Advanced Regression Methods for Independent Data

STAT/BIOST 570, 2020

Computational Examples of Bayesian Model Fitting

Mauricio Sadinle
Department of Biostatistics
University of Washington
msadinle@uw.edu

The Normal Linear Model

- Remember the normal linear model:

$$\mathbf{Y} \mid \mathbf{X} \sim N(\mathbf{X}\boldsymbol{\beta}, \sigma^2 \mathbf{I}_n).$$

- Under this model we were able to obtain exact, finite sample frequentist inferences

- This model is also convenient for Bayesian inference, since it has a conjugate prior!

- The Normal-Inverse Gamma (NIG) prior is given by

$$p(\boldsymbol{\beta}, \sigma^2) = p(\boldsymbol{\beta} \mid \sigma^2) p(\sigma^2),$$

  where

$$\begin{aligned} \boldsymbol{\beta} \mid \sigma^2 &\sim \text{Normal}(\boldsymbol{\mu}_\beta, \sigma^2 V_\beta), \\ \sigma^2 &\sim \text{Inv-Gamma}(a, b), \end{aligned}$$

  where the last line is equivalent to $\sigma^{-2} \sim \text{Gamma}(a, b)$

The Normal Linear Model

- Using the NIG prior, we obtain a NIG posterior

$$\begin{aligned} \boldsymbol{\beta} \mid \sigma^2, \boldsymbol{y} &\sim \text{Normal}(\boldsymbol{\mu}^*, \sigma^2 V^*), \\ \sigma^2 \mid \boldsymbol{y} &\sim \text{Inv-Gamma}(a^*, b^*), \end{aligned}$$

  with

$$\begin{aligned} \boldsymbol{\mu}^* &= (V_\beta^{-1} + \mathbf{X}^T\mathbf{X})^{-1}(V_\beta^{-1}\boldsymbol{\mu}_\beta + \mathbf{X}^T\boldsymbol{y}), \\ V^* &= (V_\beta^{-1} + \mathbf{X}^T\mathbf{X})^{-1}, \\ a^* &= a + n/2, \\ b^* &= b + (\boldsymbol{\mu}_\beta^T V_\beta^{-1}\boldsymbol{\mu}_\beta + \boldsymbol{y}^T\boldsymbol{y} - \boldsymbol{\mu}^{*T} V^{*-1}\boldsymbol{\mu}^*)/2 \end{aligned}$$

- To check the "gory" details of these derivations, see the lecture notes of Sudipto Banerjee: http://www.biostat.umn.edu/~ph7440/pubh7440/BayesianLinearModelGoryDetails.pdf

- Inferences are pretty easy using the NIG prior, otherwise we need to rely on MCMC or other numerical methods

A Simple Normal Linear Model

We now illustrate with a simple example

- Consider the simple linear regression model

$$Y_i = \beta_0 + \beta_1 x_i + \epsilon_i,$$

with $\epsilon_i \mid \sigma^2 \sim_{iid} N(0, \sigma^2)$, $i = 1, ..., n$.

- From this, the likelihood function is obtained as

$$L(\beta_0, \beta_1, \sigma^2 \mid \boldsymbol{y}) \propto \frac{1}{(\sigma^2)^{n/2}} \exp\left[-\frac{1}{2\sigma^2} \sum_i (y_i - \beta_0 - \beta_1 x_i)^2 \right],$$

dropping terms that do not depend on parameters

A Simple Normal Linear Model

- Suppose the prior is of the form

$$p(\beta_0, \beta_1, \sigma^2) = p(\beta_0, \beta_1) \times p(\sigma^2)$$

where the prior for $\boldsymbol{\beta}$ is

$$\begin{bmatrix} \beta_0 \\ \beta_1 \end{bmatrix} \sim N\left(\boldsymbol{m} = \begin{bmatrix} m_0 \\ m_1 \end{bmatrix}, V = \begin{bmatrix} v_{00} & v_{01} \\ v_{01} & v_{11} \end{bmatrix}\right)$$

and the prior for σ^2 is

$$\sigma^2 \sim \text{Inv-Gamma}(a, b)$$

- Unlike the NIG prior, here $\boldsymbol{\beta}$ and σ^2 are independent apriori

