



Theses and Dissertations

---

2011-03-07

## Hierarchical Bayesian Methods for Evaluation of Traffic Project Efficacy

Andrew Nolan Olsen  
*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

Olsen, Andrew Nolan, "Hierarchical Bayesian Methods for Evaluation of Traffic Project Efficacy" (2011).  
*Theses and Dissertations*. 2922.  
<https://scholarsarchive.byu.edu/etd/2922>

This Selected Project is brought to you for free and open access by BYU ScholarsArchive. It has been accepted for inclusion in Theses and Dissertations by an authorized administrator of BYU ScholarsArchive. For more information, please contact [scholarsarchive@byu.edu](mailto:scholarsarchive@byu.edu), [ellen\\_amatangelo@byu.edu](mailto:ellen_amatangelo@byu.edu).

Hierarchical Bayesian Methods for Evaluation of  
Traffic Project Efficacy

Andrew N. Olsen

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  
Grant G. Schultz  
Candace Berrett

Department of Statistics  
Brigham Young University

April 2011

Copyright © 2011 Andrew N. Olsen

All Rights Reserved



## ABSTRACT

### Hierarchical Bayesian Methods for Evaluation of Traffic Project Efficacy

Andrew N. Olsen  
Department of Statistics, BYU  
Master of Science

A main objective of Departments of Transportation is to improve the safety of the roadways over which they have jurisdiction. Safety projects, such as cable barriers and raised medians, are utilized to reduce both crash frequency and crash severity. The efficacy of these projects must be evaluated in order to use resources in the best way possible. Five models are proposed for the evaluation of traffic projects: (1) a Bayesian Poisson regression model; (2) a hierarchical Poisson regression model building on model (1) by adding hyperpriors; (3) a similar model correcting for overdispersion; (4) a dynamic linear model; and (5) a traditional before-after study model. Evaluation of these models is discussed using various metrics including DIC.

Using the models selected for analysis, it was determined that cable barriers are quite effective at reducing severe crashes and cross-median crashes on Utah highways. Raised medians are also largely effective at reducing severe crashes. The results of before and after analyses are highly valuable to Departments of Transportation in identifying effective projects and in determining which roadway segments will benefit most from their implementation.

Keywords: hierarchical model, Poisson regression, dynamic linear model



## ACKNOWLEDGMENTS

First I must thank my advisor Dr. Reese for his invaluable mentoring throughout this project. Furthermore, I also wish to thank Dr. Schultz for the opportunity to be involved with this project and for his careful tutoring concerning the civil engineering aspects of this project. I also wish to acknowledge the support of the Utah Department of Transportation who provided motivation, funding, and data for this project. This project was also supported by an Office of Research and Creative Activities grant from the university. Finally, I wish to acknowledge my wife Andrea who has spent countless hours supporting me as I have worked on this project.



## CONTENTS

Contents . . . . .	vii
1 Introduction . . . . .	1
2 Literature Review . . . . .	3
2.1 Traditional Before-After Study Methods . . . . .	3
2.2 Bayesian Methods . . . . .	5
2.3 Hierarchical Poisson Regression Models . . . . .	8
2.4 MCMC Methods . . . . .	11
2.5 Model Comparison . . . . .	13
2.6 Goodness of Fit . . . . .	15
3 Methods . . . . .	19
3.1 Cable Barriers and Raised Medians . . . . .	19
3.2 Models for Analyzing Efficacy of Traffic Hazard Remediation . . . . .	21
Model 1: Standard Poisson Regression . . . . .	21
Model 2: Hyperpriors . . . . .	22
Model 3: Overdispersion . . . . .	23
Model 4: Dynamic Linear Model . . . . .	24
Model 5: Traditional Approach . . . . .	25
4 Results . . . . .	27
4.1 Prior Sensitivity Simulation Study . . . . .	27
4.2 Model Comparison . . . . .	29

4.3	Goodness of Fit . . . . .	31
4.4	Posterior Summarization . . . . .	33
4.5	Before-After Study Results . . . . .	36
4.6	Traditional Results . . . . .	43
5	Conclusion . . . . .	45
	Bibliography . . . . .	47
	Appendices . . . . .	51
	Appendix A: Hierarchical Model Results . . . . .	53
	A.1 Cable Barriers . . . . .	53
	A.2 Raised Medians . . . . .	64
	Appendix B: Traditional Method Results . . . . .	75
	B.1 Cable Barriers . . . . .	76
	B.2 Raised Medians . . . . .	78
	Appendix C: Complete Conditional Distributions . . . . .	81
	C.1 Model 1 . . . . .	81
	C.2 Model 2 . . . . .	81
	C.3 Model 3 . . . . .	82
	C.4 Model 4 . . . . .	82
	Appendix D: Code . . . . .	85
	D.1 Model 1 . . . . .	85
	D.2 Model 2 . . . . .	91
	D.3 Model 3 . . . . .	99
	D.4 Model 4 . . . . .	110

## CHAPTER 1

---

### INTRODUCTION

Departments of Transportation (DOTs) are consistently striving to improve the safety of the roadways over which they have jurisdiction. Safety projects, such as installing cable barriers and raised medians, are continually implemented in order to reduce crashes and fatalities on particularly dangerous roadways. In order to allocate resources effectively, these projects must be carefully evaluated in order to determine the extent to which crashes, severe injuries, or fatalities are reduced. Projects that are less effective may be either modified or abandoned, whereas projects that produce desired results may be further implemented and improved.

One traditional method of before-after analysis is the empirical Bayes (EB) method discussed in detail by Hauer (1997). This method estimates expected crashes for a roadway segment using a weighted average between the predicted number of crashes and the actual number of crashes that occurred on the segment. This method has become the industry standard for traffic safety analyses and is widely used.

Hierarchical Bayesian modeling expands the EB framework, resulting in several statistical advantages. Bayesian methods allow incorporation of previously known information into the model through prior distributions. This allows inclusion of results obtained from previous analyses in the model. The use of prior information also allows for better results when few data are available.

The purpose of this project is to develop the use of hierarchical Bayesian methods in traffic safety analyses. Chapter 2 reviews the literature relevant to this project including traditional before-after analysis methods, Bayesian methodology, hierarchical Poisson regression, Markov chain Monte Carlo (MCMC) methods, model comparison techniques, and

methods to measure goodness of fit. Chapter 3 outlines the methods used in this project including a discussion of cable barriers and raised medians and a description of each of the five models used to analyze the crash data. Chapter 4 presents the results of the analyses including the results of the simulation study, model comparison, goodness of fit, and a summarization of results from the models chosen for analysis. Chapter 5 summarizes the project and presents several conclusions from this research.

The methods discussed in this project are extremely effective in allowing DOTs to determine which safety projects are best for reducing fatalities and incapacitating injuries due to crashes on roadways worldwide. They also provide the foundation for the development of further methods that will similarly improve safety.

### LITERATURE REVIEW

To set a foundation for the methodologies used in this project, several key aspects of the literature are reviewed. First, traditional methods of analyzing before-after studies are discussed. Then, Bayesian methods are introduced and described. Further, hierarchical Poisson regression is outlined and discussed. MCMC methods, which are necessary for computation of posterior distributions, are then presented. Finally, a discussion about model comparison and goodness of fit is provided.

#### 2.1 TRADITIONAL BEFORE-AFTER STUDY METHODS

The mainstream method for evaluating traffic safety is the EB method. Hauer (1997) provides an extensive discussion about this method in *Observational Before-After Studies in Road Safety*. A tutorial to these methods by Hauer, Harwood, Council, and Griffith (2002) was published in the *Transportation Research Record: Journal of the Transportation Research Board*. In that paper, the authors made the following assertion:

“The time has come for the EB method to be the standard of professional practice; it should be used whenever the need to estimate road safety arises, whether in the search for sites with promise, the evaluation of the safety effects of interventions, or the assessment of potential safety savings due to site improvements” (Hauer et al. 2002, p. 126).

This quote illustrates the depth of the use of EB methods within the traffic safety analysis industry. The American Association of State Highway and Transportation Officials (AASHTO) (2010) recently released the *Highway Safety Manual* (HSM), which advocates

and further explains the use of EB methods. This can be considered the national standard for safety studies to be used by DOTs and other research entities.

The HSM provides a method for estimating expected crashes on a roadway which is based on EB methodology. The formula for calculating the expected number of crashes is

$$N_{\text{expected}} = w \times N_{\text{predicted}} + (1 - w) \times N_{\text{observed}}, \quad (2.1)$$

where

$$w = \frac{1}{1 + k \times \left( \sum_{\text{all study years}} N_{\text{predicted}} \right)}, \quad (2.2)$$

and

$N_{\text{expected}}$  = estimate of expected average crash frequency for the study period,

$N_{\text{predicted}}$  = predictive model estimate of average crash frequency for the study period under the given conditions,

$N_{\text{observed}}$  = observed crash frequency at the site over the study period,

$w$  = weighted adjustment to be placed on the predictive model estimate, and

$k$  = overdispersion parameter of the associated safety performance function (SPF) used to estimate  $N_{\text{predicted}}$  (AASHTO 2010).

Where an SPF is not readily available, negative binomial regression may be utilized to estimate the overdispersion parameter and to create a custom SPF for determining  $N_{\text{predicted}}$ . Thus, from Equation 2.1, EB methods estimate the expected number of crashes on a roadway as a combination of crash counts for that specific roadway and the predicted number of crash counts based on a model for similar roadways. The advantage of this method over utilizing crash counts alone includes accounting for regression to the mean bias. The expected number of crashes may then be compared with those of a reference population of similar characteristics where the crash prevention project was not implemented in order to determine whether or not the project was effective.

The Bayesian framework builds upon the advantages of the traditional EB method by allowing prior information to be incorporated into the model. Additional advantages to Bayesian methodology include obtaining a posterior distribution rather than simply a point estimate with appropriate standard error as is done in this implementation of EB methods. Furthermore, inferences from the posterior distribution within Bayesian methods are flexible and highly interpretable.

## 2.2 BAYESIAN METHODS

The process of inferring population parameters from sample data involves first formulating a postulated model from which the data were drawn. The postulated model for the  $i^{th}$  observation is a likelihood function  $f(y_i|\boldsymbol{\theta})$  where  $y_i$  is the data value and  $\boldsymbol{\theta}$  is the collection of parameters required to specify the likelihood fully. Frequentist methods of analysis affirm that model parameters  $\boldsymbol{\theta}$  are fixed, though unknown, constants. They estimate  $\boldsymbol{\theta}$  and typically provide an estimate of the error surrounding the estimate for  $\boldsymbol{\theta}$ .

The Bayesian philosophy about  $\boldsymbol{\theta}$  differs from the classical paradigm. Bayesian methods regard population model parameters  $\boldsymbol{\theta}$  as distributions rather than fixed constants. Each potential parameter value has a likelihood of occurring. Two views of  $\boldsymbol{\theta}$  are critical in Bayesian analysis: the view of  $\boldsymbol{\theta}$  before data collection, also known as the *a priori* view of  $\boldsymbol{\theta}$ , and the view of  $\boldsymbol{\theta}$  after incorporating information from the data, known as the *a posteriori* view of  $\boldsymbol{\theta}$ . The combination of the data and the prior distribution of  $\boldsymbol{\theta}$  to form the posterior distribution of  $\boldsymbol{\theta}$  is the central theory behind Bayesian methodology. Identification of an appropriate data model and prior distribution are crucial to Bayesian analysis.

Data are incorporated into Bayesian inference through a likelihood function. The likelihood dictates how likely the data are given specific parameter values. It is essential to choose a likelihood that corresponds well with the data in terms of support. For example, the normal distribution likelihood is a poor choice for crash counts because a normally distributed random variable can take values anywhere from  $-\infty$  to  $\infty$ , but crash counts

can only be integer values greater than or equal to zero. The Poisson likelihood is more appropriate because the support of the likelihood function matches the support of the data. Failing to select an appropriate likelihood can lead to poor results.

Bayesian methodology also preserves the likelihood principle, which states that inference from proportional likelihoods should lead to the same conclusions (Casella and Berger 2002). This principle is not always preserved in classical inference; experimental design can often change the conclusions made in classical analysis. Bayesian use of the data through the likelihood prevents this problem from occurring.

The information or intuition possessed about a situation independent of any data collection is summarized into prior distributions for the data likelihood parameters. These distributions assign a likelihood for every value the parameter can take based upon the analyst's belief prior to collection and analysis of the data. If the analyst is uncertain about his or her belief, the variance of the prior distribution may be increased comparatively, making the prior distribution more diffuse. Likewise, if the analyst is more confident about the prior information or is basing his or her belief on prior experimentation, a distribution with smaller variance may be selected. When few data are available, the prior will generally play a large role in the inference made. Thus, care should be taken to specify prior distributions in a thoughtful and meaningful way. If many data are available, the importance of the prior distribution is reduced as the prior holds less weight compared to the data.

The prior and the data likelihood are combined together to form the posterior distribution. As each person may uniquely define a prior distribution, so can each posterior distribution vary. Although this prevents researchers from arriving at one unique answer with the same data, if a prior is selected thoughtfully, the inference made from the posterior distribution will be quite useful.

The prior distribution, data likelihood, and posterior distribution are related using conditional probability rules found in the literature (Casella and Berger 2002). Consider

simple events  $A$  and  $B$ . The conditional probability of  $A$  given  $B$  is

$$P(A|B) = \frac{P(A \cap B)}{P(B)}, \quad (2.3)$$

where  $P(B) > 0$ . Using this fact and realizing similarly that  $P(A \cap B) = P(B|A)P(A)$ , the resulting formula is

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} = \frac{P(B|A)P(A)}{P(B|A)P(A) + P(B|A^c)P(A^c)}, \quad (2.4)$$

with the final equality holding by the law of total probability. This expression is known as Bayes' Theorem. Extending Bayes' Theorem beyond the simple events  $A$  and  $B$  in the previous statement, let  $A$  be the population model parameters  $\boldsymbol{\theta}$ , and  $B$  be the data  $\mathbf{y}$ . The resulting expression is given by

$$\pi(\boldsymbol{\theta}|\mathbf{y}) = \frac{f(\mathbf{y}|\boldsymbol{\theta})\pi(\boldsymbol{\theta})}{\int f(\mathbf{y}|\boldsymbol{\theta})\pi(\boldsymbol{\theta})d\boldsymbol{\theta}}, \quad (2.5)$$

where the following definitions hold:

$\mathbf{y}$  = the vector of data,

$\boldsymbol{\theta}$  = the vector of parameters required for the data likelihood,

$f(\mathbf{y}|\boldsymbol{\theta})$  = the likelihood of the data given the parameters  $\boldsymbol{\theta}$ ,

$\pi(\boldsymbol{\theta})$  = the prior distribution of  $\boldsymbol{\theta}$ , and

$\pi(\boldsymbol{\theta}|\mathbf{y})$  = the posterior distribution of  $\boldsymbol{\theta}$  given the data  $\mathbf{y}$ .

This expression of Bayes theorem provides the theoretical framework for Bayesian statistics (Carlin and Louis 2009) constituting the method for combining the prior and likelihood into the posterior distribution.

Note that the integral in the denominator of the posterior distribution in Equation 2.5 is equivalent to  $P(\mathbf{y})$  or the probability of the entire dataset integrated over the distribution of  $\boldsymbol{\theta}$ . This quantity is fixed and is referred to as the normalization constant because it allows the posterior distribution to become a proper probability density function, which

must, by definition, integrate or sum to one (Casella and Berger 2002). In many cases, the integral may be extremely difficult to evaluate. For this reason, Bayesian modeling did not become widespread until relatively recently when computing resources became sufficiently advanced to support methods for obtaining the posterior distribution without utilizing the normalization constant. This is accomplished through MCMC methods, which are discussed in greater detail in Section 2.4.

## 2.3 HIERARCHICAL POISSON REGRESSION MODELS

### *The Poisson Distribution*

Before outlining the method of Poisson regression, characteristics of the Poisson distribution should be considered. The Poisson distribution is commonly utilized to model count data (Bolstad 2010), and thus is a good candidate for modeling the number of crashes on a specific roadway segment over a specified period of time. The probability mass function of a Poisson-distributed random variable (Casella and Berger 2002) is

$$f(y|\theta) = \frac{e^{-\theta}\theta^y}{y!}, \quad (2.6)$$

where  $y$  is a realization of the random variable  $Y$ ,  $\theta$  is the mean of the distribution, and  $y \in \{0, 1, 2, 3, \dots\}$ . The support of this density allows for any number of crashes to occur on a particular segment.

Another characteristic of the Poisson distribution is that the mean of the distribution is equal to the variance of the distribution (Casella and Berger 2002). While this detail is useful in many situations, correction may be required for situations where this equal mean and variance assumption is violated. Miranda-Moreno, Fu, Saccomanno, and Labbe (2005) discuss approaches that account for this overdispersion, one of which is implemented in model 3 of this project.

## *Regression Models*

Regression is the process by which a random variable, such as the number of crashes on a roadway segment, is modeled by a set of covariates. Frequently, the covariates are related to the random variable linearly through the mean (Hocking 2003). This implies that the mean of the random variable ( $\theta$ ) can be represented as a linear combination of covariates ( $x_j$ )

$$\theta = \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_p x_p, \quad (2.7)$$

where  $p$  is the number of covariates included in the model.

Often, however, this relationship is constructed with a link function. A link function is frequently required because the linear combination of covariates can range from  $-\infty$  to  $\infty$ , but the mean of the random variable of interest may be limited to the interval  $[0, \infty)$  or require other constraints (Bolstad 2010). This is indeed the case for the Poisson distribution where  $\theta \geq 0$ . Bolstad (2010) describes that the most common link function for Poisson random variables is the log-link function, which creates the relationship

$$\log(\theta) = \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_p x_p. \quad (2.8)$$

This link plays a crucial role in both Bayesian and traditional Poisson regression analyses.

## *Bayesian Hierarchical Poisson Regression Models*

As the methods employed in this project are inherently Bayesian, both the likelihood and prior distributions for the Poisson regression model will be developed.

The joint likelihood of the data  $f(\mathbf{y}|\boldsymbol{\theta})$  is found by assuming that each of the observations is exchangeable and by multiplying the likelihoods together (Casella and Berger 2002) such that

$$\begin{aligned} f(\mathbf{y}|\boldsymbol{\theta}) &= \prod_{i=1}^n \frac{e^{\theta_i} \theta_i^{y_i}}{y_i!} \\ &= \frac{e^{\sum_{i=1}^n \theta_i} \prod_{i=1}^n \theta_i^{y_i}}{\prod_{i=1}^n y_i!}, \end{aligned} \quad (2.9)$$

where  $\theta_i$  is the mean of the Poisson distribution for the  $i^{th}$  observation  $y_i$ , and  $n$  is the total number of observations. Knowing that  $\log(\theta_i) = \beta_1 x_{i1} + \beta_2 x_{i2} + \dots + \beta_p x_{ip}$ , the joint likelihood can be expressed as

$$\begin{aligned} f(\mathbf{y}|\boldsymbol{\beta}) &= \exp \left[ \sum_{i=1}^n \exp(\eta_i) \right] \prod_{i=1}^n \frac{\exp(\eta_i)^{y_i}}{y_i!} \\ &\propto \exp \left[ \sum_{i=1}^n \exp(\eta_i) \right] \prod_{i=1}^n \exp(\eta_i)^{y_i}, \end{aligned} \quad (2.10)$$

where  $\eta_i = \beta_1 x_{i1} + \beta_2 x_{i2} + \dots + \beta_p x_{ip}$ .

As was mentioned in Section 2.2, the normalization constant is not required for MCMC methods of obtaining the posterior distribution; thus, the joint likelihood up to a constant is sufficient to obtain the posterior distribution. The proportional likelihood expression in Equation 2.10 will therefore form the data portion of the Bayesian analysis.

A common choice of prior for  $\boldsymbol{\beta}$  is multivariate normal with zero cross-covariances (Bolstad 2010), which is equivalent to a series of independent normal priors for each  $\beta_j$ . The multivariate normal density function (Hocking 2003) for a random variable vector  $\mathbf{X}$  in  $k$  dimensions is

$$\pi(\boldsymbol{\beta}) = \frac{1}{(2\pi)^{k/2} |\boldsymbol{\Sigma}|^{1/2}} \exp \left[ -\frac{1}{2} (\mathbf{x} - \boldsymbol{\mu})' \boldsymbol{\Sigma}^{-1} (\mathbf{x} - \boldsymbol{\mu}) \right], \quad (2.11)$$

where

$k$  = the dimension of  $\boldsymbol{\beta}$  (generally the number of covariates in the model),

$\mathbf{x}$  = the vector of  $k$  realizations of the multivariate normal random variable,

$\boldsymbol{\mu}$  = the mean vector of the multivariate normal random variable,

$\boldsymbol{\Sigma}$  = the variance-covariance matrix of the multivariate normal random variable, and

$|\boldsymbol{\Sigma}|$  = the determinant of the variance-covariance matrix.

When the covariances are equal to zero, the prior simplifies into the product of  $k$  normal distributions as follows:

$$\pi(\boldsymbol{\beta}) = \frac{1}{(2\pi)^{k/2} \prod_{i=1}^k \sigma_i} \exp \left[ -\frac{1}{2} \sum_{i=1}^k \frac{(\beta_i - \mu_i)^2}{\sigma_i^2} \right], \quad (2.12)$$

where the realization, mean, and variance for the  $i^{th}$  dimension are denoted by  $\beta_i$ ,  $\mu_i$ , and  $\sigma_i^2$  respectively.

With the likelihood derivations in Equation 2.10 and the prior derivations in Equation 2.11, the posterior distribution, up to a constant, is given by

$$\begin{aligned}\pi(\boldsymbol{\beta}|\mathbf{y}) &\propto f(\mathbf{y}|\boldsymbol{\beta})\pi(\boldsymbol{\beta}) \\ &= \exp \left[ \sum_{i=1}^n \exp(\eta_i) \right] \left[ \prod_{i=1}^n \exp(\eta_i)^{y_i} \right] \frac{1}{(2\pi)^{p/2} \prod_{i=1}^p \sigma_i} \exp \left[ -\frac{1}{2} \sum_{i=1}^p \frac{(x_i - \mu_i)^2}{\sigma_i^2} \right].\end{aligned}\tag{2.13}$$

As the marginal posterior densities are not available in closed form, MCMC methods are utilized to obtain marginal posterior distributions for each of the  $\beta$  parameters from the joint posterior distribution expressed in Equation 2.13.

## 2.4 MCMC METHODS

Though MCMC methods are increasingly used with improved technology, they have foundations of more than half a century ago. Metropolis, Rosenbluth, Rosenbluth, Teller, and Teller (1953) proposed using Monte Carlo methods for multiple complex integrals. This involved integrating over a random sampling of points rather than a specific array of points. Hastings (1970) extended the use of Monte Carlo methods using Markov chains. He thoroughly discussed the theory and implications behind this method. Geman and Geman (1984) extended these principles by introducing the Gibbs sampler, which allows for marginal distributions to be calculated using conditional distributions. Gelfand and Smith (1990) further discuss Gibbs sampling and provide several examples that utilize this and other MCMC methods. MCMC methods are currently utilized in many fields and are the key to Bayesian analyses with difficult posterior forms.

### *Gibbs Sampling*

Gibbs sampling is widely used and is referenced in several sources of literature (Gelman, Carlin, Stern, and Rubin 2004; Hamada, Wilson, Reese, and Martz 2008; Carlin and Louis 2009; Bolstad 2010). The Gibbs sampler algorithm allows for posterior draws to be obtained from the complete conditional distributions  $[\theta_i]$  of the parameters in  $\boldsymbol{\theta}$ . The algorithm is as follows:

- (1) Start with  $\boldsymbol{\theta}^{(0)} = (\theta_1^{(0)}, \theta_2^{(0)}, \dots, \theta_r^{(0)})$ , where  $r$  is the number of parameters in the model. Note that the algorithm is quite robust to starting values.
- (2) Set  $i = 1$ .
- (3) Generate an observation  $\theta_1^{(i)}$  from  $\pi(\theta_1 | \theta_2^{(i-1)}, \dots, \theta_r^{(i-1)}, \mathbf{y}) = [\theta_1]$ .
- (4) Generate an observation  $\theta_2^{(i)}$  from  $[\theta_2]$ .
- (5) Continue sequential generation until  $\theta_r^{(i)}$  from  $[\theta_r]$  is obtained.
- (6) Repeat steps (3) through (5)  $T + B$  times, where  $B$  is the number of burn-in iterations, and  $T$  is the desired number of draws from the posterior distribution.

The resulting draws are considered random realizations from the joint posterior distribution of  $\boldsymbol{\theta} | \mathbf{y}$ .

### *Metropolis Hastings Algorithm*

The random walk Metropolis Hastings algorithm is used in conjunction with Gibbs sampling when complete conditional distributions are available only up to a constant. The algorithm also allows for drawing from the full posterior distribution  $\pi(\boldsymbol{\theta} | \mathbf{y})$  utilizing the posterior distribution up to a constant  $g(\boldsymbol{\theta}) = f(\mathbf{y} | \boldsymbol{\theta})\pi(\boldsymbol{\theta})$ , if desired. Some of the draws are rejected, allowing the sample to converge to the posterior distribution. A candidate density based

on the previous value,  $q(\boldsymbol{\theta}^*|\boldsymbol{\theta}^{(t-1)})$  is used for obtaining draws. The Metropolis Hastings algorithm is outlined as follows, after an initial  $\boldsymbol{\theta}^{(0)}$  is obtained (Carlin and Louis 2009):

For  $t = 1, \dots, T$ , where  $T$  is the number of desired draws, repeat:

1. Draw a candidate  $\boldsymbol{\theta}^*$  from  $q(\cdot|\boldsymbol{\theta}^{(t-1)})$ .

2. Calculate

$$r = \min \left( \frac{g(\boldsymbol{\theta}^*)q(\boldsymbol{\theta}^*|\boldsymbol{\theta}^{(t-1)})}{g(\boldsymbol{\theta}^{(t-1)})q(\boldsymbol{\theta}^{(t-1)}|\boldsymbol{\theta}^*)} \right). \quad (2.14)$$

3. Accept  $\boldsymbol{\theta}^*$  as  $\boldsymbol{\theta}^{(t)}$  with probability of  $r$ .

4. Repeat until convergence.

With the conditions that are typically satisfied for a majority of statistical models,  $\boldsymbol{\theta}^{(t)}$  converges in distribution to a draw from the posterior  $\pi(\boldsymbol{\theta}|\mathbf{y})$  (Carlin and Louis 2009).

## 2.5 MODEL COMPARISON

Model selection in statistical analysis is critical. In the Bayesian paradigm, three methods for evaluating models include Bayes' Factors, Bayesian Information Criterion (BIC), and Deviance Information Criterion (DIC). Hamada et al. (2008) give meaningful tutorials to these three model selection criteria, which are summarized below.

### *Bayes' Factors*

Bayesian methodology evaluates models based on the odds that a model is true. Consider the case where there are two probability models  $M_1$  and  $M_2$  that are considered as approximating the process that produced the observed data set. Let the likelihood functions and prior distributions of these probability models be  $f(\mathbf{y}|\boldsymbol{\theta}_1, M_1)p(\boldsymbol{\theta}_1|M_1)$  and  $f(\mathbf{y}|\boldsymbol{\theta}_2, M_2)p(\boldsymbol{\theta}_2|M_2)$  respectively. Let  $P(M_1)$  be the prior probability for model  $M_1$  and  $P(M_2) = 1 - P(M_1)$  be the prior probability of the second probability model. The posterior odds that model  $M_1$  is

true are then

$$\begin{aligned}\frac{P[M_1|\mathbf{y}]}{P[M_2|\mathbf{y}]} &= \frac{P(\mathbf{y}|M_1)P(M_1)}{P(\mathbf{y}|M_2)P(M_2)} \\ &= \frac{P(M_1)}{P(M_2)} \times \frac{m_1(\mathbf{y}|M_1)}{m_2(\mathbf{y}|M_2)},\end{aligned}\tag{2.15}$$

where  $m_i(\mathbf{Y}|M_i)$  is the marginal distribution of  $\mathbf{y}$  given the  $M_i$  sampling model. Note then that Equation 2.15 states that the posterior odds is the product of the prior odds multiplied by the Bayes' factor, where the Bayes' factor is defined as the ratio of the marginal densities of the data under two potential sampling models (Hamada et al. 2008).

If the posterior odds, as shown in Equation 2.15, are greater than one, the sampling model  $M_1$  is more likely. If the posterior odds are less than one, the sampling model  $M_2$  is more probable. The posterior odds reduce to the Bayes factor if the prior probabilities for each model are 0.5. Thus, the marginal distributions of the data under sampling model  $i$ ,  $m_i(\mathbf{y}|M_i)$   $i = 1, 2$ , play an important role in determining *a posteriori* which model is more likely through Bayes' factors.

As Bayes' factors are difficult to compute, the Laplace approximation is used, which is discussed by DiCiccio, Kass, Raftery, and Wasserman (1997).

