	Bayes	20	0	0.5	5	0.25	1.0	-0.30536	2.18130	-1.81818	0.24127	0.92707
3	MLE-I	20	0	0.5	15	0.50	1.0	0.09382	-0.79228	1.45019	-0.00296	-0.19650
3	MLE-II	20	0	0.5	15	0.50	1.0	0.05658	-0.38289	0.69311	-0.00116	-0.18402
3	MLE-TZ	20	0	0.5	15	0.50	1.0	0.09604	-0.70296	1.15343	-0.00432	-0.19454
3	Bayes	20	0	0.5	15	0.50	1.0	-0.29261	0.86097	-1.84956	-0.00304	1.05538
4	MLE-I	20	0	10.0	-15	0.50	1.0	-0.06548	0.63807	-1.50922	0.00343	-0.19192
4	MLE-II	20	0	10.0	-15	0.50	1.0	-0.03979	0.32563	-0.67018	-0.00049	-0.17778
4	MLE-TZ	20	0	10.0	-15	0.50	1.0	-0.06632	0.55026	-1.17713	0.00183	-0.18966
4	Bayes	20	0	10.0	-15	0.50	1.0	0.51389	-1.86483	2.73620	0.01986	1.10083
5	MLE-I	20	0	10.0	5	0.75	0.1	0.03389	-0.29063	0.94448	-0.03129	-0.02387
5	MLE-II	20	0	10.0	5	0.75	0.1	0.02195	-0.05614	0.48894	0.00023	0.04962
5	MLE-TZ	20	0	10.0	5	0.75	0.1	0.03146	-0.18360	1.17538	-0.02448	-0.02212
5	Bayes	20	0	10.0	5	0.75	0.1	0.08661	-0.88781	-1.14913	-0.22622	1.04860
6	MLE-I	20	0	0.5	-15	0.75	0.1	-0.00568	0.01628	-0.17098	-0.00142	-0.02159
6	MLE-II	20	0	0.5	-15	0.75	0.1	-0.00461	0.01233	-0.14537	-0.00114	-0.01748
6	MLE-TZ	20	0	0.5	-15	0.75	0.1	-0.00632	0.02023	-0.15480	-0.00166	-0.02156
6	Bayes	20	0	0.5	-15	0.75	0.1	0.07428	-0.00864	4.36061	-0.08413	1.13510
7	MLE-I	50	0	10.0	-15	0.25	1.0	-0.09137	1.61860	-1.90660	0.00664	-0.07501
7	MLE-II	50	0	10.0	-15	0.25	1.0	-0.06626	0.92469	-0.96351	0.00074	-0.06945
7	MLE-TZ	50	0	10.0	-15	0.25	1.0	-0.08583	1.32084	-1.42364	0.00526	-0.07392
7	Bayes	50	0	10.0	-15	0.25	1.0	0.48582	-3.44919	3.30999	0.06212	0.41163
8	MLE-I	50	0	0.5	-15	0.25	0.1	-0.00830	0.04669	-0.06787	0.00132	-0.00774
8	MLE-II	50	0	0.5	-15	0.25	0.1	-0.00402	0.05942	-0.07358	0.00044	0.05479
8	MLE-TZ	50	0	0.5	-15	0.25	0.1	-0.01060	0.07568	-0.09978	0.00111	-0.00767
8	Bayes	50	0	0.5	-15	0.25	0.1	0.13505	-0.94863	0.96820	0.01168	0.41400
9	MLE-I	50	0	10.0	5	0.50	0.1	0.00730	-0.04808	0.10376	0.00007	-0.00781
9	MLE-II	50	0	10.0	5	0.50	0.1	0.00118	-0.03583	0.08167	-0.00097	0.01699
9	MLE-TZ	50	0	10.0	5	0.50	0.1	0.01302	-0.08285	0.16184	-0.00075	-0.00677
9	Bayes	50	0	10.0	5	0.50	0.1	-0.00099	-0.28489	0.33447	-0.01210	0.41754

Table B.6: Estimated biases of all estimators. Combinations 10 to 18

		Parameter Combination						Estimated Bias				
		n	β_0	β_1	β_2	γ	σ^2	$\hat{\beta}_0$	$\hat{\beta}_1$	$\hat{\beta}_2$	$\hat{\gamma}$	$\hat{\sigma}^2$
10	MLE-I	50	0	0.5	5	0.50	0.1	0.01125	-0.05518	0.14168	0.00218	-0.00781
10	MLE-II	50	0	0.5	5	0.50	0.1	0.00666	-0.04620	0.11319	0.00082	0.03402
10	MLE-TZ	50	0	0.5	5	0.50	0.1	0.01278	-0.06256	0.15643	0.00207	-0.00745
10	Bayes	50	0	0.5	5	0.50	0.1	-0.02750	-0.08704	0.22876	0.00259	0.41705
11	MLE-I	50	0	10.0	15	0.75	1.0	0.02626	-0.14079	1.76713	-0.00581	-0.08368
11	MLE-II	50	0	10.0	15	0.75	1.0	0.01798	-0.09281	0.79860	-0.00269	-0.07839
11	MLE-TZ	50	0	10.0	15	0.75	1.0	0.02734	-0.13888	1.34547	-0.00576	-0.08175
11	Bayes	50	0	10.0	15	0.75	1.0	0.02431	-0.26565	-2.16651	-0.05399	0.39131
12	MLE-I	50	0	0.5	15	0.75	1.0	0.03445	-0.18385	1.86339	-0.00789	-0.08114
12	MLE-II	50	0	0.5	15	0.75	1.0	0.01920	-0.08430	0.85774	-0.00159	-0.07528
12	MLE-TZ	50	0	0.5	15	0.75	1.0	0.02886	-0.13457	1.22569	-0.00537	-0.07914
12	Bayes	50	0	0.5	15	0.75	1.0	-0.00722	-0.13278	-2.30117	-0.04903	0.39353
13	MLE-I	100	0	0.5	15	0.25	0.1	-0.00081	-0.01759	0.03285	0.00048	-0.00373
13	MLE-II	100	0	0.5	15	0.25	0.1	-0.00364	-0.02564	0.03617	-0.00028	0.01332
13	MLE-TZ	100	0	0.5	15	0.25	0.1	0.00080	-0.03755	0.05691	0.00040	-0.00363
13	Bayes	100	0	0.5	15	0.25	0.1	-0.04400	0.27359	-0.27299	0.00329	0.20242
14	MLE-I	100	0	10.0	5	0.25	1.0	0.11397	-6.38787	8.40780	0.07902	-0.04332
14	MLE-II	100	0	10.0	5	0.25	1.0	0.03465	-1.02776	1.12896	-0.00498	-0.03467
14	MLE-TZ	100	0	10.0	5	0.25	1.0	0.12367	-4.20577	4.92756	0.04506	-0.04152
14	Bayes	100	0	10.0	5	0.25	1.0	-0.10186	-0.40770	0.79177	0.10490	0.18735
15	MLE-I	100	0	10.0	15	0.50	0.1	0.00040	0.00757	-0.00080	0.00060	-0.00371
15	MLE-II	100	0	10.0	15	0.50	0.1	0.00077	0.00397	-0.00065	0.00026	-0.00207
15	MLE-TZ	100	0	10.0	15	0.50	0.1	0.00384	-0.01420	0.03894	0.00050	-0.00366
15	Bayes	100	0	10.0	15	0.50	0.1	-0.00462	0.01959	-0.05727	0.00002	0.20245
16	MLE-I	100	0	0.5	-15	0.50	1.0	-0.01793	0.08080	-0.18895	0.00091	-0.04496
16	MLE-II	100	0	0.5	-15	0.50	1.0	-0.01372	0.06274	-0.15363	0.00073	-0.04372
16	MLE-TZ	100	0	0.5	-15	0.50	1.0	-0.01912	0.08764	-0.20254	0.00090	-0.04420
16	Bayes	100	0	0.5	-15	0.50	1.0	0.03191	-0.09360	0.15093	0.00109	0.18039
17	MLE-I	100	0	0.5	5	0.75	1.0	0.10669	-3.50856	10.77104	-0.07926	-0.04777
17	MLE-II	100	0	0.5	5	0.75	1.0	0.02857	-0.12458	2.14725	0.00297	-0.04061
17	MLE-TZ	100	0	0.5	5	0.75	1.0	0.04501	-0.21718	2.89042	-0.01425	-0.04269
17	Bayes	100	0	0.5	5	0.75	1.0	0.00730	-0.29058	-0.08600	-0.11955	0.18522
18	MLE-I	100	0	10.0	-15	0.75	0.1	-0.00264	0.00824	-0.06640	0.00000	-0.00377
18	MLE-II	100	0	10.0	-15	0.75	0.1	-0.00300	0.00974	-0.05836	-0.00015	-0.00374
18	MLE-TZ	100	0	10.0	-15	0.75	0.1	-0.00472	0.01620	-0.09972	-0.00004	-0.00364
18	Bayes	100	0	10.0	-15	0.75	0.1	0.00607	-0.01299	0.27730	-0.00305	0.20255

Table B.7: Estimated MSEs of all estimators. Combinations 1 to 9

		Parameter Combination						MSE				
		n	β_0	β_1	β_2	γ	σ^2	$\hat{\beta}_0$	$\hat{\beta}_1$	$\hat{\beta}_2$	$\hat{\gamma}$	$\hat{\sigma}^2$
1	MLE-I	20	0	10.0	15	0.25	0.1	0.05609	3.555	3.532	0.00062	0.00120
1	MLE-II	20	0	10.0	15	0.25	0.1	0.05294	3.116	3.164	0.00054	0.00249
1	MLE-TZ	20	0	10.0	15	0.25	0.1	0.05791	3.872	3.822	0.00064	0.00120
1	Bayes	20	0	10.0	15	0.25	0.1	0.12383	4.752	5.090	0.00164	1.18131
2	MLE-I	20	0	0.5	5	0.25	1.0	0.83042	145.147	260.463	0.12102	0.13834
2	MLE-II	20	0	0.5	5	0.25	1.0	0.49394	97.581	102.222	0.02651	0.12962
2	MLE-TZ	20	0	0.5	5	0.25	1.0	0.58020	349.478	382.963	0.05016	0.13558
2	Bayes	20	0	0.5	5	0.25	1.0	0.33650	7.475	10.468	0.06043	0.95982
3	MLE-I	20	0	0.5	15	0.50	1.0	0.38613	14.296	21.433	0.00903	0.12112
3	MLE-II	20	0	0.5	15	0.50	1.0	0.31431	4.883	8.728	0.00495	0.12065
3	MLE-TZ	20	0	0.5	15	0.50	1.0	0.37035	9.300	12.422	0.00729	0.12150
3	Bayes	20	0	0.5	15	0.50	1.0	0.35019	3.194	9.306	0.00241	1.22405
4	MLE-I	20	0	10.0	-15	0.50	1.0	0.41247	13.195	23.746	0.00961	0.11241
4	MLE-II	20	0	10.0	-15	0.50	1.0	0.33235	5.063	9.013	0.00477	0.11201
4	MLE-TZ	20	0	10.0	-15	0.50	1.0	0.38566	9.525	13.477	0.00753	0.11275
4	Bayes	20	0	10.0	-15	0.50	1.0	0.52835	5.975	14.104	0.00258	1.32126
5	MLE-I	20	0	10.0	5	0.75	0.1	0.03539	2.619	11.476	0.01936	0.00129
5	MLE-II	20	0	10.0	5	0.75	0.1	0.03478	0.156	4.996	0.00539	0.61712
5	MLE-TZ	20	0	10.0	5	0.75	0.1	0.03139	0.495	25.816	0.01503	0.00123
5	Bayes	20	0	10.0	5	0.75	0.1	0.03462	1.214	2.246	0.05228	1.10068
6	MLE-I	20	0	0.5	-15	0.75	0.1	0.02429	0.137	3.969	0.00064	0.00126
6	MLE-II	20	0	0.5	-15	0.75	0.1	0.02394	0.132	3.517	0.00052	0.01612
6	MLE-TZ	20	0	0.5	-15	0.75	0.1	0.02418	0.135	3.816	0.00059	0.00126
6	Bayes	20	0	0.5	-15	0.75	0.1	0.03003	0.159	20.234	0.00759	1.28976
7	MLE-I	50	0	10.0	-15	0.25	1.0	0.36225	58.289	82.183	0.00574	0.04465
7	MLE-II	50	0	10.0	-15	0.25	1.0	0.27011	21.029	20.754	0.00256	0.04450
7	MLE-TZ	50	0	10.0	-15	0.25	1.0	0.32977	36.457	34.751	0.00431	0.04454
7	Bayes	50	0	10.0	-15	0.25	1.0	0.43978	17.831	17.299	0.00697	0.21481
8	MLE-I	50	0	0.5	-15	0.25	0.1	0.02898	1.627	1.634	0.00025	0.00043
8	MLE-II	50	0	0.5	-15	0.25	0.1	0.04176	1.474	1.495	0.00035	0.63030
8	MLE-TZ	50	0	0.5	-15	0.25	0.1	0.02916	1.653	1.659	0.00025	0.00043
8	Bayes	50	0	0.5	-15	0.25	0.1	0.04622	2.200	2.272	0.00037	0.17181
9	MLE-I	50	0	10.0	5	0.50	0.1	0.01707	0.236	0.410	0.00191	0.00043
9	MLE-II	50	0	10.0	5	0.50	0.1	0.02367	0.202	0.393	0.00261	0.16755
9	MLE-TZ	50	0	10.0	5	0.50	0.1	0.01704	0.229	0.421	0.00169	0.00043
9	Bayes	50	0	10.0	5	0.50	0.1	0.01932	0.581	0.587	0.00235	0.17478