- We rely on MCMC or other numerical methods to approximate the posterior

Conditional Posterior Distributions

We are still able to identify the conditional posterior distributions

- First, by properties of the multivariate normal we know

$$\beta_0 \mid \beta_1 \sim N\left(m_{0|1} = m_0 + \frac{v_{01}}{v_{11}}(\beta_1 - m_1), v_{0|1} = v_{00} - \frac{v_{01}^2}{v_{11}}\right)$$

- Based on this,

$$\begin{aligned} p(\beta_0 \mid \beta_1, \sigma^2, \boldsymbol{y}) &\propto L(\beta_0, \beta_1, \sigma^2 \mid \boldsymbol{y}) p(\beta_0 \mid \beta_1) \\ &\propto \exp\left[-\frac{1}{2\sigma^2}\sum_i (y_i - \beta_0 - \beta_1 x_i)^2\right] \times \exp\left[-\frac{1}{2v_{0|1}}(\beta_0 - m_{0|1})^2\right] \end{aligned}$$

after some algebra (which you'll need to figure out for HW4), we find that

$$\beta_0 \mid \beta_1, \sigma^2, \boldsymbol{y} \sim N\left(m_0^* = v_0^*\left(\frac{1}{\sigma^2}\sum_i (y_i - \beta_1 x_i) + \frac{m_{0|1}}{v_{0|1}}\right), v_0^* = \left(\frac{n}{\sigma^2} + \frac{1}{v_{0|1}}\right)^{-1}\right).$$

Conditional Posterior Distributions

- Similarly,

$$\beta_1 \mid \beta_0, \sigma^2, \boldsymbol{y} \sim N\left(m_1^* = v_1^*\left(\frac{1}{\sigma^2}\sum_i x_i(y_i - \beta_0) + \frac{m_{1|0}}{v_{1|0}}\right), v_1^* = \left(\frac{\sum_i x_i^2}{\sigma^2} + \frac{1}{v_{1|0}}\right)^{-1}\right).$$

- Finally, for σ^2, we have

$$p(\sigma^2 \mid \beta_0, \beta_1, \boldsymbol{y}) \propto L(\beta_0, \beta_1, \sigma^2 \mid \boldsymbol{y}) \times p(\sigma^2)$$

and we get

$$\sigma^2 \sim \text{Inv-Gamma}\left(a^* = n/2 + a, b^* = \frac{1}{2}\sum_i (y_i - \beta_0 - \beta_1 x_i)^2 + b\right)$$

Gibbs Sampler

We can now use these pieces to implement a Gibbs sampler as follows:

- Start with initial values $\beta_0^{(0)}, \beta_1^{(0)}, (\sigma^2)^{(0)}$ (e.g., taken to be the MLEs)

- For $t = 1, \ldots, T$ iterate:

  – $\beta_0^{(t+1)} \sim p(\beta_0 \mid \beta_1^{(t)}, (\sigma^2)^{(t)}, \boldsymbol{y})$