## BIC

Hamada et al. (2008) define BIC as

$$BIC = -2\log[f(\mathbf{y}|\hat{\boldsymbol{\theta}})] + \log(n)k,\tag{2.16}$$

where  $\mathbf{y}$  are the data,  $\hat{\boldsymbol{\theta}}$  is an estimate of  $\boldsymbol{\theta}$  from the posterior distribution (such as the posterior mean, median, or mode),  $n$  is the number of observations, and  $k$  is the number of parameters in  $\boldsymbol{\theta}$ . A lower BIC indicates a better model; fit is penalized by model complexity through  $\log(n)k$ . Comparing the BIC of different models is a useful way of determining which models are better.

## *DIC*

DIC is especially helpful in evaluating the superiority of one model to another in a hierarchical setting. For this reason, when the other two criterion do not agree, final decisions in this project are made using DIC. The advantage lies in the fact that DIC does not require the analyst to specify the number of parameters in the model as required for BIC. An effective number of parameters is calculated. Hamada et al. (2008) define model deviance as

$$D(\boldsymbol{\theta}) = -2\log[f(\mathbf{y}|\boldsymbol{\theta})]. \quad (2.17)$$

The expected deviance is then defined as

$$\bar{D} = E_{\boldsymbol{\theta}}[D]. \quad (2.18)$$

This can be found using a Monte Carlo sample from the posterior distribution as the mean of  $-2$  times the log likelihood of  $\mathbf{y}$  given each individual draw from the joint posterior distribution. The number of effective parameters  $p_D$  is then calculated as

$$p_D = \bar{D} - D[\bar{\boldsymbol{\theta}}], \quad (2.19)$$

where  $\bar{\boldsymbol{\theta}}$  is the vector of marginal posterior means or medians from the Monte Carlo sample of the joint posterior distribution. DIC is then defined by Hamada et al. (2008) as

$$DIC = \bar{D} + p_D. \quad (2.20)$$

As with BIC, models with smaller values of DIC relatively are preferable to alternate models.

## 2.6 GOODNESS OF FIT

Two measures of goodness of fit are utilized in this project. The first is the Bayesian  $\chi^2$  goodness-of-fit test. The second utilizes predictive distributions and quantiles to assess fit.

### *Bayesian $\chi^2$ Goodness-of-Fit*

This test is rooted in the Pearson's  $\chi^2$  goodness of fit test. The interval from 0 to 1 is broken into  $K$  evenly-spaced intervals. Hamada et al. (2008) suggest using  $K \approx n^{0.4}$ . For each draw of the posterior distribution, compute the cumulative distribution function for each observation  $y_i$ . For all  $i$ , these numbers fall between 0 and 1. Let  $m_k$ ,  $k = 1, \dots, K$  be the total number of cumulative distribution function values among all  $n$  observations that fall in the  $k^{th}$  interval between 0 and 1. The test statistic for this posterior draw is then computed as

$$R^B(\tilde{\theta}) = \sum_{k=1}^K \frac{(m_k - n/K)^2}{n/K}, \quad (2.21)$$

which is distributed  $\chi^2_{K-1}$  under the null hypothesis that the model fits well. The  $p$ -value for each of the draws from the posterior distribution may then be calculated. The proportion of  $p$ -values below 0.05 ought to be close to 0.05 under the null hypothesis. A model with a higher proportion of  $p$ -values below 0.05 fits more poorly.

This method will be principally used to determine whether or not models fit appropriately as it is a numeric value that can easily be compared to other models.

### *Predictive Distributions and Quantiles*

Gelman et al. (2004) propose comparing the posterior predictive distribution for each observation with the corresponding data value as a method of evaluating goodness of fit. The quantile of the datapoint from the posterior predictive distribution may be obtained. The distribution of quantiles of all the datapoints ought to be uniformly distributed according to the probability integral transformation theorem which states that a continuous cumulative distribution function is uniformly distributed from zero to one (Casella and Berger 2002).

Figure 2.1 shows a posterior predictive distribution and the corresponding data point. Figure 2.2 compares the distributions of differences of two different models; the better-fitting model produces the distribution on the left, which follows a uniform distribution relatively

well; the somewhat poor-fitting model produces the distribution on the right, which is not uniformly distributed.

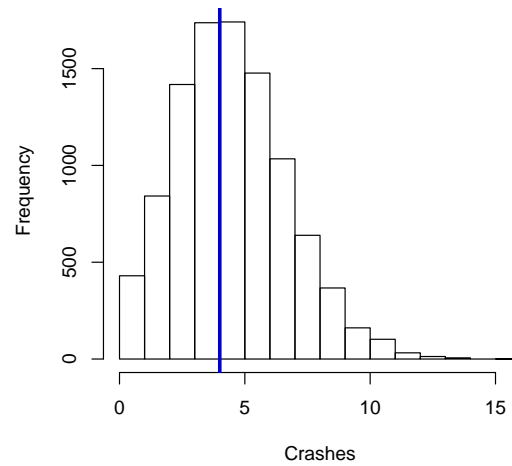


Figure 2.1: Comparison of data value with a posterior predictive distribution.

With these foundations in Bayesian methodology and crash analysis, the methods and models used in evaluating before and after traffic studies may be developed and discussed.

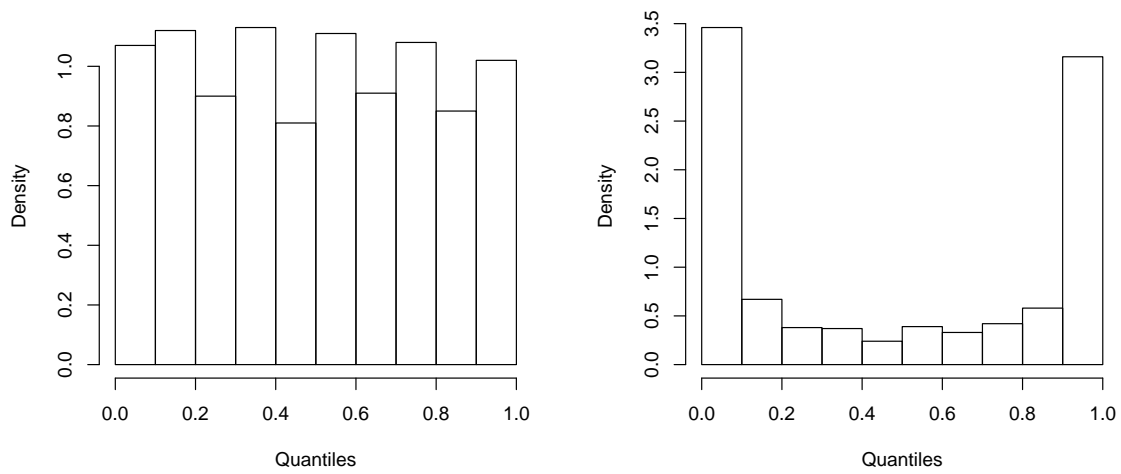


Figure 2.2: Comparison of distributions of quantiles for two models. The model whose distribution is on the left fits well. The model whose distribution is on the right fits more poorly.

## CHAPTER 3

---

### METHODS

This chapter first outlines the two types of traffic projects (cable barriers and raised medians) that will be considered in the analysis of this project. The five models that will be utilized to fit the data will then be introduced and briefly discussed.

#### 3.1 CABLE BARRIERS AND RAISED MEDIANS

##### *Cable Barriers*

Cable barriers are constructed of wire rope and small posts and are frequently used in medians of highway systems. They are relatively inexpensive to install and are quite effective at reducing cross-median crashes, which are generally quite severe as both directions of traffic are usually flowing at high speeds (AASHTO 2006).

The before-after crash analysis of cable barriers will focus on three different types of crashes: cross-median, severe, and overall. Cross-median crashes ought to be less prevalent as their reduction is the underlying goal of cable barriers. Severe crashes are also considered because it is expected that reduction of cross-median crashes should also reduce severe crashes. Severe crashes are those that resulted in either a fatality or incapacitating injury. Overall crashes are also of interest to determine whether cable barriers may have influenced overall crash frequency at each of the sites of interest.

In Utah, cable barriers have been installed primarily on the freeway system. Seven sites were considered for this study from either I-15 or I-215. A summary of the sites is provided in Table 3.1 (Schultz, Thurgood, Olsen, and Reese 2010). The site index will be used to identify sites throughout the presentation of the results.

Data from all sites were obtained between 2002 and 2008. One data point summarizing the number of crashes occurring on the roadway segment is available for each year. The installation year is indicated in Table 3.1. Data from the installation years are modeled separately from the before and after periods in the analysis as factors involved with the installation may have temporarily effected crash rates.

Table 3.1: Summary of Cable Barrier Sites

Site	Route	Location	Begin Mile	End Mile	Install Year
1	I-15	Provo S-curves to University Parkway	267	269	2004
2	I-15	Between Pintura and Kolob Canyons	36	38	2005
3	I-15	Spanish Fork to SR 75	257	261	2005
4	I-15	South Layton to Syracuse	330	334	2006
5	I-15	600 N to 2300 N Salt Lake	309	312	2007
6	I-215	3100 S to 3800 S Salt Lake	1.5	2.5	2007
7	I-215	2100 S to 4500 S Salt Lake	17	19.5	2007

### *Raised Medians*

A raised median is a barrier, generally constructed of concrete or landscaping, that is installed in the median of the roadway. Raised medians are often useful in reducing crashes by limiting the number of conflict points, such as the number of places where vehicles may turn left. They are also thought to be useful in reducing head-on collisions where a vehicle may cross into oncoming traffic (Transportation Research Board (TRB) 2003).

Reduction in the number of severe crashes is perhaps the most important indicator as to whether or not raised medians are effective at a particular site. An additional consideration is the reduction of cross-median crashes. Finally, the total number of crashes are considered to provide information as to whether the installation of raised medians influences the number of overall crashes.

The sites for raised medians are summarized in Table 3.2 (Schultz et al. 2010). These sites are generally located where many cars pass each day for business and other access. Note that the milepoints shown are the most current values and that milepoints varied

slightly for earlier years based on former milepoint systems. Any differences were reconciled in the analysis according to information provided by the Utah Department of Transportation (UDOT). Data are available from 1998 to 2008 for raised medians.

Table 3.2: Summary of Raised Median Sites

Site	Route	Location	Begin Mile	End Mile	Install Year
1	265	University Parkway	1.20	1.96	2002
2	74	Alpine Highway	2.40	4.29	2002
3	186	400/500 South	5.48	7.53	2001
4	71	12300 South	4.55	5.45	2004
5	34	St. George Blvd.	0.00	1.74	2006
6	36	SR 36	59.29	60.82	2005

Each of the three types of crashes for both cable barriers and raised medians will be fit with each model in this project.

### 3.2 MODELS FOR ANALYZING EFFICACY OF TRAFFIC HAZARD REMEDIATION

Five models are proposed for analyzing efficacy of crash-reduction projects. The first three of these models each are founded upon the principles of hierarchical Poisson regression. For these models, the basic modeling strategy is to create unique intercepts for before, after, and installation time periods. If enough data are available and such results are desired, unique parameters corresponding to different time periods for other covariates, such as annual average daily traffic (*AADT*), may also be used. Utilizing *AADT* is important in the analysis process because traffic flow can change between before and after periods. The fourth model fits crashes over time and comparisons may be made between years before, during, and after the implementation of the remediation project. The final model utilizes the methods discussed in the HSM (AASHTO 2010) and represents the classical approach utilized by the civil engineering community.

### *Model 1: Standard Poisson Regression*

The first model uses standard Bayesian Poisson regression and is written

$$\begin{aligned} y_i &\sim \text{Poisson}(\theta_i), \\ \log(\theta_i) &= \mathbf{X}_i\boldsymbol{\beta}, \end{aligned} \tag{3.1}$$

with prior distribution for  $\boldsymbol{\beta}$

$$\boldsymbol{\beta} \sim \text{MVN}(\mathbf{m}, \mathbf{S}),$$

where

$y_i$  = the observed number of crashes per mile for observation  $i$ ,

$\theta_i$  = the mean number of crashes per mile for observation  $i$ ,

$\mathbf{X}_i$  = the  $i^{\text{th}}$  row of the covariate matrix  $\mathbf{X}$ ,

$\boldsymbol{\beta}$  = the vector of parameters corresponding with  $\mathbf{X}$ ,

$\mathbf{m}$  = a vector of constants, and

$\mathbf{S}$  = a constant covariance matrix.

The matrix  $\mathbf{X}$  consists of all covariates desired in the analysis. Most of the analyses will include an intercept for the before period, an intercept for the after period, an intercept for the installation period, and a coefficient for *AADT* values for each observation. When more than one site is being considered, as in this analysis, these columns of the  $\mathbf{X}$  matrix are provided for each individual site.

### *Model 2: Hyperpriors*

The second model expands the first model by adding hyperpriors to the prior distribution for  $\boldsymbol{\beta}$ . This allows for the parameters from each site to come from an overall population distribution of parameters borrowing strength across sites. Rather than designate a multivariate

normal prior for  $\boldsymbol{\beta}$  with fixed mean vector  $\mathbf{m}$  and fixed covariance matrix  $\mathbf{S}$ , designate a multivariate normal prior for  $\boldsymbol{\beta}$  with mean vector  $\boldsymbol{\mu}_\beta$  and covariance matrix  $\boldsymbol{\Sigma}_\beta$  where the mean vector  $\boldsymbol{\mu}_\beta$  has a multivariate normal prior with mean vector  $\mathbf{m}$  and covariance matrix  $\mathbf{S}$ , and the diagonal elements of the covariance matrix  $\boldsymbol{\Sigma}_\beta$  are distributed inverse gamma with shape  $a$  and scale  $b$ . For this analysis, the cross-covariances of  $\boldsymbol{\Sigma}_\beta$  are chosen to be zero and thus do not require hyperpriors. The model then takes the form

$$\begin{aligned} y_i &\sim \text{Poisson}(\theta_i), \\ \log(\theta_i) &= \mathbf{X}_i \boldsymbol{\beta}, \end{aligned} \tag{3.2}$$

with prior distributions

$$\begin{aligned} \boldsymbol{\beta} &\sim \text{MVN}(\boldsymbol{\mu}_\beta, \boldsymbol{\Sigma}_\beta), \\ \boldsymbol{\mu}_\beta &\sim \text{MVN}(\mathbf{m}, \mathbf{S}), \text{ and} \\ \boldsymbol{\Sigma}_{\beta,ii} &\sim \text{IG}(a_i, b_i) \quad i = 1, \dots, p, \end{aligned}$$

where

$$\begin{aligned} p &= \text{the number of columns in } \mathbf{X}, \text{ and} \\ a_i, b_i &= \text{scalar constants greater than 0.} \end{aligned}$$

Hyperparameters  $\mathbf{m}$ ,  $\mathbf{S}$ ,  $a_i$ , and  $b_i$  may then be selected in order to represent the *a priori* knowledge about the model. The sites pool together in forming the distributions of  $\boldsymbol{\mu}_\beta$  and  $\boldsymbol{\Sigma}_\beta$ .

### *Model 3: Overdispersion*

It is often the case that for crash count data the equal mean and variance assumption of the Poisson likelihood is violated. One method proposed by Miranda-Moreno et al. (2005) compensates for this fact by letting the mean of the  $i^{th}$  Poisson distribution include not only  $\theta$ , but a multiplicative factor that has an expected value of 1, but higher or smaller variance.

This can be achieved by adding a term to the log link that has expected value of zero. The third model is based on this principle.

The model for this approach is written

$$\begin{aligned} y_i &\sim \text{Poisson}(\theta_i), \\ \log(\theta_i) &= \mathbf{X}_i\boldsymbol{\beta} + \eta, \end{aligned} \tag{3.3}$$

with

$$\begin{aligned} \boldsymbol{\beta} &\sim \text{MVN}(\boldsymbol{\mu}_\beta, \boldsymbol{\Sigma}_\beta), \\ \boldsymbol{\mu}_\beta &\sim \text{MVN}(\mathbf{m}, \mathbf{S}), \\ \boldsymbol{\Sigma}_{\beta,ii} &\sim \text{IG}(a_i, b_i) \quad i = 1, \dots, p, \\ \eta &\sim N(0, \tau_\eta), \text{ and} \\ \tau_\eta &\sim \text{IG}(a_\eta, b_\eta), \end{aligned}$$

where all symbols represent those defined in the hyperprior model. If the data are truly overdispersed, this model will likely perform better than the first two.

#### *Model 4: Dynamic Linear Model*

West and Harrison (1989) describe dynamic linear models as a flexible tool used in Bayesian forecasting. Leininger, Reese, Burraston, and Rutkowski (2010) utilize a dynamic linear model to develop criminal career trajectory latent classes. Their model can be extended to before-after studies by modeling the crash levels over time and paying particular attention to when the accident remediation was installed. This comprises the fourth model. With respect to notation, let:

- $i$  index the individual roadway segment;  $i = 1, 2, \dots, N$
- $j$  index the year of interest with  $j = 1$  corresponding to the first year of data;  $j = 1, \dots, J$

- $k$  index the sites of interest;  $k = 1, \dots, K$
- $y_{ij}$  be the number of crashes per mile for the  $i^{th}$  observation during the  $j^{th}$  time period
- $\lambda_{jk}$  be the expected mean and variance of the number of crashes per mile (after zero-inflation adjustment as shown in the likelihood) of roadway segments from site  $k$  during the  $j^{th}$  time period
- $C_i$  be an indicator for which site the  $i^{th}$  roadway segment corresponds to
- $\pi_k$  be the probability that a segment will be in a safe state of no crashes for site  $k$

The likelihood is

$$\prod_{i=1}^N \prod_{j=1}^J \left( \sum_{k=1}^K I(C_i = k) \left( \pi_k I(y_{ij} = 0) + (1 - \pi_k) \frac{e^{-\lambda_{jk}} \lambda_{jk}^{y_{ij}}}{y_{ij}!} \right) \right). \quad (3.4)$$

The priors for this analysis are

$$\theta_{1k} \sim Normal(m_1, \tau_k^2), \text{ for } k = 1, 2, \dots, K \text{ where } \log(\lambda_{1k}) = \theta_{1k},$$

$$\theta_{jk} \sim Normal(\theta_{j-1,k}, \tau_k^2), \text{ where } \log(\lambda_{jk}) = \theta_{jk}, \text{ for } j = 2, 3, \dots, J \text{ and } k = 1, 2, \dots, K,$$

$$\pi_k \sim Beta(\alpha, \beta) \text{ for } k = 1, 2, \dots, K,$$

$$\alpha \sim Gamma(a_\alpha, b_\alpha),$$

$$\beta \sim Exponential(c), \text{ and}$$

$$\tau_k^2 \sim InverseGamma(a_\tau, b_\tau) \text{ for } k = 1, 2, \dots, K,$$

where  $m_1$ ,  $a_\alpha$ ,  $b_\alpha$ ,  $c$ ,  $a_\tau$ , and  $b_\tau$  are constants. This model works as a flexible and smooth Bayesian regression method.

The complete conditional distributions required for each of these models are contained in Appendix C. The code utilized to fit each of the models is presented in Appendix D.

#### *Model 5: Traditional Approach*

This model was outlined in Section 2.1 as part of the literature review. Reference roadway segments were obtained by taking half the segment length before and half the segment length

after the segment of interest. Estimates were obtained for the expected number of crashes for all of the project sites and reference sites. Confidence intervals were then constructed for these estimates. The segments for which the project confidence interval and the reference confidence interval intersect were considered to not have a reduction. Where the project confidence interval exceeded the reference confidence interval entirely, it was determined that a reduction occurred.

With the five models outlined, analyses may be performed and results obtained for determining whether or not cable barriers and raised medians are effective at reducing crash severity and frequency on Utah roadways.

---

RESULTS

This chapter presents the results of the analyses of this project. First, the simulation study concerning prior sensitivity is discussed. Then model selection results are presented and followed by goodness of fit results. Methods of summarizing posterior distributions for before and after studies are then addressed. Results are then presented and discussed from both the hierarchical and traditional analyses approaches.

#### 4.1 PRIOR SENSITIVITY SIMULATION STUDY

##### *Simulation Study*

An analysis of posterior sensitivity to changes in the prior distributions was performed. As the amount of data for some roadway segments, particularly for after time periods, is often quite small, the prior can indeed be very important. The model was fit with different prior values in order to determine the extent to which the results change based on the prior distribution.

Model 2 was used for this simulation. For all simulations,  $\mathbf{m} = \mathbf{0}$ . The varying parameter values are shown in Table 4.1. These simulations were conducted with varying sample sizes: 4, 20, and 100. Evaluations of priors were performed with respect to whether posterior credible intervals capture the actual parameter values.

Table 4.1: Proposed parameter values for simulation study.

Parameter	Hyperparameter Values
$\mathbf{S}$	Diagonal matrix with elements 1, 10, and 100
$\Sigma_{ii}$	$a_i = 2.01$ and $b_i = 0.01, 0.10, \text{ and } 0.5$

In order to increase the efficiency of the simulations, a fractional factorial design was used as discussed by Lawson (2010). This allows the nine main effects to be estimated without needing to run all 27 parameter combinations (three parameters, including the varying sample sizes, each with three levels). A list of the runs of the simulation as produced using the `%mktruns()` and the `%mktex()` macros in SAS is shown in Table 4.2. This set of simulations produces an orthogonal design as desired. Each of the nine simulations was performed 1,000 times on randomly-generated datasets.

Table 4.2: List of Simulations Performed.

Run	$\mathbf{S}_{ii}$	$b_i$	$n$
1	1	0.50	20
2	10	0.10	20
3	10	0.01	100
4	10	0.50	4
5	100	0.10	4
6	100	0.50	100
7	100	0.01	20
8	1	0.10	100
9	1	0.01	4

The estimated coverage of the  $\beta$  parameters by 95% highest posterior density (HPD) credible intervals for each of the varying hyperparameters and sample sizes is shown in Table 4.3. As shown, all of the coverage probabilities are above 0.95. Ideally, the probabilities would be equal to 0.95. Higher values may indicate that the variances of the posterior distributions are wider than they need to be for 95% coverage. Additionally, the bias inherent with the specification of a prior distribution may lead to this result.

Table 4.3: Coverage Probabilities

$\mathbf{S}_{ii}$	Value	1	10	100
	Coverage	0.9617	0.9638	0.9620
$b_i$	Value	0.01	0.10	0.50
	Coverage	0.9533	0.9646	0.9696
$n$	Value	4	20	100
	Coverage	0.9805	0.9558	0.9512

With respect to  $\mathbf{S}_{ii}$ , the coverage is closest to 0.95 when  $\mathbf{S}_{ii} = 1$ . These three values, however, are particularly close. Concerning  $b_i$ , the coverage is closest to 0.95 with  $b_i = 0.01$ . These values, however, are also quite close. Larger sample sizes resulted in coverage probabilities closer to 0.95. Having established that the coverage does not change dramatically with respect to the prior specification, in the final analyses, the analyst may select his or her prior as deemed appropriate and reasonable.

### *Prior specification*

Based on the simulation study and the desire to obtain narrower credible intervals, it was determined to use the following prior values: for model 1, let  $\mathbf{m} = \mathbf{0}$  and let  $\mathbf{S}$  be a diagonal matrix with 1 on the diagonal; for model 2, let  $\mathbf{m}$  and  $\mathbf{S}$  be defined as in model 1; also, let  $a_i$  be 3.0 and  $b_i$  be 0.5; for model 3, let  $\mathbf{m}$ ,  $\mathbf{S}$ ,  $a_i$  and  $b_i$  be defined as in model 2, and let  $a_\eta = 5$  and  $b_\eta = 0.7$ ; finally, for model 4, let  $m_1$  be equal to the number of crashes occurring in the year previous to the first year utilized in the analysis, let  $a_\alpha = 1$ ,  $b_\alpha = 1$  and  $c = 1$ ; also, let  $a_\tau = 5$  and  $b_\tau = 0.2$ . All of these priors are fairly non-informative as this is the first such study conducted by UDOT.

## 4.2 MODEL COMPARISON

Each of the four models was fit using Gibbs sampling with Metropolis-Hastings steps where required. For each model, 500,000 draws from the posterior distribution were obtained. The Bayes' factor, BIC, and DIC for each of the models are presented in Tables 4.4 and 4.5 for cable barriers and raised medians respectively. The DIC is ultimately the criterion that was chosen to determine which model to use. This is due to the fact that DIC appropriately accounts for the hierarchy of a Bayesian hierarchical model. Model 1 performed best for severe and cross median crashes on cable barriers as well as for severe crashes on raised medians. Model 3 performed best for overall crashes on cable barriers and for cross median

crashes on raised medians. Model 4 performed best on overall crashes for raised medians. Model 2 did not outperform the other models in any case in this analysis.

The conclusion drawn from these results is that the models will perform differently dependent upon the type of crashes. In some of the cases, the data must have truly been overdispersed, and thus Model 3 performed best. This was the case with overall crashes on cable barriers where the number of crashes is larger than other crash types and on cross-median crashes for raised medians where the mean number of crashes is close to zero, but the variance is larger. In other cases, the data may not contain extra Poisson variability and Model 1 worked best. As a general procedure for the UDOT, it is recommend to fit all the models discussed and use the one that fits best for the given dataset in order to draw conclusions about whether crashes were reduced by the crash remediation project.

Table 4.4: Model Comparison for Cable Barriers

Severe Crashes				
Model	$\log(BF)$	BIC	$p_D$	DIC
1	-76.5032	227.1847	13.9729	146.1595
2	-48.4523	227.7714	16.3499	151.5002
3	-64.9612	280.8125	15.9477	149.2514
4	-247.3220	357.8889	47.2392	199.3991

Cross Median Crashes				
Model	$\log(BF)$	BIC	$p_D$	DIC
1	-65.4019	210.7403	12.2102	126.1897
2	-57.7426	211.0670	14.0148	130.1256
3	-60.9852	264.9844	12.6462	126.8204
4	-327.9390	331.0424	39.3265	156.7270

Overall Crashes				
Model	$\log(BF)$	BIC	$p_D$	DIC
1	-233.7677	399.1907	20.8409	331.9014
2	-181.7065	397.0614	22.0492	332.1889
3	-198.7380	466.9204	7.5178	318.4995
4	-227.3238	512.5594	49.5394	358.6695

Table 4.5: Model Comparison for Raised Medians

Severe Crashes				
Model	$\log(BF)$	BIC	$p_D$	DIC
1	-129.4234	310.8156	14.7889	239.8416
2	-125.6251	308.8434	17.0195	242.3307
3	-132.5646	367.6025	17.5369	243.4694
4	-460.0211	486.9270	58.4758	268.7063
Cross Median Crashes				
Model	$\log(BF)$	BIC	$p_D$	DIC
1	-54.5380	178.9118	8.7846	95.3929
2	-4.9821	171.2600	8.0946	86.8975
3	-34958.9400	227.1173	7.0213	81.9530
4	-754.9060	391.5683	18.0062	92.4083
Overall Crashes				
Model	$\log(BF)$	BIC	$p_D$	DIC
1	-440.9719	723.0915	20.2425	663.0246
2	-376.6124	715.2383	22.4042	659.4949
3	-404.8238	782.6950	10.4775	644.4431
4	-381.8761	697.3386	67.6413	496.8489

### 4.3 GOODNESS OF FIT

After determining the best models according to DIC, it is useful to determine whether or not the models fit well at all. In order to do this, Bayesian  $\chi^2$  values and corresponding  $p$ -values are computed for each of the sampled values from the posterior distribution as outlined in Section 2.6. The proportions of the  $p$ -values less than 0.05 are shown in Table 4.6. Values close to and below 0.05 indicate good fitting models. This table shows that none of the models fit cross-median crashes particularly well. One hypothesis as to why this occurred is perhaps that so many of the data values are zero (about 27% are zero for cable barriers and about 82% are zero for raised medians). Perhaps models 1 through 3 would fit better for these data if a zero-inflated Poisson likelihood were used. The same reason may apply to why none of the models fit excellently with the severe crashes (though the proportions of zero values are smaller for severe crashes). One explanation why the zero-inflated dynamic

linear model did not fit well in either of these cases is that fit of the dynamic linear model is more highly dependent upon the prior distributions specified. An additional simulation study may be required to further explore this hypothesis. For overall crashes, however, model 3 fits extremely well for cable barriers, and model 4 fits quite well for raised medians, especially when compared to the other models.

The model with the smallest DIC corresponded with the model that fit best for cross-median and overall crashes on cable barriers and for overall crashes on raised medians. For those where the same conclusion is not reached as to the preferable model, none of the models fit decidedly well. Additional models considering different likelihood and prior specifications may be explored to see if a model with better fit can be obtained.

Table 4.6: Proportion of Bayesian  $\chi^2$   $P$ -Values Less Than 0.05

Cable Barriers				
Crash Type	Model 1	Model 2	Model 3	Model 4
Severe	0.7689	0.7323	0.7822	0.9916
Cross-Median	0.8001	0.8660	0.9381	1.0000
Overall	0.1615	0.0725	0.0468	0.1136

Raised Medians				
Crash Type	Model 1	Model 2	Model 3	Model 4
Severe	0.4087	0.3921	0.3966	0.9943
Cross-Median	0.9998	1.0000	1.0000	1.0000
Overall	0.7033	0.6664	0.6995	0.0815

The second method for determining goodness of fit of a model did not lead to definitive conclusions as to which models fit well and which did not. Some problems with this method may have included the relatively small number of datapoints and the fact that the Poisson cumulative distribution function is a step function, rather than continuous. Future work with more observations may yield more success with this method.

