Section 3: Bayesian GLM

3.1 Modeling non-Gaussian observations

So far, we've assumed real-valued observations. In this setting, our likelihood model is a univariate normal, parametrized by a mean $x_i^T\beta$ and some precision that does not directly depend on the value of x_i. In general, $x_i^T\beta$ will take values in $\mathbb{R}$

If we don't want to use a Gaussian likelihood, we typically won't be able to parametrize our data using a real-valued parameter. Instead, we must transform it via an appropriate link function. This is, in essence, the generalized linear model.

As a first step into other types of data, let's consider binary valued observations. Here, the natural likelihood model is a Bernoulli random variable; however we cannot directly parametrize this by $x_i^T\beta$. Instead, we must transform $x_i^T\beta$ to lie between 0 and 1 via some function $g^{-1}:\mathbb{R}\rightarrow(0,1)$. We can then write a linear model as

$$\begin{aligned} y_i|p_i \sim& \text{Bernoulli}(p_i)\\ p_i =& g^{-1}(x_i^T\beta)\\ \beta|\theta \sim& \pi_\theta(\beta) \end{aligned}$$

where $\pi_\theta(\beta)$ is our choice of prior on β. Unfortunately, there is no choice of prior here that makes the model conjugate.

Let's start off with a normal prior on β. One appropriate function for g^{-1} is the inverse CDF of the normal distribution – known as the probit function. This is equivalent to assuming our data are generated according to

$$\begin{aligned} y_i =& \begin{cases} 1 & if z > 0\\ 0 & \text{otherwise}\end{cases}\\ z_i \sim& \text{N}(x_i^T\beta, \tau^2) \end{aligned}$$

If we put a normal-inverse gamma prior on β and τ, then we have a *latent* regression model on the (x_i, z_i) pairs, that is idential to what we had before! Conditioned on the z_i, we can easily sample values for β and τ.

Exercise 3.1 *To complete our Gibbs sampler, we must specify the conditional distribution $p(y_i|x_i, z_i, \beta, \tau)$. Write down the form of this conditional distribution, and write a Gibbs sampler to sample from the posterior distribution. Test it on the dataset* `pima.csv`*, which contains diabetes information for women of Pima indian heritage. The dataset is from National Institute of Diabetes and Digestive and Kidney Diseases, full information and explanation of variables is available at* `http://archive.ics.uci.edu/ml/datasets/Pima+Indians+Diabetes`.

Proof:

$$p(z_i|x_i, y_i, \beta, \tau) \propto \exp\{-\frac{(z_i - x_i^T\beta)^2}{2\tau^2}\}1_{\{z_i>0, y_i=1\}} + \exp\{-\frac{(z_i - x_i^T\beta)^2}{2\tau^2}\}1_{\{z_i\leq 0, y_i=0\}} \tag{3.1}$$

Therefore, we need to sample z_i from truncated normal distribution. The rest of conditional distribution has been solved in Ex 2.9-2.10. Then we can use Gibbs sample to conduct the inference. The results are shown in Figure 3.1. We can use the posterior mean $\hat{\beta}$ to do prediction, and the accuracy(percentage of correct prediction) of 0.75 can be achieved.(For classification problem, an investigation of false positive and false negative is necessary!)

■

Another choice for $g^{-1}(\theta)$ might be the logit function, $\frac{1}{1+e^{-x^T\beta}}$. In this case, it's less obvious to see how we can construct an auxiliary variable representation (it's not impossible! See **?**. But for now, we'll assume we haven't come up with something). So, we're stuck with working with the posterior distribution over β.

Exercise 3.2 *Sadly, the posterior isn't in a "known" form. As a starting point, let's find the maximum a posteriori estimator (MAP). The dataset "titantic.csv" contains survival data from the Titanic; we're going to look at probability of survival as a function of age. For now, we're going to assume the intercept of our regression is zero – i.e. that β is a scalar. Write a function (that can use a black-box optimizer! No need to reinvent the wheel. It shouldn't be a long function) to estimate the MAP of β. Note that the MAP corresponds to the frequentist estimator using a ridge regularization penalty.*

Proof:

$$p(\beta|y_1, ..., y_n) \propto e^{-\frac{(\beta-\mu)^2}{2\sigma^2}} \prod_{i=1}^{n} (\frac{1}{1+e^{-x_i\beta}})^{y_i} (\frac{1}{1+e^{x_i\beta}})^{1-y_i} \tag{3.2}$$

where μ and σ^2 are parameters in the prior of β. Let's take $\mu = 0$, and $\sigma^2 = 1$. And use some blackbox functions to find the maximum of the above objective function. The maximum is achieved when $x = -0.01101471$.