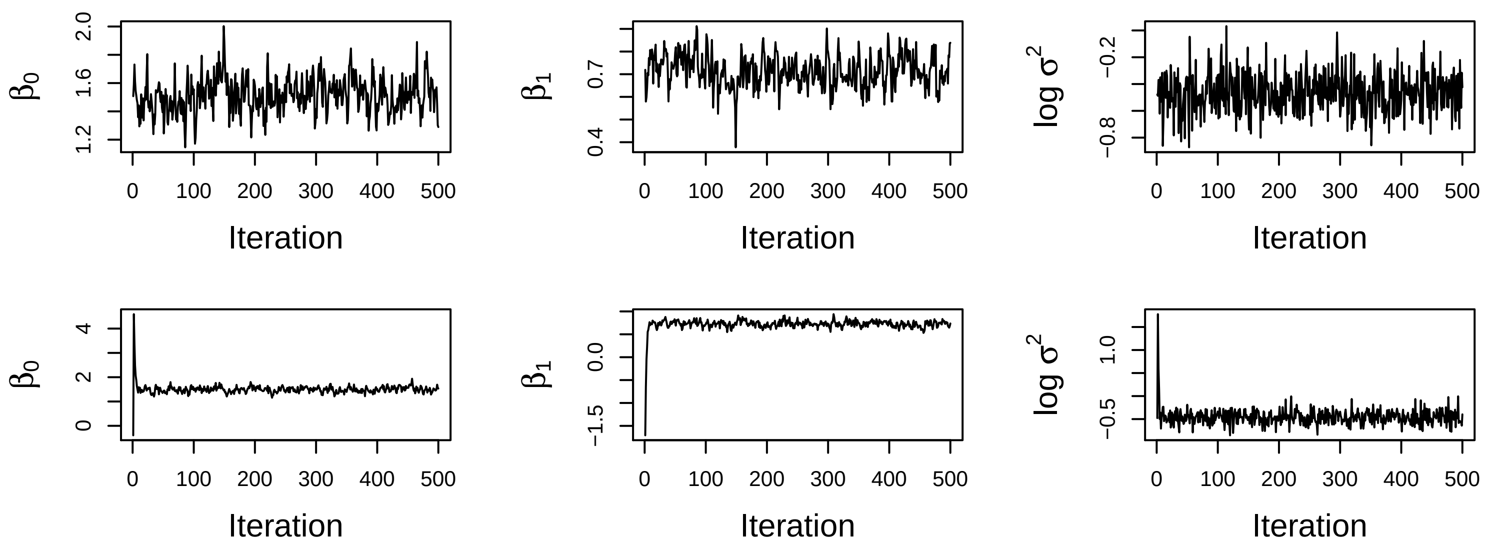
  – $\beta_1^{(t+1)} \sim p(\beta_1 \mid \beta_0^{(t+1)}, (\sigma^2)^{(t)}, \boldsymbol{y})$

  – $(\sigma^2)^{(t+1)} \sim p(\sigma^2 \mid \beta_0^{(t+1)}, \beta_1^{(t+1)}, \boldsymbol{y})$

Gibbs Sampler

- Using the `Prostate` data in the `R` package `lasso2`, we run the Gibbs sampler for the normal linear model $\log \text{PSA}_i = \beta_0 + \beta_1 \log(\text{cancer vol}_i) + \epsilon_i$

- We present the traceplots with the value of each $\beta_0^{(t)}, \beta_1^{(t)}, \log(\sigma^2)^{(t)}$ vs iteration t



First row: traceplots of Gibbs sampler started at MLEs.

Second row: traceplots of Gibbs sampler started at a random point sampled from the prior.

Practical Considerations

- *Burn-in period*: initial draws of the chain before visual convergence to be discarded

- It is also common to examine the *auto- and cross-correlation functions*

  – Given the paths of two time series $\{x_t\}_{t=0}^T$ and $\{y_t\}_{t=0}^T$, the *cross-covariance* of order s is defined as