After the goodness of fit analysis, it was determined to use those models selected by the DIC criterion in order to produce inferential results about the benefit of cable barriers and raised medians.

#### 4.4 POSTERIOR SUMMARIZATION

Once the posterior distributions for the model parameters are obtained, they must be summarized appropriately in order to be able to make informed decisions about the effectiveness of the traffic safety measure. Several meaningful summaries include the posterior distribution of factor change between before and after periods, the probability of reduction, posterior credible intervals, and distributions of differences.

##### *Posterior Distribution of Factor Change between Before and After Periods*

Comparing  $\theta_{Ai}$  with  $\theta_{Bi}$ , where  $\theta_{Ai}$  represents the mean number of crashes per mile using the covariates of the  $i^{th}$  observation in the after time period and  $\theta_{Bi}$  represents the mean number of crashes per mile using the covariates of the  $i^{th}$  observation in the before time period, is important in posterior summarization for before and after studies. One method of comparing  $\theta_{Ai}$  and  $\theta_{Bi}$  is by using the posterior distribution of the ratio  $\frac{\theta_{Ai}}{\theta_{Bi}}$ .

The posterior distribution of  $\frac{\theta_{Ai}}{\theta_{Bi}}$  can be obtained using simulation based change of variables. This principle states that if  $\boldsymbol{\theta}_1, \dots, \boldsymbol{\theta}_m$  are draws from the posterior distribution  $\pi(\boldsymbol{\theta}|y)$ , then  $g(\boldsymbol{\theta}_1), \dots, g(\boldsymbol{\theta}_m) \sim \pi(g(\boldsymbol{\theta})|y)$  which is the density of the transformed  $\boldsymbol{\theta}$  (Hamada et al. 2008). Thus, draws from the posterior distributions of  $\theta_{Ai}$  and  $\theta_{Bi}$  may be obtained for a specific level of *AADT* using the draws from the posterior distributions of  $\beta_A$ ,  $\beta_B$ , and  $\beta_1$  (the  $\beta$  coefficients in a model where there is an intercept for after and before time periods and *AADT* respectively). Draws from the posterior distribution of  $\frac{\theta_{Ai}}{\theta_{Bi}}$  may then be obtained by dividing the draws of the posterior distribution of  $\theta_{Ai}$  by the draws of the posterior distribution of  $\theta_{Bi}$ . From this distribution, several desired results can be obtained.

This method of obtaining the posterior of  $\frac{\theta_{A_i}}{\theta_{B_i}}$  requires that a specific value of  $AADT$  be used to obtain realizations from  $\theta_{A_i}$  and  $\theta_{B_i}$ . The distribution can also be obtained without using specific  $AADT$  values if the coefficient for  $AADT$  is modeled the same for before and after periods. Mathematically,  $\theta_{A_i}$  and  $\theta_{B_i}$  are denoted by

$$\theta_A = \exp(\beta_A + \beta_1 AADT_i), \text{ and} \quad (4.1)$$

$$\theta_B = \exp(\beta_B + \beta_1 AADT_i), \quad (4.2)$$

where  $\beta_B$  and  $\beta_A$  correspond to the before and after intercepts respectively, and  $\beta_1$  is the coefficient for the  $AADT$  covariate. The ratio of  $\theta_{A_i}$  and  $\theta_{B_i}$ , simplified to reflect the case where only the intercept is modeled differently for before and after time periods, is

$$\begin{aligned} \frac{\theta_{A_i}}{\theta_{B_i}} &= \frac{\exp(\beta_A + \beta_1 AADT_i)}{\exp(\beta_B + \beta_1 AADT_i)} \\ &= \exp(\beta_A - \beta_B + \beta_1 AADT_i - \beta_1 AADT_i) \\ &= \exp(\beta_A - \beta_B). \end{aligned} \quad (4.3)$$

Note that the final expression of Equation 4.3 is no longer dependent upon  $i$ . Thus, when only the intercept is changed for before and after analyses, the exponentiated distribution of the differences  $\beta_A - \beta_B$  is utilized to obtain the posterior distribution of  $\frac{\theta_{A_i}}{\theta_{B_i}}$ .

When this distribution is obtained, it may be represented graphically with a kernel density estimation. This allows the researcher to see how the after crashes compare to the before crashes in much more detail. The distribution conveys more than just a summary statistic.

### *Probability of Reduction*

The posterior distribution of  $\frac{\theta_{A_i}}{\theta_{B_i}}$  also allows the researcher to declare a probability of reduction in crashes. This is calculated as the proportion of simulated values from the posterior of  $\frac{\theta_{A_i}}{\theta_{B_i}}$  that are below one.

The reporting of this probability is a highly valuable feature of Bayesian hierarchical analysis that allows DOTs to independently determine what probability of decrease qualifies a project, from their perspective, as effective.

### *Credible Intervals*

The posterior distribution of  $\frac{\theta_{A_i}}{\theta_{B_i}}$  also results in the ability to measure the extent of the increase or decrease in crashes. This is done well by using the posterior mean or median and credible intervals.

A 95% HPD credible interval is the interval such that 95% of the posterior density is contained within the interval and the posterior density value of no point outside the HPD interval exceeds that of any point inside the interval (Carlin and Louis 2009). This interval is interpreted by stating that there is a 0.95 probability that the true factor reduction of crashes between the before and after periods is somewhere in the range of the interval. This intuitive interpretation is a key advantage of Bayesian methods.

### *Distributions of Differences*

The distribution of differences in mean number of severe crashes calculated with after minus before for a specific segment is also a useful posterior summary. Negative values suggest that the mean number of crashes was decreased between the before and after time periods.

An additional distribution of differences that is of interest is the posterior predictive distribution of differences. This distribution shows the probability that there were  $x$  fewer crashes per year per mile in the after period. From this distribution more specific probabilities, such as the probability that crashes were reduced by five or more crashes, can be calculated. This distribution is extremely useful when quantifying the true benefit of a safety project.

These posterior summaries are particularly useful for evaluating efficacy of traffic safety remediation in before-after studies. Similar results are available using the posterior distributions of the dynamic linear model.

#### 4.5 BEFORE-AFTER STUDY RESULTS

The results of the models selected using DIC for both cable barriers and raised medians for each crash type are now presented. A complete set of plots representing the posterior distribution of factor change and the posterior predictive distributions of differences for each site with respect to each type of crash is presented in Appendix A.

##### *Cable Barriers*

**SEVERE CRASHES** Model 1 was selected to analyze the severe crash data evaluating cable barriers. The results are presented in Table 4.7. Sites 2 and 6 were most successful in having a reduction of crashes after the cable barriers were installed, with probabilities of reduction equal to 0.96 and 0.94 respectively. Moderate success is also observed on sites 4, 5, and 7 where the probability of reduction is above 0.80. Though the other sites have no strong indication of reduction, none of them show indication that severe crashes were increased by the cable barriers.

Table 4.7: Results for Severe Crashes on Cable Barriers.

Site	Route	Probability of Decrease	Lower Factor Change	Median Factor Change	Upper Factor Change
1	15	0.67	0.15	0.77	2.12
2	15	0.96	0.01	0.24	0.90
3	15	0.59	0.12	0.86	2.50
4	15	0.81	0.05	0.56	1.63
5	15	0.81	0.02	0.48	1.81
6	215	0.94	0.01	0.26	1.04
7	215	0.83	0.03	0.49	1.65

CROSS-MEDIAN CRASHES Model 1 was utilized to analyze the cross-median crash data for cable barriers. The results are presented in Table 4.8. Site 2 had the most drastic result with a 0.94 probability of reduction. Sites 6 and 7 also have large probabilities of reduction. The remaining three sites have probabilities greater than or equal to 0.70. It is therefore concluded that cable barriers are quite effective at reducing cross-median crashes.

Table 4.8: Results for Cross-Median Crashes on Cable Barriers

Site	Route	Probability of Decrease	Lower Factor Change	Median Factor Change	Upper Factor Change
1	15	0.74	0.04	0.61	2.19
2	15	0.94	0.02	0.31	1.05
3	15	0.78	0.03	0.54	2.00
4	15	0.70	0.03	0.65	2.51
5	15	0.72	0.02	0.58	2.52
6	215	0.89	0.01	0.34	1.44
7	215	0.88	0.01	0.35	1.51

OVERALL CRASHES Model 3 was used to produce the results for overall crashes on cable barriers. These results are displayed in Table 4.9. The only site with an impressive probability of reduction in crashes was site 6 with a probability of 1.00. Several of the sites have very low probabilities that indicate that overall crashes are not decreased by cable-barriers.

Table 4.9: Results for Overall Crashes on Cable Barriers

Site	Route	Probability of Decrease	Lower Factor Change	Median Factor Change	Upper Factor Change
1	15	0.01	1.02	1.46	1.99
2	15	0.26	0.60	1.21	2.07
3	15	0.00	1.34	1.96	2.69
4	15	0.00	1.13	1.51	1.95
5	15	0.67	0.66	0.94	1.24
6	215	1.00	0.28	0.53	0.82
7	215	0.00	1.13	1.56	2.07

Overall, it appears that cable barriers were quite effective at reducing severe crashes and cross-median crashes, although they were ineffective at reducing overall crashes. It is therefore asserted that cable barriers should continue to be implemented at locations where severe crashes are high, particularly when many are the result of crossing over the median of the highway.

### *Raised Medians*

SEVERE CRASHES Model 1 was used to evaluate the reduction of severe crashes on raised median sites. These results are shown in Table 4.10. Sites 1 and 3 have high probabilities of reduction of severe crashes, 1.00 and 0.99 respectively. Sites 5 and 6 also have probabilities of reduction greater than 0.80. For sites 2 and 4, it is likely that there was no effect of the raised medians on severe crashes.

Table 4.10: Results for Severe Crashes on Raised Medians

Site	Route	Probability of Decrease	Lower Factor Change	Median Factor Change	Upper Factor Change
1	265	1.00	0.21	0.41	0.67
2	74	0.41	0.11	1.20	4.50
3	186	0.99	0.11	0.35	0.77
4	71	0.46	0.42	1.03	2.04
5	34	0.85	0.11	0.56	1.41
6	36	0.90	0.04	0.40	1.30

CROSS-MEDIAN CRASHES Cross-median crashes on raised medians were fit using model 3. The results of this analysis are shown in Table 4.11. None of the sites appear to have undergone a reduction in cross-median crashes due to the raised medians. Perhaps this is because of the inherently low values of cross-median crashes on raised medians. The lack of fit of this model may also be influential in obtaining this result.

Table 4.11: Results for Cross-Median Crashes on Raised Medians

Site	Route	Probability of Decrease	Lower Factor Change	Median Factor Change	Upper Factor Change
1	265	0.03	0.03	11.26	258.44
2	74	0.17	0.00	4.44	94.28
3	186	0.14	0.01	5.21	120.33
4	71	0.10	0.05	4.93	56.29
5	34	0.08	0.02	8.27	195.26
6	36	0.32	0.00	2.08	33.61

OVERALL CRASHES The dynamic linear model (model 4) was selected to model overall crashes on raised medians. This model both had a dramatically lower DIC and a better fit than the other three models for the same data. Figures 4.1 through 4.6 show the graphical representations of the dynamic linear model. As an explanation of the figures, the data are indicated with points, the center dotted band represents the posterior median, the dashed bands are 95% HPD credible intervals for the mean, and the dashed-dotted bands are 95% HPD credible intervals for the predictive distribution. The vertical solid line indicates the installation year.

The posterior medians follow the data quite well. This model, especially for site 1 due to the downward spike, may indeed be overfitting the data. It is noted that the number of overall crashes on raised medians is relatively high compared to the number of other crashes in the analysis. Perhaps the prior specification favored models where the mean number of crashes would be larger. This would potentially explain why the dynamic linear model performed so well in this case, but worse with the other response variables. For these data, the dynamic linear model appears to be very effective.

Table 4.12 presents comparable results for model 4 as were shown for data analyzed with models 1 and 3. The before and after distributions of means were created by using simulation-based change of variables to produce average distributions of the before and after periods. These were then used to compare the treatments as with models 1 and 3. The

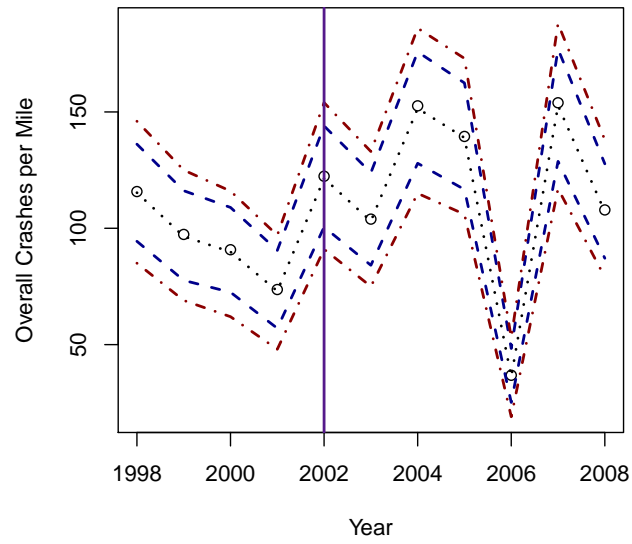


Figure 4.1: Dynamic Linear Model for Site 1.

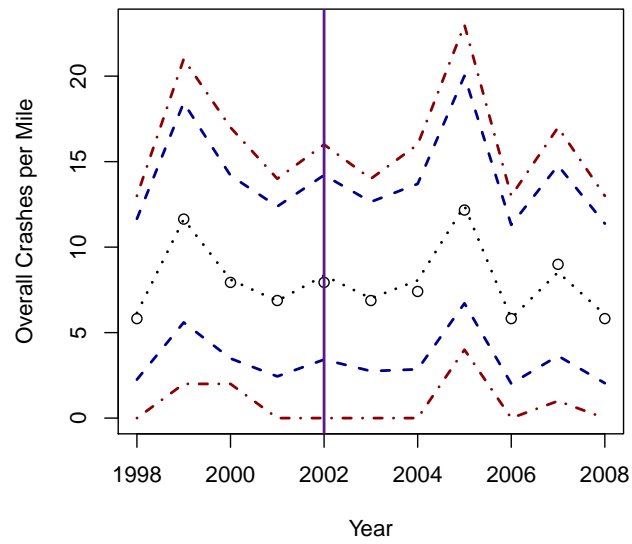


Figure 4.2: Dynamic Linear Model for Site 2.

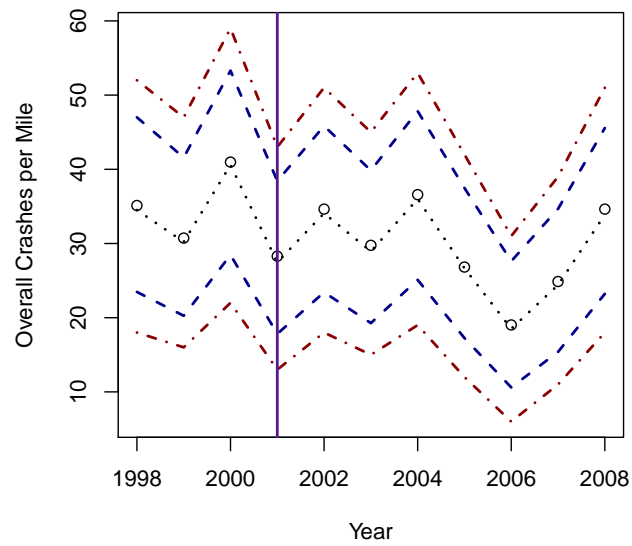


Figure 4.3: Dynamic Linear Model for Site 3.

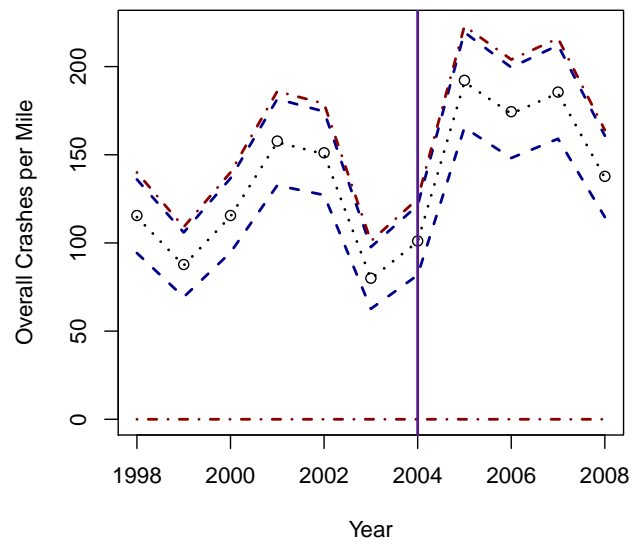


Figure 4.4: Dynamic Linear Model for Site 4.

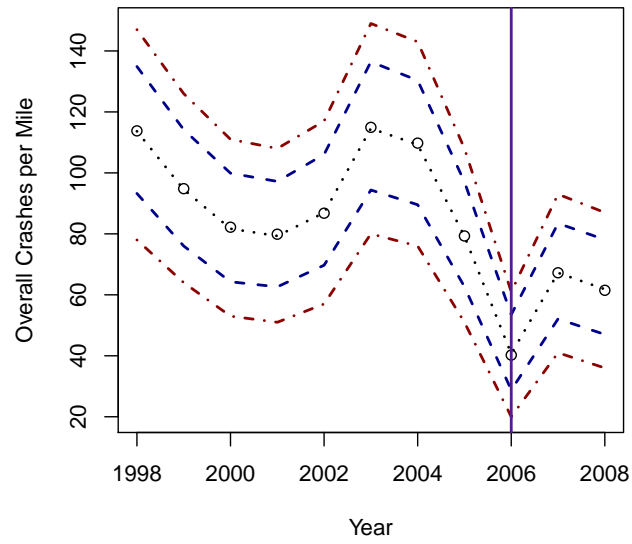


Figure 4.5: Dynamic Linear Model for Site 5.

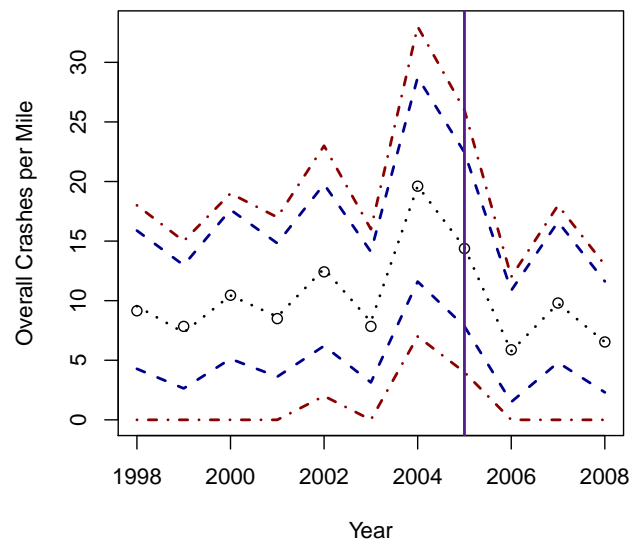


Figure 4.6: Dynamic Linear Model for Site 6.

probability of reduction in crashes is high for sites 3, 5, and 6. These sites almost certainly had a reduction of crashes in the after period, as may be observed in their respective dynamic linear model figures. In the case of sites 1 and 4, there appears to have been an increase in overall crashes. Limitations of this dynamic linear model, however, include that no covariate for *AADT* was utilized in the analysis. Perhaps crashes increased on those roadways simply because the traffic volume increased. In any case, the dynamic linear model appears to have been effective in establishing the efficacy of raised medians in reducing overall crashes on several of the study sites.

Table 4.12: Results for Overall Crashes on Raised Medians

Site	Route	Probability of Decrease	Lower Factor Change	Median Factor Change	Upper Factor Change
1	265	0.04	0.98	1.13	1.28
2	74	0.56	0.59	0.96	1.45
3	186	0.95	0.63	0.81	1.02
4	71	0.00	1.34	1.50	1.66
5	34	1.00	0.56	0.69	0.82
6	36	0.94	0.39	0.69	1.04

All crash types considered, the raised medians appear to be effective at reducing severe and, to a large extent, overall crashes. Cross-median crashes, however, do not appear to be reduced by raised medians.

#### 4.6 TRADITIONAL RESULTS

The traditional model as described in Section 2.1 was also utilized to determine which sites had a reduction of crashes due to the respective safety project. The reference population was formed from the segments of roadway just before and just after the segments where the safety project was implemented. While a more diverse and larger reference population may be obtained, this reference ought to provide detail as to whether or not any reduction of crashes is occurring due to the safety project. One potential confounding variable is the fact

that there may be residual benefits of the raised median or cable barrier on the roadway segments just before or just after where the safety measure was installed.

The SPFs and overdispersion parameters were estimated with negative binomial regression using *AADT* and segment length as predictor variables. Only the after period was considered. Separate analyses were conducted for each year as the number of after years for each site was not uniform.

The results of this analysis are presented in Appendix B. While this analysis is somewhat simplified as compared to the full set of procedures described by Hauer et al. (2002) and AASHTO (2010), the results show that the only improvement on all study sites over all crash and safety project types were overall crashes for cable barriers on site 1 during all years and on site 4 during 2008. Enough tests were performed, however, that the Type I error rate is expected to be inflated. The hierarchical Bayesian approach yields more interpretable and dependable results without the need to obtain data from a reference population or construct an SPF separately from the overall analysis.

---

CONCLUSION

In this project, five models for evaluating efficacy of safety projects by DOTs across the United States have been presented. Three of these models are variations of Bayesian Poisson regression. One of these models used a dynamic linear model under the zero-inflated Poisson likelihood to model the crash data. An additional model represented traditional approaches used commonly by civil engineers in evaluating safety projects.

The model that fits best is not always clear before modeling the data itself. It is recommended that DOTs fit several models for their specific data, perhaps all five discussed in this project, and use DIC and goodness of fit measures in order to select the best model for their situation. This process can be largely automated for improvement of efficiency and for ease of implementation. The results obtained from Bayesian analyses provides strong flexibility and interpretation in making decisions about how to utilize safety budgets most effectively to save lives on U.S. roadways.

The analyses show that cable barriers are very effective at reducing severe and cross-median crashes. Raised medians are also quite effective at reducing severe crashes. Now efficacy is established, these models may be extended and used to identify sites that will likely benefit most from these and other safety projects.

Future work with this project involves determining the prior sensitivity of the dynamic linear models and developing methods for eliciting effective prior distributions for such a model. Also, it would be interesting to extend the Poisson regression in models 1 through 3 to a zero-inflated Poisson likelihood in certain situations—such as with cross-median crashes. These models should be compared to those presented in this research using DIC and goodness

of fit procedures in order to determine which models perform best for before-after studies of traffic remediation projects.

Overall, the effects of this project on safety can be extremely beneficial in determining the efficacy of existing and new safety projects. DOTs can use these principles to continually evaluate their efforts in order to determine how to best serve the public and save lives of the citizens who drive on their roadways each day.

## BIBLIOGRAPHY

- American Association of State Highway and Transportation Officials (AASHTO) (2006), *Roadside Design Guide* (3rd ed.), Washington D.C.: Author.
- (2010), *Highway Safety Manual* (Vol. 2, 1st ed.), Washington D.C.: Author.
- Bolstad, W. M. (2010), *Understanding Computational Bayesian Statistics*, Hoboken, NJ: John Wiley & Sons, Inc.
- Carlin, B. P., and Louis, T. A. (2009), *Bayesian Methods for Data Analysis* (3rd ed.), Boca Raton, FL: Chapman & Hall/CRC.
- Casella, G., and Berger, R. L. (2002), *Statistical Inference* (2nd ed.), Pacific Grove, CA: Duxbury.
- DiCiccio, T. J., Kass, R. E., Raftery, A., and Wasserman, L. (1997), “Computing Bayes Factors by Combining Simulation and Asymptotic Approximations,” *Journal of the American Statistical Association*, 92, 439, 903–915.
- Gelfand, 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. B., Stern, H. S., and Rubin, D. B. (2004), *Bayesian Data Analysis* (2nd ed.), Boca Raton, FL: Chapman & Hall/CRC.
- Geman, S., and Geman, D. (1984), “Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6, 721–741.

- Hamada, M. S., Wilson, A. G., Reese, C. S., and Martz, H. F. (2008), *Bayesian Reliability*, New York: Springer Science+Business Media, LLC.
- Hastings, W. K. (1970), “Monte Carlo Sampling Methods Using Markov Chains and Their Applications,” *Biometrika*, 57, 97–109.
- Hauer, E. (1997), *Observational Before–After Studies in Road Safety*, Tarrytown, New York: Elsevier Science Inc.
- Hauer, E., Harwood, D. W., Council, F. M., and Griffith, M. S. (2002), “Estimating Safety by the Empirical Bayes Method,” *Transportation Research Record: Journal of the Transportation Research Board*, 1784, 126–131.
- Hocking, R. R. (2003), *Methods and Applications of Linear Models: Regression and the Analysis of Variance* (2nd ed.), Hoboken, NJ: John Wiley & Sons, Inc.
- Lawson, J. (2010), *Design and Analysis of Experiments with SAS*, Boca Raton, FL: Chapman & Hall/CRC.
- Leininger, T. J., Reese, C. S., Burraston, B., and Rutkowski, M. (2010), “Criminal Career Trajectory Latent Class Analysis using a Bayesian Hierarchical Dynamic Linear Model,” Brigham Young University, Technical Report.
- Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., and Teller, E. (1953), “Equation of State Calculations by Fast Computing Machines,” *The Journal of Chemical Physics*, 21, 1087–1091.
- Miranda-Moreno, L. F., Fu, L., Saccomanno, F. F., and Labbe, A. (2005), “Alternative Risk Models for Ranking Locations for Safety Improvement,” *Transportation Research Record: Journal of the Transportation Research Board*, 1908, 1–8.
- Schultz, G. G., Thurgood, D. J., Olsen, A. N., and Reese, C. S. (2010), *Transportation Safety Data and Analysis: Volume 1: Analyzing the Effectiveness of Safety Measures Using*

*Bayesian Methods.*, Salt Lake City, UT: UDOT Report No. UT-10.12a, Utah Department of Transportation, Research Division.

Transportation Research Board (TRB) (2003), *Access Management Manual*, Washington D.C.: Author.

West, M., and Harrison, J. (1989), *Bayesian Forecasting and Dynamic Models*, New York: Springer-Verlag New York Inc.



## APPENDICES



## HIERARCHICAL MODEL RESULTS

This appendix contains a complete set of results for each type of crash for both cable barriers and raised medians. The plot on the left represents the factor change in crashes between the before and after periods. Values larger than 1 indicate an increase in crashes while values less than 1 indicate a reduction in crashes. The plot on the right is the posterior predictive distribution of differences, which is the probability of a certain number of crashes being reduced (negative numbers) or increased (positive numbers) between the before and after time periods.

## A.1 CABLE BARRIERS

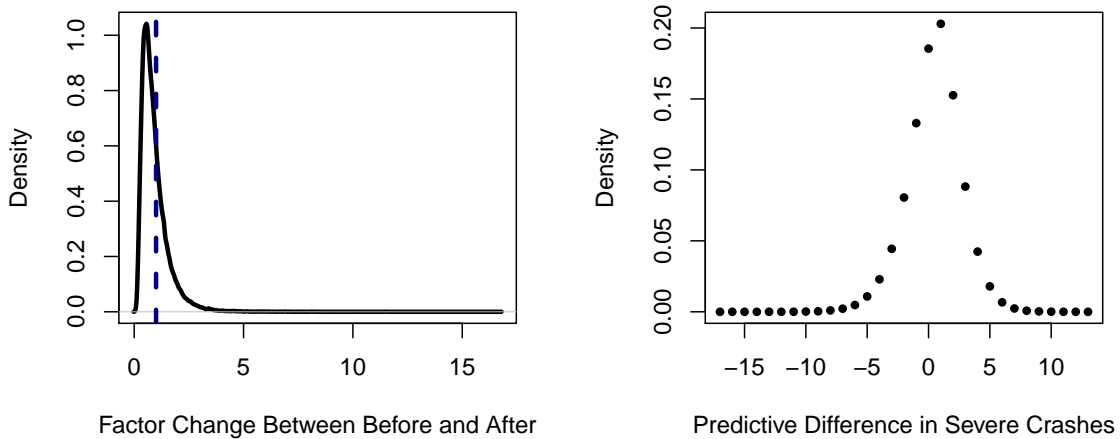
*Severe Crashes*

Figure A.1: Severe Crash Results on Cable Barriers for Site 1.