Table B.8: Estimated MSEs of all estimators. Combinations 10 to 18

		Parameter Combination						Estimated MSE				
		n	β_0	β_1	β_2	γ	σ^2	$\hat{\beta}_0$	$\hat{\beta}_1$	$\hat{\beta}_2$	$\hat{\gamma}$	$\hat{\sigma}^2$
10	MLE-I	50	0	0.5	5	0.50	0.1	0.01704	0.243	0.393	0.00216	0.00043
10	MLE-II	50	0	0.5	5	0.50	0.1	0.02320	0.204	0.375	0.00290	0.55289
10	MLE-TZ	50	0	0.5	5	0.50	0.1	0.01713	0.243	0.401	0.00208	0.00044
10	Bayes	50	0	0.5	5	0.50	0.1	0.01841	0.325	0.406	0.00209	0.17436
11	MLE-I	50	0	10.0	15	0.75	1.0	0.11978	0.851	87.363	0.00503	0.04295
11	MLE-II	50	0	10.0	15	0.75	1.0	0.11106	0.636	16.649	0.00237	0.04258
11	MLE-TZ	50	0	10.0	15	0.75	1.0	0.11583	0.738	31.405	0.00420	0.04271
11	Bayes	50	0	10.0	15	0.75	1.0	0.11239	0.818	12.885	0.00646	0.19452
12	MLE-I	50	0	0.5	15	0.75	1.0	0.11994	1.365	93.226	0.00617	0.04321
12	MLE-II	50	0	0.5	15	0.75	1.0	0.10523	0.685	31.369	0.00228	0.04263
12	MLE-TZ	50	0	0.5	15	0.75	1.0	0.11025	0.849	32.320	0.00404	0.04316
12	Bayes	50	0	0.5	15	0.75	1.0	0.10518	0.789	13.706	0.00568	0.19642
13	MLE-I	100	0	0.5	15	0.25	0.1	0.01449	0.732	0.734	0.00011	0.00020
13	MLE-II	100	0	0.5	15	0.25	0.1	0.02136	0.669	0.673	0.00019	0.20961
13	MLE-TZ	100	0	0.5	15	0.25	0.1	0.01457	0.741	0.742	0.00011	0.00020
13	Bayes	100	0	0.5	15	0.25	0.1	0.01686	0.807	0.812	0.00012	0.04117
14	MLE-I	100	0	10.0	5	0.25	1.0	0.39314	1258.539	1871.910	0.06549	0.02000
14	MLE-II	100	0	10.0	5	0.25	1.0	0.12494	64.899	65.067	0.00830	0.01968
14	MLE-TZ	100	0	10.0	5	0.25	1.0	0.25377	326.147	338.816	0.03334	0.01999
14	Bayes	100	0	10.0	5	0.25	1.0	0.14030	8.489	8.677	0.02641	0.05411
15	MLE-I	100	0	10.0	15	0.50	0.1	0.00751	0.096	0.192	0.00007	0.00021
15	MLE-II	100	0	10.0	15	0.50	0.1	0.00741	0.093	0.188	0.00009	0.00151
15	MLE-TZ	100	0	10.0	15	0.50	0.1	0.00754	0.096	0.193	0.00007	0.00021
15	Bayes	100	0	10.0	15	0.50	0.1	0.00764	0.099	0.195	0.00007	0.04119
16	MLE-I	100	0	0.5	-15	0.50	1.0	0.07919	1.070	1.822	0.00108	0.01965
16	MLE-II	100	0	0.5	-15	0.50	1.0	0.07265	0.911	1.652	0.00096	0.01962
16	MLE-TZ	100	0	0.5	-15	0.50	1.0	0.07924	1.072	1.826	0.00108	0.01965
16	Bayes	100	0	0.5	-15	0.50	1.0	0.07516	0.953	1.704	0.00092	0.05099
17	MLE-I	100	0	0.5	5	0.75	1.0	0.32654	1473.529	2949.487	0.06836	0.02133
17	MLE-II	100	0	0.5	5	0.75	1.0	0.05115	0.346	152.462	0.00942	0.02105
17	MLE-TZ	100	0	0.5	5	0.75	1.0	0.05545	0.614	107.840	0.01494	0.02107
17	Bayes	100	0	0.5	5	0.75	1.0	0.05693	1.411	6.605	0.02577	0.05470
18	MLE-I	100	0	10.0	-15	0.75	0.1	0.00539	0.030	0.831	0.00011	0.00020
18	MLE-II	100	0	10.0	-15	0.75	0.1	0.00535	0.030	0.782	0.00010	0.00020
18	MLE-TZ	100	0	10.0	-15	0.75	0.1	0.00542	0.030	0.842	0.00011	0.00020
18	Bayes	100	0	10.0	-15	0.75	0.1	0.00541	0.029	0.816	0.00011	0.04122

COMPUTER CODE

This section includes code used in the simulation study and application.

C.1 ESTIMATORS AND FUNCTIONS USED IN SIMULATION

```
#####
###      GENERATE SEGMENTED DATA      ###
#####
# Generates a sequence from 0 to 1
segregx<-function(n){ rep(seq(0,1,length=n)) }

# Generate n y's corresponding to the x's.
# p = [beta_0, beta_1, beta_2, gamma (needs to be in [0,1]), sig^2]

segregy<-function(p,x){
  (p[1] + p[2]*x)*(x<p[4]) + (p[1] + p[2]*x + p[3]*(x-p[4]))*(x>=p[4]) +
  rnorm(length(x),0,as.numeric(sqrt(p[5])))
}

#####
###      MSE/BIAS FUNCTIONS      ###
#####
# A function to calculate the Bias for the n/beta/gamma combo for all nsims
# Checks for NA's in case of no convergence for any nsim in 1 combo.
BIAS<-function(param,estvec){
  round(mean(estvec-param,na.rm=T),digits=6)
}

# A function to calculate MSE for the n/beta/gamma combo.
# Checks for NA's in case of no convergence for any nsim in 1 combo.
MSE<-function(param,estvec){
  if(sum(estvec,na.rm=T)==0){NA} else
  {round(var(estvec,na.rm=T)+(mean(estvec-param,na.rm=T))^2,digits=6)}
}

#####
###      PARAMETER COMBINATIONS      ###
#####

frac.design <- rbind(
c(20,    0, 10.0,   15,  0.25,   0.1),
c(20,    0,  0.5,    5,  0.25,   1.0),
```

```

c(20, 0, 0.5, 15, 0.50, 1.0),
c(20, 0, 10.0, -15, 0.50, 1.0),
c(20, 0, 10.0, 5, 0.75, 0.1),
c(20, 0, 0.5, -15, 0.75, 0.1),
c(50, 0, 10.0, -15, 0.25, 1.0),
c(50, 0, 0.5, -15, 0.25, 0.1),
c(50, 0, 10.0, 5, 0.50, 0.1),
c(50, 0, 0.5, 5, 0.50, 0.1),
c(50, 0, 10.0, 15, 0.75, 1.0),
c(50, 0, 0.5, 15, 0.75, 1.0),
c(100, 0, 0.5, 15, 0.25, 0.1),
c(100, 0, 10.0, 5, 0.25, 1.0),
c(100, 0, 10.0, 15, 0.50, 0.1),
c(100, 0, 0.5, -15, 0.50, 1.0),
c(100, 0, 0.5, 5, 0.75, 1.0),
c(100, 0, 10.0, -15, 0.75, 0.1)
)

```

```
#####
```

```
#### ESTIMATORS ####
```

```
#####
```

```

# This section contains all of the code for estimators used in simulation:
# MAXIMUM LIKELIHOOD I (maximize discrete marginal likelihood)
# MAXIMUM LIKELIHOOD II (numerically approximate gradient and Hessian)
# TISHLER AND ZANG MAXIMUM LIKELIHOOD ALTERNATIVE (a bent cable approach)
# BAYESIAN POSTERIOR MEAN (a Bayesian approach)

```

```
#####
#####          MAXIMUM LIKELIHOOD I AND II          #####
#####
```

```
#####
```

```

# MLE-I
# Maximize marginal likelihood as a function of m (all possible divisions)
# of data into 2 segments. For each m, use constrained NR to estimate
# between gradient discontinuities, constraining  $x_i \leq \gamma \leq x_{i+1}$ 

```

```

MLE.I<-function(p,x,y){
  # Maximizes discrete log-likelihood for all 5 params simultaneously
  # Inputs: p (starting values), x, y (data)
  #         where p = [beta_0, beta_1, beta_2, gamma, sig^2]
  # Outputs: param.est (MLEs), like.min (marginal likelihood)
  #
  like.min <- rep(NA,length(x))
  params <- matrix(NA,length(x),5)

  # For each possible division, find the constrained MLEs
  for (i in 2:(length(x)-2)){
temp <- nlmnrb(start=p,objective=fx,x=x,y=y,gradient=G,hessian=H,
              lower=c(-Inf,-Inf,-Inf,x[i],0),upper=c(Inf,Inf,Inf,x[i+1],Inf))
    params[i,] <- temp$par
    like.min[i] <- temp$objective
  }

  use <- min(which(like.min==min(like.min,na.rm=T))) # Multiple optima: use first
  list(param.est=params[use,],like.min=like.min)
}

```

```

# The Negative log-likelihood function (objective function)
fx <- function(x,y,p) {
  .5*length(x)*log(2*pi)+.5*length(x)*log(p[5]) +
  .5/p[5]*sum((y-p[1]-p[2]*x-p[3]*(x-p[4])*(x>p[4]))^2)
}

# Returns the gradient of the negative log-likelihood
G <- function(p,x,y){
  G <- rep(NA,5)
  G[1] <- 1/p[5]*sum(y - p[1] - p[2]*x - p[3]*(x-p[4])*(x>p[4]))
  G[2] <- 1/p[5]*sum(x*(y - p[1] - p[2]*x - p[3]*(x-p[4])*(x>p[4])))
  G[3] <- 1/p[5]*sum((x-p[4])*(x>p[4])*(y - p[1] - p[2]*x - p[3]*(x-p[4])*(x>p[4])))
  G[4] <- -1/p[5]*sum(p[3]*(x>p[4])*(y - p[1] - p[2]*x - p[3]*(x-p[4])))
  G[5] <- -length(x)/(2*p[5]) + 1/(2*p[5]^2)*sum((y-p[1]-p[2]*x-p[3]*(x-p[4])*(x>p[4]))^2)
  -G
}

# Returns the Hessian of the negative log-likelihood
H <- function(p,x,y){
  H <- matrix(0,5,5)
  H[1,1]<- -length(x)/p[5]
  H[1,2]<- -1/p[5]*sum(x)
  H[1,3]<- -1/p[5]*sum((x>p[4])*(x-p[4]))
  H[1,4]<- 1/p[5]*p[3]*sum(x>p[4])
  H[1,5]<- -1/(p[5]^2)*sum(y - p[1] - p[2]*x - p[3]*(x-p[4])*(x>p[4]))

  H[2,2]<- -1/p[5]*sum(x^2)
  H[2,3]<- -1/p[5]*sum((x>p[4])*x*(x-p[4]))
  H[2,4]<- 1/p[5]*p[3]*sum(x*(x>p[4]))
  H[2,5]<- -1/(p[5]^2)*sum(x*(y - p[1] - p[2]*x - p[3]*(x-p[4])*(x>p[4])))

  H[3,3]<- -1/p[5]*sum(((x>p[4])*(x-p[4]))^2)
  H[3,4]<- -1/p[5]*sum((x>p[4])*(y-p[1]-p[2]*x - 2*p[3]*(x - p[4])))
  H[3,5]<- -1/(p[5]^2)*sum((x-p[4])*(x>p[4])*(y - p[1] - p[2]*x - p[3]*(x-p[4])))

  H[4,4]<- -1/p[5]*sum(x>p[4])*p[3]^2
  H[4,5]<- 1/(p[5]^2)*sum(p[3]*(x>p[4])*(y - p[1] - p[2]*x - p[3]*(x-p[4])))

  H[5,5]<- length(x)/(2*p[5]^2) - 1/(p[5]^3)*sum((y-p[1]-p[2]*x-p[3]*(x-p[4])*(x>p[4]))^2)

  H <- H + t(H*upper.tri(H))
  -H
}

```

```

#####
# MLE-II
# Maximize the log-likelihood using Newton-Raphson and numeric approximations
# to the gradient and hessian

MLE.II<-function(p,x,y){
  # Maximize log-likelihood for all 5 params simultaneously using numeric
  # gradient and hessian.
  # Inputs: p (starting/true values), x, y (data)
  # where p = [beta_0, beta_1, beta_2, gamma, sig^2]
  # Outputs: param.est (MLEs), conv (a code indicating if the algorithm converged),

  storecode <- 6
  while(storecode>=3){
    start <- ifelse(storecode==6,0,runif(1,-.1,.1))

```

```

    store <- nlm(fx2,p=p+start,x=x,y=y)
    storecode<- as.numeric(store$code)
  }
  list(param.est=store$estimate,conv=storecode,start=start)
}

# The Negative log-likelihood function (objective function)
fx2 <- function(p,x,y) {
  nml <- .5*length(x)*log(2*pi)+.5*length(x)*log(p[5]) +
    .5/p[5]*sum((y-p[1]-p[2]*x-p[3]*(x-p[4])*(x>p[4]))^2)

  if (p[4]<0 | p[4]>1 | p[5]<0) nml <- 100000000 # Constrains gamma, sigma to appropriate ranges
  return(nml)
}

#####
##### TISHLER AND ZANG MAXIMUM LIKELIHOOD #####
#####

# A bent-cable model replaces the original "sharp" model, facilitating continuous
# differentiation of the likelihood function

MLETZ<-function(p,x,y){
# Maximizes log-likelihood for all 5 params simultaneously using bent cable
# approximation to likelihood
# Inputs: p (starting/true values), x, y (data)
#       where p = [beta_0, beta_1, beta_2, gamma, sig^2]
# NOTE: function transforms to parameterization with 2 intersecting lines
# to do optimization

change <- 20 # This must be positive and even
r <- ifelse(p[3]<0,-change,change) # Choose initial p depending on concave/convex
B <- 1.1*max(abs(y)) # Add large positive value to max all positive
last <- fromPto2LineP(p) # Starting values for parameter estimates
keepgoing <- TRUE

# start each iteration with param estimates at last iteration
while(keepgoing){

# Use some error handling so simulation will record and continue after difficulties
warn.error <- rep(FALSE,2)
tryCatch( store<-nlm(fx3, p=last ,x=x, y=y, r=r, B=B,
                    iterlim=20000, check.analyticals=F)
,warning = function(warn) {keepgoing <<-FALSE; warn.error[1] <<-TRUE}
,error = function(err) {keepgoing <<-FALSE; warn.error[2] <<-TRUE}
)
current <- store$estimate
line1 <- B+current[1] + current[2]*x
line2 <- B+current[3] + current[4]*x
if(sum(c(line1,line2)<0)!=0){ # Check if all positive
B <- -1.1*min(line1-B,line2-B) # If not all positive: Adjust B and r
r <- ifelse(p[3]<0,r-change,r+change)
} else
if(sum((last-current)^2)^.5<.01){ # Check if within tolerance of last estimate
keepgoing=FALSE # If in tolerance: STOP
} else { # If not in tolerance: Adjust r
last <- current
r <- ifelse(p[3]<0,r-change,r+change)
}
}

# Returns param.est (final estimates), conv (1-5 R codes, 8 warn, 9 err), r (final r/p)

