# Appendix: deriving the conditional posteriors for linear regression

*Jasper Ginn*

*February 18, 2019*

- https://stattrek.com/ for matrix algebra
- set up thinning in MCMC
- set up multiple chains in MCMC

## 1 Preliminaries

### 1.1 Recognizing a normal distribution

A random variable $x$ is normally distributed when $X \sim N(\mu, \sigma^2)$, or:

$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\left[\frac{(x-\mu)^2}{2\sigma^2}\right]} \tag{1.1.1}$$

Many terms in the equation have the function of a normalizing constant. That is, they make the distribution a proper distribution (adhering to the laws of probability). But in the Bayesian framework, we are not necessarily concerned with these normalizing constants.

In the case of the normal, we will drop any terms that do not contain the random variable $x$

$$\begin{aligned} f(x) &\propto e^{-\left[\frac{(x-\mu)^2}{2\sigma^2}\right]} \\ &\propto e^{-\left[\frac{(x^2-2x\mu+\mu^2)}{2\sigma^2}\right]} \\ &\propto e^{-\left[\frac{x^2}{2\sigma^2} - \frac{x\mu)}{\sigma^2}\right]} \end{aligned} \tag{1.1.2}$$

Notice that we can drop $\mu^2$ terms because, with respect to $x$, this term is just a constant. Put another way, by the exponent properties we know that $f(x) = e^{x+Q} = e^x e^Q \propto e^x$. This result leads us to a general form of a normal distribution

$$e^{\left[-Ax^2+Bx\right]} \tag{1.1.3}$$

where

- $A = \frac{1}{2\sigma^2}$
- $B = \frac{\mu}{\sigma^2}$

Hence, if we see $f(x) \sim e^{\left[-Ax^2+Bx\right]}$ with $A > 0$ then we should **recognize a normal distribution**. To derive its parameters $\mu$ and $\sigma^2$ in this form, we need to manipulate the distribution such that we get $(\mu, \sigma^2)$ from $(A, B)$.

$$\begin{aligned} A = \frac{1}{2\sigma^2} &\longrightarrow \sigma^2 = \frac{1}{2A} \\ B = \frac{\mu}{\sigma^2} &\longrightarrow \mu = B\sigma^2 = \frac{B}{2A} \end{aligned} \tag{1.1.4}$$

## 1.2 Recognizing an inverse gamma distribution

The *inverse gamma* distribution is given by

$$IG(x; \alpha, \beta) = \frac{\beta^\alpha}{\Gamma(\alpha)} x^{-\alpha-1} e^{-\frac{\beta}{x}} \tag{1.2.1}$$

Dropping all terms that do not contain $x$, we get

$$IG(x; \alpha, \beta) \propto x^{-\alpha-1} e^{-\frac{\beta}{x}} \tag{1.2.2}$$

We say a random variable $x$ has an inverse gamma distribution if $f(x) \sim x^A e^{\left[\frac{B}{x}\right]}$ with $B < 0, A < 0$. In this case, we can retrieve the shape parameter $\alpha$ and the scale parameter $\beta$ by calculating

$$\begin{aligned} \alpha &= -A - 1 \\ \beta &= -B \end{aligned} \tag{1.2.3}$$

## 1.3 Linear regression equation

The basic linear regression model with two predictors is given by

$$\hat{y_i} = \beta_0 + \beta_1 X_{1i} + \beta_2 X_{2i} + e_i \tag{1.3.1}$$

Where:

- $\hat{y_i}$ is the predicted value for the outcome variable for the $i^{th}$ individual
- $\beta_0$ is the intercept of the linear model
- $\beta_1, \beta_2$ are the slope coefficients for the linear model
- $X_{1i}, X_{2i}$ are the independent variables values for the $i^{th}$ individual
- $e_i$ is the residual error associated with the $i^{th}$ individual. We assume $e \sim N(0, \sigma^2)$

The parameters of this model are the intercept $\beta_0$, the slope coefficients $\beta_1, \beta_2$ and the residual variance $\sigma^2$.

As in any bayesian model, we are looking for the posterior

$$f(\text{parameters}|\text{data}) \propto f(\text{data}|\text{parameters}) f(\text{parameters}) \tag{1.3.2}$$

where:

- $f(\text{parameters}|\text{data})$ is the *joint posterior distribution* of the parameters
- $f(\text{data}|\text{parameters})$ is the *likelihood of the data* given the parameters
- $f(\text{parameters})$ is the *joint prior distribution* of the parameters

For the linear regression case, this yields

$$f(\beta_0, \beta_1, \beta_2, \sigma^2|y, X) \propto f(y|X, \beta_0, \beta_1, \beta_2, \sigma^2) f(\beta_0, \beta_1, \beta_2, \sigma^2) \tag{1.3.3}$$