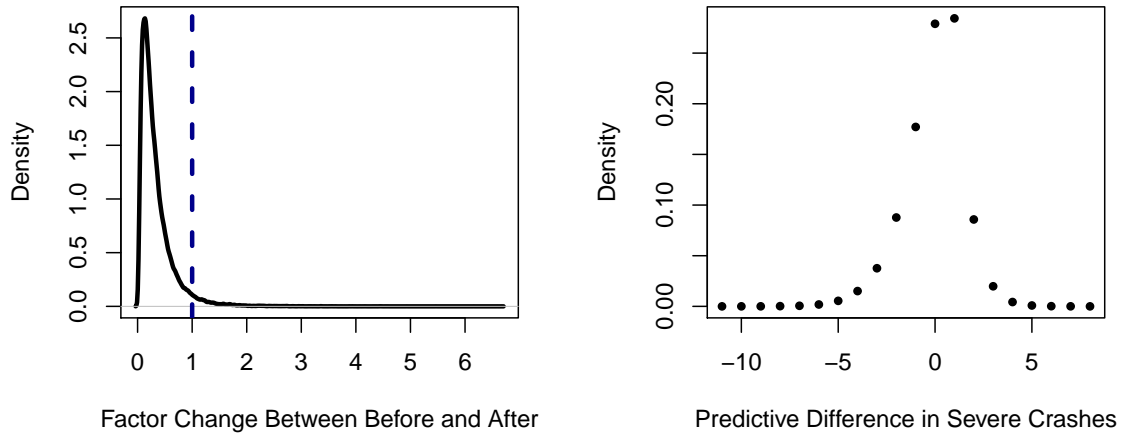


Figure A.2: Severe Crash Results on Cable Barriers for Site 2.

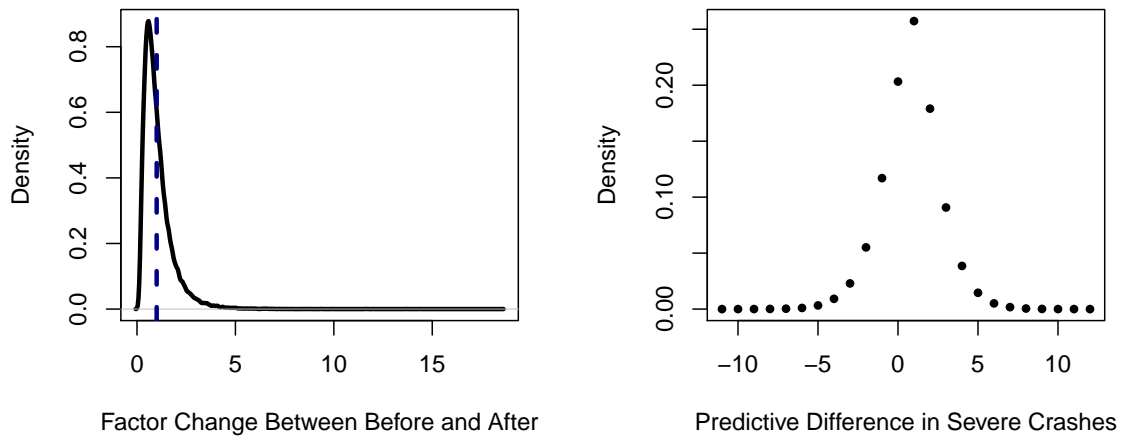


Figure A.3: Severe Crash Results on Cable Barriers for Site 3.

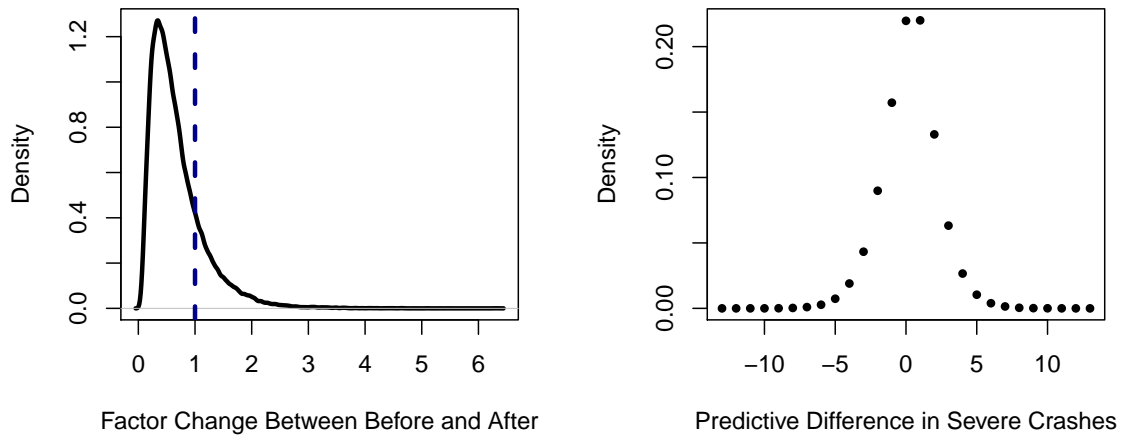


Figure A.4: Severe Crash Results on Cable Barriers for Site 4.

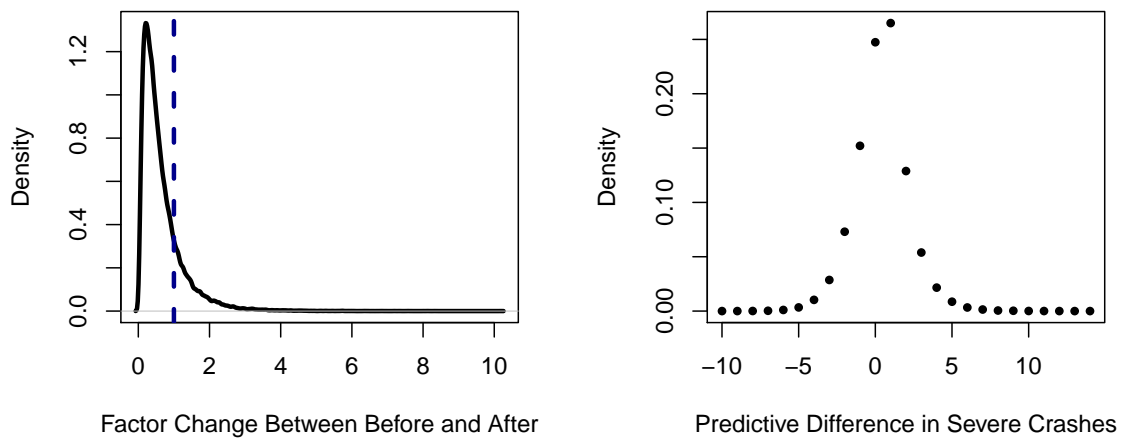


Figure A.5: Severe Crash Results on Cable Barriers for Site 5.

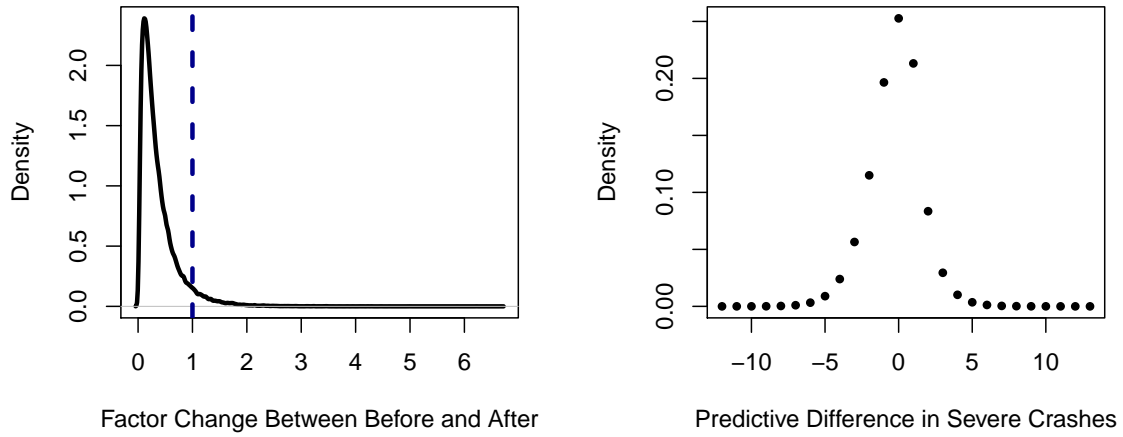


Figure A.6: Severe Crash Results on Cable Barriers for Site 6.

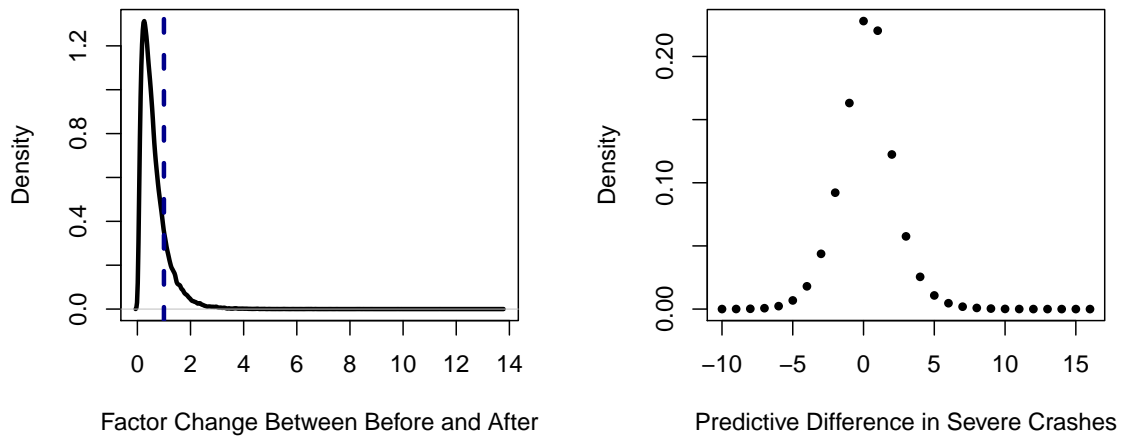


Figure A.7: Severe Crash Results on Cable Barriers for Site 7.

## Cross-Median Crashes

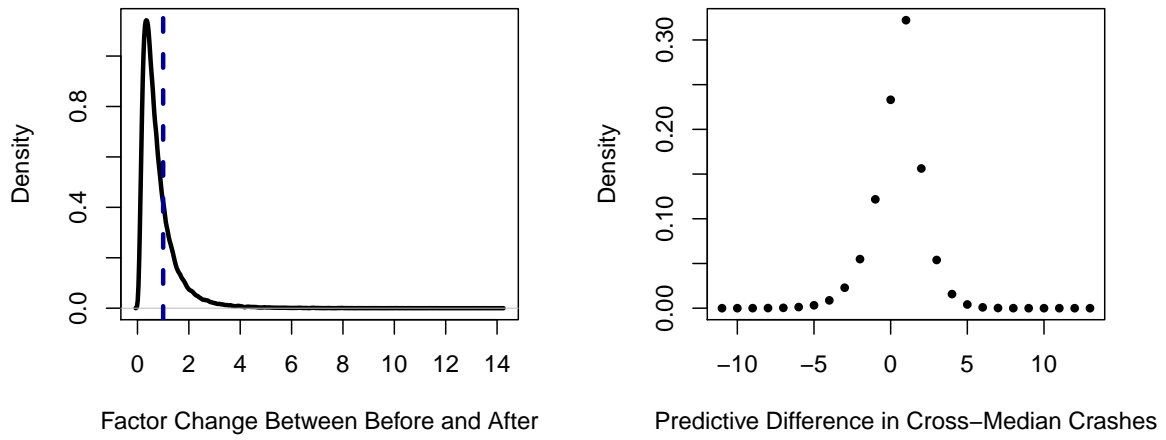


Figure A.8: Cross-Median Crash Results on Cable Barriers for Site 1.

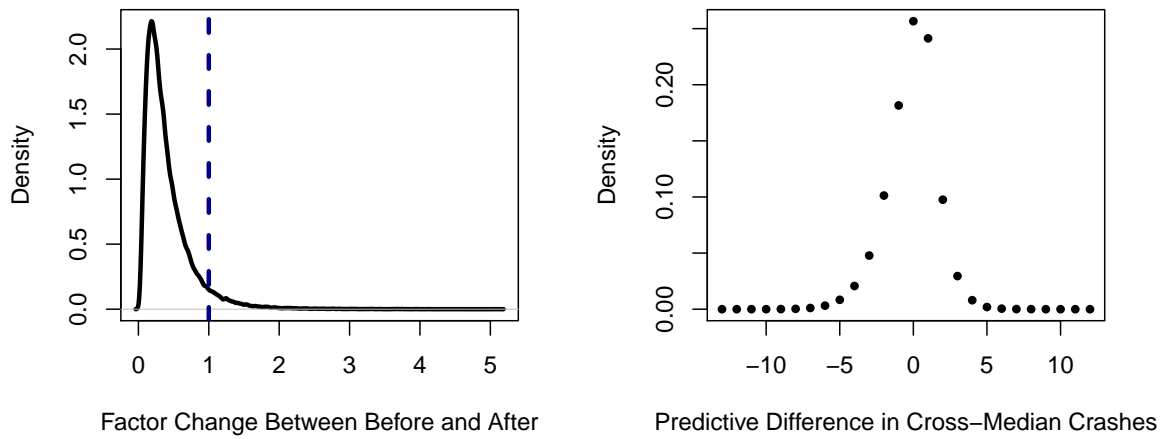


Figure A.9: Cross-Median Crash Results on Cable Barriers for Site 2.

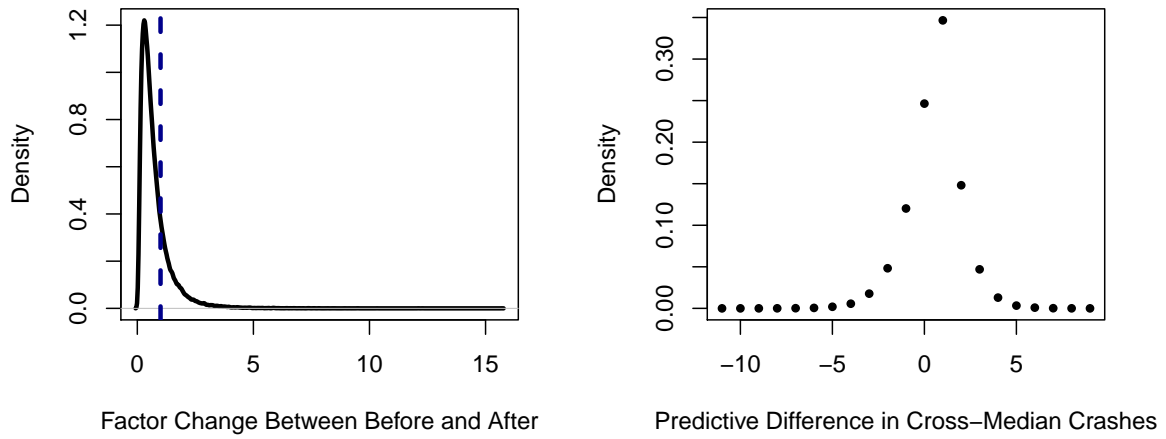


Figure A.10: Cross-Median Crash Results on Cable Barriers for Site 3.

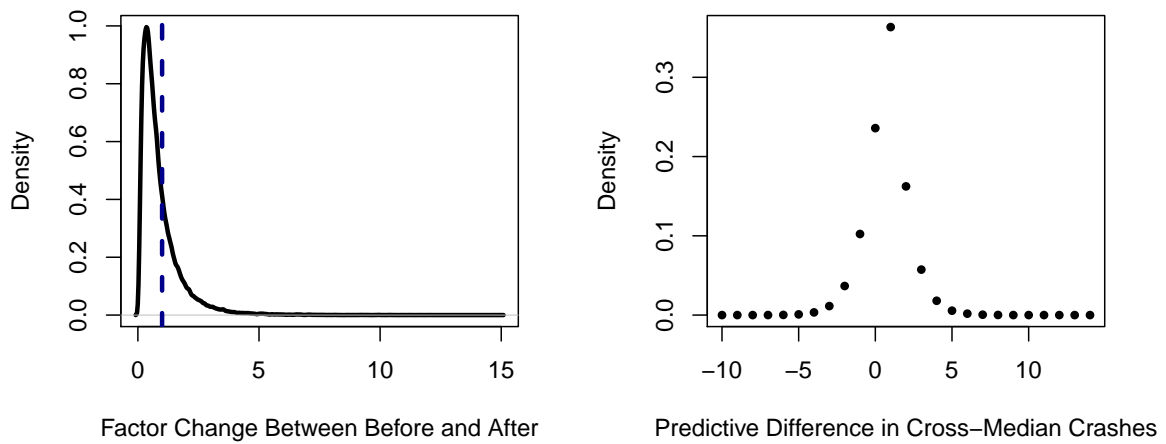


Figure A.11: Cross-Median Crash Results on Cable Barriers for Site 4.

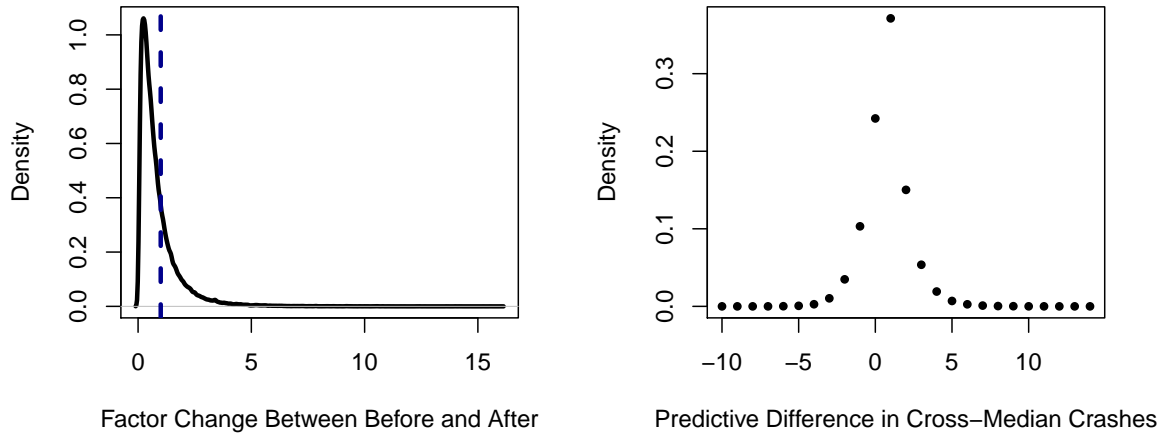


Figure A.12: Cross-Median Crash Results on Cable Barriers for Site 5.

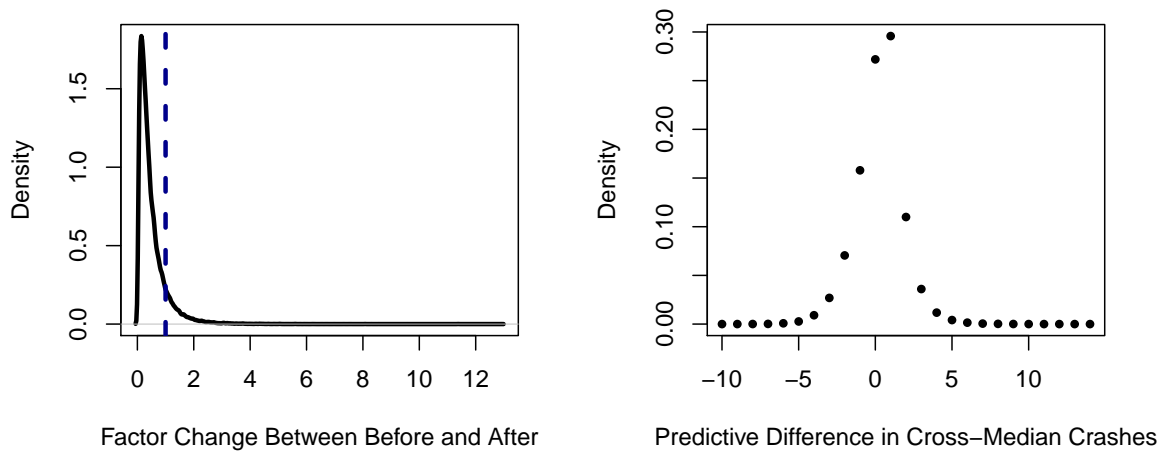


Figure A.13: Cross-Median Crash Results on Cable Barriers for Site 6.

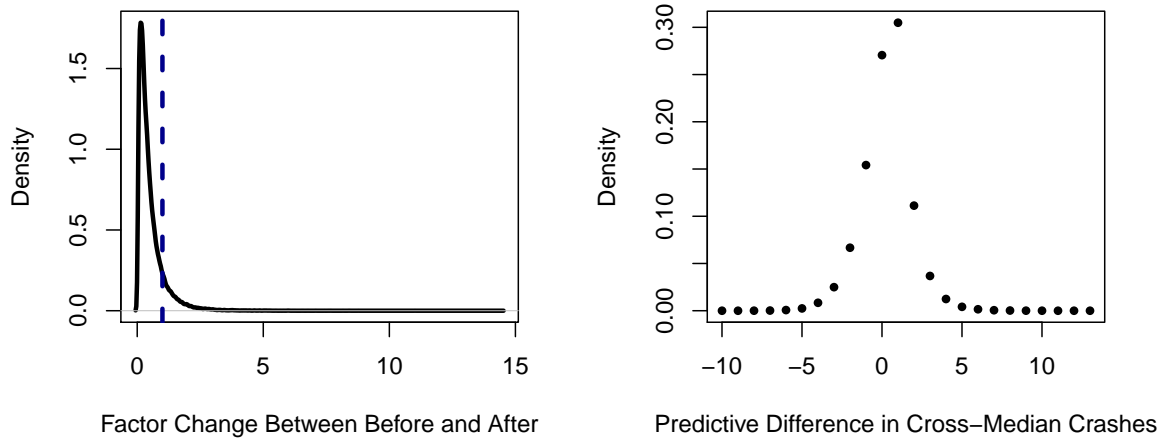


Figure A.14: Cross-Median Crash Results on Cable Barriers for Site 7.

## Overall Crashes

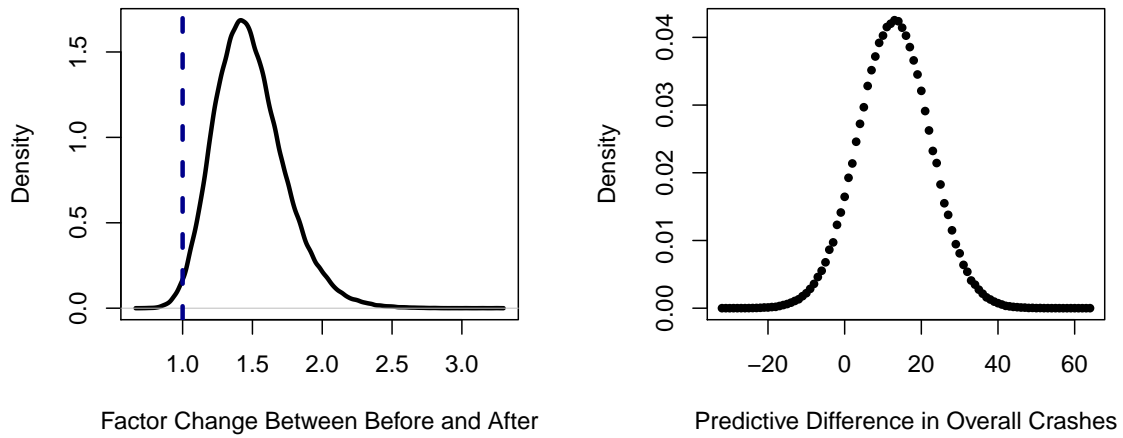


Figure A.15: Overall Crash Results on Cable Barriers for Site 1.

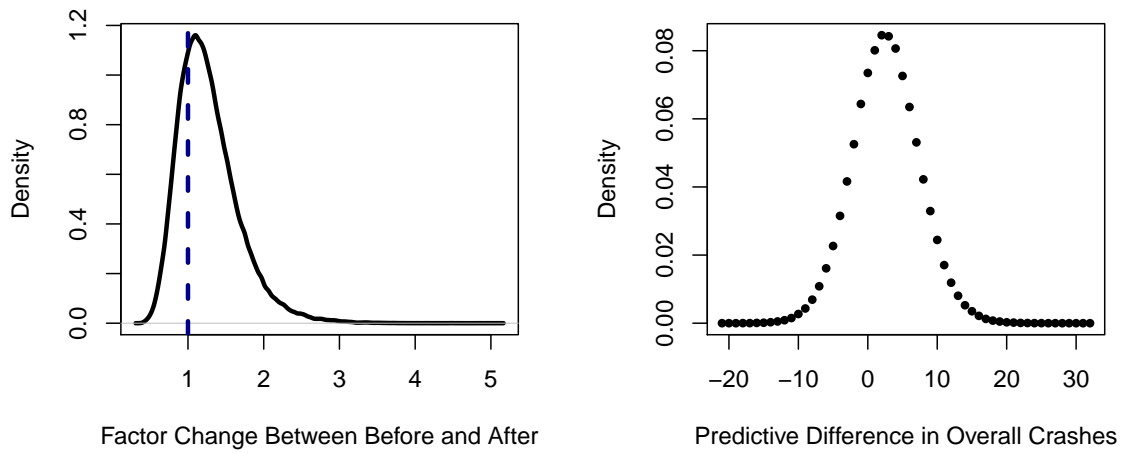


Figure A.16: Overall Crash Results on Cable Barriers for Site 2.

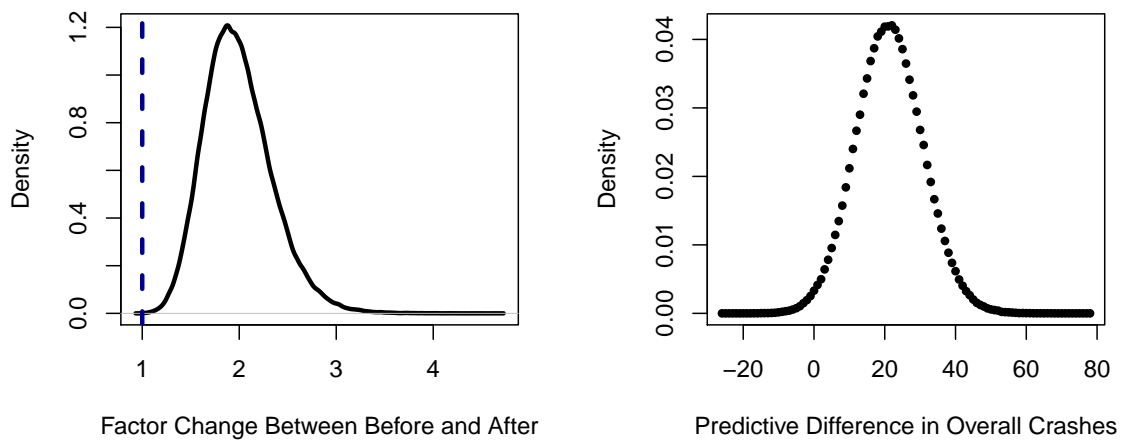


Figure A.17: Overall Crash Results on Cable Barriers for Site 3.

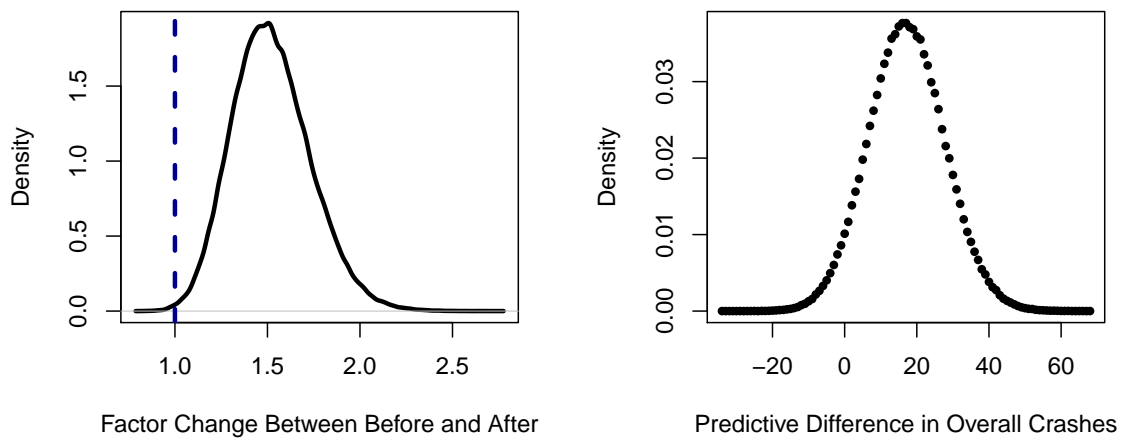


Figure A.18: Overall Crash Results on Cable Barriers for Site 4.