■

Exercise 3.3 *OK, we don't know how to sample from the posterior, but we can at least look at it. Write a function to calculate the posterior pdf $p(\beta|\mathbf{x}, \mathbf{y}, \mu, \sigma^2)$, for some reasonable hyperparameter values μ and θ (up to a normalizing constant is fine!). Plot over a reasonable range of β (your MAP from the last question should give you a hint of a reasonable range).*

Proof: The plot is shown in Figure 3.2. ■

The Laplace approximation is a method for approximating a distribution with a Gaussian, by matching the mean and variance at the mode.[1] Let P^* be the (unnormalized) PDF of a distribution we wish to approximate. We start by taking a Taylor expansion of the log (unnormalized) PDF at the global maximizing value x^*

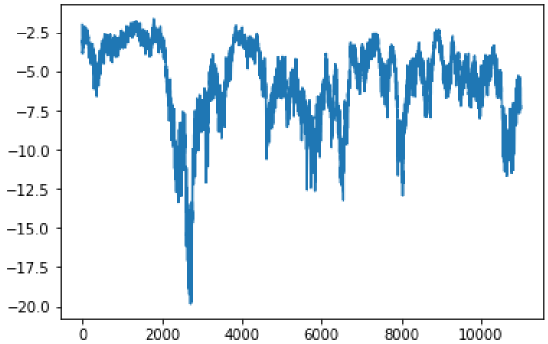
$$\log P^*(x) \approx \log P^*(x^*) - \frac{c}{2}(x - x^*)^2$$

where $c = -\frac{\delta^2}{\delta x^2} \log P^*(x)\Big|_{x=x^*}$.

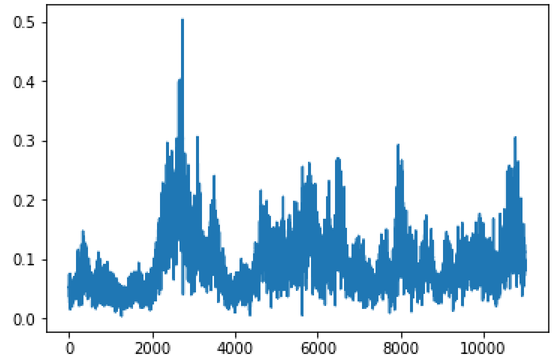
We approximate P^* with an unnormalized Gaussian, with the same mean and variance as P^*:

$$Q^*(x) = P^*(x^*) \exp\left\{-\frac{c}{2}(x - x^*)^2\right\}$$

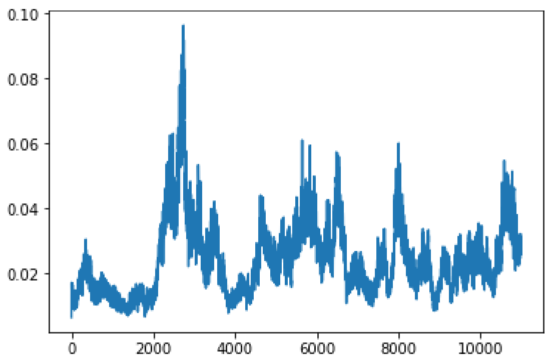
[1]More generally, the Laplace approximation is used to approximate integrands of the form $\int_A e^{Nf(x)}dx$... but for our purposes we will always be working with PDFs.



