Section 3: Bayesian GLMs

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 $g^{-1} : \mathbb{R} \to (0, 1)$. 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 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`*.*

Solution:

$$\begin{aligned} p(z|x,y,\beta,\tau) &\propto p(y|z)p(z|\beta,X) \\ &= \prod_{i=1}^{N} p(y_i|z_i)p(z_i|\beta,x_i) \\ &\propto \begin{cases} N(z_i|x_i^T\beta,\tau)\mathbb{1}(z_i>0,y_i=1) \\ N(z_i|x_i^T\beta,\tau)\mathbb{1}(z_i\le 0,y_i=0) \end{cases} \end{aligned}$$

which is a truncated normal that can be sampled easily by rejection sampling. The Gibbs sampler is presented in the R script `exercise3_1.R`. Here we set $\tau = 1$ to get a better mixing and implement the sampler 5000 times. The traceplots of β are shown in Figure 1 with `burnin` $= 1000$. The coefficients with 77.985% accuracy are listed in Table 1:

Table 1: Estimated $\beta = (\beta_1, \beta_2, \beta_3, \beta_4, \beta_5, \beta_6, \beta_7, \beta_8, \beta_9)$

Coeffs	Intercept	Pregnant	Glucose	B Pressure	Skinfold	Insulin	BMI	Pedigree	Age
Estimate	-4.870	0.073	0.020	-0.008	0.001	-0.0007	0.053	0.494	0.010