We assume that the priors are independent, and as such, we can state that

$$f(\beta_0, \beta_1, \beta_2, \sigma^2) = f(\beta_0) f(\beta_1) f(\beta_2) f(\sigma^2) \tag{1.3.4}$$

# 2 Defining the likelihood of the data

Let $k_i = \beta_0 + \beta_1 X_{1i} + \beta_2 X_{2i}$. We assume that the outcome variable $y$ is normally distributed, or $y_i \sim N(k_i, \sigma^2)$. Hence, for the $i^{th}$ example in the data, we can represent the likelihood for this example as

$$f(y_i|x_{1i}, x_{2i}, \beta_0, \beta_1, \beta_2, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\left[\frac{(y_i - k_i)^2}{2\sigma^2}\right]} \tag{2.1.1}$$

By virtue of independence, we construct the likelihood of all $N$ examples in the data as

$$f(\text{data}|\text{parameters}) = f(y|x_1, x_2, \beta_0, \beta_1, \beta_2, \sigma^2) = \prod_{i=1}^{N} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\left[\frac{(y_i - k_i)^2}{2\sigma^2}\right]} \tag{2.1.2}$$

# 3 Specifying the prior distribution

For each of the parameters $\beta_0, \beta_1, \beta_2$, we assume a normal prior distribution. For example, the slope coefficient is assumed to be distributed as $f(\beta_0) \sim N(\mu_{0,0}, \tau_{0,0}^2)$, or:

$$f(\beta_0) = \frac{1}{\sqrt{2\pi\tau_{0,0}^2}} e^{-\left[\frac{(\beta_0 - \mu_{0,0})^2}{2\tau_{0,0}^2}\right]} \tag{3.1.1}$$

Where the hyperparameters $\tau_{0,0}^2, \mu_{0,0}$ are defined as:

- $\tau_{0,0}^2$ is the prior variance for the intercept
- $\mu_{0,0}$ is the prior mean for the intercept.

For the parameter $\sigma^2$, we assume an inverse gamma prior distribution.

$$IG(x; \alpha_0, \beta_0) = \frac{\beta_0^{\alpha_0}}{\Gamma(\alpha_0)} x^{-\alpha_0 - 1} e^{-\frac{\beta_0}{x}} \tag{3.1.2}$$

Where the hyperparameters $\alpha_0$ and $\beta_0$ are defined as:

- $\alpha_0$ is the prior shape of the distribution.
- $\beta_0$ is the prior scale of the distribution.

# 4 Deriving the conditional posteriors for each parameter

We can obtain posterior distributions for each of the parameters by constructing their *conditional posterior distributions*. Hence, we want to find

$$
\begin{aligned}
&f(\beta_0|y, X, \beta_1, \ldots, \beta_j, \sigma^2) \propto f(y|X, \beta_0, \beta_1, \ldots, \beta_j, \sigma^2) \times f(\beta_0) \\
&f(\beta_1|y, X, \beta_0, \ldots, \beta_j, \sigma^2) \propto f(y|X, \beta_0, \beta_1, \ldots, \beta_j, \sigma^2) \times f(\beta_1) \\
&\ldots \\
&f(\beta_j|y, X, \beta_0, \beta_1, \ldots, \beta_{j-1}, \sigma^2) \propto f(y|X, \beta_0, \beta_1, \ldots, \beta_j, \sigma^2) \times f(\beta_j) \\
&f(\sigma^2|y, X, \beta_0, \beta_1, \ldots, \beta_j) \propto f(y|X, \beta_0, \beta_1, \ldots, \beta_j, \sigma^2) \times f(\sigma^2)
\end{aligned}
\tag{4.1}
$$

## 4.1 The conditional distribution for the intercept $\beta_0$

Here, we are making the posterior conditional on all parameters other than $\beta_0$, and hence we are effectively turning the joint prior distribution into a single prior distribution.

Plugging in the likelihood and prior distribution into (6), we get

$$
f(\beta_0|\ldots) \propto \left[\prod_{i=1}^{N} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\left[\frac{(y_i-k_i)^2}{2\sigma^2}\right]}\right] \times \frac{1}{\sqrt{2\pi\tau_{0,0}^2}} e^{-\left[\frac{(\beta_0-\mu_{0,0})^2}{2\tau_{0,0}^2}\right]} \tag{4.1.1}
$$

To arrive at the conditional posterior for $\beta_0$, we simply drop all elements that do not contain $\beta_0$. We can do this because, in the conditional posterior distribution, such terms are nothing but normalization constants.