```

```

list(param.est=from2LinePtoP(store$estimate),
      conv=ifelse(warn.error[2],9,ifelse(warn.error[1],8,store$code)),
      r=r)
}

# NOTE PARAMETERIZATION: reparameterize starting values, back transform to get estimates
# p = [a10 = beta_0
#      a11 = beta_1
#      a20 = beta_0 - beta_2*gamma
#      a21 = beta_1 + beta_2
#      sig^2]

# objective function, gradient, and hessian for MLETZ - does use scaling number to
# avoid numerical difficulties
fx3 <- function (p, x, y, B, r) {
  p1 <- p[1]
  p2 <- p[2]
  p3 <- p[3]
  p4 <- p[4]
  p5 <- ifelse(p[5]<0,.000001,p[5]) # This was a bad idea
  line1 <- B + cbind(1,x)%*%p[1:2]
  line2 <- B + cbind(1,x)%*%p[3:4]
  scalenum <- apply(cbind(line1,line2),1,function(f) if(r>0){ max(f)}else{min(f)})
  .expr5 <- (B + p1 + p2 * x)/scalenum
  .expr10 <- (B + p3 + p4 * x)/scalenum
  .expr12 <- .expr5^r + .expr10^r
  .expr13 <- 1/r
  .expr16 <- B + y - scalenum * .expr12^.expr13
  .expr18 <- .expr13 - 1
  .expr19 <- .expr12^.expr18
  .expr20 <- r - 1
  .expr21 <- .expr5^.expr20
  .expr22 <- 1/scalenum
  .expr23 <- r * .expr22
  .expr24 <- .expr21 * .expr23
  .expr25 <- .expr13 * .expr24
  .expr27 <- scalenum * (.expr19 * .expr25)
  .expr32 <- .expr12^(.expr18 - 1)
  .expr36 <- .expr20 - 1
  .expr37 <- .expr5^.expr36
  .expr38 <- .expr20 * .expr22
  .expr50 <- x/scalenum
  .expr51 <- r * .expr50
  .expr52 <- .expr21 * .expr51
  .expr54 <- .expr32 * (.expr18 * .expr52)
  .expr56 <- .expr20 * .expr50
  .expr57 <- .expr37 * .expr56
  .expr64 <- .expr13 * .expr52
  .expr66 <- scalenum * (.expr19 * .expr64)
  .expr71 <- .expr10^.expr20
  .expr72 <- .expr71 * .expr23
  .expr74 <- .expr32 * (.expr18 * .expr72)
  .expr78 <- .expr13 * .expr72
  .expr80 <- scalenum * (.expr19 * .expr78)
  .expr85 <- .expr71 * .expr51
  .expr87 <- .expr32 * (.expr18 * .expr85)
  .expr91 <- .expr13 * .expr85
  .expr93 <- scalenum * (.expr19 * .expr91)
  .expr130 <- .expr10^.expr36
  .expr143 <- .expr130 * .expr56

# Obejctive Function
.value <- .5*length(x)*log(2*pi) + .5*length(x)*log(p5) + .5/p5*sum(.expr16^2)

```

```

# Gradient
.grad <- array(0, c(1L, 5L), list(NULL, c("p1", "p2", "p3", "p4","p5")))
.grad[, "p1"] <- -1/p5*sum(.expr27 * .expr16)
.grad[, "p2"] <- -1/p5*sum(.expr66 * .expr16)
.grad[, "p3"] <- -1/p5*sum(.expr80 * .expr16)
.grad[, "p4"] <- -1/p5*sum(.expr93 * .expr16)
.grad[, "p5"] <- length(x)/(2*p5) - 1/(2*p5^2)*sum(.expr16^2)

# Hessian
.hessian <- array(0, c(5L, 5L), list( c("p1", "p2", "p3", "p4", "p5"),
                                     c("p1", "p2", "p3", "p4", "p5")))

.hessian["p1", "p1"] <- -1/p5*sum(scalenum * (.expr32 * (.expr18 *
.expr24) * .expr25 + .expr19 * (.expr13 * (.expr37 *
.expr38 * .expr23))) * .expr16 - .expr27 * .expr27)
.hessian["p1", "p2"] <- .hessian["p2", "p1"] <- -1/p5 *
sum(scalenum * (.expr54 * .expr25 + .expr19 * (.expr13 *
.expr57 * .expr23))) * .expr16 - .expr27 * .expr66)
.hessian["p1", "p3"] <- .hessian["p3", "p1"] <- -1/p5 *
sum(scalenum * (.expr74 * .expr25) * .expr16 - .expr27 * .expr80)
.hessian["p1", "p4"] <- .hessian["p4", "p1"] <- -1/p5 *
sum(scalenum * (.expr87 * .expr25) * .expr16 - .expr27 * .expr93)
.hessian["p1", "p5"] <- .hessian["p5", "p1"] <- -1/(p5)*.grad[, "p1"]

.hessian["p2", "p2"] <- -1/p5 *sum(scalenum * (.expr54 * .expr64 +
.expr19 * (.expr13 * (.expr57 * .expr51))) * .expr16 - .expr66 * .expr66)
.hessian["p2", "p3"] <- .hessian["p3", "p2"] <- -1/p5 *
sum(scalenum * (.expr74 * .expr64) * .expr16 - .expr66 * .expr80)
.hessian["p2", "p4"] <- .hessian["p4", "p2"] <- -1/p5 *
sum(scalenum * (.expr87 * .expr64) * .expr16 - .expr66 * .expr93)
.hessian["p2", "p5"] <- .hessian["p5", "p2"] <- -1/(p5)*.grad[, "p2"]

.hessian["p3", "p3"] <- -1/p5 *sum(scalenum * (.expr74 * .expr78 +
.expr19 * (.expr13 * (.expr130 * .expr38 * .expr23))) *
.expr16 - .expr80 * .expr80)
.hessian["p3", "p4"] <- .hessian["p4", "p3"] <- -1/p5 *
sum(scalenum * (.expr87 * .expr78 + .expr19 * (.expr13 *
.expr143 * .expr23))) * .expr16 - .expr80 * .expr93)
.hessian["p3", "p5"] <- .hessian["p5", "p3"] <- -1/(p5)*.grad[, "p3"]

.hessian["p4", "p4"] <- -1/p5 *sum(scalenum * (.expr87 * .expr91 +
.expr19 * (.expr13 * (.expr143 * .expr51))) * .expr16 - .expr93 * .expr93)
.hessian["p4", "p5"] <- .hessian["p5", "p4"] <- -1/(p5)*.grad[, "p4"]

.hessian["p5", "p5"] <- -length(x)/(2*p5^2) +1/(p5^3)*sum(.expr16^2)

attr(.value, "gradient") <- .grad
attr(.value, "hessian") <- .hessian
.value

}

# Functions to reparameterize to and from MLE-TZ 2 intersecting line parameterization
fromPto2LineP <- function(p){c(p[1:2],p[1]-p[3]*p[4],p[2]+p[3],p[5])}
from2LinePtoP <- function(p){c(p[1:2],p[4]-p[2],(p[3]-p[1])/(p[2]-p[4]),p[5])}

```

```

#####
##### BAYESIAN POSTERIOR MEAN #####
#####

```

```

# Takes as input x, y (data), p, the true parameter value which it uses as its
# starting values and cand.var, which are particularly tuned to the parameter
# combinations used in simulation

Bayes<-function(p,x,y,cand.var){

  # Values for the priors -- make broad enough to cover all scenarios
  # For purposes of simulation - uniform is diffuse on gamma space [0,1]
  m.b0 <- 0
  s2.b0<- 100
  m.b1 <- 0
  s2.b1<- 100
  m.b2 <- 0
  s2.b2<- 100
  a.sig2 <- 2.001
  b.sig2 <- .1

  # Starting values for MCMC iterations: true param values

  niter <- 25000
  gam<-rep(p[4],niter)
  sig2<-rep(p[5],niter)
  b0 <-rep(p[1],niter)
  b1 <-rep(p[2],niter)
  b2 <-rep(p[3],niter)

  # Candidate distribution variances - Using normals on all

  cand.b0 <- as.numeric(cand.var[1])
  cand.b1 <- as.numeric(cand.var[2])
  cand.b2 <- as.numeric(cand.var[3])
  cand.gam <- as.numeric(cand.var[4])

  ar.gam <- ar.b0 <- ar.b1 <- ar.b2 <-1

  # Run Iterations and Get Draws
  for(i in 2:niter){
  # update for gamma
  gam[i] <- gam[i-1]
  old <- gam[i-1]
  new <- rnorm(1,old,sqrt(cand.gam))
  if((new<=x[length(x)-2]) && (new>=x[3])) {
    llo <- -.5/sig2[i-1]*sum((y - (b0[i-1] + b1[i-1]*x + b2[i-1]*(x-old)*(x>old)))^2)
    lln <- -.5/sig2[i-1]*sum((y - (b0[i-1] + b1[i-1]*x + b2[i-1]*(x-new)*(x>new)))^2)
    if(log(runif(1,0,1))<(lln-llo)){ gam[i]<-new ; if(i>100) {ar.gam <-ar.gam+1}}
  }

  # update for sig2 - Complete conditional is Inverse gamma
  sig2[i] <- 1/rgamma(1,length(x)/2 + a.sig2,scale=(1/b.sig2 + .5*sum((y - (b0[i-1] +
    b1[i-1]*x + b2[i-1]*(x-gam[i])*(x>gam[i]))^2))^(-1))

  # update for b0
  b0[i] <- b0[i-1]
  old <- b0[i-1]
  new <- rnorm(1,old,sqrt(cand.b0))
  llo <- -.5/sig2[i]*sum((y - (old + b1[i-1]*x + b2[i-1]*
    (x-gam[i])*(x>gam[i]))^2) - .5/s2.b0*(old - m.b0)^2
  lln <- -.5/sig2[i]*sum((y - (new + b1[i-1]*x + b2[i-1]*(x-gam[i])*
    (x>gam[i]))^2) - .5/s2.b0*(new - m.b0)^2
  if(log(runif(1,0,1))<(lln-llo)){ b0[i]<-new ; if(i>100) {ar.b0 <-ar.b0+1}}

  # update for b1
  b1[i] <- b1[i-1]
  old <- b1[i-1]

```

```

new <- rnorm(1,old,sqrt(cand.b1))
llo <- -.5/sig2[i]*sum((y - (b0[i] + old*x + b2[i-1]*(x-gam[i]))*(x>gam[i]))^2) -
      .5/s2.b1*(old - m.b1)^2
lln <- -.5/sig2[i]*sum((y - (b0[i] + new*x + b2[i-1]*(x-gam[i]))*(x>gam[i]))^2) -
      .5/s2.b1*(new - m.b1)^2
if(log(runif(1,0,1))<(lln-llo)){ b1[i]<-new ; if(i>100) {ar.b1 <-ar.b1+1} }

# update for b2
b2[i] <- b2[i-1]
old <- b2[i-1]
new <- rnorm(1,old,sqrt(cand.b2))
llo <- -.5/sig2[i]*sum((y - (b0[i] + b1[i]*x + old*(x-gam[i]))*(x>gam[i]))^2) -
      .5/s2.b2*(old - m.b2)^2
lln <- -.5/sig2[i]*sum((y - (b0[i] + b1[i]*x + new*(x-gam[i]))*(x>gam[i]))^2) -
      .5/s2.b2*(new - m.b2)^2
if(log(runif(1,0,1))<(lln-llo)){ b2[i]<-new ; if(i>100) {ar.b2 <-ar.b2+1}}
}

out <- cbind(b0,b1,b2,gam,sig2)[-(1:100),]

# Output final parameter estimates and acceptance rates
list(param.est=apply(out,2,mean),ar=c(ar.b0,ar.b1,ar.b2,ar.gam)/(niter-100),posts=out)
}

# b0, b1, b2, gam
candidatevars <- rbind(
  c( 0.6, 2, 6, .004),
  c( 0.5, 2, 18, .2),
  c( 0.6, 2, 16, .015),
  c( 0.6, 2, 16, .015),
  c( 0.6, 2, 16, .09),
  c( 0.6, 2, 30, .04),
  c( 0.3, 1, 2, 0.005),
  c( 0.1, 0.5, 1, 0.001),
  c( 0.1, 0.5, 2, 0.01),
  c( 0.1, 0.5, 1.5, .01),
  c( 0.3, 0.7, 20, .005),
  c( 0.3, 0.5, 22, .01),
  c( 0.05, 0.1, .4, .0004),
  c( 0.1, 0.5, 1.5, .01),
  c( 0.05, 0.1, .5, .0005),
  c( 0.1, 0.4, 1.5, .001),
  c( 0.1, 0.4, 12, .02),
  c( 0.05, 0.1, 4, .001))

```

C.2 SIMULATION STUDY

```

#####

#### SIMULATIONS ####

#####

source('SimulationFunctions032410.R')

# Bias and MSE File
namestemp <- c('n','b0','b1','b2','gam','sig2')
filename1 <- 'results021210_1000_All.txt'