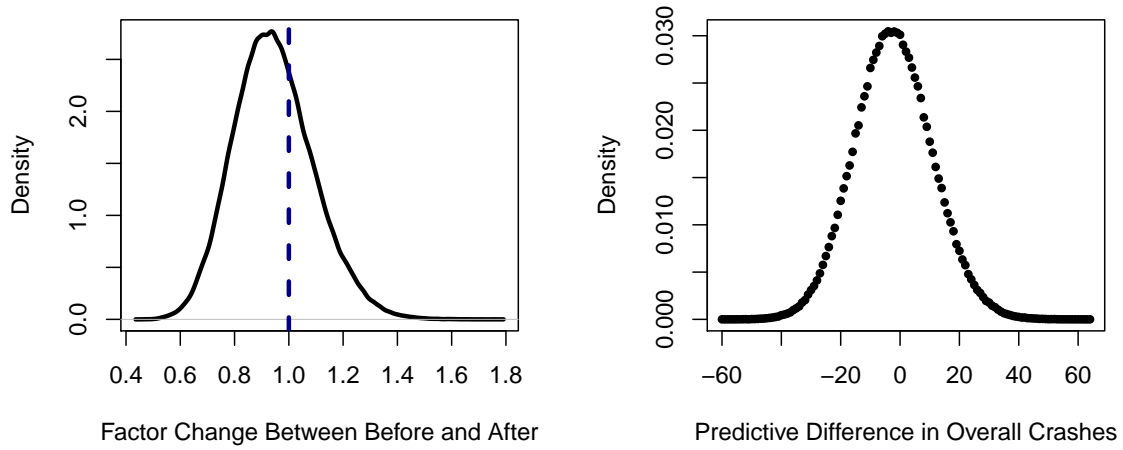


Figure A.19: Overall Crash Results on Cable Barriers for Site 5.

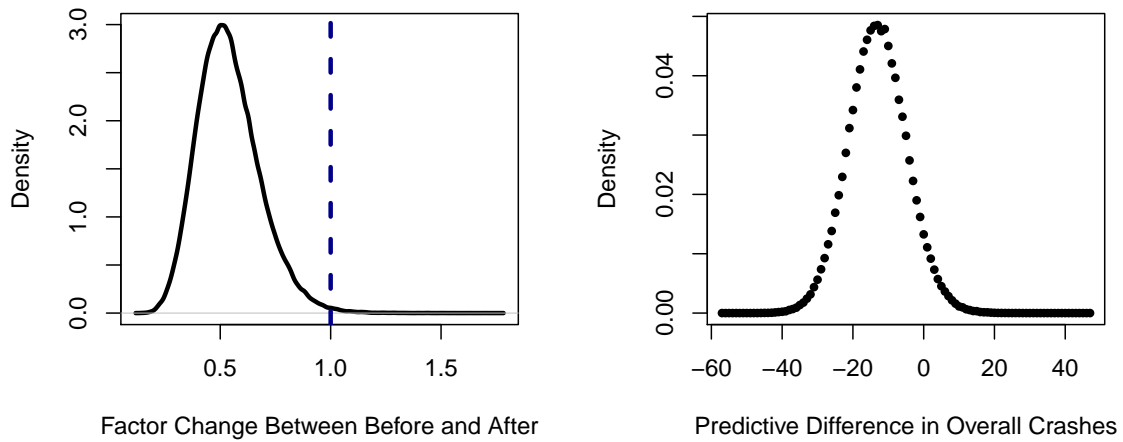


Figure A.20: Overall Crash Results on Cable Barriers for Site 6.

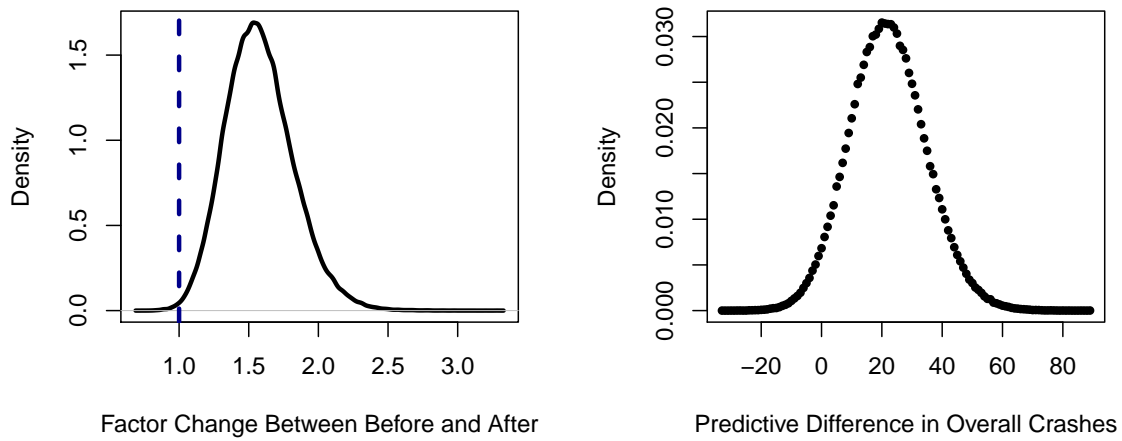


Figure A.21: Overall Crash Results on Cable Barriers for Site 7.

## A.2 RAISED MEDIANS

### *Severe Crashes*

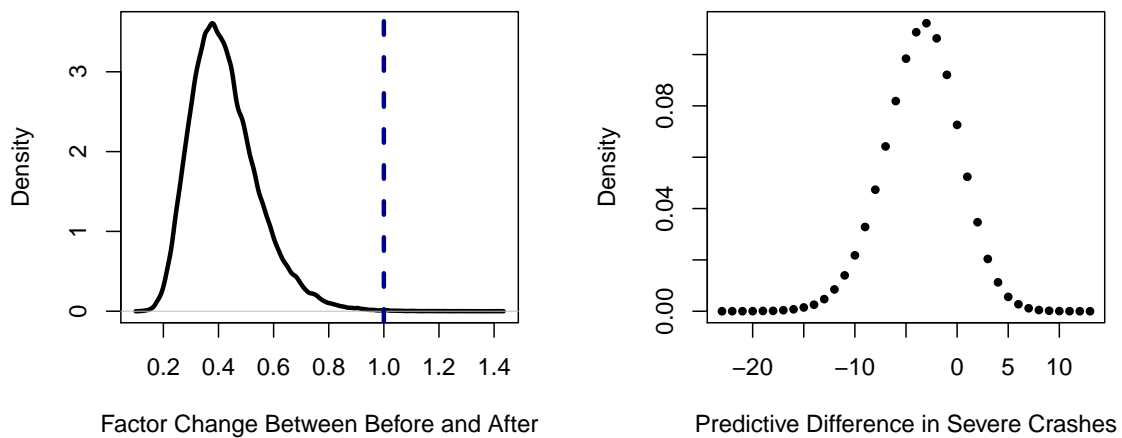


Figure A.22: Severe Crash Results on Raised Medians for Site 1.

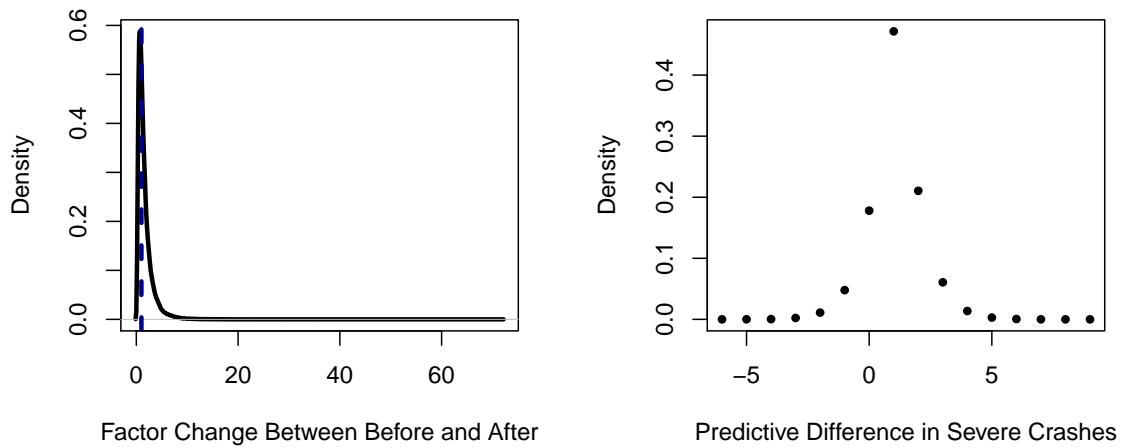


Figure A.23: Severe Crash Results on Raised Medians for Site 2.

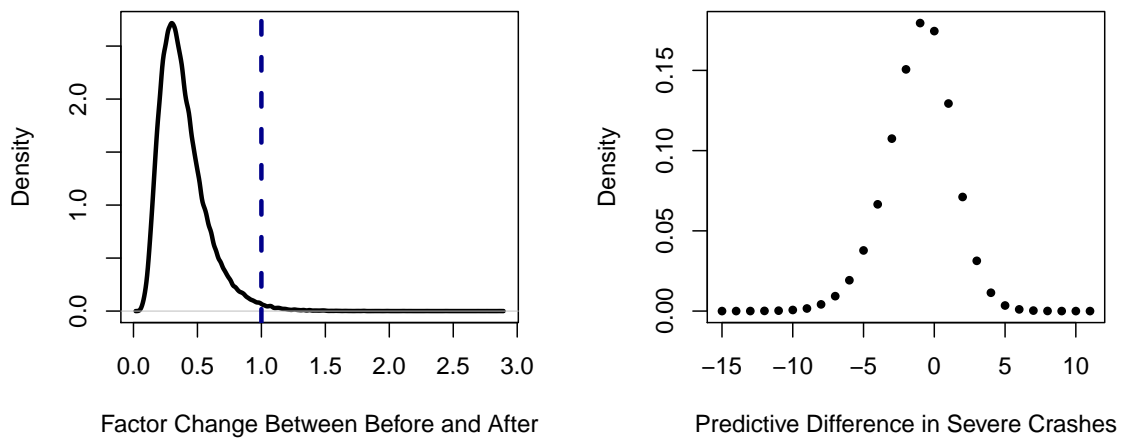


Figure A.24: Severe Crash Results on Raised Medians for Site 3.

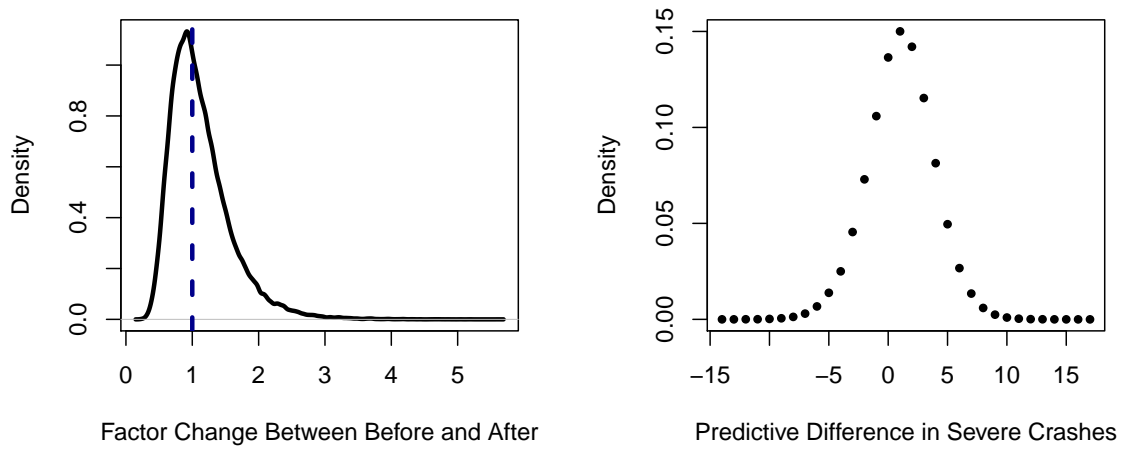


Figure A.25: Severe Crash Results on Raised Medians for Site 4.

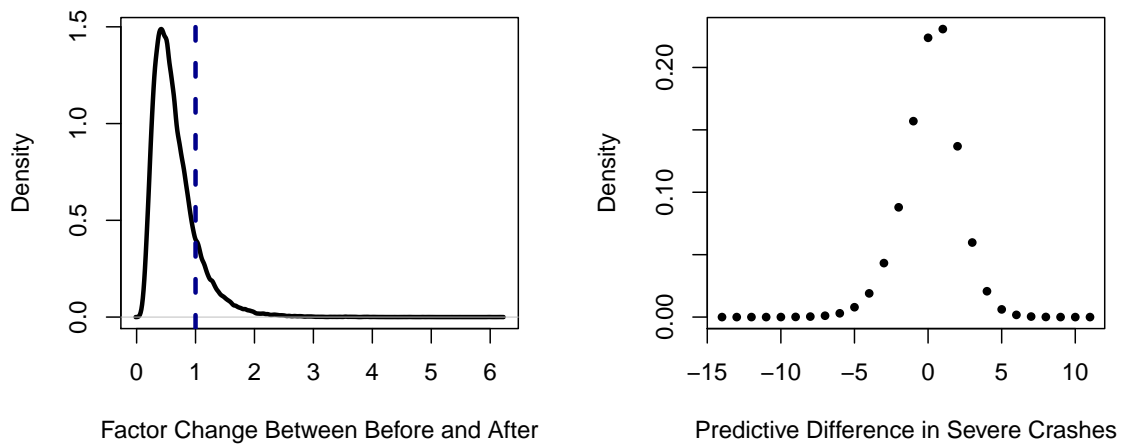


Figure A.26: Severe Crash Results on Raised Medians for Site 5.

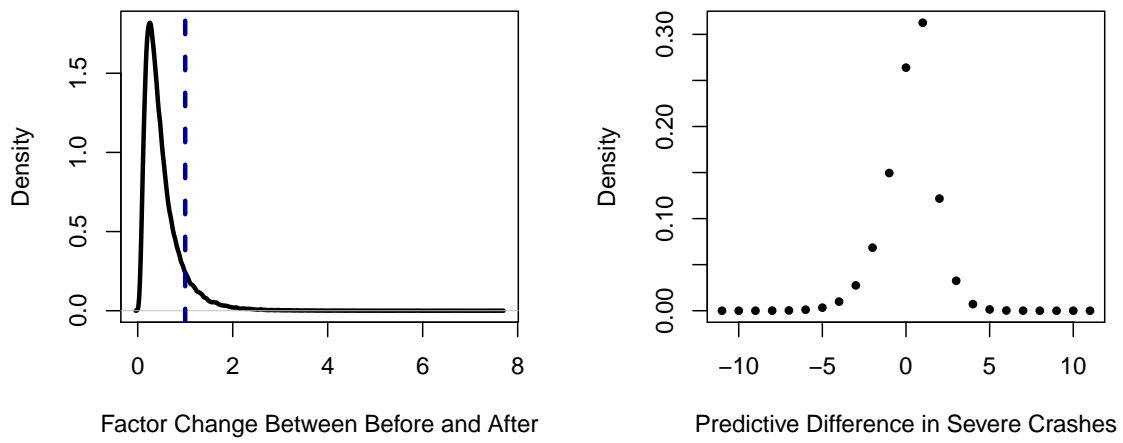


Figure A.27: Severe Crash Results on Raised Medians for Site 6.

## Cross-Median Crashes

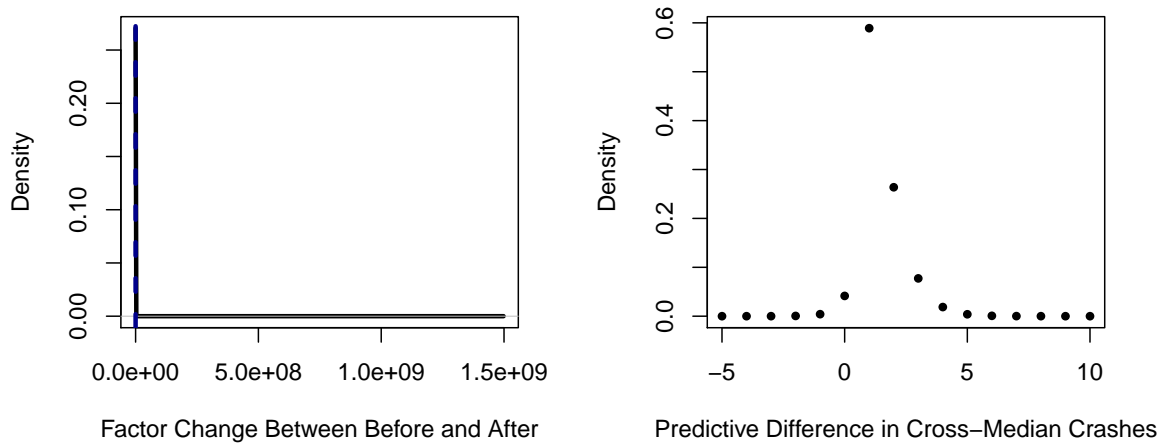


Figure A.28: Cross-Median Crash Results on Raised Medians for Site 1.

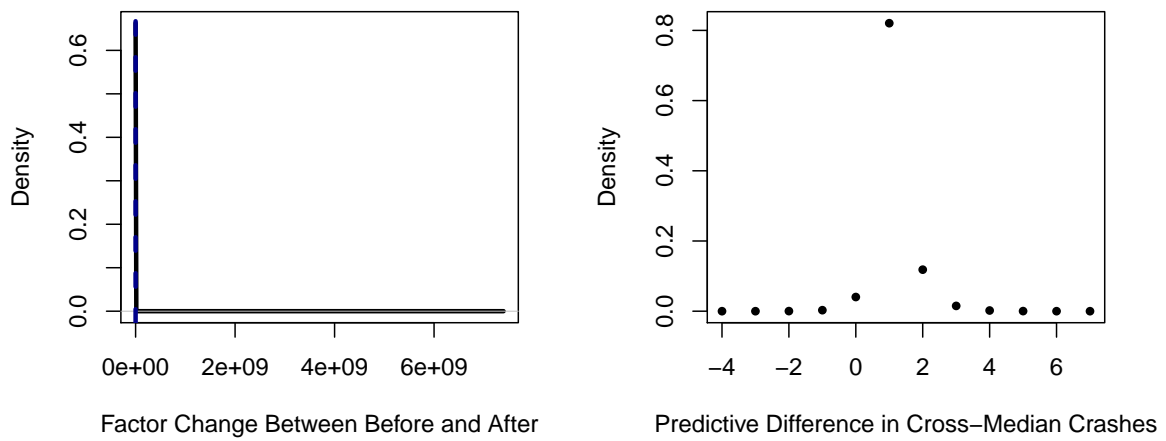


Figure A.29: Cross-Median Crash Results on Raised Medians for Site 2.

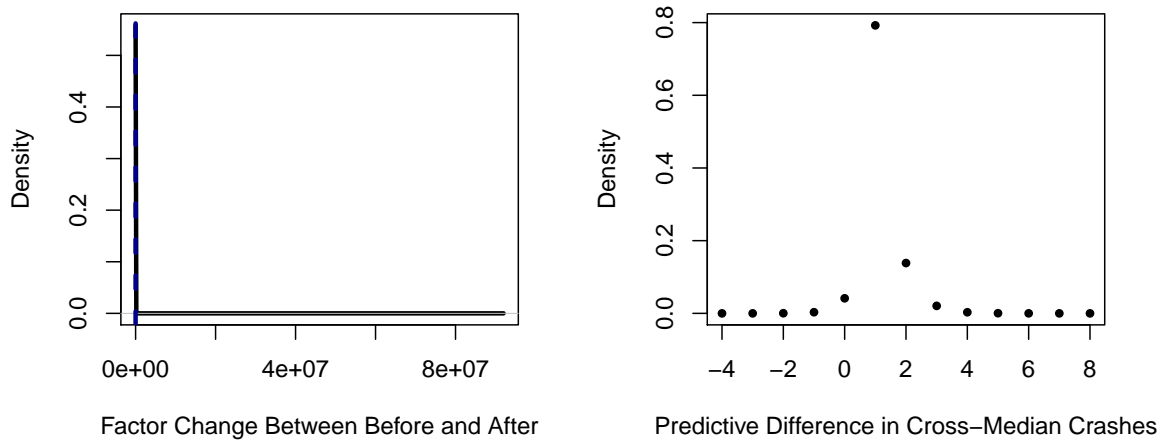


Figure A.30: Cross-Median Crash Results on Raised Medians for Site 3.

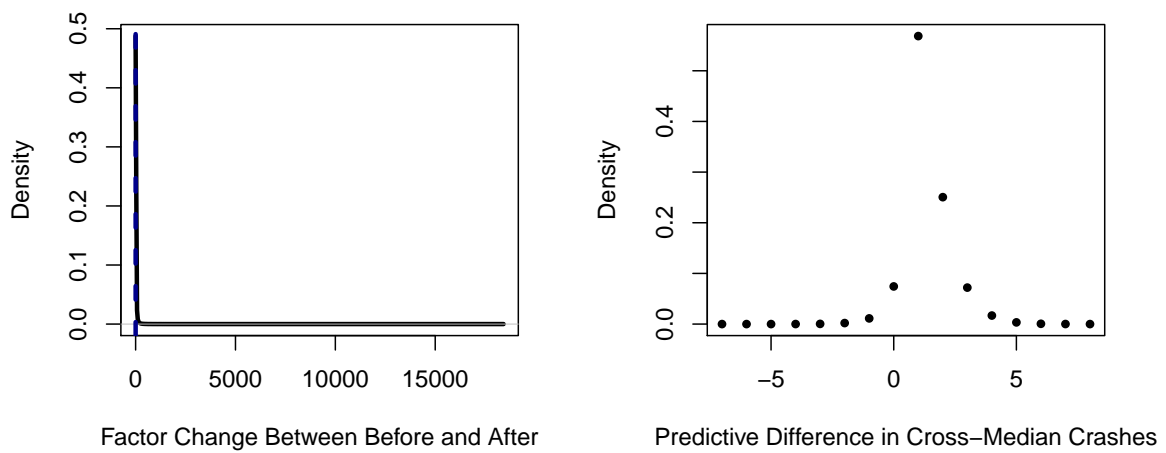


Figure A.31: Cross-Median Crash Results on Raised Medians for Site 4.

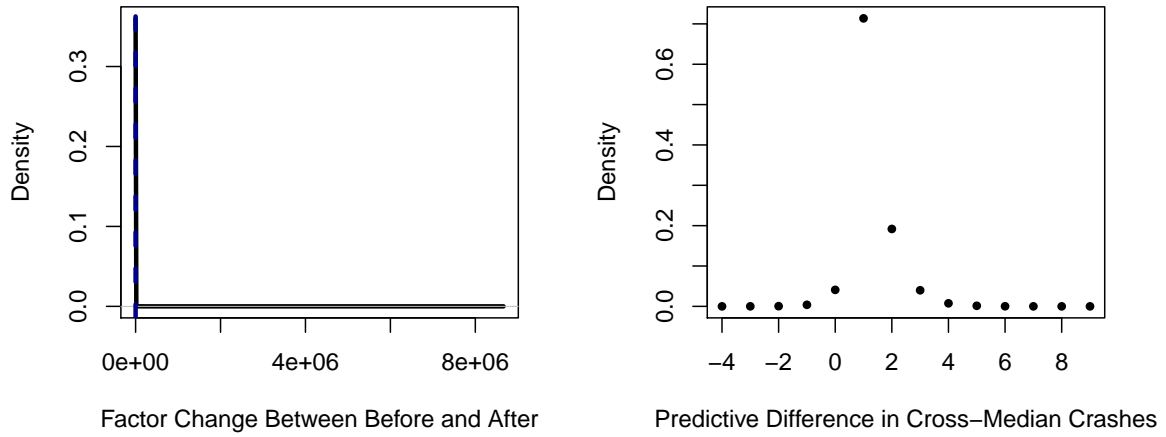


Figure A.32: Cross-Median Crash Results on Raised Medians for Site 5.

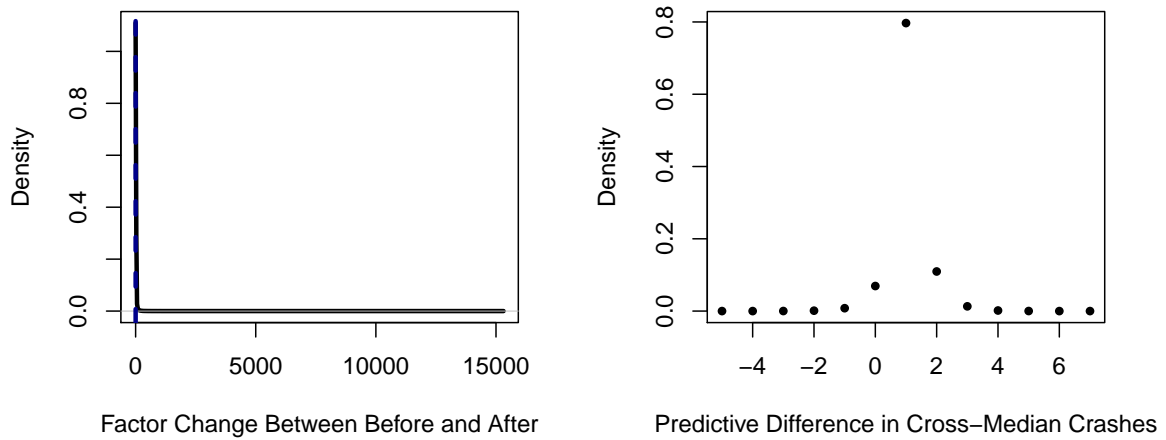


Figure A.33: Cross-Median Crash Results on Raised Medians for Site 6.

## Overall Crashes

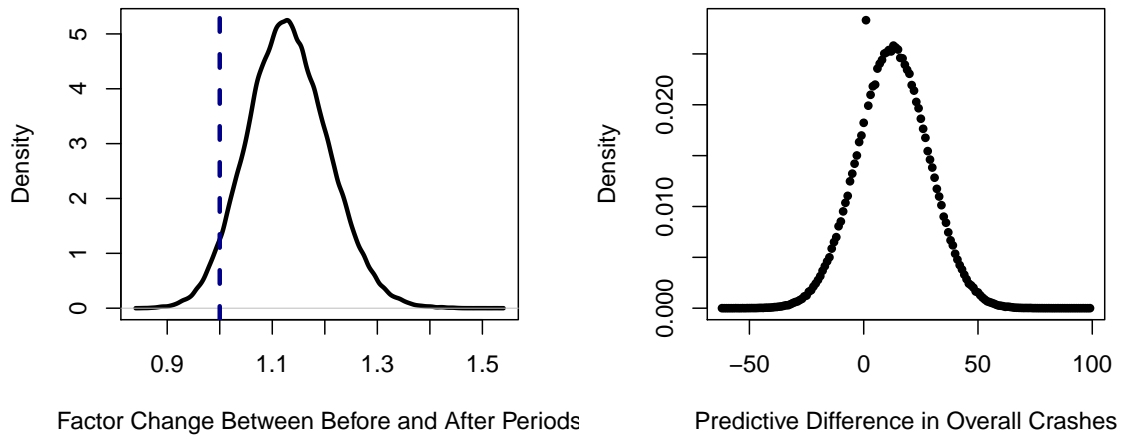


Figure A.34: Overall Crash Results on Raised Medians for Site 1.

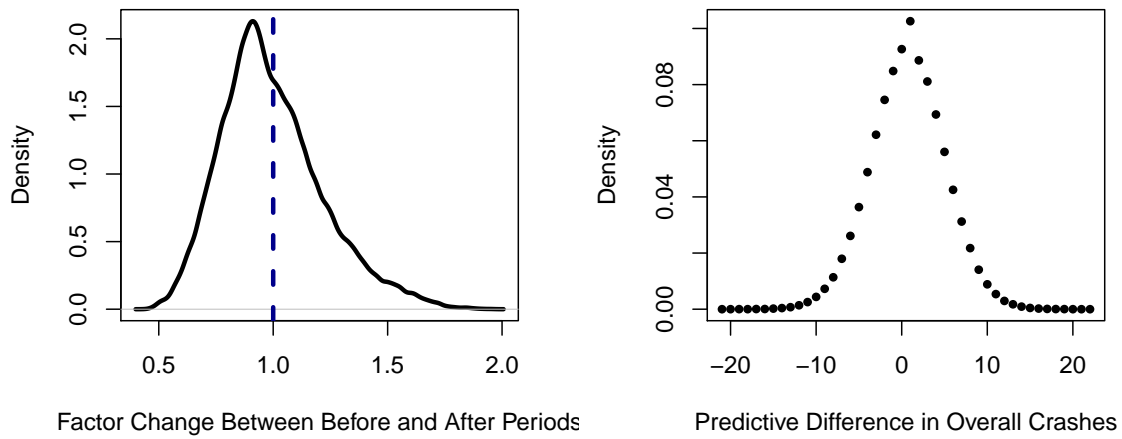


Figure A.35: Overall Crash Results on Raised Medians for Site 2.