First, we drop the leading coefficients from (7)

$$
f(\beta_0|\ldots) \propto \left[\prod_{i=1}^{N} e^{-\left[\frac{(y_i-k_i)^2}{2\sigma^2}\right]}\right] \times e^{-\left[\frac{(\beta_0-\mu_{0,0})^2}{2\tau_{0,0}^2}\right]} \tag{4.1.2}
$$

Next, we examine each of the exponents separately and expand the factored quadratic in the numerator. If we forget about the product for a moment and focus on a single example of the likelihood, we see

$$
\begin{aligned}
&e^{-\left[\frac{(y_i-k_i)^2}{2\sigma^2}\right]} \longrightarrow \\
&\frac{(y_i-k_i)^2}{2\sigma^2} = \\
&\frac{1}{2\sigma^2}\left[y_i^2 - 2y_i k_i + k_i^2\right] = \\
&\frac{1}{2\sigma^2}\left[y_i^2 - 2y_i[\beta_0 + \beta_1 X_{1i} + \beta_2 X_{2i}] + (\beta_0 + \beta_1 X_{1i} + \beta_2 X_{2i})^2\right]
\end{aligned}
\tag{4.1.3}
$$

Expanding the terms and dropping all terms without $\beta_0$ yields

$$f(y|\dots) \propto \prod_{i=1}^{N} e^{\left[-\frac{\beta_0^2}{2\sigma^2}+\beta_0\frac{y_i-\beta_1 x_{1i}-\beta_2 x_{2i}}{\sigma^2}\right]} \tag{4.1.4}$$

If we repeat the above procedure for the prior distribution $f(\beta_0)$, we get

$$f(\beta_0) \propto e^{\left[-\frac{\beta_0^2}{2\tau_{0,0}^2}+\frac{\beta_0\mu_{0,0}}{\tau_{0,0}^2}\right]} \tag{4.1.5}$$

Recall the following exponent rules

- $e^a e^b = e^{[a+b]}$
- $\prod_{i=1}^{N} e^{[i+j]} = e^{\left[\left(\sum_{i=1}^{N} i\right)+\left(\sum_{i=1}^{N} j\right)\right]} = e^{\left[\left(\sum_{i=1}^{N} i\right)+Nj\right]}$

Applying these rules to (9) and (10) and factoring the exponent yields

$$\begin{aligned} f(\beta_0|\dots) &\propto e^{\left[-\frac{\beta_0^2 N}{2\sigma^2}+\beta_0\frac{\sum_{i=1}^{N} y_i-\beta_1 x_{1i}-\beta_2 x_{2i}}{\sigma^2}\right]} \times e^{\left[-\frac{\beta_0^2}{2\tau_{0,0}^2}+\frac{\beta_0\mu_{0,0}}{\tau_{0,0}^2}\right]} \\ &\propto e^{\left[-\frac{\beta_0^2 N}{2\sigma^2}+\beta_0\frac{\sum_{i=1}^{N} y_i-\beta_1 x_{1i}-\beta_2 x_{2i}}{\sigma^2}-\frac{\beta_0^2}{2\tau_{0,0}^2}+\frac{\beta_0\mu_{0,0}}{\tau_{0,0}^2}\right]} \\ &\propto e^{\left[-\beta_0^2\left(\frac{N}{2\sigma^2}+\frac{1}{2\tau_{0,0}^2}\right)+\beta_0\left(\frac{\sum_{i=1}^{N} y_i-\beta_1 x_{1i}-\beta_2 x_{2i}}{\sigma^2}+\frac{\mu_{0,0}}{\tau_{0,0}^2}\right)\right]} \end{aligned} \tag{4.1.6}$$

This we should recognize as the form

$$e^{\left[-Ax^2+Bx\right]} \tag{4.1.7}$$

Recall that:

$$\begin{aligned} A &= \frac{1}{2\sigma^2} \longrightarrow \sigma^2 = \frac{1}{2A} \\ B &= \frac{\mu}{\sigma^2} \longrightarrow \mu = B\sigma^2 = \frac{B}{2A} \end{aligned} \tag{4.1.8}$$

And so we get

$$\tau_{0,1}^2 \frac{1}{\left(\frac{N}{2\sigma^2} + \frac{1}{2\tau_{0,0}^2}\right)} \\ \mu_{0,1} \frac{\left(\frac{\sum_{i=1}^{N} y_i - \beta_1 x_{1i} - \beta_2 x_{2i}}{\sigma^2} + \frac{\mu_{0,0}}{\tau_{0,0}^2}\right)}{\left(\frac{N}{\sigma^2} + \frac{1}{\tau_{0,0}^2}\right)} \tag{4.1.9}$$

From (12) we can see the effect of the prior distribution. If the prior variance $\tau_{0,0}^2$ is large relative to the prior mean $\mu_{0,0}$, then the posterior distribution for $\mu_{0,1}$ will depend mainly on the likelihood of the data. In the case of the posterior $\tau_{0,1}^2$, we see that it reduces to

$$\tau_{0,1}^2 = \frac{1}{\frac{N}{2\sigma^2}} = \frac{2\sigma^2}{N} = \sqrt{2}\frac{\sigma}{\sqrt{N}} \tag{4.1.10}$$

which is simply the variance scaled by the number of examples in the data and similar to the standard error obtained by the central limit theorem.