$$\sigma_{xy}(s) = \frac{1}{T-s} \sum_{t=s}^{T} (x_{t-s} - \mu_x)(y_t - \mu_y)$$

  where μ_x and μ_y are the means of the time series

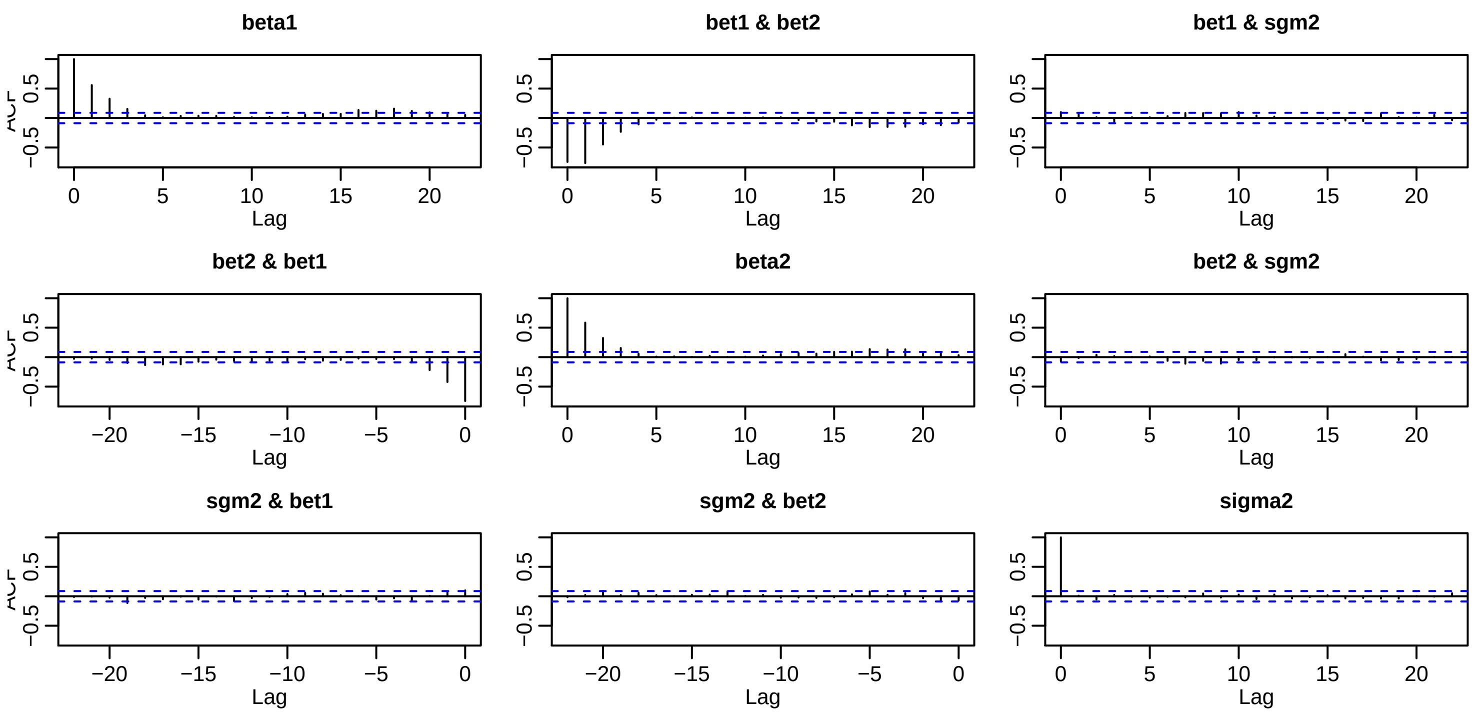
  – The *cross-correlation* is a normalized version

$$r_{xy}(s) = \frac{\sigma_{xy}(s)}{\sqrt{\sigma_{xx}(0)\sigma_{yy}(0)}}$$

  – The *auto-covariance* and *auto-correlation* of order s for time series $\{x_t\}_{t=0}^T$ are $\sigma_{xx}(s)$ and $r_{xx}(s)$, respectively