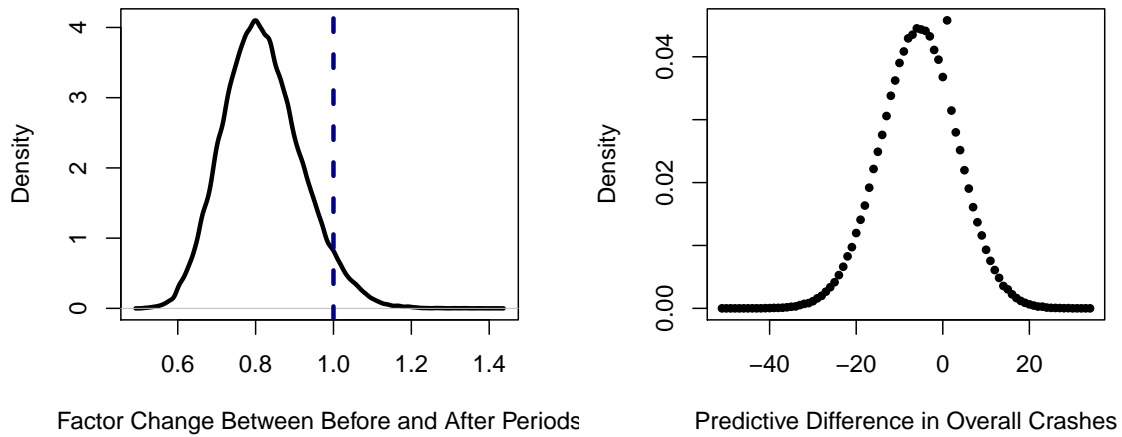


Figure A.36: Overall Crash Results on Raised Medians for Site 3.

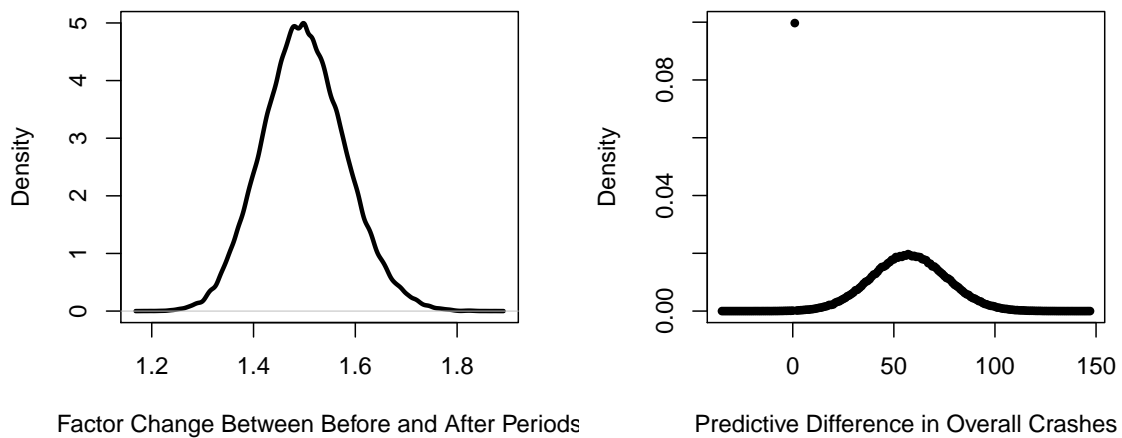


Figure A.37: Overall Crash Results on Raised Medians for Site 4.

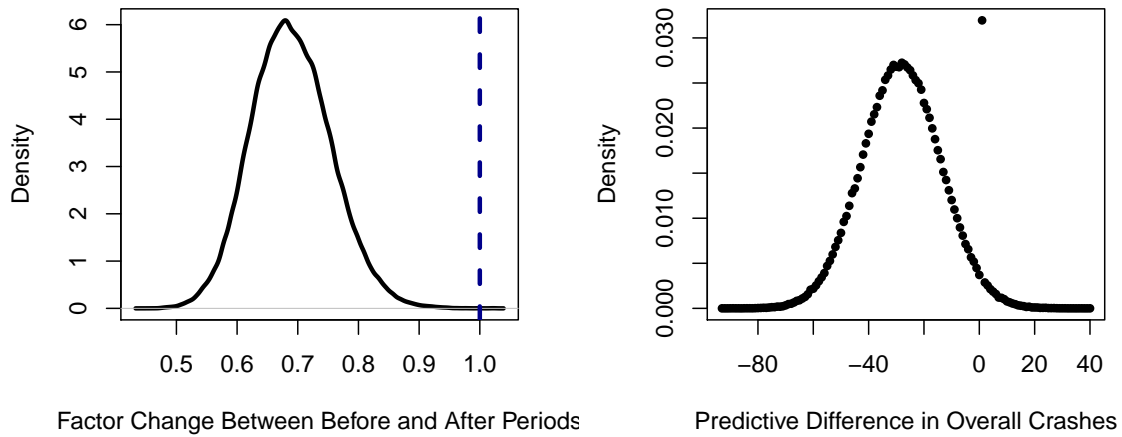


Figure A.38: Overall Crash Results on Raised Medians for Site 5.

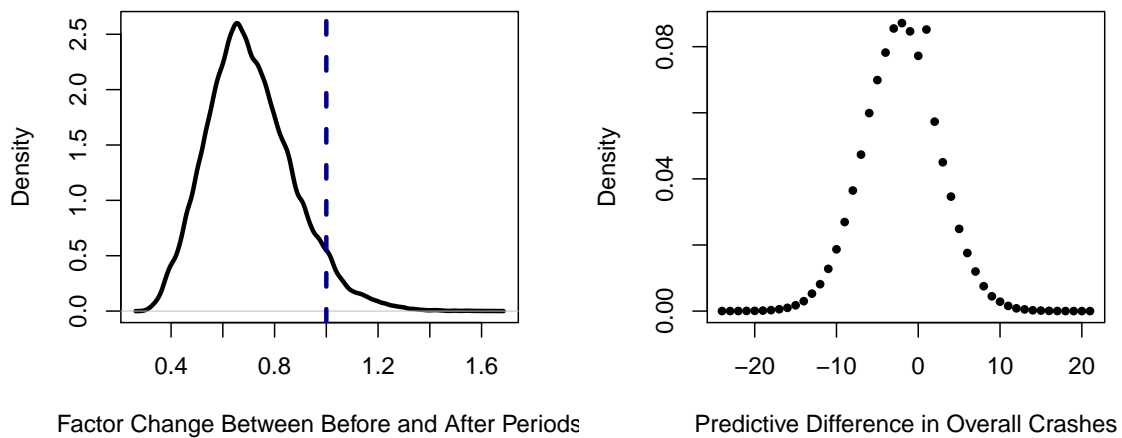


Figure A.39: Overall Crash Results on Raised Medians for Site 6.



## TRADITIONAL METHOD RESULTS

The following results were obtained using the traditional method discussed in Section 2.1. Results are presented for both raised medians and cable barriers concerning all three crash types. Improvement, which is indicated in the last column of each of the tables, was determined by whether or not the confidence intervals, which are created by

$$N_{expected} \pm 2 * \hat{\sigma}, \quad (\text{B. 1})$$

where  $\sigma$  is estimated by Hauer et al. (2002) as

$$\hat{\sigma} = \sqrt{(1 - w) \times N_{expected}}, \quad (\text{B. 2})$$

for the safety project sites and the reference sites overlapped. Using this method, the only improvement occurred on overall crashes for cable barriers.

## B.1 CABLE BARRIERS

Table B.1: Results of Traditional Analysis for Severe Crashes on Cable Barriers

		Project Site					Reference Site					Imp.
Site	Year	$w$	$N_{exp}$	$N_{obs}$	LB	UP	$w$	$N_{exp}$	$N_{obs}$	LB	UB	
1	2005	0.72	2.57	3	0.88	4.27	0.60	4.85	6	2.08	7.63	No
1	2006	0.71	3.57	6	1.53	5.62	0.58	3.42	2	1.03	5.80	No
1	2007	0.69	2.54	2	0.77	4.30	0.56	6.23	8	2.92	9.53	No
1	2008	0.73	2.26	2	0.68	3.83	0.61	4.76	6	2.04	7.49	No
2	2006	0.96	0.26	0	0.05	0.46	0.96	0.25	0	0.05	0.46	No
2	2007	0.96	0.26	0	0.05	0.47	0.96	0.30	1	0.08	0.52	No
2	2008	0.96	0.26	0	0.05	0.46	0.96	0.26	0	0.05	0.46	No
3	2006	0.67	4.01	6	1.72	6.31	0.54	6.04	7	2.72	9.36	No
3	2007	0.66	3.78	5	1.52	6.03	0.56	4.04	3	1.38	6.70	No
3	2008	0.68	3.32	4	1.24	5.40	0.58	3.90	3	1.33	6.47	No
4	2007	0.57	6.16	8	2.89	9.42	0.48	6.85	7	3.08	10.61	No
4	2008	0.61	2.41	0	0.48	4.33	0.54	4.74	4	1.78	7.71	No
5	2008	0.61	2.85	1	0.73	4.96	0.45	4.00	1	1.02	6.98	No
6	2008	0.89	0.68	0	0.13	1.22	0.91	0.59	0	0.12	1.06	No
7	2008	0.70	2.46	2	0.74	4.18	0.74	2.39	3	0.82	3.97	No

Table B.2: Results of Traditional Analysis for Cross-Median Crashes on Cable Barriers

		Project Site					Reference Site					
Site	Year	$w$	$N_{exp}$	$N_{obs}$	LB	UP	$w$	$N_{exp}$	$N_{obs}$	LB	UB	Imp.
1	2005	0.57	2.20	4	0.26	4.14	0.53	2.90	5	0.56	5.23	No
1	2006	0.57	0.94	1	-0.34	2.21	0.52	0.56	0	-0.47	1.59	No
1	2007	0.56	0.51	0	-0.43	1.46	0.52	0.57	0	-0.48	1.61	No
1	2008	0.58	0.50	0	-0.42	1.41	0.53	1.01	1	-0.36	2.39	No
2	2006	0.74	0.31	0	-0.26	0.88	0.74	0.31	0	-0.26	0.87	No
2	2007	0.74	0.57	1	-0.21	1.35	0.74	0.31	0	-0.26	0.88	No
2	2008	0.74	0.57	1	-0.20	1.34	0.74	0.57	1	-0.20	1.34	No
3	2006	0.41	0.69	0	-0.58	1.96	0.37	0.74	0	-0.63	2.10	No
3	2007	0.41	2.46	3	0.05	4.87	0.38	1.98	2	-0.24	4.20	No
3	2008	0.41	1.86	2	-0.23	3.95	0.38	1.34	1	-0.48	3.17	No
4	2007	0.38	3.22	4	0.39	6.06	0.35	2.71	3	0.06	5.36	No
4	2008	0.39	0.71	0	-0.60	2.02	0.37	3.90	5	0.76	7.05	No
5	2008	0.46	1.17	1	-0.42	2.76	0.41	0.69	0	-0.59	1.98	No
6	2008	0.73	0.32	0	-0.27	0.92	0.74	0.31	0	-0.26	0.88	No
7	2008	0.53	0.55	0	-0.47	1.56	0.55	0.53	0	-0.45	1.51	No

Table B.3: Results of Traditional Analysis for Overall Crashes on Cable Barriers

		Project Site					Reference Site					
Site	Year	$w$	$N_{exp}$	$N_{obs}$	LB	UP	$w$	$N_{exp}$	$N_{obs}$	LB	UB	Imp.
1	2005	0.13	82.23	82	65.31	99.15	0.09	123.15	123	101.99	144.31	Yes
1	2006	0.12	70.48	68	54.76	86.19	0.09	111.90	110	91.68	132.12	Yes
1	2007	0.12	86.84	86	69.34	104.34	0.08	155.73	157	131.80	179.65	Yes
1	2008	0.13	80.30	80	63.60	97.00	0.09	156.46	160	132.63	180.28	Yes
2	2006	0.43	14.40	13	8.70	20.11	0.44	14.91	14	9.11	20.70	No
2	2007	0.43	12.78	10	7.39	18.16	0.43	10.46	6	5.59	15.33	No
2	2008	0.44	23.99	30	16.63	31.35	0.44	17.21	18	10.97	23.44	No
3	2006	0.08	139.12	139	116.51	161.72	0.06	110.92	105	90.45	131.38	No
3	2007	0.08	187.29	191	161.03	213.55	0.06	126.55	122	104.73	148.38	No
3	2008	0.08	204.13	210	176.76	231.50	0.06	126.21	122	104.44	147.98	No
4	2007	0.06	206.36	207	178.50	234.21	0.05	245.39	245	214.80	275.97	No
4	2008	0.07	178.38	179	152.60	204.16	0.05	250.94	253	220.14	281.74	Yes
5	2008	0.08	153.42	154	129.64	177.20	0.05	189.53	187	162.69	216.37	No
6	2008	0.30	20.07	16	12.55	27.60	0.32	31.57	34	22.31	40.84	No
7	2008	0.11	145.93	152	123.18	168.69	0.13	109.37	113	89.84	128.89	No

## B.2 RAISED MEDIANS

Table B.4: Results of Traditional Analysis for Severe Crashes on Raised Medians

		Project Site					Reference Site					Imp.
Site	Year	$w$	$N_{exp}$	$N_{obs}$	LB	UP	$w$	$N_{exp}$	$N_{obs}$	LB	UB	
1	2003	0.40	2.60	3	0.09	5.10	0.39	3.25	4	0.43	6.08	No
1	2004	0.38	3.89	5	0.79	7.00	0.38	5.18	7	1.58	8.77	No
1	2005	0.38	3.27	4	0.43	6.12	0.38	3.30	4	0.43	6.17	No
1	2006	0.35	2.13	2	-0.22	4.47	0.38	0.80	0	-0.60	2.20	No
1	2007	0.33	2.22	2	-0.23	4.66	0.32	2.24	2	-0.23	4.71	No
1	2008	0.34	1.52	1	-0.49	3.53	0.33	0.87	0	-0.66	2.39	No
2	2003	0.41	1.35	1	-0.43	3.13	0.48	0.67	0	-0.51	1.84	No
2	2004	0.41	1.35	1	-0.43	3.13	0.48	1.19	1	-0.38	2.76	No
2	2005	0.41	2.52	3	0.09	4.96	0.46	2.30	3	0.08	4.52	No
2	2006	0.41	0.76	0	-0.58	2.11	0.46	0.70	0	-0.53	1.92	No
2	2007	0.41	0.77	0	-0.58	2.11	0.46	1.24	1	-0.40	2.87	No
2	2008	0.41	0.76	0	-0.58	2.10	0.46	0.69	0	-0.52	1.91	No
3	2002	0.35	5.42	7	1.66	9.18	0.31	4.32	5	0.88	7.76	No
3	2003	0.35	2.16	2	-0.22	4.53	0.32	3.62	4	0.47	6.77	No
3	2004	0.35	5.42	7	1.66	9.18	0.34	6.78	9	2.55	11.00	No
3	2005	0.35	2.15	2	-0.22	4.51	0.34	6.12	8	2.10	10.13	No
3	2006	0.27	1.69	1	-0.54	3.91	0.27	3.14	3	0.11	6.18	No
3	2007	0.26	1.70	1	-0.54	3.94	0.25	4.70	5	0.95	8.45	No
3	2008	0.26	1.69	1	-0.54	3.93	0.25	1.71	1	-0.55	3.97	No
4	2005	0.44	4.66	7	1.42	7.89	0.44	1.85	2	-0.19	3.89	No
4	2006	0.43	1.31	1	-0.42	3.03	0.44	0.73	0	-0.55	2.02	No
4	2007	0.43	3.60	5	0.73	6.47	0.43	2.43	3	0.09	4.78	No
4	2008	0.44	0.73	0	-0.55	2.01	0.44	1.29	1	-0.41	2.98	No
5	2007	0.35	2.13	2	-0.22	4.48	0.41	0.76	0	-0.58	2.10	No
5	2008	0.36	0.83	0	-0.63	2.28	0.42	0.75	0	-0.56	2.06	No
6	2006	0.36	1.47	1	-0.47	3.41	0.36	1.47	1	-0.47	3.41	No
6	2007	0.35	0.84	0	-0.64	2.31	0.35	0.84	0	-0.64	2.31	No
6	2008	0.40	0.78	0	-0.59	2.14	0.40	1.37	1	-0.44	3.19	No

Table B.5: Results of Traditional Analysis for Cross-Median Crashes on Raised Medians

		Project Site					Reference Site					Imp.
Site	Year	$w$	$N_{exp}$	$N_{obs}$	LB	UP	$w$	$N_{exp}$	$N_{obs}$	LB	UB	
1	2003	0.55	0.17	0	-0.39	0.72	0.53	0.17	0	-0.40	0.75	No
1	2004	0.52	0.18	0	-0.40	0.76	0.51	0.18	0	-0.41	0.77	No
1	2005	0.52	0.18	0	-0.40	0.76	0.51	0.18	0	-0.41	0.77	No
1	2006	0.48	0.19	0	-0.44	0.83	0.52	0.18	0	-0.40	0.75	No
1	2007	0.43	1.35	2	-0.40	3.10	0.42	0.22	0	-0.49	0.92	No
1	2008	0.45	0.76	1	-0.54	2.05	0.44	0.21	0	-0.48	0.90	No
2	2003	0.71	0.11	0	-0.25	0.46	0.74	0.10	0	-0.22	0.41	No
2	2004	0.71	0.11	0	-0.25	0.46	0.74	0.10	0	-0.22	0.41	No
2	2005	0.71	0.11	0	-0.25	0.46	0.72	0.10	0	-0.24	0.45	No
2	2006	0.71	0.40	1	-0.28	1.09	0.72	0.11	0	-0.24	0.45	No
2	2007	0.70	0.11	0	-0.25	0.47	0.71	0.11	0	-0.24	0.45	No
2	2008	0.71	0.11	0	-0.25	0.46	0.72	0.10	0	-0.24	0.45	No
3	2002	0.63	0.14	0	-0.31	0.58	0.58	0.16	0	-0.36	0.67	No
3	2003	0.63	0.14	0	-0.31	0.59	0.58	0.15	0	-0.35	0.66	No
3	2004	0.63	0.14	0	-0.31	0.58	0.62	0.14	0	-0.32	0.60	No
3	2005	0.64	0.14	0	-0.31	0.58	0.62	0.14	0	-0.32	0.60	No
3	2006	0.49	0.70	1	-0.50	1.90	0.49	0.19	0	-0.43	0.81	No
3	2007	0.48	0.72	1	-0.51	1.94	0.46	0.20	0	-0.46	0.86	No
3	2008	0.48	0.19	0	-0.44	0.82	0.46	0.20	0	-0.45	0.85	No
4	2005	0.63	0.14	0	-0.31	0.59	0.63	0.14	0	-0.31	0.59	No
4	2006	0.62	1.67	4	0.07	3.27	0.63	0.14	0	-0.32	0.59	No
4	2007	0.61	0.14	0	-0.33	0.61	0.62	0.14	0	-0.32	0.60	No
4	2008	0.63	0.14	0	-0.31	0.59	0.63	0.14	0	-0.31	0.59	No
5	2007	0.60	0.94	2	-0.28	2.16	0.53	0.17	0	-0.39	0.74	No
5	2008	0.62	0.53	1	-0.37	1.42	0.55	0.17	0	-0.38	0.71	No
6	2006	0.59	0.15	0	-0.35	0.65	0.59	0.56	1	-0.40	1.53	No
6	2007	0.58	0.58	1	-0.41	1.58	0.58	0.16	0	-0.36	0.67	No
6	2008	0.65	0.13	0	-0.29	0.55	0.65	0.13	0	-0.29	0.55	No

Table B.6: Results of Traditional Analysis for Overall Crashes on Raised Medians

		Project Site					Reference Site					Imp.
Site	Year	$w$	$N_{exp}$	$N_{obs}$	LB	UP	$w$	$N_{exp}$	$N_{obs}$	LB	UB	
1	2003	0.02	78.94	79	61.39	96.49	0.02	60.50	60	45.13	75.87	No
1	2004	0.02	115.26	116	94.03	136.48	0.02	104.59	105	84.36	124.82	No
1	2005	0.02	105.48	106	85.17	125.78	0.02	91.87	92	72.91	110.83	No
1	2006	0.02	29.37	28	18.63	40.10	0.02	17.53	16	9.25	25.81	No
1	2007	0.02	117.00	117	95.54	138.46	0.02	85.59	85	67.23	103.95	No
1	2008	0.02	82.48	82	64.48	100.49	0.02	47.15	46	33.53	60.76	No
2	2003	0.06	13.99	13	6.75	21.22	0.06	9.31	8	3.40	15.21	No
2	2004	0.06	14.92	14	7.45	22.40	0.06	6.51	5	1.57	11.45	No
2	2005	0.06	23.35	23	14.00	32.70	0.06	13.14	12	6.10	20.18	No
2	2006	0.06	12.13	11	5.38	18.87	0.06	18.81	18	10.39	27.24	No
2	2007	0.06	17.76	17	9.60	25.91	0.06	15.05	14	7.51	22.58	No
2	2008	0.06	12.11	11	5.38	18.85	0.06	13.14	12	6.10	20.18	No
3	2002	0.05	69.34	71	53.10	85.58	0.04	80.57	82	62.98	98.16	No
3	2003	0.05	59.87	61	44.78	74.96	0.04	62.29	63	46.83	77.75	No
3	2004	0.05	73.16	75	56.48	89.84	0.05	70.42	72	54.04	86.79	No
3	2005	0.05	54.11	55	39.77	68.45	0.05	62.79	64	47.33	78.25	No
3	2006	0.03	39.76	39	27.33	52.18	0.03	33.91	33	22.44	45.39	No
3	2007	0.03	51.46	51	37.32	65.60	0.03	35.00	34	23.32	46.67	No
3	2008	0.03	70.87	71	54.28	87.47	0.03	38.88	38	26.57	51.18	No
4	2005	0.04	168.77	173	143.25	194.29	0.04	39.50	39	27.16	51.85	No
4	2006	0.03	153.58	157	129.22	177.94	0.03	47.24	47	33.74	60.75	No
4	2007	0.03	163.32	167	138.19	188.45	0.03	62.72	63	47.15	78.28	No
4	2008	0.04	121.52	124	99.87	143.18	0.04	52.99	53	38.69	67.29	No
5	2007	0.04	114.13	117	93.20	135.06	0.02	66.46	66	50.34	82.59	No
5	2008	0.04	104.32	107	84.33	124.30	0.02	37.06	36	25.03	49.10	No
6	2006	0.04	10.55	9	4.17	16.93	0.04	21.15	20	12.12	30.18	No
6	2007	0.03	16.36	15	8.41	24.31	0.03	27.95	27	17.56	38.34	No
6	2008	0.05	11.39	10	4.80	17.99	0.05	7.58	6	2.20	12.96	No

## COMPLETE CONDITIONAL DISTRIBUTIONS

This appendix presents the complete conditional distributions required for Gibbs sampling in this project. Note that for this project and for all of the derivations of the complete conditional distributions presented here, prior covariances of multivariate normal distributions are set to be zero. This appears to be a reasonable decision based on our lack of detailed prior information.

### C.1 MODEL 1

This model does not require complete conditional distributions because the entire posterior distribution can be sampled from using the Metropolis-Hastings algorithm. The posterior distribution takes the form

$$\pi(\boldsymbol{\beta}|\mathbf{y}) \propto \exp \left[ \sum_{i=1}^n \exp(\mathbf{X}_i \boldsymbol{\beta}) \right] \left[ \prod_{i=1}^n \exp(y_i \mathbf{X}_i \boldsymbol{\beta}) \right] \exp \left[ -\frac{1}{2} \sum_{j=1}^p \frac{(\beta_j - \mu_j)^2}{\sigma_j^2} \right] \quad (\text{C. 1})$$

where the  $\mathbf{X}$  and  $\boldsymbol{\beta}$  matrices are defined appropriately and the  $\boldsymbol{\mu}$  and  $\boldsymbol{\sigma}$  parameters are prior constants.

### C.2 MODEL 2

The complete conditional distributions for this model are defined below.

$$\begin{aligned} [\beta_{jk}] &\propto \exp \left( -\sum_{i=1}^n \exp(\mathbf{X}_i \boldsymbol{\beta}) \right) \times \prod_{i=1}^n \exp(y_i \mathbf{X}_i \boldsymbol{\beta}) \times \\ &\quad \exp(-[2\sigma_k^2]^{-1}(\beta_{jk} - \mu_{\beta_k})^2) \\ [\mu_{\beta_j}] &\sim N \left( \frac{\sigma_{\mu_{\beta_k}}^2 \sum_{k=1}^K \beta_{jk} + \sigma_{\beta_k}^2 \mu_{\mu_{\beta_k}}}{K\sigma_{\mu_{\beta_k}}^2 + \sigma_{\beta_k}^2}, \frac{\sigma_{\mu_{\beta_k}}^2 \sigma_{\beta_k}^2}{K\sigma_{\mu_{\beta_k}}^2 + \sigma_{\beta_k}^2} \right) \end{aligned}$$

$$[\sigma_{\beta_j}^2] \sim IG \left( a_k + \frac{K}{2}, \left[ \frac{1}{2} \sum_{k=1}^K (\beta_{jk} - \mu_{\beta_k})^2 + \frac{1}{b_k} \right]^{-1} \right)$$

### C.3 MODEL 3

The complete conditional distributions for this model are presented in the equations that follow.

$$\begin{aligned} [\beta_{kj}] &\propto \exp \left( - \sum_{i=1}^n \exp(\mathbf{X}_i \boldsymbol{\beta}) \right) \times \prod_{i=1}^n \exp(y_i \mathbf{X}_i \boldsymbol{\beta}) \times \exp(-[2\sigma_k^2]^{-1}(\beta_{jk} - \mu_k)^2) \\ [\mu_k] &\sim Normal \left( \frac{S^2 \sum_{j=1}^J \beta_{kj} + \sigma_k^2 m}{S^2 J + \sigma_k^2}, \frac{\sigma_k^2 + S^2}{S^2 J + \sigma_k^2} \right) \\ [\sigma_k^2] &\sim IG \left( a_k + J/2, \left[ \frac{1}{2} \sum_{j=1}^J (\beta_{kj} - \mu_k)^2 + \frac{1}{b_k} \right]^{-1} \right) \\ [\eta_t] &\propto \prod_{i=1}^N Poisson_{pmf}(\theta_i \eta_t) Gamma_{pdf}(\eta_t, \phi_t, 1/\phi_y) \\ [\phi_t] &\propto Gamma_{pdf}(\eta_t, \phi_t, 1/\phi_t) Gamma_{pdf}(\phi_t, a_\phi, b_\phi) \end{aligned}$$

### C.4 MODEL 4

The complete conditional distributions for the dynamic linear model are shown below.

$$\begin{aligned} [\theta_{1k}] &\propto \prod_{i:C_i=k} \left[ \pi_{1k}(I(y_{i1} = 0)) + (1 - \pi_{1k}) \frac{\exp(-\exp(\theta_{1k})) \exp(\theta_{1k} y_{i1})}{y_{i1}!} \right] \times \\ &\quad \exp \left( -\frac{1}{2\tau_k^2} (\theta_{1k} - m_1)^2 \right) \times \exp \left( -\frac{1}{2\tau_k^2} (\theta_{2k} - \theta_{1k})^2 \right) \\ &\propto \prod_{i:C_i=k} \left[ \pi_{1k}(I(y_{i1} = 0)) + (1 - \pi_{1k}) \frac{\exp(-\exp(\theta_{1k})) \exp(\theta_{1k} y_{i1})}{y_{i1}!} \right] \times \\ &\quad \exp \left( -\frac{1}{\tau_k^2} [\theta_{1k}(\theta_{1k} - m_1 - \theta_{2k})] \right) \\ [\theta_{jk}]^* &\propto \prod_{i:C_i=k} \left[ \pi_{jk}(I(y_{ij} = 0)) + (1 - \pi_{jk}) \frac{\exp(-\exp(\theta_{jk})) \exp(\theta_{jk} y_{ij})}{y_{ij}!} \right] \times \\ &\quad \exp \left( -\frac{1}{2\tau_k^2} (\theta_{jk} - \theta_{j-1,k})^2 \right) \times \exp \left( -\frac{1}{2\tau_k^2} (\theta_{j+1,k} - \theta_{jk})^2 \right) \end{aligned}$$

$$\begin{aligned}
& \propto \prod_{i:C_i=k} \left[ \pi_{jk}(I(y_{ij}=0)) + (1-\pi_{jk}) \frac{\exp(-\exp(\theta_{jk})) \exp(\theta_{jk}y_{ij})}{y_{ij}!} \right] \times \\
& \quad \exp \left( -\frac{1}{\tau_k^2} [\theta_{jk}(\theta_{jk} - \theta_{j-1,k} - \theta_{j+1,k})] \right) \\
& \quad \text{*for } j = 2, \dots, J-1 \\
[\theta_{Jk}] & \propto \prod_{i:C_i=k} \left[ \pi_{Jk}(I(y_{iJ}=0)) + (1-\pi_{Jk}) \frac{\exp(-\exp(\theta_{Jk})) \exp(\theta_{Jk}y_{iJ})}{y_{iJ}!} \right] \times \\
& \quad \exp \left( -\frac{1}{2\tau_k^2} (\theta_{Jk} - \theta_{J-1,k})^2 \right) \\
& \propto \prod_{i:C_i=k} \left[ \pi_{Jk}(I(y_{iJ}=0)) + (1-\pi_{Jk}) \frac{\exp(-\exp(\theta_{Jk})) \exp(\theta_{Jk}y_{iJ})}{y_{iJ}!} \right] \times \\
& \quad \exp \left( -\frac{1}{2\tau_k^2} [\theta_{Jk}^2 - 2\theta_{Jk}\theta_{J-1,k}] \right) \\
[\pi_{jk}] & \propto \prod_{i:C_i=k} \left[ \pi_{Jk}(I(y_{ij}=0)) + (1-\pi_{jk}) \frac{\exp(-\exp(\theta_{jk})) \exp(\theta_{jk}y_{ij})}{y_{ij}!} \right] \pi_{jk}^{\alpha-1} (1-\pi_{jk})^{\beta-1} \\
[\alpha] & \propto \prod_{j=1}^J \prod_{k=1}^K \left[ \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)} \pi_{jk}^{\alpha-1} \right] \alpha^{a_\alpha-1} \exp \left( \frac{\alpha}{b_\alpha} \right) \\
& \propto \left[ \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)} \right]^{JK} \exp \left( -\alpha \left( -\sum_{j=1}^J \sum_{k=1}^k \log(\pi_{jk}) + \frac{1}{b_\alpha} \right) \right) \alpha^{a_\alpha-1} \\
[\beta] & \propto \prod_{j=1}^J \prod_{k=1}^K \left[ \frac{\Gamma(\alpha+\beta)}{\Gamma(\beta)} (1-\pi_{jk})^{\beta-1} \right] \exp \left( -\frac{\beta}{c} \right) \\
& \propto \left[ \frac{\Gamma(\alpha+\beta)}{\Gamma(\beta)} \right]^{JK} \exp \left( -\beta \left( -\sum_{j=1}^J \sum_{k=1}^k \log(1-\pi_{jk}) + \frac{1}{c} \right) \right) \\
[\tau_k^2] & \propto (\tau_k^2)^{-1/2} \exp \left( -\frac{1}{2\tau_k^2} (\theta_{1k} - m_1)^2 \right) \prod_{j=2}^J (\tau_k^2)^{-1/2} \exp \left( -\frac{1}{2\tau_k^2} (\theta_{jk} - \theta_{j-1,k})^2 \right) \times \\
& \quad (\tau_k^2)^{-(a_\tau+1)} \exp \left( -\frac{1}{\tau_k^2 b_\tau} \right) \\
& \Rightarrow \tau_k^2 \sim IG \left( a_\tau + \frac{J}{2}, \left[ \frac{1}{2} (\theta_{1k} - m_1)^2 + \frac{1}{2} \sum_{j=2}^J (\theta_{jk} - \theta_{j-1,k})^2 + \frac{1}{b_\tau} \right]^{-1} \right)
\end{aligned}$$



---

 CODE

This Appendix presents the code utilized for each model in this analysis. Code for models 1 through 3 is written in C. Code for model 4 is written in R.