## 4.2 The conditional distribution for the slope coefficients

The slope coefficients $\beta_1, \ldots, \beta_j$ are derived similarly to the intercept coefficient $\beta_0$. For the moment, assume that $\beta_j$ is the *last slope coefficient*. That is, assume that $j = \max(j)$

First, we define the posterior for $\beta_j$ as the product of the likelihood of the data and the prior distribution for $\beta_j$ and drop all terms in the prior distribution that do not contain this parameter

$$f(\beta_j|\beta_0, \beta_1, \ldots, \beta_j, \sigma^2) \propto \left[\prod_{i=1}^{N} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\left[\frac{(y_i - k_i)^2}{2\sigma^2}\right]}\right] \times \frac{1}{\sqrt{2\pi\tau_{j,0}^2}} e^{-\left[\frac{(\beta_j - \mu_{j,0})^2}{2\tau_{j,0}^2}\right]} \tag{1}$$

Next, we drop all terms that do not contain $\beta_j$ from the prior distribution

$$f(\beta_j|\ldots) \propto \left[\prod_{i=1}^{N} e^{-\left[\frac{(y_i - k_i)^2}{2\sigma^2}\right]}\right] \times e^{-\left[\frac{(\beta_j - \mu_{j,0})^2}{2\tau_{j,0}^2}\right]} \tag{1}$$

Just like in equation (4.1.3) we expand the part in green and keep all elements that contain $\beta_j$

$$
\begin{aligned}
(y_i - k_i)^2 &= y_i^2 - 2y_i k_i + k_i^2 \\
&= y_i^2 - 2y_i[\beta_0 + \beta_1 x_{1i} + \cdots + \beta_j x_{ji}] + (\beta_0 + \beta_1 x_{1i} + \cdots + \beta_j x_{ji})^2 \\
&\propto -2y_i\beta_j x_{ji} + \left[(\beta_0^2 + \beta_0\beta_1 x_{1i} + \cdots + \beta_0\beta_j x_{ji}) + \cdots + ([\beta_j x_{ji}]^2 + \beta_0\beta_j x_{ji} + \cdots + \beta_{j-1}x_{(j-1)i}\beta_j x_{ji})\right] \\
&\propto -2y_i\beta_j x_{ji} + (\beta_j x_{ji})^2 + 2\beta_0\beta_j x_{ji} + 2\beta_1 x_{1i}\beta_j x_{2i} + \cdots + 2\beta_{j-1}x_{(j-1)i}\beta_j x_{ji} \\
&\propto (\beta_j x_{ji})^2 - 2\beta_j x_{ji}(y_i - \beta_0 - \beta_1 x_{1i} - \cdots - \beta_{j-1}x_{(j-1)i})
\end{aligned}
$$

Plugging this back into equation XX and expanding the exponential containing the priors yields

$$
\begin{aligned}
f(\beta_j|\dots) &\propto \left[\prod_{i=1}^{N} e^{-\left[\frac{(\beta_j x_{ji})^2 - 2\beta_j x_{ji}(y_i - \beta_0 - \beta_1 x_{1i} - \cdots - \beta_{j-1}x_{(j-1)i})}{2\sigma^2}\right]}\right] \times e^{\left[-\frac{\beta_j^2}{2\tau_{j,0}^2} + \frac{\beta_j \mu_{j,0}}{\tau_{j,0}^2}\right]} \\
&\propto \left[\prod_{i=1}^{N} e^{\left[-\frac{(\beta_j x_{ji})^2}{2\sigma^2} + \beta_j x_{ji}\frac{y_i - \beta_0 - \beta_1 x_{1i} - \cdots - \beta_{j-1}x_{(j-1)i}}{\sigma^2}\right]}\right] \times e^{\left[-\frac{\beta_j^2}{2\tau_{j,0}^2} + \frac{\beta_j \mu_{j,0}}{\tau_{j,0}^2}\right]}
\end{aligned}
$$

Applying exponent properties to YY and combining the terms yields

$$
\begin{aligned}
f(\beta_j|\dots) &\propto e^{\left[-\frac{\sum_i (\beta_j x_{ji})^2}{2\sigma^2} + \beta_j x_{ji}\frac{\sum_i (y_i - \beta_0 - \left[\sum_{k=1}^{j-1}\beta_k x_{ki}\right])}{\sigma^2}\right]} \times e^{\left[-\frac{\beta_j^2}{2\tau_{j,0}^2} + \frac{\beta_j \mu_{j,0}}{\tau_{j,0}^2}\right]} \\
&\propto e^{\left[\left\{-\frac{\sum_i (\beta_j)^2 (x_{ji})^2}{2\sigma^2} - \frac{\beta_j^2}{2\tau_{j,0}^2}\right\} + \left\{\beta_j x_{ji}\frac{\sum_i (y_i - \beta_0 - \left[\sum_{k=1}^{j-1}\beta_k x_{ki}\right])}{\sigma^2} + \frac{\beta_j \mu_{j,0}}{\tau_{j,0}^2}\right\}\right]} \\
&\propto e^{\left[-\beta_j^2\left\{\frac{\sum_i x_{ji}^2}{2\sigma^2} + \frac{1}{2\tau_{j,0}^2}\right\} + \beta_j\left\{\frac{x_{ij}\sum_i (y_i - \beta_0 - \left[\sum_{k=1}^{j-1}\beta_k x_{ki}\right])}{\sigma^2} + \frac{u_{j,0}}{\tau_{j,0}^2}\right\}\right]}
\end{aligned}
$$