Practical Considerations

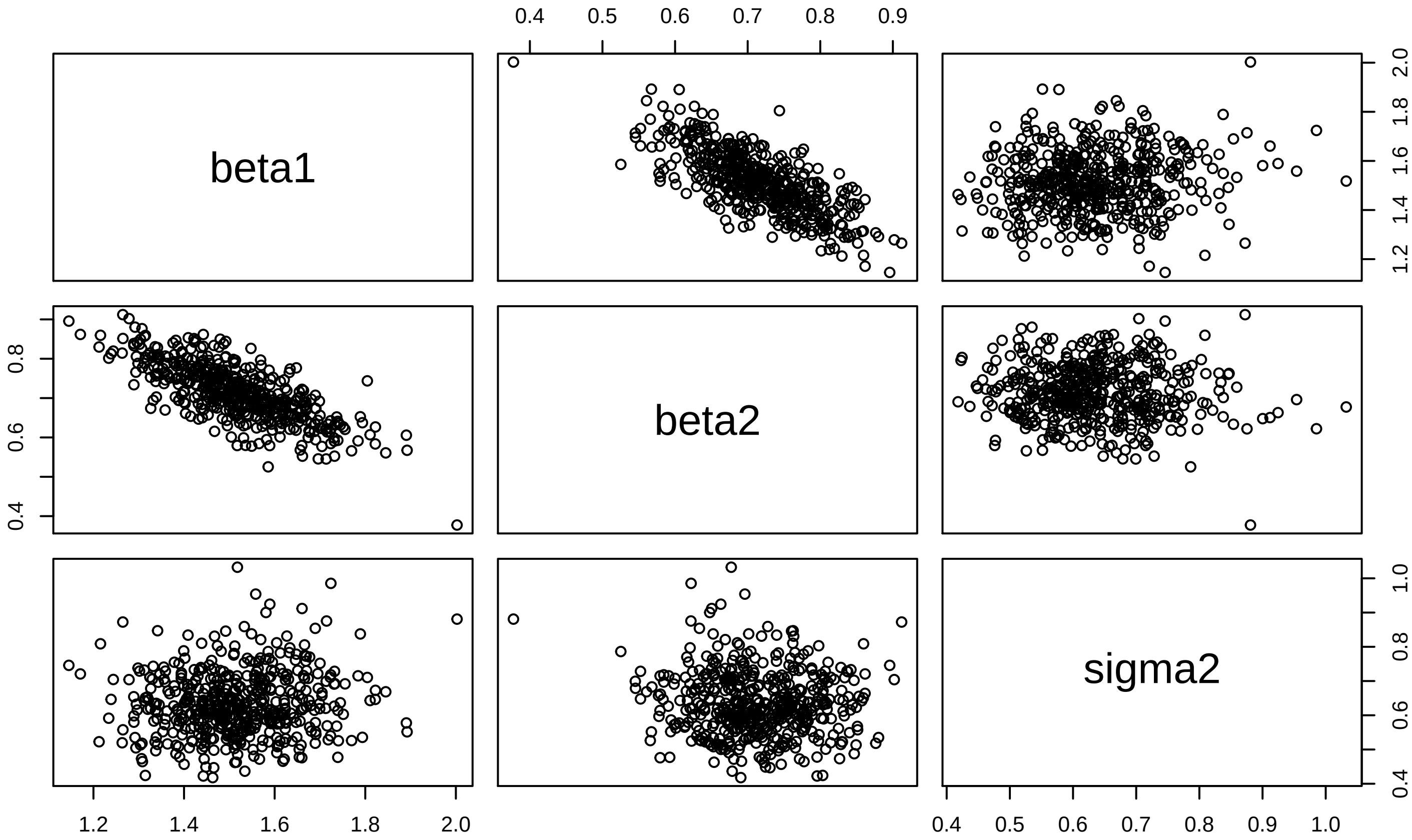
In R, use the `acf` function to obtain the auto- and cross-correlations:



According to this plot, roughly every five iterations of the Gibbs sampler we obtain an uncorrelated draw of β_0 and β_1: we can use this as an informal guidance for how long to run the chain

Practical Considerations

Posterior samples provide an approximation of the posterior distribution



Blocked Gibbs Sampler

Updating blocks of variables at a time often helps reduce the autocorrelation of the draws

- In HW4, you will show that in our example

$$\boldsymbol{\beta} \mid \boldsymbol{y}, \sigma^2 \sim N(\boldsymbol{m}^\star, V^\star),$$

where

$$\begin{aligned} \boldsymbol{m}^\star &= W \times \widehat{\boldsymbol{\beta}} + (\mathbf{I}_{k+1} - W) \times \boldsymbol{m}, \\ V^\star &= W \times \widehat{\text{var}}(\widehat{\boldsymbol{\beta}}), \end{aligned}$$