```

```

outputnames<-t(c(namestemp,'Est',paste('Bias.',sep="",namestemp[2:6]),
      paste('MSE.',sep="",namestemp[2:6])))
write.table(outputnames,filename1,col.names=F,row.names=F)

# Data File
filename3 <- 'data021210_1000_All.txt'
write.table(NULL,filename3,col.names=F,row.names=F)

# Estimator File
filename2 <- 'estimator021210_1000_All.txt'
outputnames<-t(c(namestemp,'Est','b0hat','b1hat','b2hat','gamhat','sig2hat'))
write.table(outputnames,filename2,col.names=F,row.names=F)

# Estimator Detail
filename4 <- 'estdetail021210_1000_All.txt'
outputnames<-t(c('combo','iter','MLE2.conv','MLE2.start','MLETZ.conv','MLETZ.r',
  'Bayes.arb0','Bayes.arb1','Bayes.arb2','Bayes.argam'))
write.table(outputnames,filename4,col.names=F,row.names=F)

set.seed(043004)
nsim<-1000
storeMLEI <- storeMLEII <- storeMLETZ <- storeBayes <-matrix(NA,nsim,5)

MLE2.stuff <- matrix(NA,nsim,2)
MLETZ.stuff <- matrix(NA,nsim,2)
Bayes.stuff <- matrix(NA,nsim,4)

for(i in c(1:18)){
  n <- frac.design[i,1]
  p <- as.numeric(frac.design[i,2:6])
  for (l in 1:nsim){
    x <- segregx(n)
    y <- segregy(p,x)

    # Not generating new data - just reading existing data from a table
    #thisline <- as.matrix(read.table('data021210_1000_All.txt',skip=((i-1)*1000 + (l-1)),nrow=1))
    #y <- as.vector(thisline[,9:length(thisline)])

    r1 <- MLE.I(p,x,y)
    r2 <- MLE.II(p,x,y)
    r3 <- MLETZ(p,x,y)
    r4 <- Bayes(p,x,y,candidatevars[i,])

    storeMLEI[l,] <- r1$param.est
    storeMLEII[l,] <- r2$param.est
    storeMLETZ[l,] <- r3$param.est
    storeBayes[l,] <- r4$param.est

    MLE2.stuff[l,] <- c(r2$conv,r2$start)
    MLETZ.stuff[l,] <- c(r3$conv,r3$r)
    Bayes.stuff[l,] <- r4$par

    # Write data to table
    write.table(t(c(i,l,n,p,y)),filename3,append=T,col.names=F,row.names=F)
  }

# Calculate bias and MSE

```

```

t1 <- t2 <- t3 <- t4 <- rep(NA,10)
for (k in 1:5){
t1[c(k,k+5)] <- c(BIAS(p[k],storeMLEI[,k]), MSE(p[k],storeMLEI[,k]))
t2[c(k,k+5)] <- c(BIAS(p[k],storeMLEII[,k]), MSE(p[k],storeMLEII[,k]))
t3[c(k,k+5)] <- c(BIAS(p[k],storeMLETZ[,k]), MSE(p[k],storeMLETZ[,k]))
t4[c(k,k+5)] <- c(BIAS(p[k],storeBayes[,k]), MSE(p[k],storeBayes[,k]))
}

# Write bias and MSE to table
hold <- rbind(t1,t2,t3,t4)
dimnames(hold)[[1]] <- NULL
Z <-cbind(matrix(frac.design[i,],4,6,byrow=T),c(1:4),hold)
write.table(Z,filename1,append=T,col.names=F,row.names=F)

# Write estimators
W <- cbind(matrix(frac.design[i,],4*nsim,6,byrow=T), rep(1:4,each=nsim),
           rbind(storeMLEI,storeMLEII,storeMLETZ,storeBayes))
write.table(W,filename2,append=T,col.names=F,row.names=F)

# Write estimator detail to separate file
write.table(cbind(i,rep(1:nsim),MLE2.stuff,MLETZ.stuff,Bayes.stuff),
           filename4,append=T,col.names=F,row.names=F)
}

```

C.3 PLOTTING FUNCTIONS

```

#####

##### PLOT #####

#####

# Change the pallete colors to default to something easier to read
palette(c('black',"chocolate",'blue','purple','darkgreen'))
palette()

segregplot <- function(x,y,p,labelit=p,minx=0,maxx=1,color=1,add=F){
# A function to plot data along with its regression
# x, y are data. To plot without data, give dummy inputs for x and let y=NA
# p is a vector of 5 parameters defining the linear-linear regression to be plotted
# p: b0, b1, b2 (change in slope), gamma (changepoint)
# labelit is a vector of 5 parameters to include in the header
# Defaults to print on [0,1] range, but can be changed with minx, maxx inputs

if(class(p)!="numeric") p <- as.numeric(p)
if(class(labelit)!="numeric") labelit <- as.numeric(labelit)

n<-length(x)
temp1 <- round(labelit[1],digits=2)
temp2 <- round(labelit[2],digits=2)
temp3 <- round(labelit[3],digits=2)
temp4 <- round(labelit[4],digits=2)
temp5 <- round(labelit[5],digits=2)

# Takes care of plotting window in the no-data plot case
type.plot <- ifelse(is.na(y[1]),'n','p')

```

```

y.min <- y.max <- NULL
if(is.na(y[1])){
y <- rep(NA,n)
if(p[3]>0){
y.min <- -.5 + p[1]
y.max <- .5 + p[1] + p[2]*p[4] + (p[3]+p[2])*(1-p[4])
}
if(p[3]<0){
y.min <- -.5 + min(p[1],p[1] + p[2]*p[4] + (p[3]+p[2])*(1-p[4]))
y.max <- .5 + p[1] + p[2]*p[4]
}
}

# Create labels for graph -- If labelit is different than inputs to be plotted
# this assumes that the labels on the graphs should have hats on them -- indicating
# that the line being plotted is the truth but that the estimated line is going to be
# added with a separate segadd command.
line1 <- if(sum(labelit==p)>0){
substitute(paste(beta[0],"=",temp1," ", beta[1],"=",temp2," ",beta[2],"=",temp3))}else{
substitute(paste( hat(beta)[0],"=",temp1," ", hat(beta)[1],"=",temp2," ",hat(beta)[2],"=",temp3))}
line2 <- if(sum(labelit==p)>0){
substitute(paste(gamma,"=",temp4," ",sigma^2,"=",temp5," N=",n))}else{
substitute(paste(hat(gamma),"=",temp4," ",hat(sigma)^2,"=",temp5," N=",n))}

# Plot the data
plot(x,y,pch=20,col=1,type=type.plot,ylim=c(y.min,y.max),main=line1)
# Add a title that includes the true parameter values
title(line2,line=.75)

# Add the estimated segment from p
segments(minx, p[1], p[4], p[1]+ p[4]*p[2], col=color,lwd=2)
segments(p[4], p[1]+ p[4]*p[2], maxx,p[1]+ p[2]*p[4] + (p[2] + p[3])*(maxx-p[4]),col=color,lwd=2)
}

segadd <- function(p,minx=0,maxx=1,color=1){
# Adds linear linear segments defined by p: b0, b1, b2 (change in slope), gamma (changepoint)
# Defaults to print on [0,1] range, but can be changed with minx, maxx inputs
segments(minx, p[1], p[4], p[1]+ p[4]*p[2], col=color,lwd=2)
segments(p[4], p[1]+ p[4]*p[2], maxx,p[1]+ p[2]*p[4] + (p[2] + p[3])*(maxx-p[4]),col=color,lwd=2)
}

```

C.4 SIMULATION ANALYSES

```

#####

####   LOAD FUNCTIONS FROM SIMULATIONS   ####

#####

source('SimulationFunctions032410.R')

nsim <- 1000

# Fractional Factorial Design: Hinckley (1969) would call these ill-defined cases
# 1,2,5,7,8,9,10,13,14,17 -- I think it depends also on the variance
for (i in 1:18){
  n <- frac.design[i,1]
  gam <- frac.design[i,5]
  b2 <- frac.design[i,4]
  print(b2*(n - sum(gam<seq(0,1,length=n)))/n)
}

```

```

}

source('Plots.R')

pdf('fractionalplots.pdf',width=10,height=15)
par(mfcol=c(6,3))
par(mar=c(4,4,4,2)+.01)
for(j in 1:18){
  n <- frac.design[j,1]
  p <- as.numeric(frac.design[j,2:6])
  x <- segregx(n)
  y <- segregy(p,x)

  # Plot truth
  segregplot(x,y,p=as.numeric(frac.design[j,2:6]),labelit=as.numeric(frac.design[j,2:6]))
  title(main=paste("Combination",j),line=3)
}
dev.off()
par(mfrow=c(1,1))
par(mar=c(5,4,4,2)+0.1)

#####

####   PREPARE DATA FOR ANALYSIS   ####

#####

#####
# DATA STORED FOR FUTURE REFERENCE

data.filename <- paste('data021210_1000_All.txt',sep="")

n20.data <- read.table(data.filename,nrow=nsim*6) # Sample size 20 - first 6 combos
n50.data <- read.table(data.filename,skip=nsim*6,nrow=nsim*6) # Sample size 50 - second 6 combos
n100.data <- read.table(data.filename,skip=2*nsim*6,nrow=nsim*6) # Sample size 100 - last 6 combos

n20.data$comboiter <- paste(n20.data$V1,sep="/",n20.data$V2)
n50.data$comboiter <- paste(n50.data$V1,sep="/",n50.data$V2)
n100.data$comboiter <- paste(n100.data$V1,sep="/",n100.data$V2)

#####
# ESTIMATOR DETAIL: MLETZ CONVERGENCE

# See if there were any problems on any of the estimators
det.filename <- paste('estdetail021210_1000_All.txt',sep="")
details <- read.table(det.filename,header=T)
details$comboiter <- paste(details$combo,sep="/",details$iter)

# Some of the MLETZ's look like they had trouble converging, even in the end
table(details$MLETZ.conv)
#   1   2   3   4   8
# 383 17168   59  251  139
#3:last global step failed to locate a point lower than estimate. Either estimate
# is an approximate local minimum of the function or steptol is too small.
#4:iteration limit exceeded.
#8: Threw a warning somewhere along the way and stopped

#####
# ESTIMATORS

```

```

# Read in original estimate
estimates <- read.table('estimator021210_1000_All.txt',header=T)

estimates$combo <- as.factor(rep(1:18,each=nsim*4))
estimates$iter <- rep(1:1000,times=18)
estimates$comboiter <- paste(estimates$combo,sep="/",estimates$iter)
estimates$Est <- as.factor(estimates$Est)

# Remove the estimates for MLETZ where the last convergence step was something
# other than just-peachy convergence
est.new <- estimates
badMLETZconv <- details[details$MLETZ.conv>2,]
badMLETZconv.list <- badMLETZconv$comboiter

# Where the problems occurred: 449 total
table(badMLETZconv$combo)
# 2 3 4 5 6 8 9 10 11 12 13 14 15 16 17 18
# 28 4 1 105 3 1 124 90 1 4 6 18 6 4 21 33

# Also remove the estimates for MLETZ that converge outside of the gamma range: 153 total
badMLETZconv2 <- estimates[estimates$gamhat >1 | estimates$gamhat<0,]
badMLETZconv2.list <- badMLETZconv2$comboiter

# Where they are happening
table(badMLETZconv2$combo) #153
# 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18
# 0 124 0 0 0 0 0 0 0 0 0 0 0 12 0 0 17 0

# Any overlap? No
sum(badMLETZconv.list%in%badMLETZconv2.list)
sum(badMLETZconv2.list%in%badMLETZconv.list)

est.new[(est.new$comboiter%in%c(badMLETZconv.list,badMLETZconv2.list) & est.new$Est==3),8:12] <- NA
dim(na.omit(est.new))
# 71398 15 => 602 removed. Good

# Write new estimates to table
#write.table(est.new,'estimator031610.txt',row.names=F,col.names=T)

# Read new estimates from a table
est.new <- read.table('estimator031610.txt',header=T)

# NOTE: estimates/est.new used throughout interchangeably from here on out
estimates <- est.new

new

#####
# ESTIMATOR DETAIL AGAIN

# Looking at number of updates MLETZ had to perform = r/20
plot(as.factor(details$combo),abs(details[,6])/20,pch=20)
summary(abs(details[,6])/20)[c(1,3,4,6)]
apply(matrix(1:18),1,function(f){summary(abs(details[details[,1]==f,6]))[c(1,3,4,6)]})
table(abs(details[,6])/20)

# Subsetting by the ones used only, plot the number of MLETZ had to perform
subsetdetails <- na.omit(details[!(details$comboiter%in%c(badMLETZconv.list,badMLETZconv2.list)),])
plot(as.factor(subsetdetails$combo),abs(subsetdetails[,6])/20,pch=20)
summary(abs(subsetdetails[,6])/20)[c(1,3,4,6)]
apply(matrix(1:18),1,function(f){summary(abs(subsetdetails[subsetdetails[,1]==f,6]))[c(1,3,4,6)]})