Again, we should recognize this as the form

$$
e^{\left[-Ax^2 + Bx\right]} \tag{4.1.7}
$$

Recall that:

$$
\begin{aligned}
A = \frac{1}{2\sigma^2} &\longrightarrow \sigma^2 = \frac{1}{2A} \\
B = \frac{\mu}{\sigma^2} &\longrightarrow \mu = B\sigma^2 = \frac{B}{2A}
\end{aligned} \tag{4.1.8}
$$

And so we get

$$\tau_{j,1}^2 = \frac{1}{\left(\frac{\sum_i x_{ji}^2}{2\sigma^2} + \frac{1}{2\tau_{j,0}^2}\right)}$$
$$\mu_{j,1} = \frac{\left(\frac{x_{ji}\sum_i (y_i - \beta_0 - \left[\sum_{k=1}^{j-1} \beta_k x_{ki}\right])}{\sigma^2} + \frac{u_{j,0}}{\tau_{j,0}^2}\right)}{\left(\frac{N x_{ji}^2}{\sigma^2} + \frac{1}{\tau_{0,0}^2}\right)} \tag{4.1.9}$$

Finally, we need to generalize this result to cases where $j \neq \max(j)$. Assume that we have $j = 8$ parameters, then the above equation works fine if $j = 8$. For $j < 8$ it does not work. For example, if we have $j = 4$, then the posterior mean $\mu_{4,1}$ does not depend on all coefficients except $\beta_4$ (which is what we desire). Rather, it depends on $\beta_1, \beta_2, \beta_3$ only. The simplest way to get around this problem is to rewrite $B$ such that we exclude the $j^{th}$ coefficient in the summation

$$\mu_{j,1} \frac{\left(\frac{\sum_i x_{ji}(y_i - \beta_0 - \left[\sum_{k\neq j} \beta_k x_{ki}\right])}{\sigma^2} + \frac{u_{j,0}}{\tau_{j,0}^2}\right)}{\left(\frac{N x_{ji}^2}{\sigma^2} + \frac{1}{\tau_{j,0}^2}\right)}$$

## 4.3 The conditional distribution for $sigma^2$

The process to find the posterior for $\sigma^2$ is the same as for the previous parameters, with the difference being the prior $f(\sigma^2)$, which is inverse-gamma distributed. Again, we drop all terms that do not contain the parameter $\sigma^2$, which yields

$$\begin{aligned} f(\sigma^2|\dots) &\propto \left[\prod_{i=1}^{N} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\left[\frac{(y_i - k_i)^2}{2\sigma^2}\right]}\right] \times \frac{\beta_0^{\alpha_0}}{\Gamma(\alpha_0)} x^{-\alpha_0 - 1} e^{-\frac{\beta_0}{\sigma^2}} \\ &\propto \frac{N}{\sqrt{\sigma^2}} e^{-\left[\frac{\sum_i (y_i - k_i)^2}{2\sigma^2}\right]} \times x^{-\alpha_0 - 1} e^{-\frac{\beta_0}{\sigma^2}} \end{aligned} \tag{13}$$

Recall that $k_i = \beta_0 + \beta_1 X_{1i} + \beta_2 X_{2i}$. It is helpful to redefine the part in orange in the equation above such that

$$S = \sum_{i=1}^{N} (y_i - k_i)^2 \tag{14}$$

Furthermore, we take the $1/2$ from the denominator in the exponent and divide $S$ by this constant. (look at the preliminary section on the inverse gamma distribution to see why). Hence, we get

$$f(\sigma^2|\dots) \propto \frac{N}{\sqrt{\sigma^2}} e^{-\left[\frac{S/2}{\sigma^2}\right]} \times (\sigma^2)^{-\alpha_0-1} e^{-\frac{\beta_0}{\sigma^2}} \tag{14}$$

$$k_i = \beta_0 + \beta_1 x_{1i} + \cdots + \beta_j x_{ji}$$

Now, collecting the like terms yields

$$\begin{aligned} f(\sigma^2|\dots) &\propto \left[(\sigma^2)^{-\frac{N}{2}} (\sigma^2)^{-\alpha_0-1}\right] \times \left[e^{-\left[\frac{S/2}{\sigma^2}\right]} e^{-\frac{\beta_0}{x}}\right] \\ &\propto (\sigma^2)^{[-\frac{N}{2}-\alpha_0-1]} e^{\left[-\frac{(S/2+\beta_0)}{\sigma^2}\right]} \end{aligned} \tag{14}$$