## D.1 MODEL 1

*C Code Used for Sampler*

## Main File

```
// Metropolis sampler with random walk.

#include <stdio.h>
#include <gsl/gsl_rng.h>          // required for random number generation
#include <gsl/gsl_randist.h>      // required for data generation from distributions
#include <math.h>
#include <stdlib.h>
#include "gsl_linalg.h"

// Define pi and the random digit constant

# define pi 3.141593
gsl_rng * r;

// Include specific header file for this analysis

#include "mcmcheader.txt"

// Include functions needed for the analysis

#include "poisll.h"
#include "mvnprior.h"
#include "mvnran.h"
#include "rgen2dar.h"
#include "cov_chol.h"

// Begin Main Function

int main(void){

    int ntune=1000;

// Define integers

    int i;
    int j;
    int ncol = nparam * nsite;
    int k;
    int y;
// Source the data

#include "data.txt"
```

```

#include "file.txt"

resetgen(651);

// Create the matrices that will hold the estimates.

double beta[ncol];

// Set the initial values.

for(i=0;i<nparam;i++){

    for(j=0;j<nsite;j++){
        beta[i*nsite + j] = prmu[i] ;
    }
}

double betastar[ncol];

double acceptmulti=0;

// Obtain the cholesky decomposition of the candidate covariance matrix.

double candvct[ncol][ncol];

int i2; int j2; int i3; int j3;

for(i2=0;i2<ncol;i2++){
    for(j2=0;j2<ncol;j2++){

        candvct[i2][j2] = candvc[i2][j2] * pow(tune,2);

    }
}

double chol[ncol][ncol];
cholesky(candvct, chol);

double u;

// Tune the Gibbs Sampler

int z;
for(z=0; z<15;z++){

    acceptmulti = 0;

    for (i=0; i<ntune; i++) {

        // Obtain a draw from the multivariate normal candidate distribution and determine to
        // accept or reject it.

        mvnran(beta, chol, betastar);
        u = gsl_ran_flat(r,0,1); // Random uniform
        if(log(u) < loglik(betastar,X,Y,prmu,prvar,-1) - loglik(beta,X,Y,prmu,prvar,-1)){
            acceptmulti = acceptmulti + 1;
            for(j=0;j<ncol;j++){
                beta[j] = betastar[j];
            }
        }
    }

    printf("Accept %f with tune = %f\n",acceptmulti/ntune,tune);
}

```

```

    if (z>0){

        if (acceptmulti/ntune<.25){tune = tune*.9 ;}
        if (acceptmulti/ntune>.27){tune = tune*1.1 ;}

        for (i2=0;i2<ncol;i2++){
            for (j2=0;j2<ncol;j2++){

                candvct[i2][j2] = candvc[i2][j2] * pow(tune,2);

            }
        }

        cholesky(candvct, chol);

    }
}

// Print the initial values

for (j=0;j<(ncol-1);j++){
    fprintf(mcmcf,"%f",beta[j]); }
fprintf(mcmcf,"%f",beta[ncol-1]);
fprintf(mcmcf,"\n");

acceptmulti = 0;

// Begin the Gibbs Sampler

for (i=1;i<(nsims);i++){

    // Obtain a draw from the multivariate normal candidate distribution and determine to
    // accept or reject it.

    mvnran(beta, chol, betastar);
    u = gsl_ran_flat(r,0,1); // Random uniform
    if (log(u) < loglik(betastar,X,Y,prmu,prvar,-1) - loglik(beta,X,Y,prmu,prvar,-1)){
        acceptmulti = acceptmulti + 1;
        for (j=0;j<ncol;j++){
            beta[j] = betastar[j];
        }
    }

    // Print all of the parameter values for this iteration

    for (j=0;j<(ncol-1);j++){
        fprintf(mcmcf,"%-20.20f",beta[j]); }
    fprintf(mcmcf,"%-20.20f",beta[ncol-1]);
    fprintf(mcmcf,"\n");
    // Begin the next iteration

}

// Print the acceptance rate to the terminal.

printf("Accept %f\n",acceptmulti/nsims);

```

```

    fclose(mcmcf);

    return;

}

```

## Header Files

```

FILE *mcmcf; mcmcf = fopen("beta_rm_c.txt","w");

// This is the Poisson Log Likelihood function

double loglik(double *p, double x[n][nparam*nsite], double *y, double *prmu, double *prvar,
    int condi){

    // The theta vector represents b0 + b1*x1 + b2*x2 . . .
    double theta[n];

    // Initialize loop incrementors

    int i2;
    int j2;

    // Initialize the two components of this likelihood

    double sumtheta=0;
    double ysumtheta=0;

    for(j2=0;j2<n;j2++){
        // Calculate theta for each observation
        theta[j2] = 0;
        for(i2=0;i2<nparam*nsite;i2++){
            theta[j2] = theta[j2] + x[j2][i2] * p[i2];
        }
        // Add to each of the components as needed
        sumtheta =sumtheta + exp(theta[j2]);
        //printf("Here it is %f\n",exp(theta[j2]));
        ysumtheta =ysumtheta + y[j2] * theta[j2];
    }

    if(condi>=0){
        int tempi = floor(condi/nsite);
        double normalpart = - 1/(2 * prvar[tempi]) * pow(p[condi] - prmu[tempi], 2 );

        // Return the actual likelihood value
        //printf("The sumtheta is %f\n",sumtheta);
        //printf("The ysumtheta is %f\n",ysumtheta);
        return( -sumtheta + ysumtheta + normalpart ) ;

    } else {

        int tempi;
        double normalpart = 0;
        for(i2=0;i2<nparam*nsite;i2++){
            tempi = floor(i2/nsite);
            normalpart= normalpart - 1/(2 * prvar[tempi]) * pow(p[i2] - prmu[tempi], 2 );
        }

        return( -sumtheta + ysumtheta + normalpart ) ;

    }
}

```

```

}

void mvnran(double *means, double chol[nparam*nsite][nparam*nsite], double *mvn){

int i2; int k2; int l2;
int i; int j;

double norms[nparam*nsite];

for(i2=0;i2<nparam*nsite;i2++){
norms[i2] = gsl_ran_gaussian(r,1);
//printf("Normal %d: %f ",i,norms[i]);
}
//printf("\n");

int niter = 0;

for(k2=0;k2<nparam*nsite;k2++){
mvn[k2] = 0;
for(l2=0;l2<nparam*nsite;l2++){
mvn[k2] = mvn[k2] + norms[l2] * chol[l2][k2];
// printf("Chol %d: %f ",l,chol[k][l]);
}
// printf("MVN %d: %f ",k,mvn[k]);
mvn[k2] = mvn[k2] + means[k2];
}

//for (i2 = 0; i2 < nparam; i2++) {
//    printf ("mvn %f\n", mvn[i2]);
//}

return;
}

void resetgen(int seed){

// ##### set up random number generator ##### /
const gsl_rng_type * TT;
gsl_rng_env_setup();
TT = gsl_rng_default;
r = gsl_rng_alloc (TT);
gsl_rng_set(r,seed); // set seed based on simulation number
gsl_rng_free (r); // function frees all the memory associated with the generator r
// ##### /

return;
}

double** Make2DDoubleArray(int arraySizeX, int arraySizeY) {
double** theArray;
theArray = (double**) malloc(arraySizeX*sizeof(double*));
int i;
for (i = 0; i < arraySizeX; i++)
theArray[i] = (double*) malloc(arraySizeY*sizeof(double));
return theArray;
}

/*
* cov_chol.c
*
*
* Created by Andrew Olsen on 10/16/10.
*/

```

```

*   Copyright 2010 Brigham Young University. All rights reserved.
*
*/

// Create the covariance matrix
/*
void cov(double** beta, double* EX, double sigma[nsite*nparam][nsite*nparam]){
    int ncol = nsite * nparam;
    int j;
    int i;
    int k;
//double EX[ncol];
double EXS[ncol][ncol];

for (j=0;j<ncol;j++){
    EX[j] = 0;
    for (i=(nburn/2);i<nburn;i++){
        EX[j] = EX[j] + beta[i][j];
    }
    EX[j] = EX[j]/(nburn/2);
}

for (j=0;j<ncol;j++){
    for (k=j;k<ncol;k++){
        EXS[j][k] = 0;
        for (i=(nburn/2);i<nburn;i++){
            EXS[j][k] = EXS[j][k] + beta[i][j] * beta[i][k];
        }
        EXS[j][k] = EXS[j][k];
    }
}

//double sigma[ncol][ncol];

for (j=0;j<ncol;j++){
    for (k=j;k<ncol;k++){
        sigma[j][k] = (EXS[j][k] - (nburn/2) * EX[j] * EX[k]) / ((nburn/2) - 1);
        sigma[k][j] = sigma[j][k];
        //printf("Variance Value %f\n",sigma[j][k]);
    }
}

    return;
}
*/

// Create the cholesky decomposition.

void cholesky(double sigma[nsite*nparam][nsite*nparam],double chol[nsite*nparam][nsite*
nparam]){
    int i; int j; int k;
    int ncol=nsite * nparam;

chol[0][0] = sqrt(sigma[0][0]);
for(i=1;i<ncol;i++){
    chol[0][i] = sigma[0][i]/chol[0][0];
}

double tkis;
double tktk;

for(i=1;i<ncol;i++){
    for (j=0;j<ncol;j++){
        if (j<i){chol[i][j]=0;} else{
            if (i!=j){

```

```

        tktk=0;
        for(k=0;k<(i);k++){ tktk = tktk + chol[k][i] * chol[k][j] ;}
        chol[i][j] = (sigma[i][j] - tktk) / chol[i][i];
    } else {
        tkis = 0;
        for(k=0;k<(i);k++){ tkis = tkis + pow(chol[k][i],2); }
        chol[i][i] = sqrt((sigma[i][i] - tkis));
    }
}
}
}

for(i=0;i<ncol;i++){
    for(j=0;j<ncol;j++){
        //printf("Cholesky Value %f\n",chol[i][j]);
    }
    //printf("\n");
}

return;
}

```

## D.2 MODEL 2

### *C Code Used for Sampler*

#### Main File

```

// Metropolis sampler with independent candidate density draws.

#include <stdio.h>
#include <gsl/gsl_rng.h>          // required for random number generation
#include <gsl/gsl_randist.h>      // required for data generation from distributions
#include <math.h>
#include <stdlib.h>
#include "gsl_linalg.h"

// Define pi and the random digit constant

# define pi 3.141593
gsl_rng * r;

// Include specific header file for this analysis

#include "mcmcheader.txt"

// Include functions needed for the analysis

#include "poisll.h"
#include "mvnprior.h"
#include "mvnran.h"
#include "rgen2dar.h"
#include "cov_chol.h"

// Begin Main Function

int main(void){

    int ntune=1000;

// Define integers

    int i;

```

```

    int j;
    int ncol = nparam * nsite;
    int k;
    int y;
// Source the data

#include "data.txt"

#include "file.txt"

resetgen(651);

    // Create the matrices that will hold the estimates.

    double beta[ncol];
    double mu[nparam];
    double sig[nparam];

    // Set the initial values.

    for(i=0;i<nparam;i++){

mu[i] = prmu[i];
sig[i] = 1/(pra[i] * prb[i]);

        for(j=0;j<nsite;j++){
            beta[i*nsite + j] = prmu[i] ;
        }
    }

    double betastar[ncol];

    double sumbk[nparam];
    double sumbkmms[nparam];
    double condmu;
    double condvar;
    double conda;
    double condb;

    double acceptmulti=0;

    // Obtain the cholesky decomposition of the candidate covariance matrix.

    double candvct[ncol][ncol];

    int i2; int j2; int i3; int j3;

    for(i2=0;i2<ncol;i2++){
        for(j2=0;j2<ncol;j2++){

            candvct[i2][j2] = candvc[i2][j2] * pow(tune,2);

        }
    }

    double chol[ncol][ncol];
    cholesky(candvct, chol);

    double u;

    // Tune the Gibbs Sampler

    int z;
    for(z=0; z<15;z++){

```

```

acceptmulti = 0;

for (i=0; i<ntune; i++) {

// Obtain a draw from the multivariate normal candidate distribution and determine to
// accept or reject it.

mvnran(beta, chol, betastar);
u = gsl_ran_flat(r, 0, 1); // Random uniform
// printf("ll %f", loglik(beta, X, Y, mu, sig, -1));
if (log(u) < loglik(betastar, X, Y, mu, sig, -1) - loglik(beta, X, Y, mu, sig, -1)) {
    acceptmulti = acceptmulti + 1;
    for (j=0; j<ncol; j++) {
        beta[j] = betastar[j];
    }
}

// Now sample from the other complete conditional distributions.

// Sum the betas
for (k=0; k<nparam; k++) {
    sumbk[k] = 0;
    for (j=0; j<nsite; j++) {
        sumbk[k] = sumbk[k] + beta[k*j + j];
    }
}

// Get the draws for the mu_k parameters.
for (k=0; k<nparam; k++) {
    condmu = (prvar[k] * sumbk[k] + sig[k] * prmu[k]) / (nsite * prvar[k] + sig[k]);
    condvar = (prvar[k] * sig[k]) / (nsite * prvar[k] + sig[k]);
    mu[k] = gsl_ran_gaussian(r, sqrt(condvar)) + condmu;
    // printf("mu %f\n", mu[k]);
}

// Sum the betas minus the mus squared
for (k=0; k<nparam; k++) {
    sumbkmms[k] = 0;
    for (j=0; j<nsite; j++) {
        sumbkmms[k] = sumbkmms[k] + pow(beta[k*j+1] - mu[k], 2);
    }
}

// Get the draws for the sig_k parameters.
for (k=0; k<nparam; k++) {
    conda = pra[k] + nsite/2;
    condb = 1 / (.5 * sumbkmms[k] + (1/prb[k]));
    sig[k] = 1/gsl_ran_gamma(r, conda, condb);
    // printf("sig %f\n", prb[k]);
}

}

printf("Accept %f with tune = %f\n", acceptmulti/ntune, tune);

if (z>0) {

```

```

        if (acceptmulti/ntune<.23){tune = tune*.9  ;}
        if (acceptmulti/ntune>.30){tune = tune*1.1  ;}

        for (i2=0;i2<ncol;i2++){
            for (j2=0;j2<ncol;j2++){

                candvct[i2][j2] = candvc[i2][j2] * pow(tune,2);

            }
        }

        cholesky(candvct, chol);

    }
}

// Print the initial values

for (j=0;j<(ncol);j++){
    fprintf(mcmcf,"%f",beta[j]); }

for (k=0;k<nparam;k++){
    fprintf(mcmcf,"%f,%f",mu[k],sig[k]);
    if (k<(nparam-1)){fprintf(mcmcf,",");} else {fprintf(mcmcf,"\n");}
}

acceptmulti = 0;

// Begin the Gibbs Sampler

for (i=1;i<(nsims);i++){

    // Obtain a draw from the multivariate normal candidate distribution and determine to
    // accept or reject it.

    mvnran(beta, chol, betastar);
    u = gsl_ran_flat(r,0,1); // Random uniform
    if (log(u) < loglik(betastar,X,Y,mu,sig,-1) - loglik(beta,X,Y,mu,sig,-1)){
        acceptmulti = acceptmulti + 1;
        for (j=0;j<ncol;j++){
            beta[j] = betastar[j];
        }
    }

    // Now sample from the other complete conditional distributions.

    // Sum the betas

    for (k=0;k<nparam;k++){
        sumbk[k] = 0;
        for (j=0;j<nsite;j++){
            sumbk[k] = sumbk[k] + beta[k*j + j];
        }
    }

    // Get the draws for the mu_k parameters.

    for (k=0;k<nparam;k++){

```

```

        condmu = (prvar[k] * sumbk[k] + sig[k] * prmu[k] ) / ( nsite * prvar[k] + sig[k]
        );
        condvar = ( prvar[k] * sig[k] ) / ( nsite * prvar[k] + sig[k] );
        mu[k] = gsl_ran_gaussian(r,sqrt(condvar)) + condmu;
    }

    // Sum the betas minus the mus squared

    for(k=0;k<nparam;k++){
        sumbkmms[k] = 0;
        for(j=0;j<nsite;j++){
            sumbkmms[k] = sumbkmms[k] + pow(beta[k*j+1] - mu[k],2);
        }
    }

    // Get the draws for the sig_k parameters.

    for(k=0;k<nparam;k++){
        conda = pra[k] + nsite/2;
        condb = 1 / ( .5 * sumbkmms[k] + (1/prb[k]) );
        sig[k] = 1/gsl_ran_gamma(r,conda,condb);
    }

    // Print all of the parameter values for this iteration

    for(j=0;j<(ncol);j++){
        fprintf(mcmcf,"%-20.20f ",beta[j]); }

    for(k=0;k<nparam;k++){
        fprintf(mcmcf,"%f,%f",mu[k],sig[k]);
        if(k<(nparam-1)){fprintf(mcmcf,",");} else {fprintf(mcmcf,"\n");}
    }

    // Begin the next iteration
}

// Print the acceptance rate to the terminal.

printf("Accept %f\n",acceptmulti/nsims);

fclose(mcmcf);

return;
}

```

## Header Files

```

FILE *mcmcf; mcmcf = fopen("beta_rm_c.txt","w");

// This is the Poisson Log Likelihood function

double loglik(double *p, double x[n][nparam*nsite], double *y, double *prmu, double *prvar,
    int condi){

    // The theta vector represents b0 + b1*x1 + b2*x2 . . .
    double theta[n];

    // Initialize loop incrementors

    int i2;
    int j2;

```

```

// Initialize the two components of this likelihood

double sumtheta=0;
double ysumtheta=0;

for(j2=0;j2<n;j2++){
    // Calculate theta for each observation
    theta[j2] = 0;
    for(i2=0;i2<nparam*nsite;i2++){
        theta[j2] = theta[j2] + x[j2][i2] * p[i2];
    }
    // Add to each of the components as needed
    sumtheta =sumtheta + exp(theta[j2]);
    //printf("Here it is %f\n",exp(theta[j2]));
    ysumtheta =ysumtheta + y[j2] * theta[j2];
}

if(condi>=0){
    int tempi = floor(condi/nsite);
    double normalpart = - 1/(2 * prvar[tempi]) * pow(p[condi] - prmu[tempi], 2 );

    // Return the actual likelihood value
    //printf("The sumtheta is %f\n",sumtheta);
    //printf("The ysumtheta is %f\n",ysumtheta);
    return( -sumtheta + ysumtheta + normalpart ) ;

} else {

    int tempi;
    double normalpart = 0;
    for(i2=0;i2<nparam*nsite;i2++){
        tempi = floor(i2/nsite);
        //printf("tempi %f",prmu[tempi]);
        normalpart= normalpart - 1/(2 * prvar[tempi]) * pow(p[i2] - prmu[tempi], 2 );
        //printf("stuff %f\n",pow(p[i2] - prmu[tempi], 2 ));
    }

    return( -sumtheta + ysumtheta + normalpart ) ;

}

}

void mvnran(double *means, double chol[nparam*nsite][nparam*nsite], double *mvn){

int i2; int k2; int l2;
int i; int j;

double norms[nparam*nsite];

for(i2=0;i2<nparam*nsite;i2++){
    norms[i2] = gsl_ran_gaussian(r,1);
    //printf("Normal %d: %f ",i2,norms[i2]);
}
//printf("\n");

int niter = 0;

for(k2=0;k2<nparam*nsite;k2++){
    mvn[k2] = 0;
    for(l2=0;l2<nparam*nsite;l2++){
        mvn[k2] = mvn[k2] + norms[l2] * chol[l2][k2];
    }
    // printf("Chol %d: %f ",l, chol[k][l]);
}
}

```

```

// printf("MVN %d: %f ",k2,mvn[k2]);
mvn[k2] = mvn[k2] + means[k2];
}

//for (i2 = 0; i2 < nparam; i2++) {
//    printf ("mvn %f\n", mvn[i2]);
//}

return;
}

void resetgen(int seed){

// ##### set up random number generator ##### /
const gsl_rng_type * TT;
gsl_rng_env_setup();
TT = gsl_rng_default;
r = gsl_rng_alloc (TT);
gsl_rng_set(r,seed); // set seed based on simulation number
gsl_rng_free (r); // function frees all the memory associated with the generator r
// ##### /

return;
}

double** Make2DDoubleArray(int arraySizeX, int arraySizeY) {
double** theArray;
theArray = (double**) malloc(arraySizeX*sizeof(double*));
int i;
for (i = 0; i < arraySizeX; i++)
theArray[i] = (double*) malloc(arraySizeY*sizeof(double));
return theArray;
}

/*
*   cov_chol.c
*
*
*   Created by Andrew Olsen on 10/16/10.
*   Copyright 2010 Brigham Young University. All rights reserved.
*
*/

// Create the covariance matrix
/*
void cov(double** beta, double* EX, double sigma[nsite*nparam][nsite*nparam]){
    int ncol = nsite * nparam;
    int j;
    int i;
    int k;
//double EX[ncol];
double EXS[ncol][ncol];

for (j=0;j<ncol;j++){
    EX[j] = 0;
    for (i=(nburn/2);i<nburn;i++){
        EX[j] = EX[j] + beta[i][j];
    }
    EX[j] = EX[j]/(nburn/2);
}

for (j=0;j<ncol;j++){

```

```

    for (k=j; k<ncol; k++){
        EXS[j][k] = 0;
        for (i=(nburn/2); i<nburn; i++){
            EXS[j][k] = EXS[j][k] + beta[i][j] * beta[i][k];
        }
        EXS[j][k] = EXS[j][k];
    }
}

//double sigma[ncol][ncol];

for (j=0; j<ncol; j++){
    for (k=j; k<ncol; k++){
        sigma[j][k] = (EXS[j][k] - (nburn/2) * EX[j] * EX[k]) / ((nburn/2) - 1);
        sigma[k][j] = sigma[j][k];
        //printf("Variance Value %20.20f\n", sigma[j][k]);
    }
}

return;
}
*/

// Create the cholesky decomposition.

void cholesky(double sigma[nsite*nparam][nsite*nparam], double chol[nsite*nparam][nsite*
nparam]){
    int i; int j; int k;
    int ncol=nsite * nparam;

    chol[0][0] = sqrt(sigma[0][0]);
    for (i=1; i<ncol; i++){
        chol[0][i] = sigma[0][i] / chol[0][0];
    }

    double tkis;
    double tktk;

    for (i=1; i<ncol; i++){
        for (j=0; j<ncol; j++){
            if (j<i){ chol[i][j]=0; } else {
                if (i!=j){
                    tktk=0;
                    for (k=0; k<(i); k++){ tktk = tktk + chol[k][i] * chol[k][j] ; }
                    chol[i][j] = (sigma[i][j] - tktk) / chol[i][i];
                } else {
                    tkis = 0;
                    for (k=0; k<(i); k++){ tkis = tkis + pow(chol[k][i], 2); }
                    chol[i][i] = sqrt((sigma[i][i] - tkis));
                }
            }
        }
    }
}

for (i=0; i<ncol; i++){
    for (j=0; j<ncol; j++){
        //printf("Cholesky Value %20.20f\n", chol[i][j]);
    }
    //printf("\n");
}

return;
}

```

## D.3 MODEL 3

### *C Code Used for Sampler*

#### Main File

```
// Metropolis sampler with independent candidate density draws.

#include <stdio.h>
#include <gsl/gsl_rng.h>      // required for random number generation
#include <gsl/gsl_randist.h>   // required for data generation from distributions
#include <math.h>
#include <stdlib.h>
#include "gsl_linalg.h"

// Define pi and the random digit constant

# define pi 3.141593
gsl_rng * r;

// Include specific header file for this analysis

#include "mcmcheader.txt"

// Include functions needed for the analysis

#include "poisll.h"
#include "mvnprior.h"
#include "mvnran.h"
#include "rgen2dar.h"
#include "cov_chol.h"

// Begin Main Function

int main(void){

    int ntune=1000;

// Define integers

    int i;
    int j;
    int ncol = nparam * nsite;
    int k;
    int y;
// Source the data

#include "data.txt"

#include "file.txt"

resetgen(651);

    // Create the matrices that will hold the estimates.

    double beta[ncol];
    double mu[nparam];
    double sig[nparam];
    double eta[nz];
    double etamu[nz];
    double etasig[nz];

// Set the initial values.
```

```

    for ( i=0; i<nparam; i++){

    mu[i] = prmu[i];
    sig[i] = 1/(pra[i] * prb[i]);

    for (j=0; j<nsite; j++){
    beta[i*nsite + j] = prmu[i] ;
    }
    }

    for ( i=0; i<nz; i++){
        etamu[i] = 0;
        etasig[i] = 1;
    }

    double betastar[ncol];
    double etastar[nz];

    double sumbk[nparam];
    double sumbkmms[nparam];
    double condmu;
    double condvar;
    double conda;
    double condb;

    double acceptmulti=0;
    double accepteta=0;

    // Obtain the cholesky decomposition of the candidate covariance matrix.

    double candvct[ncol][ncol];
    double candvcetaprime[nz][nz];

    int i2; int j2; int i3; int j3;

    for ( i2=0; i2<ncol; i2++){
        for ( j2=0; j2<ncol; j2++){

            candvct[i2][j2] = candvc[i2][j2] * pow(tune,2);

        }
    }

    for ( i2=0; i2<nz; i2++){
        for ( j2=0; j2<nz; j2++){

            candvcetaprime[i2][j2] = candvceta[i2][j2];

        }
    }

    double chol[ncol][ncol];
    cholesky(candvct, chol);

    double choleta[nz][nz];

    choleskyeta(candvceta, choleta);

    double u;

    // Tune the Gibbs Sampler

    int z;
    for (z=0; z<15; z++){