```

```

# Mixing of Bayesian looks okay for the most part -- if generous

# means of acceptance rates
aggregate(details[,7:10],list(details[,1]),mean)
#   Group.1 Bayes.arb0 Bayes.arb1 Bayes.arb2 Bayes.aram
#1      1  0.3513045  0.3329532  0.3237071  0.4073882
#2      2  0.4492460  0.4011161  0.4205154  0.2611075
#3      3  0.4304706  0.4101467  0.4100612  0.3484659
#4      4  0.4334661  0.4133557  0.4331865  0.3769340
#5      5  0.3470357  0.3289850  0.4077556  0.3383955
#6      6  0.3566280  0.3380802  0.4303070  0.2770733
#7      7  0.3468768  0.3306693  0.3984330  0.3202768
#8      8  0.3605106  0.2903412  0.3178955  0.3193526
#9      9  0.3614918  0.2911452  0.3937643  0.3459790
#10     10 0.3612485  0.2911172  0.4466987  0.3544488
#11     11 0.3448180  0.3793526  0.4287412  0.4035124
#12     12 0.3451441  0.4304221  0.4213025  0.3163165
#13     13 0.2899379  0.3432180  0.2764382  0.2624389
#14     14 0.3820496  0.3099108  0.3899237  0.3561296
#15     15 0.2900479  0.3433333  0.4103433  0.2782987
#16     16 0.3811404  0.3393191  0.4527644  0.3722596
#17     17 0.3819199  0.3398510  0.4047135  0.3729541
#18     18 0.2901327  0.3435010  0.4014785  0.2821084

# Percentage that are in the 20-55 generous range
bayesian.mix <- cbind((details[,7:10]>.2)*(details[,7:10]<.55),rep(1:18,each=1000))
aggregate(bayesian.mix[,1:4],list(bayesian.mix[,5]),mean)

#   Group.1 Bayes.arb0 Bayes.arb1 Bayes.arb2 Bayes.aram
#1      1      1.000      1      1.000      1.000
#2      2      0.999      1      0.997      0.928
#3      3      1.000      1      0.998      0.998
#4      4      1.000      1      0.991      0.990
#5      5      1.000      1      1.000      1.000
#6      6      1.000      1      1.000      1.000
#7      7      1.000      1      0.989      0.963
#8      8      1.000      1      1.000      1.000
#9      9      1.000      1      1.000      1.000
#10     10      1.000      1      0.988      1.000
#11     11      1.000      1      0.976      0.974
#12     12      1.000      1      0.978      0.983
#13     13      1.000      1      1.000      1.000
#14     14      1.000      1      0.913      0.845
#15     15      1.000      1      1.000      1.000
#16     16      1.000      1      0.994      1.000
#17     17      1.000      1      0.919      0.988
#18     18      1.000      1      1.000      1.000

#####
# BIAS AND MSE TO ANALYZE

# Read in bias and MSE from original results. Will NOT be using this
results <- read.table('results021210_1000_All.txt',header=T)

# Recalculate the means, bias, and MSE with the modifications made in the estimate section
means <- aggregate(est.new[,8:12],list(est.new$combo:est.new$Est),function(f){mean(f,na.rm=T)})

stretchmat <- model.matrix(~-1 + as.factor(rep(1:18,each=4)))
bias <- cbind(unique(est.new$Est:est.new$combo), means[,2:6]-stretchmat%*%frac.design[1:18,2:6])

MSE.temp <- matrix(NA,72,5)

```

```

for (i in 1:18){
  for(j in 1:4){
    for(k in 1:5){
      MSE.temp[(i-1)*4+j,k] <- MSE(matrix(frac.design[i,k+1],1000,1),
        est.new[(est.new$combo==i & est.new$Est==j),7+k])
    }
  }
}

# Write new results to table
results.new <- results
results.new[,8:17] <- cbind(bias[,2:6],MSE.temp)
#write.table(results.new,'results031610.txt',row.names=F,col.names=T)

# Read new results from table
results.new <- read.table('results031610.txt',header=T)

#####
      ### ANALYSIS ###
#####

#####
# TABLES OF RESULTS

library(xtable)

# By estimator--update for == for each estimator
# table of bias
temp <-results.new[results.new$Est==3,-c(7,13:17)]
row.names(temp) <- 1:18
xtable(temp,digits=c(0,0,0,1,0,2,1,5,5,5,5))

# table of MSE
temp <-results.new[results.new$Est==3,-c(7:12)]
row.names(temp) <- 1:18
xtable(temp,digits=c(0,0,0,1,0,2,1,5,3,3,5,5))

# All four estimators together
# table of bias
temp <- results.new[,c(7,1:6,8:12)]
temp[,1] <- rep(c("MLE-I","MLE-II","MLE-TZ","Bayes"),times=9)
xtable(temp,digits=c(0,0,0,0,1,0,2,1,5,5,5,5))

# table of MSE
temp <- results.new[,c(7,1:6,13:17)]
temp[,1] <- rep(c("MLE-I","MLE-II","MLE-TZ","Bayes"),times=9)
xtable(temp,digits=c(0,0,0,0,1,0,2,1,5,3,3,5,5))

#####
# PLOTS OF MEAN ESTIMATES

source('Plots.R')

legend.location <- c('topleft','topleft','topleft','bottomright','topleft','bottomleft',
  'bottomleft','bottomleft','topleft','topleft','topleft','topleft',
  'topleft','topleft','topleft','bottomleft','topleft','topleft','bottomright' )

```

```

pdf('ComboEstimates.pdf',width=8.5,height=15)
par(mfcol=c(6,3))
par(mar=c(3,3,4,2)+.01)
for(i in 1:18){
  # Plot truth
  segregplot(x=seq(0,1,length=frac.design[i,1]),y=NA,p=as.numeric(frac.design[i,2:6]),
             labelit=as.numeric(frac.design[i,2:6]))

# Add each estimator
  for(j in 1:4){
    segadd(p=as.numeric(means[(i-1)*4 + j,2:6]),color=(j+1))
  }
  title(main=paste("Combination",i),line=3)
  legend(legend.location[i],c("Truth","MLE-I","MLE-II","MLE-TZ","Bayes"),
        lwd=2,col=1:5,bty="n",y.intersp=.9,cex=1)
}
dev.off()
par(mfrow=c(1,1))
par(mar=c(5,4,4,2)+0.1)

#####
### SEEING HOW BIAS, MSE, ESTIMATES CORRELATE WITHIN ESTIMATOR

# Correlation of biases by estimator
for(i in 1:4){print(cor(results.new$results.new$Est==i,8:12))}

# Correlation of MSEs by estimator
for(i in 1:4){print(cor(results.new$results.new$Est==i,13:17))}

# Correlation of all results by estimator
for(i in 1:4){print(cor(results.new$results.new$Est==i,8:17))}

# Correlation of estimates by estimator
for(i in 1:4){print(cor(estimated[estimated$Est==i,8:12],use="complete.obs"))}

#####
# EXPLORATION OF SIGMA^2 BIASES

# Correct estimators by *n/(n-4)
MLEsig2means <- means$sig2hat[-seq(4,72,by=4)]
MLEsig2biases <- MLEsig2means-rep(frac.design[,6],each=3)
MLEsig2unbiased <- means$sig2hat[-seq(4,72,by=4)]*rep(c(20,50,100),each=18)/
                (rep(c(20,50,100),each=18)-4) - rep(frac.design[,6],each=3)

# Plot of uncorrected and corrected estimate bias
pdf('mleBiassig2.pdf',width=10,height=5)
par(mfrow=c(1,2))
plot(as.factor(frac.design[,1]),MLEsig2biases[seq(1,52,by=3)],
     ylab=expression(paste("Bias of ",hat(sigma)^2)),col=2,axes=F,at=c(1,4,7),
     xlim=c(.5,9.5),ylim=c(-.24,.085),main='Maximum Likelihood Estimators, biased')
plot(as.factor(frac.design[,1]),MLEsig2biases[seq(2,53,by=3)],add=T,axes=T,col=3,
     at=c(2,5,8),xlim=c(0,10))
plot(as.factor(frac.design[,1]),MLEsig2biases[seq(3,54,by=3)],add=T,axes=F,col=4,
     at=c(3,6,9),xlim=c(0,10))
abline(v=c(3.5,6.5))
abline(h=0,lty=2)
legend('bottomright',c('MLE-I','MLE-II','MLE-TZ'),lwd=5,col=2:4,bty='n')

```

```

plot(as.factor(frac.design[,1]),MLEsig2unbiased[seq(1,52,by=3)],ylab=
  expression(paste("Bias of ",hat(sigma)^2)),col=2,axes=F,at=c(1,4,7),
  xlim=c(.5,9.5),ylim=c(-.24,.085),main='Maximum Likelihood Estimators, unbiased')
plot(as.factor(frac.design[,1]),MLEsig2unbiased[seq(2,53,by=3)],add=T,axes=T,
  col=3,at=c(2,5,8),xlim=c(0,10))
plot(as.factor(frac.design[,1]),MLEsig2unbiased[seq(3,54,by=3)],add=T,axes=F,
  col=4,at=c(3,6,9),xlim=c(0,10))
abline(v=c(3.5,6.5))
abline(h=0,lty=2)
legend('bottomright',c('MLE-I','MLE-II',"MLE-TZ"),lwd=5,col=2:4,bty='n')
dev.off()
par(mfrow=c(1,1))

# Are the unbiased results really unbiased??? 1 sample t-tests
t.test(MLEsig2unbiased[seq(1,52,by=3)],mu=0) # t = -1.105, df = 17, p-value = 0.2845 YES
t.test(MLEsig2unbiased[seq(2,53,by=3)],mu=0) # t = 3.1628, df = 17, p-value = 0.005685 NO
t.test(MLEsig2unbiased[seq(3,52,by=3)],mu=0) # t = -0.1481, df = 16, p-value = 0.8841 YES

#####
# EXPLORATION OF MLE-I HIGH BIAS AND VARIANCE

# Why are some of these combos so off? Exploring combo17
# n b0 b1 b2 gam sig2
# 100 0 0.5 5 0.75 1

combo17 <- n100.data[n100.data$V1==17,9:108]

par(mfrow=c(3,3))
# beta1hat by beta2hat
plot(estimates[(estimates$combo==17 & estimates$Est==1),9:10],pch=20)
plot(estimates[(estimates$combo==17 & estimates$Est==1),9:10],xlim=c(-20,20),ylim=c(-10,20),pch=20)

# beta0hat by beta1hat
plot(estimates[(estimates$combo==17 & estimates$Est==1),8:9],pch=20)

# beta0hat by beta2hat
plot(estimates[(estimates$combo==17 & estimates$Est==1),c(8,10)],pch=20)

# slope 1 by slope 2
plot(estimates[(estimates$combo==17 & estimates$Est==1),9],estimates[(estimates$combo==17 &
  estimates$Est==1),9] + estimates[(estimates$combo==17 & estimates$Est==1),10],main="Estimate of
  first slope vs. Estimate of second slope", xlab=expression(hat(beta[1])),ylab=
  expression(hat(beta[1])+hat(beta[2])),pch=20)

# slope 1 by slope 2
plot(estimates[(estimates$combo==17 & estimates$Est==1),9],estimates[(estimates$combo==17 &
  estimates$Est==1),9] + estimates[(estimates$combo==17 & estimates$Est==1),10],main="Estimate
  of first slope vs. Estimate of second slope", xlab=expression(hat(beta[1])),ylab=
  expression(hat(beta[1])+hat(beta[2])),pch=20,ylim=c(-10,10),xlim=c(-10,10))

# Gamma by slope 1
plot(estimates[(estimates$combo==17 & estimates$Est==1),11],estimates[(estimates$combo==17 &
  estimates$Est==1),9], main="Estimate of change-point vs. Estimate of first slope",xlab=
  expression(hat(gamma)),ylab=expression(hat(beta[1])),pch=20)

# Gamma by beta2hat
plot(estimates[(estimates$combo==17 & estimates$Est==1),11],estimates[(estimates$combo==17 &
  estimates$Est==1),10], main="Estimate of change-point vs. Estimate of change in slope",
  xlab=expression(hat(gamma)),ylab=expression(hat(beta[2])),pch=20)

# Gamma by slope 2
plot(estimates[(estimates$combo==17 & estimates$Est==1),11],estimates[(estimates$combo==17 &

```

```

estimates$Est==1),9]+ estimates[(estimates$combo==17 & estimates$Est==1),10],main="Estimate
of change-point vs. Estimate of second slope", xlab=expression(hat(gamma)),ylab=
expression(hat(beta[1]) + hat(beta[2])),pch=20)

par(mfrow=c(1,1))

#####
# An example of one dataset that does the weird MLE-I thing
par(mfrow=c(1,1))
x <- seq(0,1,length=100)
y <- as.numeric(combo17[4,])
p <- as.numeric(frac.design[17,2:6])
phat <- as.numeric(estimates[(estimates$combo==17 & estimates$iter==4 & estimates$Est==1),8:12])

fx(x,y,p) #146.0016
fx(x,y,phat) #142.8433 Has a smaller negative log-likelihood
logLik(lm(y~x)) #143.4294 Segmented model still better than non

# Is this a reasonable way to test changepointing/no changepointing -- out of scope of paper
1-pchisq(142.8433/143.4294,2) #.60 Not compelling evidence to keep change

# Plotting the likelihoods (partial -- holding others constant)
# Black lines hold constant and the true values and colored lines indicate the estimated values
par(mfrow=c(3,2))

  segregplot(x,y,p,color=1)
  segadd(phat,color=2)
  # -----
  beta0try <- seq(-5,5,length=100)
  fxtry <- sapply(beta0try,function(f) fx(x,y,p=c(f,phat[2:5])))
  plot(beta0try,fxtry,pch=20,type='l',col=2)

  beta0try <- seq(-5,5,length=100)
  fxtry <- sapply(beta0try,function(f) fx(x,y,p=c(f,p[2:5])))
  lines(beta0try,fxtry,pch=20)
  # -----
  beta1try <- seq(-40,10,length=100)
  fxtry <- sapply(beta1try,function(f) fx(x,y,p=c(phat[1],f,phat[3:5])))
  plot(beta1try,fxtry,pch=20,type='l',col=2)

  beta1try <- seq(-40,10,length=100)
  fxtry <- sapply(beta1try,function(f) fx(x,y,p=c(p[1],f,p[3:5])))
  lines(beta1try,fxtry,pch=20)
  # -----
  beta2try <- seq(-10,60,length=100)
  fxtry <- sapply(beta2try,function(f) fx(x,y,p=c(phat[1:2],f,phat[4:5])))
  plot(beta2try,fxtry,pch=20,type='l',col=2)

  beta2try <- seq(-10,60,length=100)
  fxtry <- sapply(beta2try,function(f) fx(x,y,p=c(p[1:2],f,p[4:5])))
  lines(beta2try,fxtry,pch=20)
  # -----
  gammatry <- seq(0,1,length=1000)
  fxtry <- sapply(gammatry,function(f) fx(x,y,p=c(phat[1:3],f,phat[5])))
  plot(gammatry,fxtry,pch=20,type='l',col=2)

  gammatry <- seq(0,1,length=1000)
  fxtry <- sapply(gammatry,function(f) fx(x,y,p=c(p[1:3],f,p[5])))
  lines(gammatry,fxtry,pch=20)
  # -----
  sigma2try <- seq(0,2,length=100)
  fxtry <- sapply(sigma2try,function(f) fx(x,y,p=c(phat[1:4],f)))
  plot(sigma2try,fxtry,pch=20,type='l',col=2)