Which you should recognize as an inverse gamma distribution with $A = -\frac{N}{2} - \alpha_0 - 1$ and $B = -(S/2 + \beta_0)$. We can retrieve the posterior shape $\alpha_1$ and scale $\beta_1$ by using

$$\begin{aligned} \alpha_1 &= -A - 1 = -(-\frac{N}{2} - \alpha_0 - 1) - 1 = \frac{N}{2} + \alpha_0 \\ \beta_1 &= -B = -(-(S/2 + \beta_0)) = \frac{S}{2} + \beta_0 \end{aligned}$$

# 5 Posterior functions in R

## 5.1 Posterior distributions in vector notation

The notation we used for the posteriors can become a little unwieldy when we implement it in $\mathbb{R}$. This has to do with the summations in the calculations for the posterior means of the coefficients and intercepts. If we would implement it as is, we would have to create a for loop in which we loop over the $j$ parameters.

Alternatively, we can use some linear algebra to simplify the equations. Recall that $k_i = \beta_0 + \beta_1 x_{1i} + \cdots + \beta_j x_{ji}$. This is a system of linear equations for $n$ examples. Accordingly, we can rewrite $k_i$ as

$$\vec{k} = \mathbf{X}\vec{w}$$

Where

- $\vec{k}$ is a column vector of length $n$
- $\mathbf{X}$ is a matrix with $n$ rows and $m$ columns

- $\vec{w}$ is a column vector of length $m$

$$\begin{bmatrix} k_1 \\ k_2 \\ \vdots \\ k_n \end{bmatrix} = \begin{bmatrix} 1 & x_{11} & x_{12} & \dots & x_{1m} \\ 1 & x_{21} & x_{22} & \dots & x_{2m} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n1} & x_{n2} & \dots & x_{nm} \end{bmatrix} \cdot \begin{bmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_j \end{bmatrix} = \begin{bmatrix} \beta_0 + \beta_1 x_{11} + \beta_2 x_{12} + \cdots + \beta_j x_{1m} \\ \beta_0 + \beta_1 x_{21} + \beta_2 x_{22} + \cdots + \beta_j x_{2m} \\ \dots \\ \beta_0 + \beta_1 x_{n1} + \beta_2 x_{n2} + \cdots + \beta_j x_{nm} \end{bmatrix}$$

Note that the subscripts in $x$ have switched to accomodate the rows $\times$ columns notation that is usual in linear algebra. Instead of $x_{\text{coefficient}\times\text{observation}} = x_{ji}$ we now have $x_{\text{observation}\times\text{coefficient}} = x_{ij} = x_{nm}$.

We will use the following notation to denote that the $j^{th}$ *column* of the design matrix $\mathbf{X}$ and the $j^{th}$ row of the coefficient matrix $\vec{w}$ have been removed

$$\vec{k}_{*j} = \mathbf{X}^{-j}\vec{w}_{-j}$$

We will use $x_j$ to denote the $j^{th}$ column vector from the design matrix $\mathbf{X}$

### 5.1.1 Intercept and slope posteriors using linear algebra notation

When we examine the posterior means for the intercept and slope coefficients, we should recognize that we can rewrite both equations as one using vector notation.

$$\mu_{j,1} = \frac{1}{\frac{\vec{x}_j^T \cdot \vec{x}_j}{\sigma^2} + \frac{1}{\tau_{j,0}^2}} \times \left( \frac{\sum \vec{x}_j \circ [\vec{y} - \vec{k}_{*j}]}{\sigma^2} + \frac{u_{j,0}}{\tau_{j,0}^2} \right)$$

Where:

- $\vec{x}_j^T \vec{x}_j$ is the sum of the squared elements of $\vec{x}_j$
- $X^{-j}$ is the design matrix without the $j^th$ column corresponding to $\beta_j$
- $w_{-j}$ is the weight matrix without the $j_th$ coefficient

For the variance, we get

$$\tau_{j,1}^2 = \frac{1}{\left( \frac{\vec{x}_j^T \cdot \vec{x}_j}{2\sigma^2} + \frac{1}{2\tau_{j,0}^2} \right)}$$