where $\widehat{\boldsymbol{\beta}}$ is the OLS estimator, and

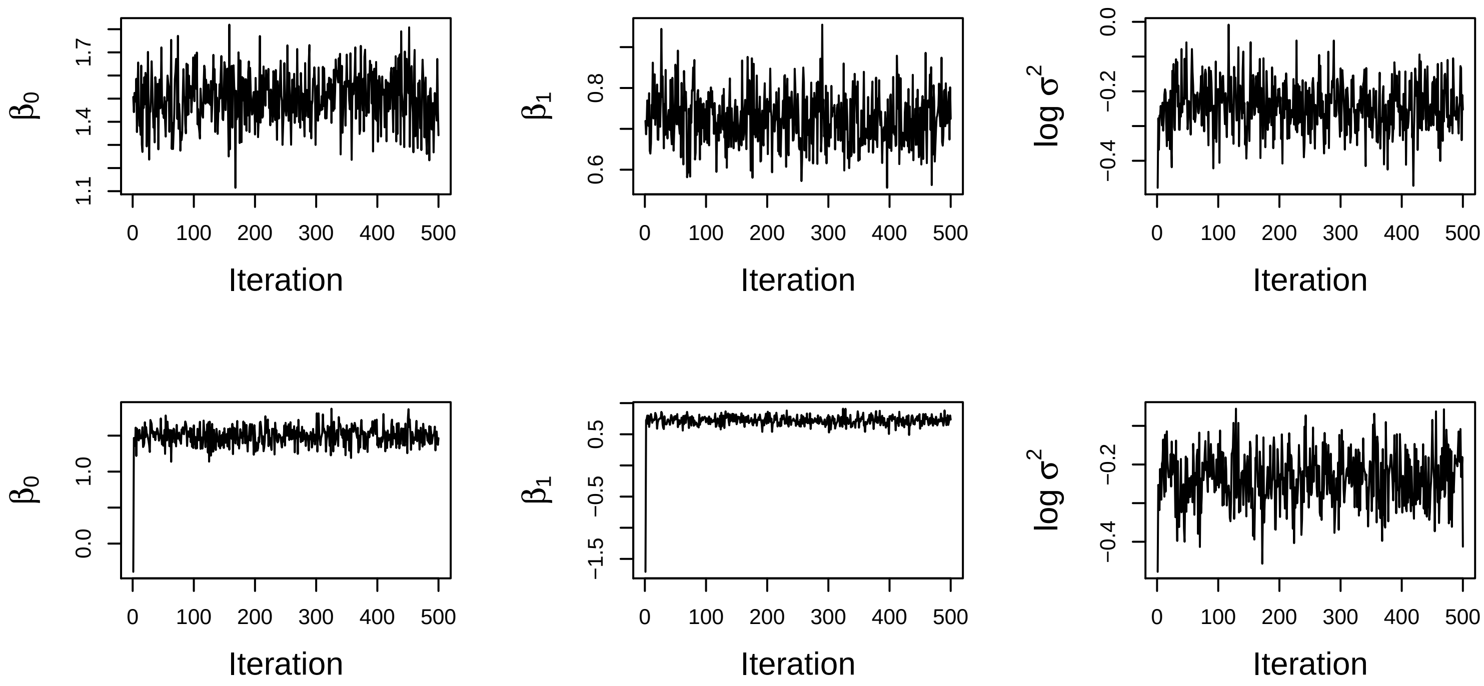
$$W = (\mathbf{X}^T\mathbf{X} + V^{-1}\sigma^2)^{-1}(\mathbf{X}^T\mathbf{X}),$$

and

$$\sigma^2 \mid \boldsymbol{y}, \boldsymbol{\beta} \sim \text{Inv-Gamma}\left(a + \frac{n}{2}, b + \frac{1}{2}(\boldsymbol{y} - \mathbf{X}\boldsymbol{\beta})^T(\boldsymbol{y} - \mathbf{X}\boldsymbol{\beta})\right)$$