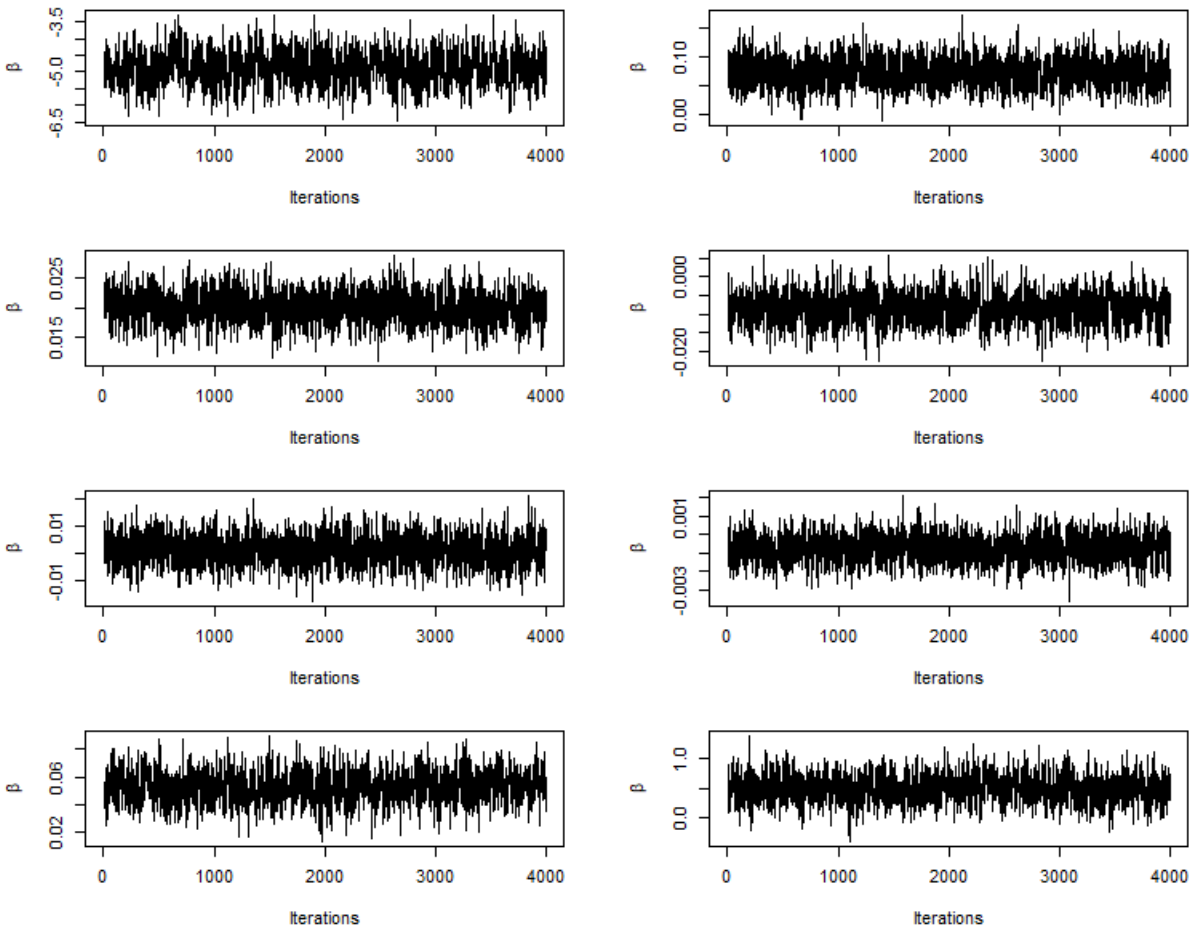

Figure 1: Traceplots of β

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 `Polson et al.(2013)`. 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.*

Solution: First, we estimate the posterior:

$$p(\beta|y,X) \propto p(\beta)p(y|\beta,x)$$

Using a $N(1,0)$ prior on β, we have

$$p(\beta|y,X) \propto e^{-\frac{1}{2}\beta^T\beta}\prod_{i=1}^{N} p(x_i)^{y_i}(1-p(x_i))^{1-y_i} \propto e^{-\frac{1}{2}\beta^T\beta}p(x)^y(1-p(x))^{(1-y)}$$

Now estimate the likelihood:

$$L(\beta|y_i) = \prod_{i=1}^{N} p(\beta)p(\beta|y_i) = e^{-\frac{1}{2}\beta^2}\prod_{i=1}^{N} p(x_i)^{y_i}(1-p(x_i))^{1-y_i} = e^{-\frac{1}{2}\beta^2}\prod_{i=1}^{N}\left(\frac{1}{1+e^{-x_i\beta}}\right)^{y_i}\left(\frac{e^{-x_i\beta}}{1+e^{-x_i\beta}}\right)^{1-y_i}$$

$$\log(L(\beta|y_i)) = -\frac{1}{2}\beta^2 + \sum_{i=1}^{N}\left[-y_i\log(1+e^{-x_i\beta}) + (1-y_i)(-x_i\beta) + (1-y_i)(-\log(1+e^{-x_i\beta}))\right]$$

$$\log(L(\beta|y_i)) = -\frac{1}{2}\beta^2 - \sum_{i=1}^{N}\left[(1-y_i)x_i\beta + \log(1+e^{-x_i\beta})\right]$$

Then we get,

$$\hat{\beta}_{MAP} = arg\min_{\beta}\left\{\frac{\beta^2}{2} + \sum_{i=1}^{N}\left[(1-y_i)x_i\beta + \log(1+e^{-x_i\beta})\right]\right\}$$

After implementing the R script `exercise3_2&3.R`, the estimate is $\hat{\beta} = -0.011$.

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).*

Solution: See the R script `exercise3_2&3.R` for the plot shown in Figure 2.