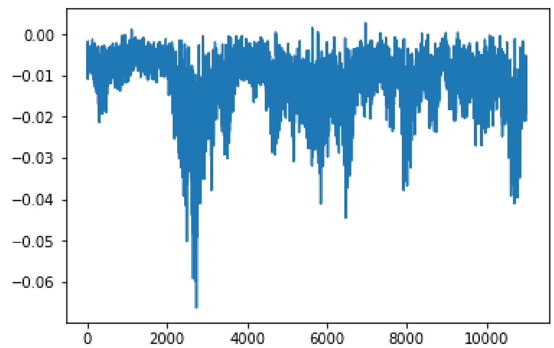
(a) Markov Chain for β_0



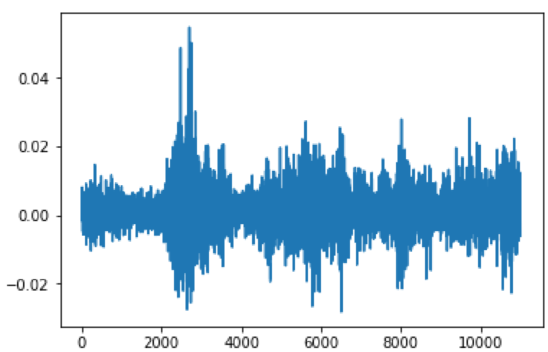
(b) Markov Chain for β_1



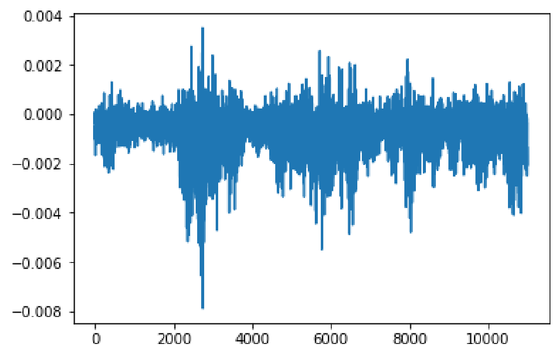
(c) Markov Chain for β_2



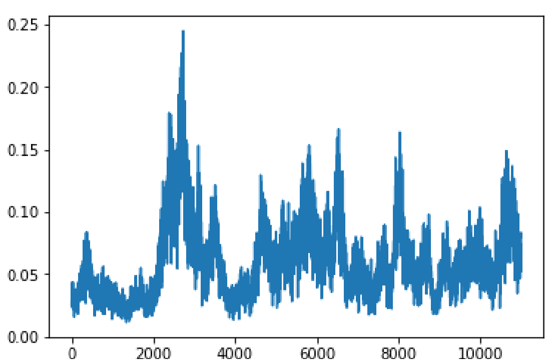
(d) Markov Chain for β_3



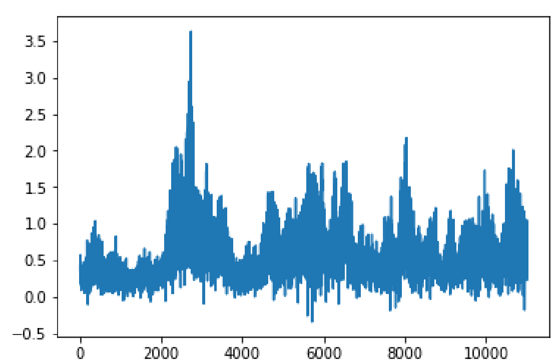
(e) Markov Chain for β_4



(f) Markov Chain for β_5



(g) Markov Chain for β_6



(h) Markov Chain for β_7

Figure 3.1: Results of Gibbs sample for Probit regression.

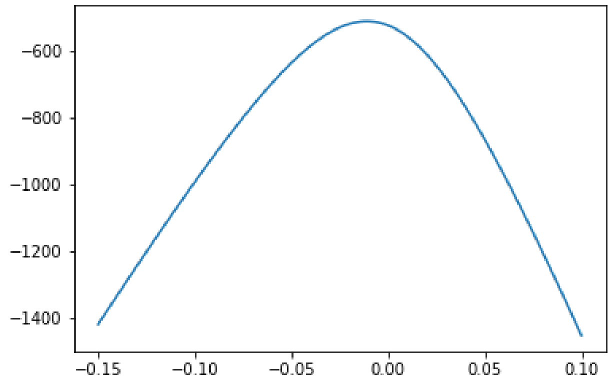


Figure 3.2: Plot of the unnormalized posterior

Exercise 3.4 *Find the mean and precision of a Gaussian that can be used in a Laplace approximation to the posterior distribution over β.*