```

```

sigma2try <- seq(max(0,2,length=100)
fxtry <- sapply(sigma2try,function(f) fx(x,y,p=c(p[1:4],f)))
lines(sigma2try,fxtry,pch=20)

par(mfrow=c(1,1))

ttemp <- MLE.I(p,x,y)

pdf('ChangepointMLEIeffect.pdf',width=10,height=5)
par(mfrow=c(1,2))
segregplot(x,y,p=p,labelit=phat,col=1)
segadd(phat,col=2)
legend('bottomright',c("Truth","MLE-I"),bty="n",lwd=2,col=1:2)

plot(-ttemp$like.min,ylab="Log Likelihood",main='Discrete Log Likelihood',pch=20)
points(3,-ttemp$like.min[3],col=2,pch=6,cex=2)
dev.off()
par(mfrow=c(1,1))

#####
# Posterior examples of 1 of each combination -- new data

set.seed(032410)
pdf('BayesGamPost.pdf',width=10,height=15)
par(mfcol=c(6,3))
par(mar=c(3,3,4,2)+.01)
for(j in 1:18){
  n <- frac.design[j,1]
  p <- as.numeric(frac.design[j,2:6])
  x <- segregx(n)
  y <- segregy(p,x)
  ttemp2 <- Bayes(as.numeric(p),x,y,as.numeric(candidatevars[j,]))
  plot(density(ttemp2$postests[,4]),main=paste("Combo ",j," Change-point at ",p[4],sep=""),
       xlab="",ylab="",xlim=c(0,1))
  hist(ttemp2$postests[,4],freq=FALSE,add=T)
  abline(v=p[4],lwd=3)
  abline(v=ttemp2$param.est[4],col=5,lwd=3)
}
dev.off()
par(mfrow=c(1,1))
par(mar=c(5,4,4,2)+.01)

#####
##### MAIN CHUNK OF ANALYSIS #####
#####

# HOW EACH ESTIMATOR IS AFFECTED BY EACH FACTOR: EACH ESTIMATOR SEPARATELY

#####

##### Plot Raw Effects: These are used to identify possible trends in what is
# happening after ANOVA indicates an effect in one case. Ideally, we should do tests
# to see if these effects are significant (ie a linear or quadratic effect, etc)
# but these graphs will have to suffice for now.

outs <- names(results.new[,8:17])
ests <- c('MLEI','MLE2','MLETZ',"BAYES")

```

```

##### BIAS
graphics.off()
for(i in 1:5){
  pdf(c('BiasByn.pdf', 'BiasByb1.pdf', 'BiasByb2.pdf', 'BiasBygamma.pdf',
        'BiasBysigma2.pdf')[i], width=12, height=9)
  par(mfrow=c(4,5))
}

for (i in 1:4){
for (j in 1:5){

dev.set(2)
plot(as.factor(results.new$n[results.new$Est==i]), results.new[results.new$Est==i, (j+7)],
pch=20, xlab="n", ylab=paste(ests[i], main=outs[j]), main=outs[j])
abline(h=0, col='red')

dev.set(3)
plot(as.factor(results.new$b1[results.new$Est==i]), results.new[results.new$Est==i, (j+7)],
pch=20, xlab="b1", ylab=paste(ests[i], main=outs[j]), main=outs[j])
abline(h=0, col='red')

dev.set(4)
plot(as.factor(results.new$b2[results.new$Est==i]), results.new[results.new$Est==i, (j+7)],
pch=20, xlab="b2", ylab=paste(ests[i], main=outs[j]), main=outs[j])
abline(h=0, col='red')

dev.set(5)
plot(as.factor(results.new$gam[results.new$Est==i]), results.new[results.new$Est==i, (j+7)],
pch=20, xlab="gam", ylab=paste(ests[i], main=outs[j]), main=outs[j])
abline(h=0, col='red')

dev.set(6)
plot(as.factor(results.new$sig2[results.new$Est==i]), results.new[results.new$Est==i, (j+7)],
pch=20, xlab="sig2", ylab=paste(ests[i], main=outs[j]), main=outs[j])
abline(h=0, col='red')
}
}
for (i in 1:5) dev.off(i+1)

##### MSE
graphics.off()
for(i in 1:5){
  pdf(c('MSEByn.pdf', 'MSEByb1.pdf', 'MSEByb2.pdf', 'MSEBygamma.pdf',
        'MSEBysigma2.pdf')[i], width=12, height=9)
  par(mfrow=c(4,5))
}

for (i in 1:4){
for (j in 1:5){

dev.set(2)
plot(as.factor(results.new$n[results.new$Est==i]), results.new[results.new$Est==i, (j+12)],
pch=20, xlab="n", ylab=paste(ests[i], main=outs[j+5]), main=outs[j+5])
abline(h=0, col='red')

dev.set(3)
plot(as.factor(results.new$b1[results.new$Est==i]), results.new[results.new$Est==i, (j+12)],
pch=20, xlab="b1", ylab=paste(ests[i], main=outs[j+5]), main=outs[j+5])
abline(h=0, col='red')

dev.set(4)
plot(as.factor(results.new$b2[results.new$Est==i]), results.new[results.new$Est==i, (j+12)],

```

```

pch=20,xlab="b2",ylab=paste(ests[i],main=outs[j+5]),main=outs[j+5])
abline(h=0,col='red')

dev.set(5)
plot(as.factor(results.new$gam[results.new$Est==i]),results.new[results.new$Est==i,(j+12)],
pch=20,xlab="gam",ylab=paste(ests[i],main=outs[j+5]),main=outs[j+5])
abline(h=0,col='red')

dev.set(6)
plot(as.factor(results.new$sig2[results.new$Est==i]),results.new[results.new$Est==i,(j+12)],
pch=20,xlab="sig2",ylab=paste(ests[i],main=outs[j+5]),main=outs[j+5])
abline(h=0,col='red')
}
}
for (i in 1:5) dev.off(i+1)

#####

# This model doesn't account for the fact that these were run over different combos.
# But it is kind of nice because we can do a cell means and we DO want to test if the cell
# means are significantly different than 0 -- problem though is that if it is positively
# biased on one combo and negatively on another, it may appear unbiased

fit.multiple1 <- lm(as.matrix(results.new[,8:17]) ~ -1 + as.factor(Est),data=results.new)
summary(fit.multiple1)

par(mfrow=c(3,2))
for(i in 1:5) plot(as.factor(results.new$Est),fit.multiple1$residuals[,i])

#####

# 40 UNIVARIATE ANOVA ANALYSES (4 ESTIMATORS x 10 RESPONSES)
# obtain p-values -- not going to adjust for multiple testing
# Each anova is performed on the set of 5 explanatory variables that were
# varied in the analysis (n, b0, b1, b2, gam)

pvalues.biasMSE <- array(dim=c(5,10,4))
dimnames(pvalues.biasMSE)[[1]] <- c('n','beta_1','beta_2','gamma','sigma^2')
dimnames(pvalues.biasMSE)[[2]] <- c('beta_0','beta_1','beta_2','gamma','sigma^2',
'beta_0','beta_1','beta_2','gamma','sigma^2')
dimnames(pvalues.biasMSE)[[3]] <- c('MLE-I',"MLE-II","MLE-TZ","Bayes")

# Just doing good-ole ANOVA (TYPE I, but balanced so same as TYPE III)
for (i in 1:4){
for (j in 1:10){
fit.single2 <- lm(results.new[, (j+7)] ~ as.factor(n) + as.factor(b1) + as.factor(b2) +
as.factor(gam) + as.factor(sig2),data=results.new,subset=results.new$Est==i)
pvalues.biasMSE[,j,i] <- round(t(t(anova(fit.single2)[1:5,5])),3)
}
}
print(pvalues.biasMSE,digits=3)

# Create a table to print in project
pvalues.toprint <- pvalues.biasMSE
for(k in 1:4){
pvalues.toprint <- as.data.frame(pvalues.biasMSE[,k])
for (i in 1:5){
for(j in 1:10){
temp <- ifelse(pvalues.toprint[i,j]<.05,"*","")
pvalues.toprint[i,j] <- paste(pvalues.toprint[i,j],temp)
}
}
}

```

```

    }
  }
  print("#####")
  print(xtable(pvalues.toprint))
}

#####
# Is ANOVA appropriate -- have assumptions of normality been met?
# Here we will plot the residuals
# Transformations tried -- absolute value on bias which then requires a log transform

graphics.off()
for(i in 1:3){quartz(); par(mfrow=c(4,5))}

# BIAS: ANOVA + RESIDUAL PLOTS
for (i in 1:4){
print(paste("#####",ests[i],"#####"))
for (j in 1:5){

print(paste("#####",i,",",j,"#####"))
fit.single2 <- lm(results.new[, (j+7)] ~ as.factor(n) + as.factor(b1) + as.factor(b2) +
as.factor(gam) + as.factor(sig2),data=results.new,subset=results.new$Est==i)
print(anova(fit.single2))

dev.set(2)
plot(residuals(fit.single2),pch=20,ylab=paste(ests[i],"Residuals"),main=outs[j])

dev.set(3)
plot(rstandard(fit.single2),pch=20,ylab=paste(ests[i],"Standardized"),main=outs[j])

dev.set(4)
plot(rstudent(fit.single2),pch=20,ylab=paste(ests[i],"Studentized Del"),main=outs[j])
}
}

# MSE: ANOVA + RESIDUAL PLOTS

graphics.off()
for(i in 1:3){quartz(); par(mfrow=c(4,5))}

# Just doing good-ole ANOVA (TYPE I, but balanced so same as TYPE III)
for (i in 1:4){
for (j in 1:5){
fit.single2 <- lm(results.new[, (j+12)] ~ as.factor(n) + as.factor(b1) + as.factor(b2) +
as.factor(gam) + as.factor(sig2),data=results.new,subset=results.new$Est==i)

dev.set(2)
plot(residuals(fit.single2),pch=20,ylab="Residuals")

dev.set(3)
plot(rstandard(fit.single2),pch=20,ylab="Standardized")

dev.set(4)
plot(rstudent(fit.single2),pch=20,ylab="Studentized Del")
}
}
}

```

```

#####
# Get some plots to illustrate effects

# Effect of b2 on BIAS of b0, b1, b2
ests <- c('MLE-I', 'MLE-II', 'MLE-TZ', "Bayes")

pdf('BiasByb2illustrate.pdf', width=8, height=10)
par(mfrow=c(4,3), mar=c(5,4.5,4,2)+.1)
for (i in 1:4){

plot(as.factor(results.new$b2[results.new$Est==i]), results.new[results.new$Est==i, (1+7)],
pch=20, xlab=expression(beta[2]), ylab=expression(paste("Bias ", hat(beta)[0])), main=ests[i])
abline(h=0, col='red')

plot(as.factor(results.new$b2[results.new$Est==i]), results.new[results.new$Est==i, (2+7)],
pch=20, xlab=expression(beta[2]), ylab=expression(paste("Bias ", hat(beta)[1])), main=ests[i])
abline(h=0, col='red')

plot(as.factor(results.new$b2[results.new$Est==i]), results.new[results.new$Est==i, (3+7)],
pch=20, xlab=expression(beta[2]), ylab=expression(paste("Bias ", hat(beta)[2])), main=ests[i])
abline(h=0, col='red')
}
dev.off()

par(mar=c(5,4,4,2)+.1)

#### MSE
graphics.off()
for(i in 1:5){
  pdf(c('MSEByn.pdf', 'MSEByb1.pdf', 'MSEByb2.pdf', 'MSEBygamma.pdf',
        'MSEBysigma2.pdf')[i], width=12, height=9)
  par(mfrow=c(4,5))
}

for (i in 1:4){
for (j in 1:5){

dev.set(2)
plot(as.factor(results.new$n[results.new$Est==i]), results.new[results.new$Est==i, (j+12)],
pch=20, xlab="n", ylab=paste(ests[i], main=outs[j+5]), main=outs[j+5])
abline(h=0, col='red')

dev.set(3)
plot(as.factor(results.new$b1[results.new$Est==i]), results.new[results.new$Est==i, (j+12)],
pch=20, xlab="b1", ylab=paste(ests[i], main=outs[j+5]), main=outs[j+5])
abline(h=0, col='red')

dev.set(4)
plot(as.factor(results.new$b2[results.new$Est==i]), results.new[results.new$Est==i, (j+12)],
pch=20, xlab="b2", ylab=paste(ests[i], main=outs[j+5]), main=outs[j+5])
abline(h=0, col='red')

dev.set(5)
plot(as.factor(results.new$gam[results.new$Est==i]), results.new[results.new$Est==i, (j+12)],
pch=20, xlab="gam", ylab=paste(ests[i], main=outs[j+5]), main=outs[j+5])
abline(h=0, col='red')

dev.set(6)
plot(as.factor(results.new$sig2[results.new$Est==i]), results.new[results.new$Est==i, (j+12)],
pch=20, xlab="sig2", ylab=paste(ests[i], main=outs[j+5]), main=outs[j+5])
abline(h=0, col='red')
}
}

```

```

}
}
for (i in 1:5) dev.off(i+1)