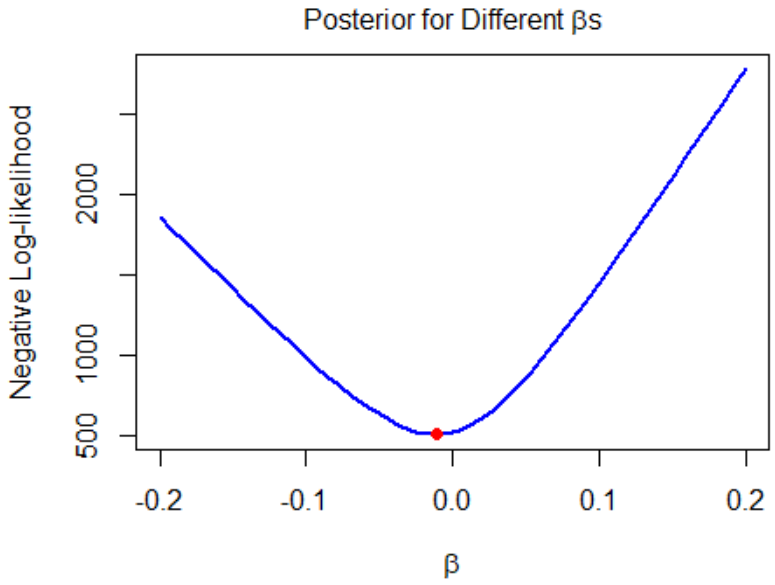

Figure 2: Negative Log-Likelihood vs. β's

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^*:

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

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.*

Solution:

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

The the mean an precision are obtained directly from the Gaussian form:

- $x* = \hat{\beta}_{MAP}$ because it is the global maximum of the distribution of beta.
- $c = -\frac{\delta^2}{\delta x^2} \log P^*(x)\Big|_{x=x^*}$, we need to estimate this.

[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.

$$c = -\frac{\delta^2}{\delta x^2} \log P^*(x)\Big|_{x=x^*} = 1 + \sum_{i=1}^{N} \frac{x_i^2 e^{-x_i\beta}}{(1+e^{-x_i\beta})^2}$$

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.

Solution:

$$\log p(\beta|X,\mathbf{y},\lambda) = \mathbf{y}^T \log\sigma(X\beta) + (1-\mathbf{y})^T \log(1-\sigma(X\beta)) - \frac{\lambda}{2}\beta^T\beta$$
$$\frac{\partial \log p(\beta|X,\mathbf{y},\lambda)}{\partial\beta} = X^T(\mathbf{y} - \sigma(X\beta)) - \lambda\beta$$
$$H = \frac{\partial^2 \log p(\beta|X,\mathbf{y},\lambda)}{\partial\beta\beta^T} = -X^T diag\left(\sigma(X\beta)(1-\sigma(X\beta))\right) - \lambda$$

So the prior can be approximated by a multivariate Gaussian with mean equal to the MAP value found, and covariance equal to the inverse of the negative Hessian:

$$\tilde{P}(\beta|x,\mathbf{y},\lambda) = \left(\frac{-H}{2\pi}\right)^{1/2} \exp\left\{-\frac{1}{2}(\beta-\beta_{MAP})^T(-H)(\beta-\beta_{MAP})\right\}$$

For interpretability, we have scaled the data (see the R script `exercise3_5.R`). The maximum a posterior estimates and 95% marginal credible interval are

Table 2: Maximum a posterior estimates and corresponding 95% marginal credible interval

Coefficient	MAP	Marginal Credible Interval
Intercept	-0.3469	[-0.4915, -0.2022]
Age	-0.1247	[-0.2704, 0.0211]

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

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

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) accross 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.*

Solution: From the histogram of `ACTIONS` shown in Figure 3, the distribution is highly skewed and there are several concerns:

- The support of the Poisson is the non-negative integers, and despite the data being censored, it is highly skewed to the right.
- In Poisson-distributed data the mean and variance should be identical, whereas here $E[x] = 15.93$ and $Var[x] = 460.91$. The data is highly over-dispersed, suggesting that a Negative Binomial fit might be a better choice.
- The mode of the data is 5, which is far from the mean.