Proof: The unnormalized posterior is shown in Equation 3.2.

$$\log p(\beta|y_1,...,y_n,\mu,\sigma^2) = \text{constant} - \frac{(\beta-\mu)^2}{2\sigma^2} - \sum_i (y_i \log(1+e^{-x_i\beta}) + (1-y_i)\log(1+e^{x_i\beta})) \tag{3.3}$$

Therefore,

$$\frac{d}{d\beta}\log p(\beta|\cdots) = -\frac{\beta-\mu}{\sigma^2} - \sum_i (y_i x_i \frac{-e^{-x_i\beta}}{1+e^{-x_i\beta}} + (1-y_i)x_i\frac{e^{x_i\beta}}{1+e^{x_i\beta}}),$$

$$\frac{d^2}{d\beta^2}\log p(\beta|\cdots) = -\frac{1}{\sigma^2} - \sum_i (y_i x_i^2 \frac{e^{-x_i\beta}}{(1+e^{-x_i\beta})^2} + (1-y_i)x_i^2\frac{e^{-x_i\beta}}{(1+e^{-x_i\beta})^2}) = -\frac{1}{\sigma^2} - \sum_i x_i^2 \frac{e^{-x_i\beta}}{(1+e^{-x_i\beta})^2}.$$

Do Tyler Expansion around the mode of posterior β^*, then $\log p(\beta|\cdots) \approx \log p(\beta^*|\cdots) - \frac{c}{2}(\beta-\beta^*)^2$, where $c = -\frac{d^2}{d\beta^2}\log p(\beta|\cdots)|_{\beta=\beta^*} = \frac{1}{\sigma^2} + \sum_i x_i^2 \frac{e^{-x_i\beta^*}}{(1+e^{-x_i\beta^*})^2}$.

Therefore, the mean of the approximating Gaussian is β^*, and the precision is c.

■

Exercise 3.5 *That's all well and good... but we probably have a non-zero intercept. We can extend the Laplace approximation to multivariate PDFs. This amounts to estimating the precision matrix of the approximating Gaussian using the negative of the Hessian – the matrix of second derivatives*

$$H_{ij} = \frac{\delta^2}{\delta x_i \delta x_j} \log P^*(x)\Big|_{x=x^*}$$

Use this to approximate the posterior distribution over β. Give the form of the approximating distribution, plus 95% marginal credible intervals for its elements.

Proof: We can generalize the result from the last exercise into multivariate case. The Laplacian approximation of the posterior of β would be $\mathcal{N}(\beta^*, \Lambda^{-1})$, where

$$\Lambda = \frac{1}{\sigma^2}I + \sum_i \frac{e^{-x_i^T\beta^*}}{(1+e^{-x_i^T\beta^*})^2} x_i x_i^T.$$

With the data, we found $\beta^* = [-1.93792428, 0.03916847]$, and $\Lambda = \begin{bmatrix} 1.67 \cdot 10^2 & 5.69 \cdot 10^3 \\ 5.69 \cdot 10^3 & 2.18 \cdot 10^5 \end{bmatrix}$.

The 95% interval for β_0 is $[-2.09253152, -1.78331704]$, for β_1 is $[0.03488676, 0.04345018]$.

■

Let's try the same thing with a Poisson likelihood. Here, the obvious transformation is to let $g^{-1}(\theta) = e^\theta$, i.e.

$$\begin{aligned} y_i|p_i \sim & \text{Poisson}(\lambda_i) \\ \lambda_i = & e^{x_i^T \beta} \end{aligned}$$

We're going to work with the dataset `tea_discipline_oss.csv`, a dataset gathered by Texas Appleseed, looking at the number of out of school suspensions (ACTIONS) across schools in Texas. The data is censored for privacy reasons – data points with fewer than 5 actions are given the code "-99". For now, we're going to exclude these data points.

Exercise 3.6 *We're going to use a Poisson model on the counts. Ignoring the fact that the data is censored, why is this not quite the right model? Hint: there are several answers to this – the most fundamental involve considering the support of the Poisson.*

Proof: One reason is that Poisson assigns a considerable part of mass to numbers less than 5, which are absent from data.