Blocked Gibbs Sampler

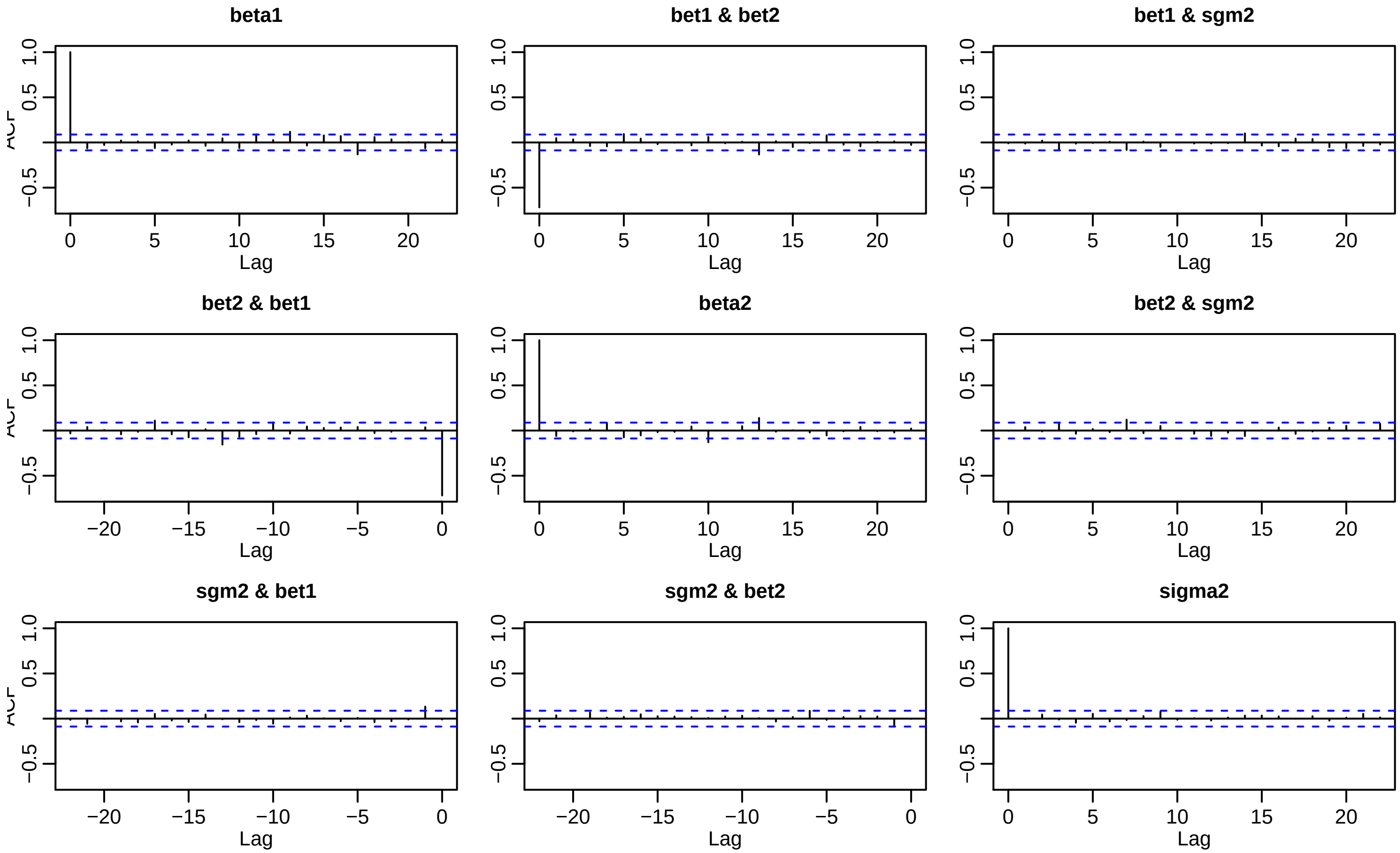
Here are the traceplots of this blocked Gibbs sampler



- First row: traceplots of blocked Gibbs sampler started at MLEs
- Second row: traceplots of blocked Gibbs sampler started at a random point from the prior