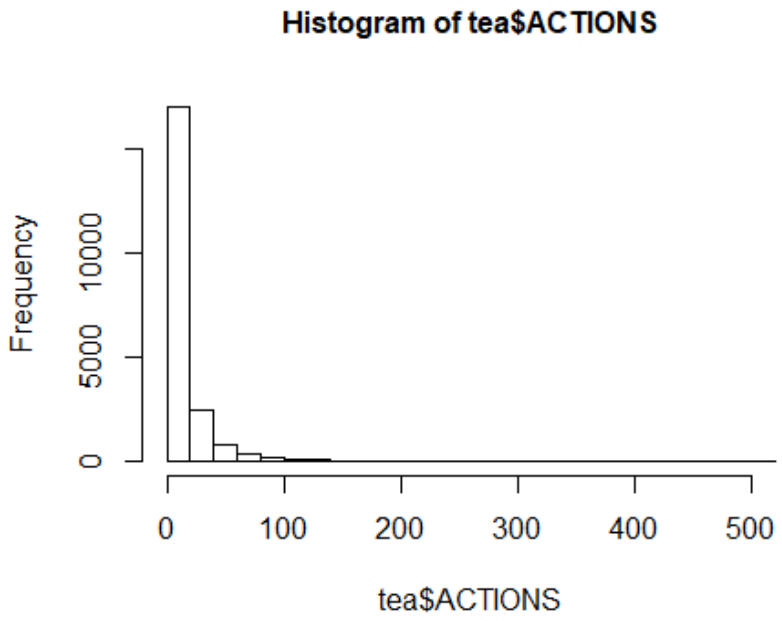

Figure 3: Histogram of `ACTIONS`

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.*

Solution: The Laplace approximation to the Poisson log posterior is

$$\log p(\beta|X, \mathbf{y}, \lambda) \propto \mathbf{y}^T X\beta - \sum_{i=1}^{n} e^{X\beta} - \frac{\lambda}{2}\beta^T\beta$$

$$\frac{\partial \log p(\beta|X, \mathbf{y}, \lambda)}{\partial \beta} = X^T(\mathbf{y} - e^{X\beta})) - \lambda\beta$$

$$\frac{\partial^2 \log p(\beta|X, \mathbf{y}, \lambda)}{\partial \beta \beta^T} = -(X^T diag(e^{X\beta})X + \lambda)$$

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

After implementing the code (see the R script `exercise3_6.R`). The maximum a posterior estimates and 95% marginal credible interval are

Table 3: Maximum a posterior estimates and corresponding 95% marginal credible interval

Coefficient	MAP	Marginal Credible Interval
Intercept	2.7575	[2.7541, 2.7609]
GRADE	0.1469	[0.1433, 0.1505]

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.html` 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* ***`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 script to include more variables, and present the results.*

Solution: The results are shown in Figure 4 and Figure 5. In Figure 4, Markov chains of the `Intercept` coefficient and `GRADE` coefficient are plotted and the result is consistent with the Laplacian approximation results. In Figure 5, we 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.