```

C.5 APPLICATION

```

#####

#####          DATA          #####

#####
crack <- read.table('mydata.dat')
x <- crack$X
y <- crack$data1

#####

#####          INITIAL PLOT          #####

#####

source('Plots.R')

plot(x,y,pch=20,main='Aging of Weapons',xlab='Age (masked)',ylab='Crack Length')

#pdf('CrackApp.pdf',width=6,height=4.5)
#plot(x,y,pch=20,main='Aging of Weapons',xlab='Age (masked)',ylab='Crack Length')
#dev.off()

#####

#####          ESTIMATORS          #####

#####

# Section contains code for estimators used for the application -- tuned to the application
# MAXIMUM LIKELIHOOD I (maximize discrete marginal likelihood)
# MAXIMUM LIKELIHOOD II (numerically approximate gradient and Hessian)
# TISHLER AND ZANG MAXIMUM LIKELIHOOD ALTERNATIVE (a bent cable approach)
# BAYESIAN POSTERIOR MEAN (a Bayesian approach)

#####
#####          MAXIMUM LIKELIHOOD I AND II          #####
#####

# Maximize log-likelihood for all 5 params together
# p = [beta_0
#       beta_1
#       beta_2
#       gamma
#       sig^2]

# The Negative log-likelihood function (objective function)
fx <- function(x,y,p) {
  .5*length(x)*log(2*pi)+.5*length(x)*log(p[5]) +
  .5/p[5]*sum((y-p[1]-p[2]*x-p[3]*(x-p[4])*(x>p[4]))^2)
}

```

```

# Returns the gradient of the negative log-likelihood
G <- function(p,x,y){
G <- rep(NA,5)
G[1] <- 1/p[5]*sum(y - p[1] - p[2]*x - p[3]*(x-p[4])*(x>p[4]))
G[2] <- 1/p[5]*sum(x*(y - p[1] - p[2]*x - p[3]*(x-p[4])*(x>p[4])))
G[3] <- 1/p[5]*sum((x-p[4])*(x>p[4])*(y - p[1] - p[2]*x - p[3]*(x-p[4])*(x>p[4])))
G[4] <- -1/p[5]*sum(p[3]*(x>p[4])*(y - p[1] - p[2]*x - p[3]*(x-p[4])))
G[5] <- -length(x)/(2*p[5]) + 1/(2*p[5]^2)*sum((y-p[1]-p[2]*x-p[3]*(x-p[4])*(x>p[4]))^2)
-G
}

```

```

# Returns the Hessian of the negative log-likelihood
H <- function(p,x,y){
H <- matrix(0,5,5)
H[1,1]<- -length(x)/p[5]
H[1,2]<- -1/p[5]*sum(x)
H[1,3]<- -1/p[5]*sum((x>p[4])*(x-p[4]))
H[1,4]<- 1/p[5]*p[3]*sum(x>p[4])
H[1,5]<- -1/(p[5]^2)*sum(y - p[1] - p[2]*x - p[3]*(x-p[4])*(x>p[4]))

```

```

H[2,2]<- -1/p[5]*sum(x^2)
H[2,3]<- -1/p[5]*sum((x>p[4])*x*(x-p[4]))
H[2,4]<- 1/p[5]*p[3]*sum(x*(x>p[4]))
H[2,5]<- -1/(p[5]^2)*sum(x*(y - p[1] - p[2]*x - p[3]*(x-p[4])*(x>p[4])))

```

```

H[3,3]<- -1/p[5]*sum((x>p[4])*(x-p[4])^2)
H[3,4]<- -1/p[5]*sum((x>p[4])*(y-p[1]-p[2]*x - 2*p[3]*(x - p[4])))
H[3,5]<- -1/(p[5]^2)*sum((x-p[4])*(x>p[4])*(y - p[1] - p[2]*x - p[3]*(x-p[4])))

```

```

H[4,4]<- -1/p[5]*sum(x>p[4])*p[3]^2
H[4,5]<- 1/(p[5]^2)*sum(p[3]*(x>p[4])*(y - p[1] - p[2]*x - p[3]*(x-p[4])))

```

```

H[5,5]<- length(x)/(2*p[5]^2) - 1/(p[5]^3)*sum((y-p[1]-p[2]*x-p[3]*(x-p[4])*(x>p[4]))^2)

```

```

H <- H + t(H*upper.tri(H))
-H
}

```

```

#####

```

```

# MLE-I
# Maximize marginal likelihood as a function of m (all possible divisions)
# of data into 2 segments. For each m, use constrained NR to estimate
# between gradient discontinuities, constraining  $x_i \leq \text{gamma} \leq x_{i+1}$ 

```

```

MLE.I<-function(p,x,y){
like.min <- rep(NA,length(x))
params <- matrix(NA,length(x),5)

#Helps to fix scaling problems (params on diff scales, some large some small)
scalingfacs <- c(1/0.01,1/-.01,1/.04,1/40,1/.05)

# For each possible division, find the constrained MLEs
for (i in 2:(length(x)-2)){
temp <- nlmnb(start=p,objective=fx,x=x,y=y,gradient=G,hessian=H,scale=scalingfacs,
lower=c(-Inf,-Inf,-Inf,x[i],0),upper=c(Inf,Inf,Inf,x[i+1],Inf))
params[i,] <- temp$par
like.min[i] <- temp$objective
p <- temp$par
scalingfacs <- 1/abs(temp$par)
}
}

```

```

use <- min(which(like.min==min(like.min,na.rm=T))) #Multiple optima: use first
list(param.est=params[use,],like.min=like.min)
}

#####
# MLE-II
# Using Newton-Raphson and numerically calculating gradient and hessian

# The Negative log-likelihood function (objective function)
fx2 <- function(p,x,y) {

nnl <- .5*length(x)*log(2*pi)+.5*length(x)*log(p[5]) +
      .5/p[5]*sum((y-p[1]-p[2]*x-p[3]*(x-p[4])*(x>p[4]))^2)
if (p[4]<min(x) | p[4]>max(x) | p[5]<0) {nnl <- 100000000}
# Constrains gamma and sigma to be in appropriate range

return(nnl)
}

MLE.II<-function(p,x,y){
storecode <- 6
while(storecode>=3){
start <- ifelse(storecode==6,0,runif(1,-.1,.1))
store <- nlm(fx2,p=p+start,x=x,y=y)
storecode<- as.numeric(store$code)
}
list(param.est=store$estimate,conv=storecode,start=start)
}

#####
##### TISHLER AND ZANG MAXIMUM LIKELIHOOD #####
#####

# Maximize log-likelihood for all 5 params together
# NOTE PARAMETERIZATION: reparameterize starting values, back transform to get estimates
# p = [a10 = beta_0
#      a11 = beta_1
#      a20 = beta_0 - beta_2*gamma
#      a21 = beta_1 + beta_2
#      sig^2]

# objective function, gradient, and hessian for MLETZ - does use scaling number to
# avoid numerical difficulties
f.1.nlm <- function (p, x, y, B, r) {
p1 <- p[1]
p2 <- p[2]
p3 <- p[3]
p4 <- p[4]
p5 <- p[5]
line1 <- B + cbind(1,x)%*%p[1:2]
line2 <- B + cbind(1,x)%*%p[3:4]
scalenum <- apply(cbind(line1,line2),1,function(f) if(r>0){ max(f)}else{min(f)})
.expr5 <- (B + p1 + p2 * x)/scalenum
.expr10 <- (B + p3 + p4 * x)/scalenum
.expr12 <- .expr5^r + .expr10^r
.expr13 <- 1/r
.expr16 <- B + y - scalenum * .expr12^.expr13
.expr18 <- .expr13 - 1
.expr19 <- .expr12^.expr18
.expr20 <- r - 1

```

```

.expr21 <- .expr5^.expr20
.expr22 <- 1/scalenum
.expr23 <- r * .expr22
.expr24 <- .expr21 * .expr23
.expr25 <- .expr13 * .expr24
.expr27 <- scalenum * (.expr19 * .expr25)
.expr32 <- .expr12^(.expr18 - 1)
.expr36 <- .expr20 - 1
.expr37 <- .expr5^.expr36
.expr38 <- .expr20 * .expr22
.expr50 <- x/scalenum
.expr51 <- r * .expr50
.expr52 <- .expr21 * .expr51
.expr54 <- .expr32 * (.expr18 * .expr52)
.expr56 <- .expr20 * .expr50
.expr57 <- .expr37 * .expr56
.expr64 <- .expr13 * .expr52
.expr66 <- scalenum * (.expr19 * .expr64)
.expr71 <- .expr10^.expr20
.expr72 <- .expr71 * .expr23
.expr74 <- .expr32 * (.expr18 * .expr72)
.expr78 <- .expr13 * .expr72
.expr80 <- scalenum * (.expr19 * .expr78)
.expr85 <- .expr71 * .expr51
.expr87 <- .expr32 * (.expr18 * .expr85)
.expr91 <- .expr13 * .expr85
.expr93 <- scalenum * (.expr19 * .expr91)
.expr130 <- .expr10^.expr36
.expr143 <- .expr130 * .expr56

# Objective Function
if(p5<0) {.value<-100000000000} else{
.value <- .5*length(x)*log(2*pi) + .5*length(x)*log(p5) + .5/p5*sum(.expr16^2)}

# Gradient
.grad <- array(0, c(1L, 5L), list(NULL, c("p1", "p2", "p3", "p4", "p5")))
.grad[, "p1"] <- -1/p5*sum(.expr27 * .expr16)
.grad[, "p2"] <- -1/p5*sum(.expr66 * .expr16)
.grad[, "p3"] <- -1/p5*sum(.expr80 * .expr16)
.grad[, "p4"] <- -1/p5*sum(.expr93 * .expr16)
.grad[, "p5"] <- length(x)/(2*p5) - 1/(2*p5^2)*sum(.expr16^2)

# Hessian
.hessian <- array(0, c(5L, 5L), list( c("p1", "p2", "p3", "p4", "p5"),
c("p1", "p2", "p3", "p4", "p5")))

.hessian["p1", "p1"] <- -1/p5*sum(scalenum * (.expr32 * (.expr18 *
.expr24) * .expr25 + .expr19 * (.expr13 * (.expr37 *
.expr38 * .expr23))) * .expr16 - .expr27 * .expr27)
.hessian["p1", "p2"] <- .hessian["p2", "p1"] <- -1/p5 *
sum(scalenum * (.expr54 * .expr25 + .expr19 * (.expr13 *
(.expr57 * .expr23))) * .expr16 - .expr27 * .expr66)
.hessian["p1", "p3"] <- .hessian["p3", "p1"] <- -1/p5 *
sum(scalenum * (.expr74 * .expr25) * .expr16 - .expr27 * .expr80)
.hessian["p1", "p4"] <- .hessian["p4", "p1"] <- -1/p5 *
sum(scalenum * (.expr87 * .expr25) * .expr16 - .expr27 * .expr93)
.hessian["p1", "p5"] <- .hessian["p5", "p1"] <- -1/(p5)*.grad[, "p1"]

.hessian["p2", "p2"] <- -1/p5 *sum(scalenum * (.expr54 * .expr64 +
.expr19 * (.expr13 * (.expr57 * .expr51))) * .expr16 - .expr66 * .expr66)
.hessian["p2", "p3"] <- .hessian["p3", "p2"] <- -1/p5 *
sum(scalenum * (.expr74 * .expr64) * .expr16 - .expr66 * .expr80)
.hessian["p2", "p4"] <- .hessian["p4", "p2"] <- -1/p5 *
sum(scalenum * (.expr87 * .expr64) * .expr16 - .expr66 * .expr93)

```

```

.hessian["p2", "p5"] <- .hessian["p5", "p2"] <- -1/(p5)*.grad[, "p2"]

.hessian["p3", "p3"] <- -1/p5 *sum(scalenum * (.expr74 * .expr78 +
.expr19 * (.expr13 * (.expr130 * .expr38 * .expr23))) *
.expr16 - .expr80 * .expr80)
.hessian["p3", "p4"] <- .hessian["p4", "p3"] <- -1/p5 *
sum(scalenum * (.expr87 * .expr78 + .expr19 * (.expr13 *
(.expr143 * .expr23))) * .expr16 - .expr80 * .expr93)
.hessian["p3", "p5"] <- .hessian["p5", "p3"] <- -1/(p5)*.grad[, "p3"]

.hessian["p4", "p4"] <- -1/p5 *sum(scalenum * (.expr87 * .expr91 +
.expr19 * (.expr13 * (.expr143 * .expr51))) * .expr16 - .expr93 * .expr93)
.hessian["p4", "p5"] <- .hessian["p5", "p4"] <- -1/(p5)*.grad[, "p4"]

.hessian["p5", "p5"] <- -length(x)/(2*p5^2) +1/(p5^3)*sum(.expr16^2)

attr(.value, "gradient") <- .grad
attr(.value, "hessian") <- .hessian
.value

}

fromPto2LineP <- function(p){c(p[1:2],p[1]-p[3]*p[4],p[2]+p[3],p[5])}
from2LinePtoP <- function(p){c(p[1:2],p[4]-p[2],(p[3]-p[1])/(p[2]-p[4]),p[5])}

MLETZ<-function(p,x,y){
change <- 20 # This must be positive and even
r <- ifelse(p[3]<0,-change,change) # Choose initial p depending on concave/convex
B <- 1.01*max(abs(y)) # Add large positive value to max all positive
last <- fromPto2LineP(p) # Starting values for parameter estimates
keepgoing <- TRUE

# start each iteration with param estimates at last iteration
while(keepgoing){
# Use some error handling so simulation will record and continue after difficulties
warn.error <- FALSE
# Including "typical size" from transformation of one of the other estimators
# to help convergence happen quicker
tryCatch( store<-nlm(f.1.nlm, p=last ,x=x, y=y, r=r, B=B,typsize=c(.0001,.001,.04,0.02,.03),
iterlim=20000, check.analyticals=F)
,error = function(err) {keepgoing <-FALSE; warn.error <-TRUE}
)
current <- store$estimate
line1 <- B+current[1] + current[2]*x
line2 <- B+current[3] + current[4]*x
if(sum(c(line1,line2)<0)!=0){ # Check if all positive
B <- -1.01*min(line1-B,line2-B) # If not all positive: Adjust B and r
r <- ifelse(p[3]<0,r-change,r+change)
} else
if(sum((last-current)^2)^.5<.01){ # Check if within tolerance of last estimate
keepgoing=FALSE # If in tolerance: STOP
} else { # If not in tolerance: Adjust r
last <- current
r <- ifelse(p[3]<0,r-change,r+change)
}
}

# Returns param.est (final estimates), conv (1-5 R codes, 8 warn, 9 err), r (final r/p)
list(param.est=from2LinePtoP(store$estimate),
conv=ifelse(warn.error,9,store$code),r=r)
}