## 5.2 R Functions for the posterior distributions

Create some data

```
rm(list=ls())
n <- 1000 # Number of examples
j <- 5 # Number of coefficients
# Empty matrix
X <- matrix(0L, ncol=j, nrow=n)
# Populate
for(i in 1:j) {
  if(i != 3) {
    X[,i] <- runif(n, -1, 5)
  } else if(i == 5) {
    X[,i] <- runif(n, 1, 2)^2
  } else {
    X[,i] <- rbinom(n, 1, 0.5)
  }
}
# Add intercept
X <- cbind(rep(1, n), X)
# Coefficients
b0 <- 3.4
b1 <- -1.5
b2 <- 1.7
b3 <- 3 # This will be a dummy var
b4 <- 1.2
b5 <- 1.1 # Polynomial
sigma <- 0.3
# Create y
y <- rnorm(n, mean = b0 + b1*X[,2] + b2*X[,3] + b3*X[,4] + b4*X[,5] + b5*X[,6],
           sd = 1 / sqrt(sigma))
# List of real values (for later)
```

```
real <- list(
  b0,b1,b2,b3,b4,b5,sigma
)
```

We want to make sure to center the data to remove autocorrelation

```
X[,2:6] <- apply(X[,2:6], 2, function(x) x - mean(x))
```

We run a normal linear regression model for comparison purposes

```
# Quick linear regression
lrd <- data.frame(cbind(y, X[,-1]))
linreg <- lm("y ~ V2 + V3 + V4 + V5 + I(V6^2)", data=lrd)
summary(linreg)
```

```
##
## Call:
## lm(formula = "y ~ V2 + V3 + V4 + V5 + I(V6^2)", data = lrd)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -7.4600 -1.8716  0.0986  1.7506  7.9168
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  9.76704    0.12233  79.844   <2e-16 ***
## V2          -1.56155    0.04675 -33.404   <2e-16 ***
## V3           1.65972    0.04683  35.444   <2e-16 ***
## V4           3.18193    0.16545  19.232   <2e-16 ***
## V5           1.11775    0.04671  23.930   <2e-16 ***
## I(V6^2)      0.05826    0.03156   1.846   0.0651 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 2.602 on 994 degrees of freedom
## Multiple R-squared:  0.7762, Adjusted R-squared:  0.7751
## F-statistic: 689.4 on 5 and 994 DF,  p-value: < 2.2e-16
```

The posterior for the intercept mean and variance are

```r
# Posterior mean helper function
posterior_mu <- function(X, xj, y, w, sigma, mu0, tau0) {

  # Numerator
  num1 <- sum(xj * (y - X %*% w)) / sigma
  num2 <- mu0 / tau0

  # Denominator
  den1 <- t(xj) %*% xj / sigma
  den2 <- 1 / tau0

  # Return
  return( ( num1 + num2 ) / ( den1 + den2 ) )

}

# Posterior tau helper function
posterior_tau <- function(xj, sigma, tau0) {

  return( 1 / ( (t(xj) %*% xj/sigma) + (1/tau0) ) )

}

# Wrapper for the helper functions
posterior_coef <- function(X, y, w, j, priors, sigma) {

  # Rows and columns
```

```r
  n <- nrow(X)
  m <- ncol(X)

  # Extract vectors
  xj <- matrix(X[,(j+1)])
  wj <- w[(j+1),]

  # Remove the jth column from X and row from w
  w <- matrix(w[-(j+1),])
  X <- matrix(X[,-(j+1)], nrow=n, ncol=m-1)

  # Calculate mu
  mu1 <- posterior_mu(X, xj, y, w, sigma, priors$mu, priors$tau)

  # Calculate tau
  tau1 <- posterior_tau(xj, sigma, priors$tau)

  # Return
  # (in reality, create an object)
  return(list(
    "mu_posterior" = mu1,
    "tau_posterior" = tau1
  ))

}

# Calculate posterior sigma
posterior_sigma <- function(X, y, w, priors) {

  # Get number of data points
  n <- nrow(X)
```

```
  # Calculate alpha1
  alpha1 <- (n / 2) + priors$alpha


  # Calculate beta1
  beta1 <- (sum((y - X %*% w)^2) / 2) + priors$beta


  # Return
  return(
    list(
      "alpha_posterior" = alpha1,
      "beta_posterior" = beta1
    )
  )


}
```

The code below defines the starting values and priors and samples the posterior 10.000 times.

```
# Starting values
sigma <- 0.1
b0 <- 0
b1 <- 0
b2 <- 0
b3 <- 0
b4 <- 0
b5 <- 0
# Priors
priors <- list(
  "b0" = list("mu"=0, "tau"=1000),
  "b1" = list("mu"=0, "tau"=1000),
  "b2" = list("mu"=0, "tau"=1000),
  "b3" = list("mu"=0, "tau"=1000),
  "b4" = list("mu"=0, "tau"=1000),
```

```
  "b5" = list("mu"=0, "tau"=1000),
  "sigma" = list("alpha"=0.001, "beta"=0.001)
)
# To column vector
w <- matrix(c(b0,b1,b2,b3,b4,b5), ncol=1)
# Iterations
k <- 10000
# Matrix to store data
results <- matrix(0L, nrow=k, ncol = length(priors))
# For each
for(i in 1:(k+1)) {

  # Save values from previous iteration
  if(i > 1) {
    results[i-1,1:6] <- w[,1]
    results[i-1,7] <- 1/sqrt(sigma)
  }

  # Update the params
  for(j in 0:5) {

    # Get posterior
    posterior_values <- posterior_coef(X, y, w, j, priors[[paste0("b", j)]], sigma)

    # Draw from posterior
    w[j+1,1] <- rnorm(1, posterior_values$mu_posterior, posterior_values$tau_posterior)

  }

  # Get posterior for sigma
  posterior_sigma_values <- posterior_sigma(X, y, w, priors[["sigma"]])
```