Also, the histogram shows that the distribution is multi-modal. Furthermore, over-dispersed is also a problem, i.e., the mean not equal to standard deviation. ■

Exercise 3.7 *Let's assume our only covariate of interest is GRADE*[2] *and put a normal prior on β. Using a Laplace approximation and an appropriately vague prior, find 95% marginal credible intervals for the entries of β. You'll probably want to use an intercept.*

Proof: Suppose, the prior for β is $\mathcal{N}(\mu, \sigma^2 I)$. The unnormalized posterior is

$$p(\beta|\cdots) \propto e^{-\frac{(\beta-\mu)^T(\beta-\mu)}{2\sigma^2}} \prod_i e^{-e^{x_i^T\beta}} \frac{e^{y_i x_i^T \beta}}{y_i!}$$

Then,

$$\log p(\beta|\cdots) = \text{constant} - \frac{(\beta-\mu)^T(\beta-\mu)}{2\sigma^2} - \sum_i (e^{x_i^T\beta} - y_i x_i^T \beta) \tag{3.4}$$

Therefore,

$$\begin{aligned} \frac{d}{d\beta} \log p(\beta|\cdots) &= -\frac{\beta-\mu}{\sigma^2} - \sum_i (e^{x_i^T\beta} x_i - y_i x_i), \\ \frac{d^2}{d\beta^2} \log p(\beta|\cdots) &= -\frac{I}{\sigma^2} - \sum_i (e^{x_i^T\beta} x_i x_i^T). \end{aligned}$$

Therefore, the mean of the approximating Gaussian is $\beta^* = [2.38935902, 0.0502636]$, and the precision is $\Lambda = \begin{bmatrix} 33177 & 2576058 \\ 2576058 & 22480441 \end{bmatrix}$. The 95% interval for β_0 is $[2.38589411, 2.39282393]$, for β_1 is $[0.04984178, 0.05068542]$.

[2] I have manually replaced Kindergarten and Pre-K with Grades 0 and -1, respectively.

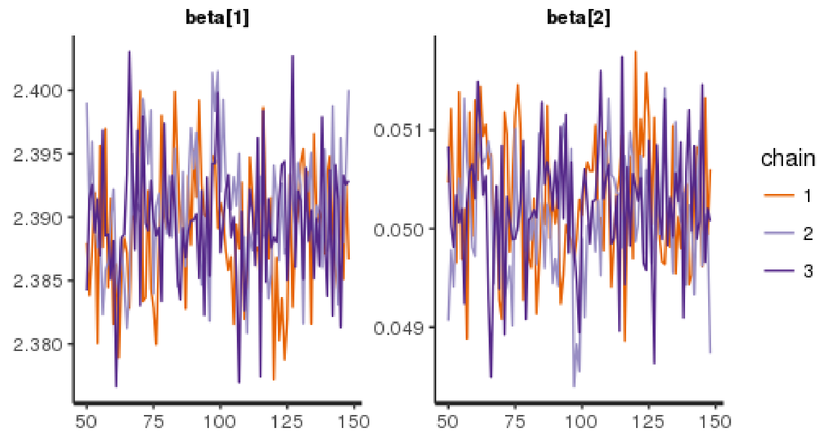


Figure 3.3

■

Exercise 3.8 (Optional) *Repeat the analysis using a set of variables that interest you.*

Even though we don't have conjugacy, we can still use MCMC methods – we just can't use our old friend the Gibbs sampler. Since this isn't an MCMC course, let's use STAN, a probabilistic programming language available for R, python and Matlab. I'm going to assume herein that we're using RStan, and give appropriate scripts; it should be fairly straightforward to use if you're an R novice, or if you want to use a different language, there are hints on translating to PyStan at `http://pystan.readthedocs.io/en/latest/differences_pystan_rstan.htm` and info on MatlabStan (which seems much less popular) at `http://mc-stan.org/users/interfaces/matlab-stan`.