```

```

#####
##### BAYESIAN POSTERIOR MEAN #####
#####

# Takes as input x, y (data), p, the true parameter value which it uses as its
# starting values and cand.var, which are particularly tuned to the parameter
# combinations used in simulation

Bayes<-function(p,x,y,cand.var=c( 0.004, 0.000005, .00001, 25)){

# Values for the priors -- diffuse considering the values of the parameters
# For purposes of application - gamma is done with a normal distribution
  m.gam<- 20      #Normal
  s2.gam<-44
  m.b0 <- 0       #Normal
  s2.b0<- 10
  m.b1 <- 0       #Normal
  s2.b1<- 10
  m.b2 <- 0       #Normal
  s2.b2<- 10
  a.sig2 <- 2.01  #Inverse Gamma
  b.sig2 <- 10
  #1/(a.sig2-1)/b.sig2
  #1/((a.sig2-1)^2*b.sig2^2*(a.sig2-2))

  #Plot priors
  #plot(seq(0,40,length=100), dnorm(seq(0,40,length=100),m.gam,sqrt(s2.gam)),type='l') #gamma
  #plot(seq(-20,20,length=100), dnorm(seq(-20,20,length=100),m.b0,sqrt(s2.b0)),type='l') #b0
  #plot(seq(-20,20,length=100), dnorm(seq(-20,20,length=100),m.b1,sqrt(s2.b1)),type='l') #b1
  #plot(seq(-20,20,length=100), dnorm(seq(-20,20,length=100),m.b2,sqrt(s2.b2)),type='l') #b2
  #temp1=1/rgamma(100000,a.sig2,yscale=b.sig2)
  #plot(density(temp1)) #sig2

# Starting values for MCMC iterations: Solution from segmented library -- Same
# start as all other estimators receive
  niter <- 110000
  gam<-rep(p[4],niter)
  sig2<-rep(p[5],niter)
  b0 <-rep(p[1],niter)
  b1 <-rep(p[2],niter)
  b2 <-rep(p[3],niter)

# Candidate distribution variances - Using normals on all (and truncation on gamma)
  cand.gam <- as.numeric(cand.var[4])
  cand.b0 <- as.numeric(cand.var[1])
  cand.b1 <- as.numeric(cand.var[2])
  cand.b2 <- as.numeric(cand.var[3])

  # Initialize vectors for acceptance rates
  ar.gam <- ar.b0 <- ar.b1 <- ar.b2 <-1

# Run Iterations and Get Draws
for(i in 2:niter){
# update for gamma
gam[i] <- gam[i-1]
old <- gam[i-1]
new <- rnorm(1,old,sqrt(cand.gam))
if(new>=min(x) && new<=max(x)) {
llo <- -.5/sig2[i-1]*sum((y - (b0[i-1] + b1[i-1]*x + b2[i-1]*(x-old)*(x>old)))^2)-
.5/s2.gam*(old - m.gam)^2
lln <- -.5/sig2[i-1]*sum((y - (b0[i-1] + b1[i-1]*x + b2[i-1]*(x-new)*(x>new)))^2)-
.5/s2.gam*(new - m.gam)^2
if(log(runif(1,0,1))<(lln-llo)){ gam[i]<-new ; if(i>100) {ar.gam <-ar.gam+1}}
}
}

```

```

# update for sig2 - Complete conditional is Inverse gamma
sig2[i] <- 1/rgamma(1,length(x)/2 + a.sig2,scale=(1/b.sig2 + .5*sum((y - (b0[i-1] +
b1[i-1]*x + b2[i-1]*(x-gam[i])*(x>gam[i]))^2))^-1))

# update for b0
b0[i] <- b0[i-1]
old <- b0[i-1]
new <- rnorm(1,old,sqrt(cand.b0))
llo <- -.5/sig2[i]*sum((y - (old + b1[i-1]*x + b2[i-1]*
(x-gam[i])*(x>gam[i]))^2) - .5/s2.b0*(old - m.b0)^2
lln <- -.5/sig2[i]*sum((y - (new + b1[i-1]*x + b2[i-1]*(x-gam[i])*
(x>gam[i]))^2) - .5/s2.b0*(new - m.b0)^2
if(log(runif(1,0,1))<(lln-llo)){ b0[i]<-new ; if(i>100) {ar.b0 <-ar.b0+1}}

# update for b1
b1[i] <- b1[i-1]
old <- b1[i-1]
new <- rnorm(1,old,sqrt(cand.b1))
llo <- -.5/sig2[i]*sum((y - (b0[i] + old*x + b2[i-1]*(x-gam[i])*(x>gam[i]))^2) -
.5/s2.b1*(old - m.b1)^2
lln <- -.5/sig2[i]*sum((y - (b0[i] + new*x + b2[i-1]*(x-gam[i])*(x>gam[i]))^2) -
.5/s2.b1*(new - m.b1)^2
if(log(runif(1,0,1))<(lln-llo)){ b1[i]<-new ; if(i>100) {ar.b1 <-ar.b1+1} }

# update for b2
b2[i] <- b2[i-1]
old <- b2[i-1]
new <- rnorm(1,old,sqrt(cand.b2))
llo <- -.5/sig2[i]*sum((y - (b0[i] + b1[i]*x + old*(x-gam[i])*(x>gam[i]))^2) -
.5/s2.b2*(old - m.b2)^2
lln <- -.5/sig2[i]*sum((y - (b0[i] + b1[i]*x + new*(x-gam[i])*(x>gam[i]))^2) -
.5/s2.b2*(new - m.b2)^2
if(log(runif(1,0,1))<(lln-llo)){ b2[i]<-new ; if(i>100) {ar.b2 <-ar.b2+1}}
}

out <- cbind(b0,b1,b2,gam,sig2)

#par(mfrow=c(3,2))
#for (i in 1:5) plot(out[,i],type='l')
#par(mfrow=c(1,1))

# Output final parameter estimates and acceptance rates
list(param.est=apply(out[-(1:10000)],,2,mean),param.posts=out[-(1:10000)],,
ar=c(ar.b0,ar.b1,ar.b2,ar.gam)/(niter-10000))
}

#####

#### DATA ANALYSIS ####

#####

##### Obtain reasonable starting values

#psi=20
library(segmented)
out <- lm(y~x)
out.seg <- segmented(out,seg.Z=~x,psi=list(x=20))
out.seg

# 0.007589350 -0.001368922 0.023193395
# Estimated Break-Point(s) psi1.x : 17.42

```

```

# 0.03885 on 36 degrees of freedom
#> .03885*36/40 = 0.034965

p <- c(0.007, -0.001, 0.023, 17.400, 0.035)

##### Exploring the likelihood and gradient

# Partial likelihoods, holding other variables constant at starting values

par(mfrow=c(2,3))

# Hold constant over p, starting values

beta0try <- seq(-.1,.1,length=100)
fxtry <- sapply(beta0try,function(f) fx(x,y,p=c(f,p[2:5])))
plot(beta0try,fxtry,pch=20,xlab=expression(beta[0]),type='l')

beta1try <- seq(-.1,.1,length=100)
fxtry <- sapply(beta1try,function(f) fx(x,y,p=c(p[1],f,p[3:5])))
plot(beta1try,fxtry,pch=20,xlab=expression(beta[1]),type='l')

beta2try <- seq(-.2,.2,length=100)
fxtry <- sapply(beta2try,function(f) fx(x,y,p=c(p[1:2],f,p[4:5])))
plot(beta2try,fxtry,pch=20,xlab=expression(beta[2]),type='l')

gammatry <- seq(0,40,length=1000)
fxtry <- sapply(gammatry,function(f) fx(x,y,p=c(p[1:3],f,p[5])))
plot(gammatry,fxtry,pch=20,cex=.2,xlab=expression(gamma),type='l')

# Hone in in gradient But the gradient is discontinuous (along the gamma range)
gammatry <- seq(10,30,length=1000)
fxtry <- sapply(gammatry,function(f) G(x,y,p=c(p[1:3],f,p[5])))
plot(t(t(gammatry)),t(-fxtry)[,4],pch=20,cex=.4,xlab=expression(gamma),ylab="Marginal Gradient")

sigma2try <- seq(0,.01,length=100)
fxtry <- sapply(sigma2try,function(f) fx(x,y,p=c(p[1:4],f)))
plot(sigma2try,fxtry,pch=20,xlab=expression(sigma^2),type='l')

par(mfrow=c(1,1))

##### Fit each estimator

r1 <- MLE.I(p=c(0.007,-0.001,0.023,17.4,.035),x,y)
r2 <- MLE.II(p=c(0.007,-0.001,0.023,17.4,.035),x,y)
r3 <- MLETZ(p=c(0.007,-0.001,0.023,17.4,.035),x,y)
r4 <- Bayes(p=c(0.007,-0.001,0.023,17.4,.035),x,y,cand.var=c(0.0012,0.000002,.000012,6))

r1$param.est
r2$param.est
r3$param.est
r4$param.est
all <- rbind(r1$param.est,r2$param.est,r3$param.est,r4$param.est)

library(xtable)
xtable(all,digits=6)

par(mfrow=c(1,1))
pdf('CrackAppEsts.pdf',width=6,height=4.5)
plot(x,y,pch=20,main='Aging of Weapons',xlab='Age (masked)',ylab='Crack Length')
segadd(r1$param.est,col=2,minx=0,maxx=40)
segadd(r2$param.est,col=3,minx=0,maxx=40)

```

```

segadd(r3$param.est,col=4,minx=0,maxx=40)
segadd(r4$param.est,col=5,minx=0,maxx=40)
legend(1,.5,c('MLE-I','MLE-II','MLE-TZ','Bayes'),lwd=2,col=2:5)
dev.off()

##### Seeing what would happen with different starting values

MLE.I(p=c(0.007,-0.001,0.023,17.4,.035),x,y)

# DOES AN OKAY JOB OF CONVERGING WHEN
MLE.II.start <- matrix(NA,40,5)
for (i in 1:40){
MLE.II.start[i,] <- MLE.II(p=c(0.007,-0.001,0.023,i,.035),x,y)$param.est
}
plot(MLE.II.start[,5])

MLETZ(p=c(0.007,-0.001,0.023,17.4,.035),x,y)

##### Some additional goodies from the Bayesian posterior distributions

round(t(apply(r4$param.posts,2,function(f){quantile(f,c(.025,.975))})),digits=4)

par(mar=c(5, 4, 4, 2) + 0.1)
pdf('BayesAppPosteriors.pdf',width=6,height=8.3)
par(mfrow=c(3,2))
plot(density(r4$param.posts[,1]),main=expression(paste('Posterior for ',beta[0])),xlab="")
abline(v=r4$param.est[1],lwd=2,col=5)
plot(density(r4$param.posts[,2]),main=expression(paste('Posterior for ',beta[1])),xlab="")
abline(v=r4$param.est[2],lwd=2,col=5)
plot(density(r4$param.posts[,3]),main=expression(paste('Posterior for ',beta[2])),xlab="")
abline(v=r4$param.est[3],lwd=2,col=5)
plot(density(r4$param.posts[,4]),main=expression(paste('Posterior for ',gamma)),xlab="")
abline(v=r4$param.est[4],lwd=2,col=5)
plot(density(r4$param.posts[,5]),main=expression(paste('Posterior for ',sigma^2)),xlab="")
abline(v=r4$param.est[5],lwd=2,col=5)
par(mar=c(1,1,6,1))
plot(1,1,axes=F,type='n',xlab="",ylab="",ylim=c(-.2,2),xlim=c(.5,1.3), main="95% Credible Intervals")
text(1.2,1.9," ( 2.5% , 97.5% )",pos=2,cex=1.5)
text(1.2,1.5,expression(paste(beta[0]," (-0.0812 , 0.0818)")),pos=2,cex=1.5)
text(1.2,1.1,expression(paste(beta[1]," (-0.0090 , 0.0069)")),pos=2,cex=1.5)
text(1.2,.7,expression(paste(beta[2]," ( 0.0132 , 0.0322)")),pos=2,cex=1.5)
text(1.21,.3,expression(paste(gamma," (13.3976 , 23.7551)")),pos=2,cex=1.5)
text(1.2,-.1,expression(paste(sigma^2," ( 0.0043 , 0.0104)")),pos=2,cex=1.5)
par(mfrow=c(1,1))
dev.off()

# To get some confidence bands on the predicted values
fhat <- matrix(NA,100000,100)
x12 <- seq(0,40,length=100)
for(i in 1:100000){
temp.est <- as.numeric(r4$param.posts[i,])
x1 <- x12[(x12<=temp.est[4])]
x2 <- x12[(x12>temp.est[4])]
fhat[i,1:length(x1)] <- temp.est[1] + x1*temp.est[2]
fhat[i,(length(x1)+1):100] <- fhat1[i,length(x1)] + (x2 -temp.est[4])*(temp.est[2]+temp.est[3])
}

plot(x,y,pch=20,main='Aging of Weapons',xlab='Age (masked)',ylab='Crack Length')
# Plug in mean and 95% estimators to get lines
# Don't think this makes much sense, not sure what you would call it
segadd(r4$param.est,col=5,minx=0,maxx=40)
segadd(apply(r4$param.posts,2,function(f){quantile(f,.275)}),col=5,minx=0,maxx=40)

```

```
segadd(apply(r4$param.posts,2,function(f){quantile(f,.975)}),col=5,minx=0,maxx=40) #expand ,ylim=c(-.25,1)

# Posterior something - not predictive because doesn't generate normal data.
# Posterior prediction bands?
# (not sharp): Calculate lines then take mean along the line and 95% values of those lines
matplot(x12,t(apply(fhat1,2,function(f){quantile(f,c(.275,.975))})),type='l',col=5,lty=2,add=T)
lines(x12,apply(fhat1,2,mean),type='l',col=5,lty=2)
```