```

```

acceptmulti = 0;
acceptteta = 0;

for (i=0; i<ntune; i++) {

// Obtain a draw from the multivariate normal candidate distribution and determine to
// accept or reject it.

mvnran(beta, chol, betastar);
u = gsl_ran_flat(r, 0, 1); // Random uniform
if (log(u) < loglik(betastar, eta, X, Z, Y, mu, sig, -1) - loglik(beta, eta, X, Z, Y, mu, sig, -1)) {
    acceptmulti = acceptmulti + 1;
    for (j=0; j<ncol; j++) {
        beta[j] = betastar[j];
    }
}

// Now sample from the other complete conditional distributions.

// Update the variance components.
mvnraneta(eta, choleta, etastar);
u = gsl_ran_flat(r, 0, 1); // Random uniform
if (log(u) < logliktwo(beta, etastar, X, Z, Y, etamu, etasig, -1) - logliktwo(beta, eta, X, Z, Y,
    etamu, etasig, -1)) {
    acceptteta = acceptteta + 1;
    for (j=0; j<nz; j++) {
        eta[j] = etastar[j];
    }
}

// Sum the betas
for (k=0; k<nparam; k++) {
    sumbk[k] = 0;
    for (j=0; j<nsite; j++) {
        sumbk[k] = sumbk[k] + beta[k*j + j];
    }
}

// Get the draws for the mu_k parameters.
for (k=0; k<nparam; k++) {
    condmu = (prvar[k] * sumbk[k] + sig[k] * prmu[k]) / (nsite * prvar[k] + sig[k]);
    condvar = (prvar[k] * sig[k]) / (nsite * prvar[k] + sig[k]);
    mu[k] = gsl_ran_gaussian(r, sqrt(condvar)) + condmu;
}

// Sum the betas minus the mus squared
for (k=0; k<nparam; k++) {
    sumbkmms[k] = 0;
    for (j=0; j<nsite; j++) {
        sumbkmms[k] = sumbkmms[k] + pow(beta[k*j+1] - mu[k], 2);
    }
}

// Get the draws for the sig_k parameters.
for (k=0; k<nparam; k++) {
    conda = pra[k] + nsite/2;

```

```

        condb = 1 / ( .5 * sumbkmms[k] + (1/prb[k]) );
        sig[k] = 1/gsl_ran_gamma(r, conda, condb);
    }

    // Update the sigma parameter for the added variance components
    sumbkmms[k] = 0;
    for(k=0;k<nz;k++){
        sumbkmms[k] = sumbkmms[k] + pow(eta[k],2);
    }

    for(k=0;k<nz;k++){
        conda = etapra[k] + nz/2;
        condb = 1 / ( .5 * sumbkmms[k] + (1/etaprb[k]) );
        etasig[k] = 1/gsl_ran_gamma(r, conda, condb);
    }

}

printf("Accept betas %f w/ %f and etas %f w/ %f\n", acceptmulti/ntune, tune, accepteta/
ntune, tuneeta);

if(z>0){

    if(acceptmulti/ntune<.23){tune = tune*.9 ;}
    if(acceptmulti/ntune>.30){tune = tune*1.1 ;}
    if(accepteta/ntune<.23){tuneeta = tuneeta * .9;}
    if(accepteta/ntune>.30){tuneeta = tuneeta * 1.1;}

    for(i2=0;i2<ncol;i2++){
        for(j2=0;j2<ncol;j2++){

            candvct[i2][j2] = candvc[i2][j2] * pow(tune,2);

        }
    }

    for(i2=0;i2<nz;i2++){
        for(j2=0;j2<nz;j2++){

            candvceta[i2][j2] = candvcetaprime[i2][j2] * pow(tuneeta,2);

        }
    }

    cholesky(candvct, chol);
    choleskyeta(candvceta, choleta);

}

}

// Print the initial values

for(j=0;j<(ncol);j++){
    fprintf(mcmcf,"%f, ", beta[j]); }

for(j=0;j<(nz);j++){
    fprintf(mcmcf,"%f, ", eta[j]); }

for(k=0;k<nparam;k++){
    fprintf(mcmcf,"%f,%f, ", mu[k], sig[k]);
}

```

```

}

for(k=0;k<nz;k++){
    fprintf(mcmcf,"%f",etasig[k]);
    if(k<(nz-1)){fprintf(mcmcf,",");} else {fprintf(mcmcf,"\n");}
}

acceptmulti = 0;
accepteta = 0;

// Begin the Gibbs Sampler

for(i=1;i<(nsims);i++){

    // Obtain a draw from the multivariate normal candidate distribution and determine to
    // accept or reject it.

    mvnran(beta, chol, betastar);
    u = gsl_ran_flat(r, 0, 1); // Random uniform
    if(log(u) < loglik(betastar, eta, X, Z, Y, mu, sig, -1) - loglik(beta, eta, X, Z, Y, mu, sig, -1)){
        acceptmulti = acceptmulti + 1;
        for(j=0;j<ncol;j++){
            beta[j] = betastar[j];
        }
    }

    // Now sample from the other complete conditional distributions.

    // Update the variance components.

    mvnraneta(eta, choleta, etastar);
    u = gsl_ran_flat(r, 0, 1); // Random uniform
    if(log(u) < logliktwo(beta, etastar, X, Z, Y, etamu, etasig, -1) - logliktwo(beta, eta, X, Z, Y,
    etamu, etasig, -1)){
        accepteta = accepteta + 1;
        for(j=0;j<nz;j++){
            eta[j] = etastar[j];
        }
    }

    // Sum the betas

    for(k=0;k<nparam;k++){
        sumbk[k] = 0;
        for(j=0;j<nsite;j++){
            sumbk[k] = sumbk[k] + beta[k*j + j];
        }
    }

    // Get the draws for the mu_k parameters.

    for(k=0;k<nparam;k++){
        condmu = (prvar[k] * sumbk[k] + sig[k] * prmu[k]) / (nsite * prvar[k] + sig[k]);
        condvar = (prvar[k] * sig[k]) / (nsite * prvar[k] + sig[k]);
        mu[k] = gsl_ran_gaussian(r, sqrt(condvar)) + condmu;
    }

    // Sum the betas minus the mus squared

    for(k=0;k<nparam;k++){
        sumbkms[k] = 0;

```

```

        for (j=0;j<nsite;j++){
            sumbkmms[k] = sumbkmms[k] + pow(beta[k*j+1] - mu[k],2);
        }
    }

    // Get the draws for the sig_k parameters.

    for (k=0;k<nparam;k++){
        conda = pra[k] + nsite/2;
        condb = 1 / ( .5 * sumbkmms[k] + (1/prb[k]) );
        sig[k] = 1/gsl_ran_gamma(r,conda,condb);
    }

    // Get the draws for the variance component variance.

    sumbkmms[k] = 0;
    for (k=0;k<nz;k++){
        sumbkmms[k] = sumbkmms[k] + pow(eta[k],2);
    }

    for (k=0;k<nz;k++){
        conda = etapra[k] + nz/2;
        condb = 1 / ( .5 * sumbkmms[k] + (1/etaprb[k]) );
        etasig[k] = 1/gsl_ran_gamma(r,conda,condb);
    }

    // Print all of the parameter values for this iteration

    for (j=0;j<(ncol);j++){
        fprintf(mcmcf,"%f",beta[j]); }

    for (j=0;j<(nz);j++){
        fprintf(mcmcf,"%f",eta[j]); }

    for (k=0;k<nparam;k++){
        fprintf(mcmcf,"%f,%f",mu[k],sig[k]);
    }

    for (k=0;k<nz;k++){
        fprintf(mcmcf,"%f",etasig[k]);
        if (k<(nz-1)){fprintf(mcmcf,",");} else {fprintf(mcmcf,"\n");}
    }

    // Begin the next iteration
}

// Print the acceptance rate to the terminal.

printf("Accept beta %f\nAccept eta %f\n",acceptmulti/nsims,accepteta/nsims);

fclose(mcmcf);

return;
}

```

Header Files

```

FILE *mcmcf; mcmcf = fopen("beta_rm_c.txt","w");

// This is the Poisson Log Likelihood function

double loglik(double *p, double *pq, double x[n][nparam*nsite], double z[n][nz], double *y,
double *prmu, double *prvar, int condi){

    // The theta vector represents  $b_0 + b_1*x_1 + b_2*x_2 \dots$ 
    double theta[n];

    // Initialize loop incrementors

    int i2;
    int j2;

    // Initialize the two components of this likelihood

    double sumtheta=0;
    double ysumtheta=0;

    for(j2=0;j2<n;j2++){
        // Calculate theta for each observation
        theta[j2] = 0;
        for(i2=0;i2<nparam*nsite;i2++){
            theta[j2] = theta[j2] + x[j2][i2] * p[i2];
        }
        for(i2=0;i2<nz;i2++){
            theta[j2] = theta[j2] + z[j2][i2] * pq[i2];
        }
        // Add to each of the components as needed
        sumtheta =sumtheta + exp(theta[j2]);
        //printf("Here it is %f\n",exp(theta[j2]));
        ysumtheta =ysumtheta + y[j2] * theta[j2];
    }

    if(condi>=0){
        int tempi = floor(condi/nsite);
        double normalpart = - 1/(2 * prvar[tempi]) * pow(p[condi] - prmu[tempi], 2 );

        // Return the actual likelihood value
        //printf("The sumtheta is %f\n",sumtheta);
        //printf("The ysumtheta is %f\n",ysumtheta);
        return( -sumtheta + ysumtheta + normalpart ) ;

    } else {

        int tempi;
        double normalpart = 0;
        for(i2=0;i2<nparam*nsite;i2++){
            tempi = floor(i2/nsite);
            normalpart= normalpart - 1/(2 * prvar[tempi]) * pow(p[i2] - prmu[tempi], 2 );
        }

        return( -sumtheta + ysumtheta + normalpart ) ;

    }

}

double logliktwo(double *p, double *q, double x[n][nparam*nsite], double z[n][nz], double *y
, double *prmu, double *prvar, int condi){

    // The theta vector represents  $b_0 + b_1*x_1 + b_2*x_2 \dots$ 
    double theta[n];

```

```

// Initialize loop incrementors

int i2;
int j2;

// Initialize the two components of this likelihood

double sumtheta=0;
double ysumtheta=0;

for(j2=0;j2<n;j2++){
    // Calculate theta for each observation
    theta[j2] = 0;
    for(i2=0;i2<nparam*nsite;i2++){
        theta[j2] = theta[j2] + x[j2][i2] * p[i2];
    }
    for(i2=0;i2<nz;i2++){
        theta[j2] = theta[j2] + z[j2][i2] * q[i2];
    }
    // Add to each of the components as needed
    sumtheta =sumtheta + exp(theta[j2]);
    //printf("Here it is %f\n",exp(theta[j2]));
    ysumtheta =ysumtheta + y[j2] * theta[j2];
}

if(condi>=0){

} else {

    int tempi;
    double normalpart = 0;
    for(i2=0;i2<nz;i2++){
        tempi = floor(i2/nsite);
        normalpart= normalpart - 1/(2 * prvar[tempi]) * pow(p[i2] - prmu[tempi], 2 );
    }

    return( -sumtheta + ysumtheta + normalpart ) ;

}

}

void mvnran(double *means, double chol[nparam*nsite][nparam*nsite], double *mvn){

int i2; int k2; int l2;
int i; int j;

double norms[nparam*nsite];

for(i2=0;i2<nparam*nsite;i2++){
norms[i2] = gsl_ran_gaussian(r,1);
//printf("Normal %d: %f ",i,norms[i]);
}
//printf("\n");

int niter = 0;

for(k2=0;k2<nparam*nsite;k2++){
    mvn[k2] = 0;
    for(l2=0;l2<nparam*nsite;l2++){
        mvn[k2] = mvn[k2] + norms[l2] * chol[l2][k2];
    }
    // printf("Chol %d: %f ",l,chol[k][l]);
}
// printf("MVN %d: %f ",k,mvn[k]);

```

```

    mvn[k2] = mvn[k2] + means[k2];
}

//for (i2 = 0; i2 < nparam; i2++) {
//    printf ("mvn %f\n", mvn[i2]);
//}

return;
}

void mvnraneta(double *means, double chol[nz][nz], double *mvn){

    int i2; int k2; int l2;
    int i; int j;

    double norms[nz];

    for(i2=0;i2<nz;i2++){
        norms[i2] = gsl_rng_gaussian(r,1);
        //printf("Normal %d: %f ",i,norms[i]);
    }
    //printf("\n");

    int niter = 0;

    for(k2=0;k2<nz;k2++){
        mvn[k2] = 0;
        for(l2=0;l2<nz;l2++){
            mvn[k2] = mvn[k2] + norms[l2] * chol[l2][k2];
            // printf("Chol %d: %f ",l,chol[k][l]);
        }
        // printf("MVN %d: %f ",k,mvn[k]);
        mvn[k2] = mvn[k2] + means[k2];
    }

    return;
}

void resetgen(int seed){

// ##### set up random number generator ##### /
const gsl_rng_type * TT;
gsl_rng_env_setup();
TT = gsl_rng_default;
r = gsl_rng_alloc (TT);
gsl_rng_set(r,seed); // set seed based on simulation number
gsl_rng_free (r); // function frees all the memory associated with the generator r
// ##### /

return;
}

double** Make2DDoubleArray(int arraySizeX, int arraySizeY) {
double** theArray;
theArray = (double**) malloc(arraySizeX*sizeof(double*));
int i;
for (i = 0; i < arraySizeX; i++)
    theArray[i] = (double*) malloc(arraySizeY*sizeof(double));
return theArray;
}

```

```

/*
 * cov_chol.c
 *
 *
 * Created by Andrew Olsen on 10/16/10.
 * Copyright 2010 Brigham Young University. All rights reserved.
 *
 */

// Create the covariance matrix
/*
void cov(double** beta, double* EX, double sigma[nsite*nparam][nsite*nparam]){
    int ncol = nsite * nparam;
    int j;
    int i;
    int k;
//double EX[ncol];
double EXS[ncol][ncol];

for (j=0;j<ncol;j++){
    EX[j] = 0;
    for (i=(nburn/2);i<nburn;i++){
        EX[j] = EX[j] + beta[i][j];
    }
    EX[j] = EX[j]/(nburn/2);
}

for (j=0;j<ncol;j++){
    for (k=j;k<ncol;k++){
        EXS[j][k] = 0;
        for (i=(nburn/2);i<nburn;i++){
            EXS[j][k] = EXS[j][k] + beta[i][j] * beta[i][k];
        }
        EXS[j][k] = EXS[j][k];
    }
}

//double sigma[ncol][ncol];

for (j=0;j<ncol;j++){
    for (k=j;k<ncol;k++){
        sigma[j][k] = (EXS[j][k] - (nburn/2) * EX[j] * EX[k]) / ((nburn/2) - 1);
        sigma[k][j] = sigma[j][k];
        //printf("Variance Value %-.20f\n",sigma[j][k]);
    }
}

    return;
}
*/

// Create the cholesky decomposition.

void cholesky(double sigma[nsite*nparam][nsite*nparam],double chol[nsite*nparam][nsite*
nparam]){
    int i; int j; int k;
    int ncol=nsite * nparam;

chol[0][0] = sqrt(sigma[0][0]);
for (i=1;i<ncol;i++){
    chol[0][i] = sigma[0][i]/chol[0][0];
}

double tkis;
double tktk;

```

```

for (i=1;i<ncol;i++){
    for (j=0;j<ncol;j++){
        if (j<i){chol[i][j]=0;} else{
            if (i!=j){
                tktk=0;
                for(k=0;k<(i);k++){ tktk = tktk + chol[k][i] * chol[k][j] ;}
                chol[i][j] = (sigma[i][j] - tktk) / chol[i][i];
            } else {
                tkis = 0;
                for(k=0;k<(i);k++){ tkis = tkis + pow(chol[k][i],2); }
                chol[i][i] = sqrt((sigma[i][i] - tkis));
            }
        }
    }
}

```

```

for (i=0;i<ncol;i++){
    for (j=0;j<ncol;j++){
        //printf("Cholesky Value %20.20f\n",chol[i][j]);
    }
    //printf("\n");
}

return;
}

```

```

void choleskyeta(double sigma[nz][nz],double chol[nz][nz]){
    int i; int j; int k;

    chol[0][0] = sqrt(sigma[0][0]);
    for (i=1;i<nz;i++){
        chol[0][i] = sigma[0][i]/chol[0][0];
    }

    double tkis;
    double tktk;

    for (i=1;i<nz;i++){
        for (j=0;j<nz;j++){
            if (j<i){chol[i][j]=0;} else{
                if (i!=j){
                    tktk=0;
                    for(k=0;k<(i);k++){ tktk = tktk + chol[k][i] * chol[k][j] ;}
                    chol[i][j] = (sigma[i][j] - tktk) / chol[i][i];
                } else {
                    tkis = 0;
                    for(k=0;k<(i);k++){ tkis = tkis + pow(chol[k][i],2); }
                    chol[i][i] = sqrt((sigma[i][i] - tkis));
                }
            }
        }
    }

    return;
}

```

## D.4 MODEL 4

### *R Code Used for Sampler*

```
sampler <- function(Y,X,P,nsims,burn,tunetheta,tunept,tunealpha,tunebeta,data,savefile,m1){
  set.seed(651651)
  # Load needed libraries
  library(MASS)
  # Compute the number of sites and the number of years.
  K <- nsite <- length(unique(data$site))
  J <- nyear <- max(data$Year) - min(data$Year) + 1
  if(length(m1)==1){m1 <- rep(m1,K)}
  # Compute the overall number of parameters, etc.
  ntheta <- ncol(X)
  np <- ncol(P)
  ntau <- nsite
  nparam <- ntheta + np + ntau + 2
  # Create the posterior matrix
  post <- matrix(nrow=nsims+burn,ncol=nparam)
  # Create names for the posterior matrix.
  t1 <- paste('theta_site ',1:K,sep='')
  t2 <- paste(rep(t1,each=J), '_year ',rep(1:J,K),sep='')
  t3 <- paste('pi_site ',1:K,sep='')
  t4 <- paste(rep(t3,each=J), '_year ',rep(1:J,K),sep='')
  t5 <- paste('tau_site ',1:K,sep='')
  t6 <- c('alpha','beta')
  titles <- c(t2,t3,t5,t6)
  colnames(post) <- titles
  # Create a new poisson function that allows for the gamma distribution to be used rather
  # than factorial.
  mydpois <- function(x,lambda,log=FALSE){
    if(log==F){exp(-lambda) * lambda^x / gamma(x+1)} else {
      -lambda + x * log(lambda) - lgamma(x+1)
    }
  }
  # Create the likelihood function
  ylik <- function(y,lambda,pi){
    pi * (y==0) + (1-pi) * exp(mydpois(y,lambda,log=T))
  }
  # Create the candidate covariance matrix for the theta.
  # This matrix puts correlation between the years of a specific site, less and less the
  # further they are apart.
  rho <- .33
  arlfit <- matrix(0,nyear,nyear)
  for(i in 1:nyear){
    for(j in 1:nyear){
      arlfit[i,j] <- rho^(abs(i-j))
    }
  }
```

```

}

cstheta <- kronecker(diag(tunetheta,nsite,nsite),ar1fit)

# Set the candidate standard deviation for the truncated normal for the pi

cspi <- tuneapi

csa <- tunealpha
csb <- tunebeta

# Set the initial values. We let the theta begin at the prior for the first theta. We let
  the pi begin at .15. We let the variances and alpha and beta begin at 1.

post[1,] <- c(rep(m1,each=J),rep(.15,K),rep(1,K+2))

# Write the initial values to the file

write.table(rbind(post[1,]),savefile,row.names=F)

# This makes an index of the first and last years of each site.

firsts <- seq(1,by=nyear,length=nsite)
lasts <- seq(nyear,by=nyear,length=nsite)

# This stores initial values to the appropriate vectors.

theta <- post[1,1:ntheta]
pi <- post[1,(ntheta+1):(ntheta+np)]
tau <- post[1,(ntheta+np+1):(ntheta+np+ntau)]
alpha <- post[1,ntheta+np+ntau+1]
beta <- post[1,ntheta+np+ntau+2]

# Initialize the acceptance rates to 0.

atheta <- api <- aalpha <- abeta <- rep(0,nsims+burn)

# Begin the for loop here.

for(i in 2:(nsims+burn)){

  thetastar <- mvrnorm(1,mu=theta,Sigma=cstheta)

# Update all of the thetas

  llstar <- sum(log(ylik(Y,exp(X %*% thetastar),P%*%pi))) + sum(dnorm(c(thetastar[firsts] - m1
    ,thetastar[-firsts] - theta[-lasts]),0,c(tau,rep(tau,each=6)),log=T))

  ll <- sum(log(ylik(Y,exp(X %*% theta),P%*%pi))) + sum(dnorm(c(theta[firsts] - m1,theta[-
    firsts] - theta[-lasts]),0,c(tau,rep(tau,each=6)),log=T))

  if(log(runif(1)) < llstar - ll){
    theta <- post[i,1:ntheta] <- thetastar; atheta[i] <- 1
  } else {post[i,1:ntheta] <- theta}

# Now update the pis

  pistar <- rtnorm(np,pi,cspi,0,1)

  #if(sum(pistar==0)==0){

  llstar <- sum(log(ylik(Y,exp(X %*% theta),P%*%pistar))) + sum(dbeta(pistar,alpha,beta,log=T)
    ) + sum(dtnorm(pi,pistar,cspi,0,1,log=T))

  ll <- sum(log(ylik(Y,exp(X %*% theta),P%*%pi))) + sum(dbeta(pi,alpha,beta,log=T)) + sum(
    dtnorm(pistar,pi,cspi,0,1,log=T))

```

```

if(log(runif(1)) < llstar - ll){
  pi <- post[i,(ntheta+1):(ntheta + np)] <- pistar; api[i] <- 1
} else {post[i,(ntheta+1):(ntheta + np)] <- pi}
#} else {post[i,(ntheta+1):(ntheta + np)] <- pi}

# Now update tau's

tempa <- at + nyear/2

diffs <- c(theta[firsts] - ml, theta[-firsts] - theta[-lasts])

for(k in 1:nsite){
idx <- c(1:nsite, rep(1:nsite, each=(nyear-1)))

idx2 <- which(idx==k)

tempb <- 1/(.5 * sum((diffs[idx2])^2) + 1/bt)

post[i, ntheta + np + k] <- tau <- 1/rgamma(1, shape=tempa, scale=tempb)
}

# Now update alpha

alphastar <- rnorm(1, alpha, csa)

if(alphastar > 0){
llstar <- sum(dbeta(pi, alphastar, beta, log=T)) + dgamma(alphastar, shape=aa, scale=ba, log=T)

ll <- sum(dbeta(pi, alpha, beta, log=T)) + dgamma(alpha, shape=aa, scale=ba, log=T)

if(log(runif(1)) < llstar - ll){
  alpha <- post[i, (ntheta+np+nsite+1)] <- alphastar; aalpha[i] <- 1
} else {post[i, (ntheta+np+nsite+1)] <- alpha}
} else {post[i, (ntheta+np+nsite+1)] <- alpha}

# Finally, update Beta

betastar <- rnorm(1, beta, csb)
if(betastar > 0){
llstar <- sum(dbeta(pi, alpha, betastar, log=T)) + dexp(betastar, 1/c, log=T)

ll <- sum(dbeta(pi, alpha, beta, log=T)) + dexp(beta, 1/c, log=T)

if(log(runif(1)) < llstar - ll){
  beta <- post[i, (ntheta+np+nsite+2)] <- betastar; abeta[i] <- 1
} else {post[i, (ntheta+np+nsite+2)] <- beta}
} else {post[i, (ntheta+np+nsite+2)] <- beta}

write.table(rbind(post[i,]), savefile, row.names=F, append=T, col.names=F)

if(i/10000==round(i/10000,0)){print(paste('Iteration ', i))}

}

accept <- cbind(atheta, api, aalpha, abeta)
print(colMeans(accept[-(1:burn),]))

return(post[-(1:burn),])

}

atime <- Sys.time()

library(msm)

nsims= 500000

```

```

burn = 5000

# Set the working directory to the C directory

setwd('~ / Documents / Masters_Project / Code / Project / model4 ')

source('sampler.R')
source('functions.R')

#####
#   Cable Barriers   #
#####

# Read in the data

cbdata <- read.csv('~ / Documents / Masters_Project / Code / Project / data / cbdata.csv ')

# Read in the index

idxcb <- read.csv('~ / Documents / Masters_Project / Code / Project / data / cbidx.csv ', header=TRUE)

# Get the 1997 data to use as priors.

cbdata97 <- read.csv('~ / Documents / Masters_Project / Code / Project / data / cbdata97.csv ')

mlcbc <- log( cbdata97$crashes / (idxcb$EndMile - idxcb$BegMile))
mlcbs <- log( cbdata97$severe / (idxcb$EndMile - idxcb$BegMile))
mlbcm <- log( cbdata97$crossmed / (idxcb$EndMile - idxcb$BegMile))

aa <- 1
ba <- 1
c <- 1
at <- 5
bt <- .2

cbdata$year <- cbdata$Year - 2001

Xcb <- model.matrix(~ as.factor(site):as.factor(year)-1,cbdata)
Pcb <- model.matrix(~ as.factor(site)-1,cbdata)

Ycbc <- t(t(cbdata$crash))/cbdata$nmil
Ycbs <- t(t(cbdata$severe))/cbdata$nmil
Ybcm <- t(t(cbdata$crossmed))/cbdata$nmil

post_cb_s <- sampler(Ycbs,Xcb,Pcb,nsims,burn,.06,.0025,.35,3.25,cbdata,'estimates_cb_s.txt',
mlcbs)

system('rm estimates_cb_s.txt.gz')
system('gzip estimates_cb_s.txt')

post_cb_cm <- sampler(Ybcm,Xcb,Pcb,nsims,burn,.1,.008,.36,3.25,cbdata,'estimates_cb_cm.txt',
mlbcm)

system('rm estimates_cb_cm.txt.gz')
system('gzip estimates_cb_cm.txt')

post_cb_c <- sampler(Ycbc,Xcb,Pcb,nsims,burn,.003,.004,.35,3.25,cbdata,'estimates_cb_c.txt',
mlcbc)

system('rm estimates_cb_c.txt.gz')
system('gzip estimates_cb_c.txt')

compCBS <- compmodel4(post_cb_s,Xcb,Ycbs,Pcb,mlcbs,cbdata)
compCBcm <- compmodel4(post_cb_cm,Xcb,Ybcm,Pcb,mlbcm,cbdata)
compCBc <- compmodel4(post_cb_c,Xcb,Ycbc,Pcb,mlcbc,cbdata)

```

```
#####
#   Raised Medians   #
#####

# Read in the data

rmdata <- read.csv('~ / Documents / Masters_Project / Code / Project / data / rmdata.csv ')

# Read in the index

idxrm <- read.csv('~ / Documents / Masters_Project / Code / Project / data / medidx.csv ', header=TRUE)

idxrm97 <- read.csv('~ / Documents / Masters_Project / Code / Project / data / medidx97.csv ', header=TRUE
)

rmdata97 <- read.csv('~ / Documents / Masters_Project / Code / Project / data / rmdata97.csv ')

mlrmc <- log( rmdata97$crashes / (idxrm97$EndMile - idxrm97$BegMile))
mlrms <- log( rmdata97$severe / (idxrm97$EndMile - idxrm97$BegMile))
# Use a 1 here because I don't believe the prior value should be -Inf.
mlrmcm <- log( 1 / (idxrm97$EndMile - idxrm97$BegMile))

aa <- 1
ba <- 1
c <- 1
at <- 5
bt <- .2

rmdata$year <- rmdata$Year - 2001

Xrm <- model.matrix(~ as.factor(site):as.factor(year)-1, rmdata)
Prm <- model.matrix(~ as.factor(site)-1, rmdata)

# Define the Y matrices

Yrmc <- t(t(rmdata$crash))/rmdata$nmil
Yrms <- t(t(rmdata$severe))/rmdata$nmil
Yrmcm <- t(t(rmdata$crossmed))/rmdata$nmil

post_rm_s <- sampler(Yrms, Xrm, Prm, nsims, burn, .03, .0042, .35, 2.7, rmdata, 'estimates_rm_s.txt',
mlrms)

system('rm estimates_rm_s.txt.gz')
system('gzip estimates_rm_s.txt')

post_rm_cm <- sampler(Yrmcm, Xrm, Prm, nsims, burn, .35, .24, 1.5, 1.5, rmdata, 'estimates_rm_cm.txt',
mlrmcm)

system('rm estimates_rm_cm.txt.gz')
system('gzip estimates_rm_cm.txt')

post_rm_c <- sampler(Yrmc, Xrm, Prm, nsims, burn, .0012, .0034, .35, 3.2, rmdata, 'estimates_rm_c.txt',
mlrmc)

system('rm estimates_rm_c.txt.gz')
system('gzip estimates_rm_c.txt')

comprms <- compmodel4(post_rm_s, Xrm, Yrms, Prm, mlrms, rmdata)
comprmc <- compmodel4(post_rm_cm, Xrm, Yrmcm, Prm, mlrmcm, rmdata)
comprmc <- compmodel4(post_rm_c, Xrm, Yrmc, Prm, mlrmc, rmdata)

save.image('model4.Rdata')

save(compcbcm, compcbcs, compcbc, comprmc, comprms, comprmc, file='model4comp.Rdata')
```