Exercise 3.9 *Download the sample STAN script* `poisson.stan` *and corresponding R script* `run_poisson_stan.R`*. The R script should run the regression vs GRADE from earlier (feel free to change the prior parameters). Run it and see how the results differ from the Laplace approximation. Modify the scripy to include more variables, and present your results.*

Proof: The results are shown in Figure 3.3 and Figure 3.4. In Figure 3.3, Markov chains of the intercept coefficient and grade coefficient are plotted and the result is consistent with the Laplacian approximation results. In Figure 3.4, I plotted the Markov chains of the model including sex information. It does not change dramatically. We can compare these two models in terms of predictive errors. If we use the mean of poisson distribution as the prediction, then the predictive error (on the training set) for the first model is 237237.5. The predictive error for the second model is 237228.5, so the second model is slightly better because we incorporate the sex information.

■

Exercise 3.10 *Consider ways you might improve your regression (still, using the censored data) - while staying in the GLM framework. Ideas might include hierarchical error modeling (as we looked at in the last set of exercises), interaction terms... or something else! Looking at the data may give you inspiration. Implement this in STAN.*

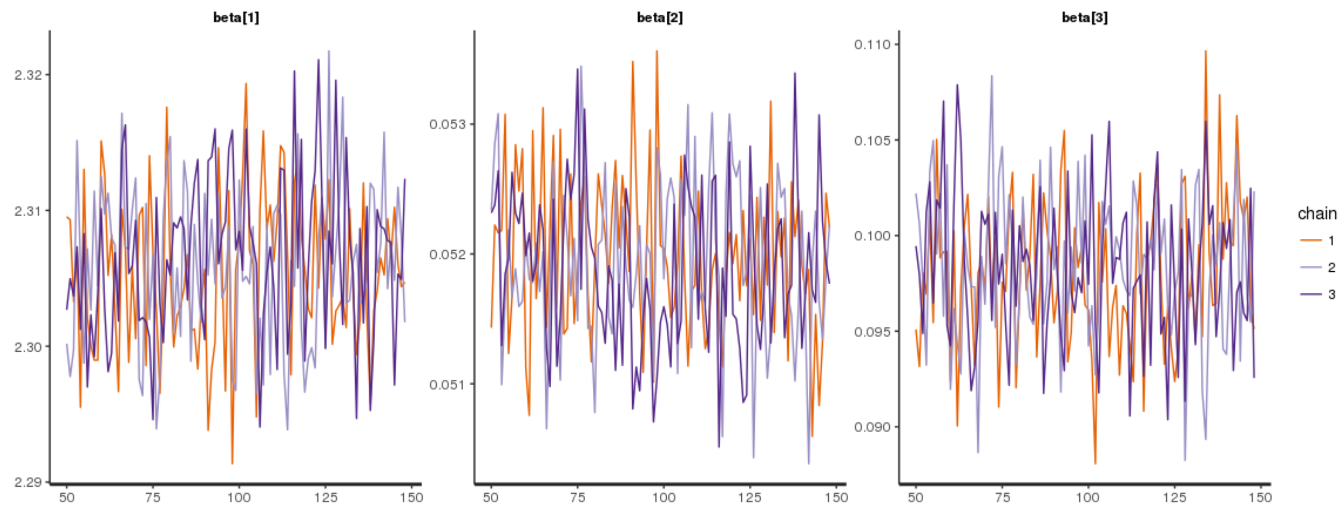


Figure 3.4

Proof: We can take the interaction between sex variable and grade variable into our model. With much change in the code, we can have the posterior of the coefficients in the new model as shown in Figure 3.5. The posterior does not change much from the previous model. Also, we can compute the predictive error, which in this case is 237056.3, slightly better than the previous models.

■

Exercise 3.11 *We are throwing away a lot of information by not using the censored data. Come up with a strategy, and write down how you would alter your model/sampler. Bonus points for actually implementing it in STAN (hint: look up the section on censored data in the STAN manual).*

Proof: We can view data y_i as the truncated variable of some unknown parameter/latent variable z_i as follows:

$$\begin{aligned} y_i =& z_i \mathbf{1}_{\{z_i > 5\}} - 99\mathbf{1}_{\{z_i \leq 5\}} \\ z_i | p_i \sim& \text{Poisson}(\lambda_i) \\ \lambda_i =& e^{x_i^T \beta} \end{aligned}$$

We can either integrate out z_i and do likelihood maximization, or do Gibbs sampling(condition on z_i sample other parameters, and condition on other parameters sample z_i). ■