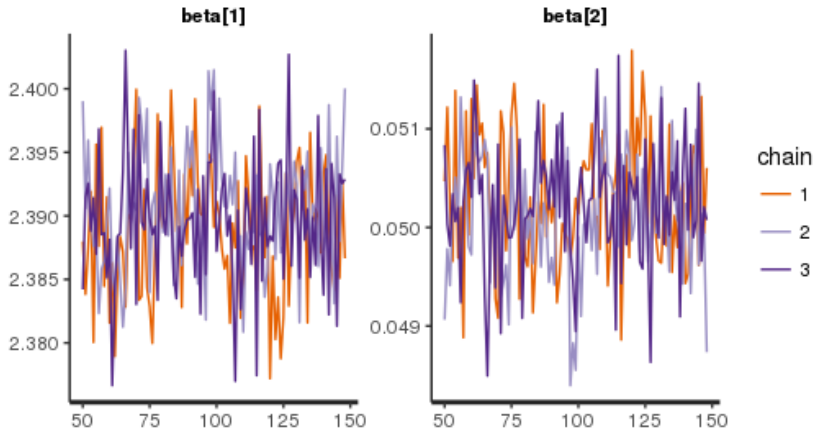

Figure 4: Traceplots of `Intercept` and `GRADE`

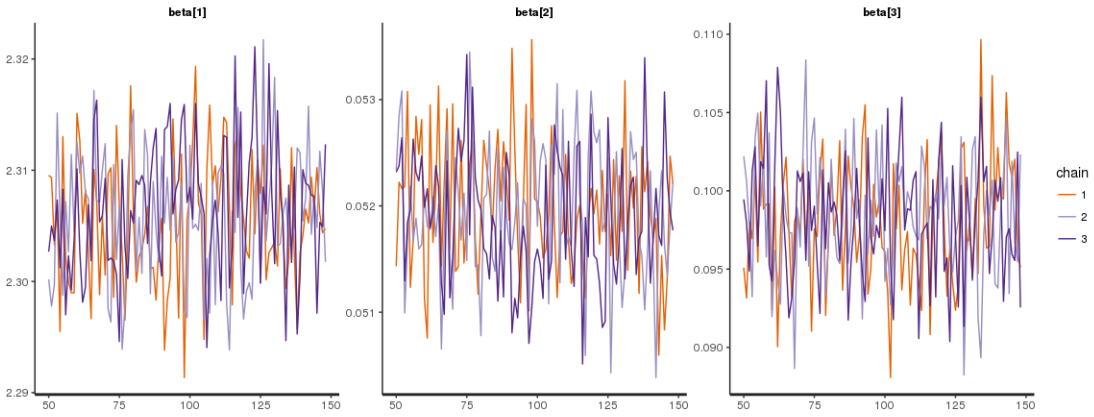

Figure 5: Traceplots of `Intercept`, `GRADE`, and `SEX`

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.*

Solution:We can take the interaction between `SEX` variable and `GRADE` variable into our model. After implementation, we have the posterior of the coefficients in the new model as shown in Figure 6. 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.

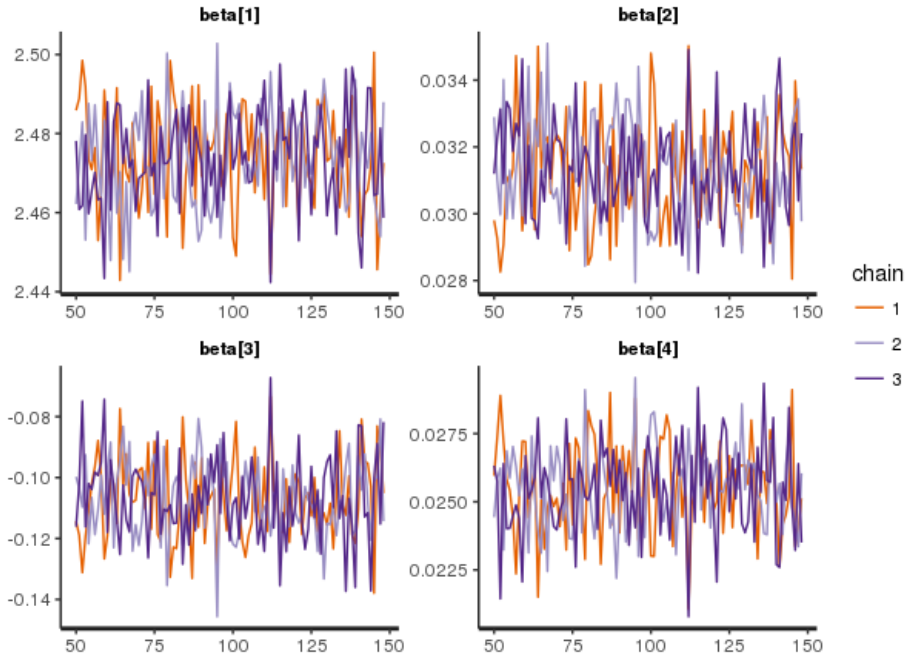

Figure 6: Traceplots of `Intercept`, `GRADE`, `SEX`, and interaction between `SEX` variable and `GRADE`

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).*

Solution: We can view data y_i as the truncated variable of some latent variable z_i as follows:

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

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