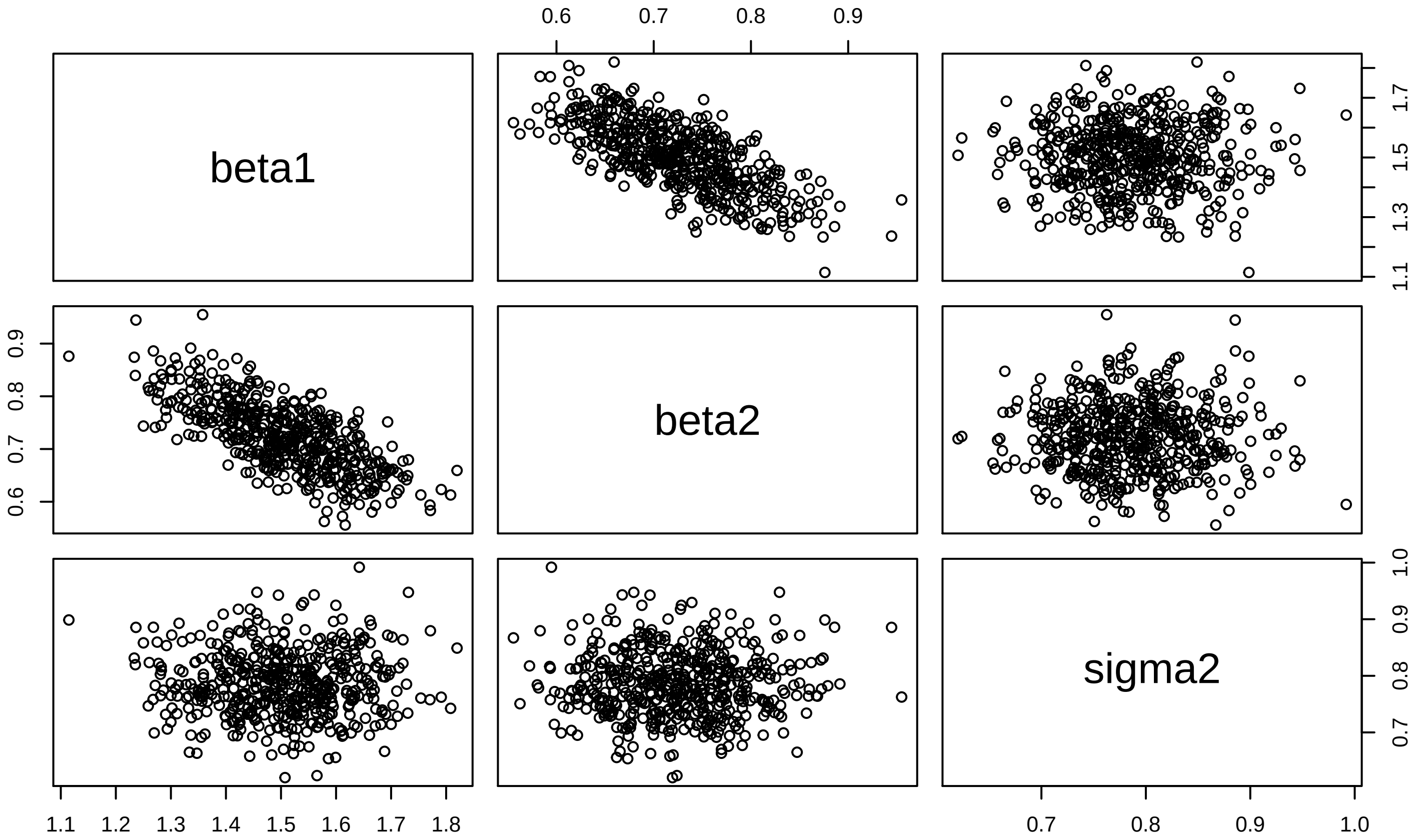
Blocked Gibbs Sampler

We obtain almost no auto-correlation in this case:



Blocked Gibbs Sampler

Scatterplot of the posterior samples:



Blocked Gibbs Sampler

We now run a longer Gibbs sampler to use for approximating functionals of the posterior

```
> Gibbs_samples <- blocked.gibbs.samp(
+          5000, y, X, beta.hat, sigma2.hat, m, V, a, b
+          )
> head(Gibbs_samples,3)

        beta1     beta2    sigma2
[1,] 1.507297 0.7193204 0.6201556
[2,] 1.465795 0.6595277 0.7632096
[3,] 1.361221 0.7515430 0.8197836

> Gibbs_samples <- Gibbs_samples[-c(1:500),] # discard burn-in period
```

Blocked Gibbs Sampler

We can now answer questions that would seem nonsensical from a frequentist point of view, such as *what is the probability that* $|\beta_1| > \delta$ for a value δ that measures practical significance

```
> mean( abs(Gibbs_samples[,"beta1"]) > 0.1 )

[1] 1
```

Some Final Comments

Here we emphasized implementing the algorithms ourselves to get a sense of how they work, but there are many *off the shelf* implementations of MCMC:

- The `R` packages `MCMC` and `MCMCpack`
- `WinBUGS`
- `OpenBUGS`
- Just Another Gibbs Sampler: `JAGS`
- `NIMBLE`
- `Stan`

Also, there are lots of topics and issues we didn't cover, but hopefully you have the background to pick those up on your own!