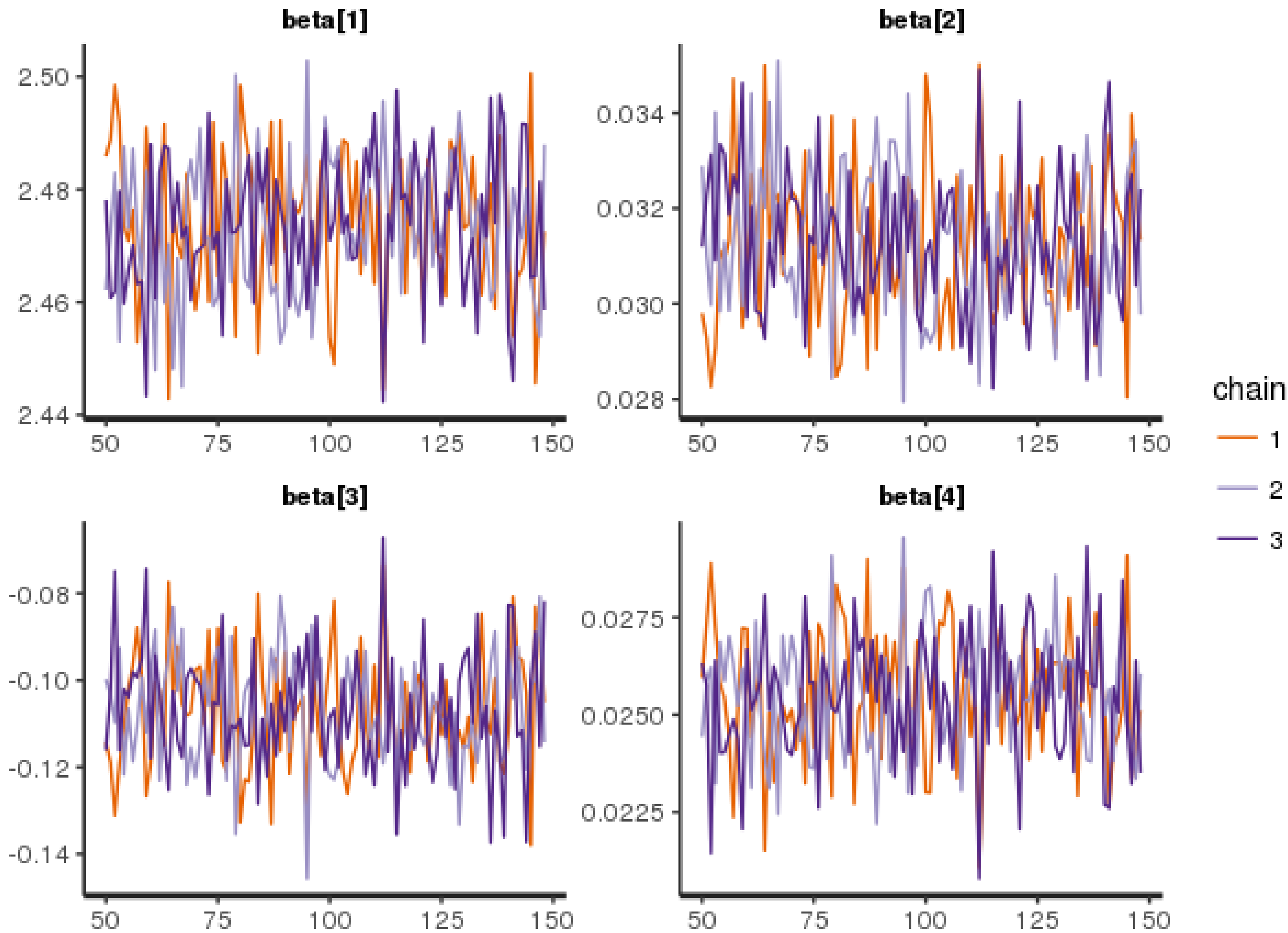

beta[1]
2.50
2.48
2.46
2.44
50
75
100
125
150
beta[2]
0.034
0.032
0.030
0.028
beta[3]
-0.08
-0.10
-0.12
-0.14
beta[4]
0.0275
0.0250
0.0225
chain
1
2
3

Figure 3.5