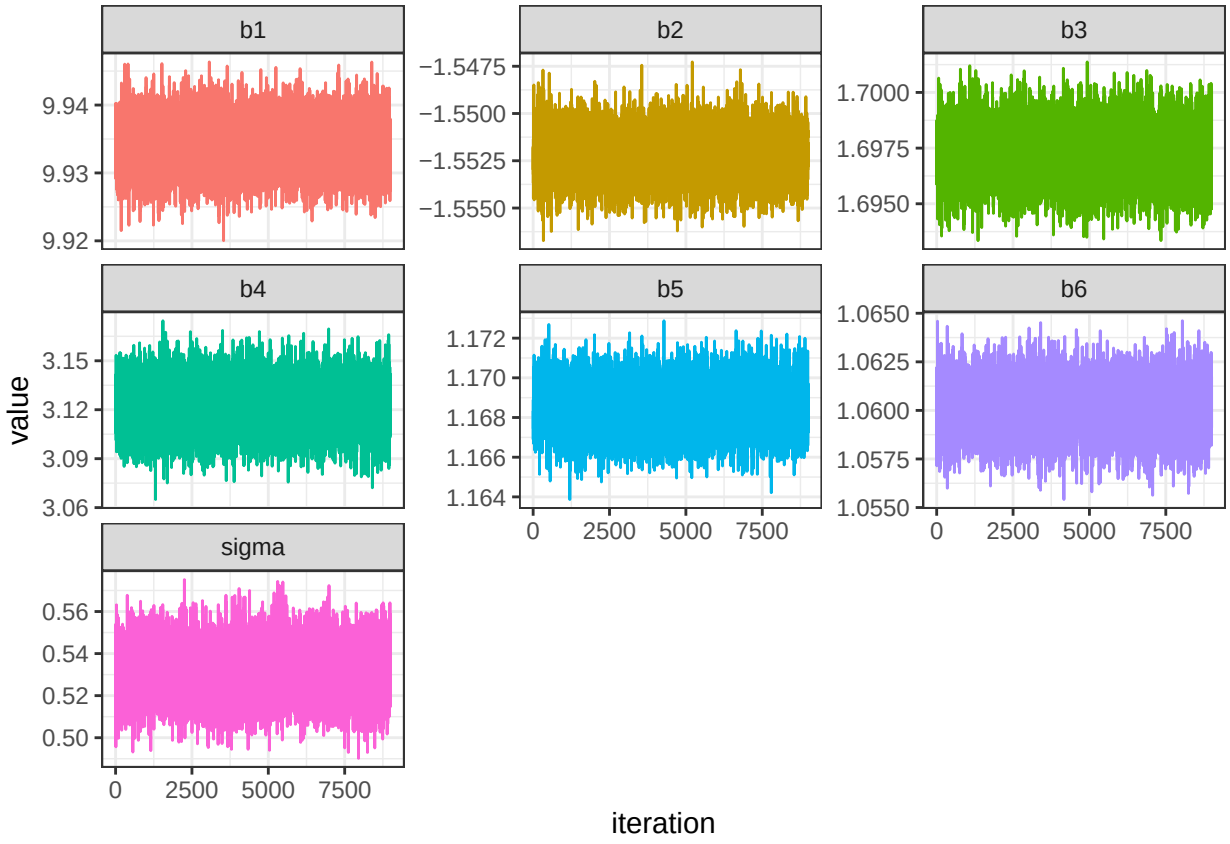
```
  # Draw
  sigma <- 1 / rgamma(1, posterior_sigma_values$alpha_posterior,
                      posterior_sigma_values$beta_posterior)

}

# Remove the first 1.000 observations (burn-in period)
results <- results[-1:-1000,]
```

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(tidyr)
library(ggplot2)
library(forcats)

results %>%
  as_tibble() %>%
  mutate(iteration = 1:n()) %>%
  gather(variable, value, -iteration) %>%
  mutate(variable = factor(variable, labels = c(paste0("b", 1:6), "sigma"))) %>%
  ggplot(aes(x=iteration, y=value, group=variable, color=variable)) +
    geom_line() +
    theme_bw() +
```

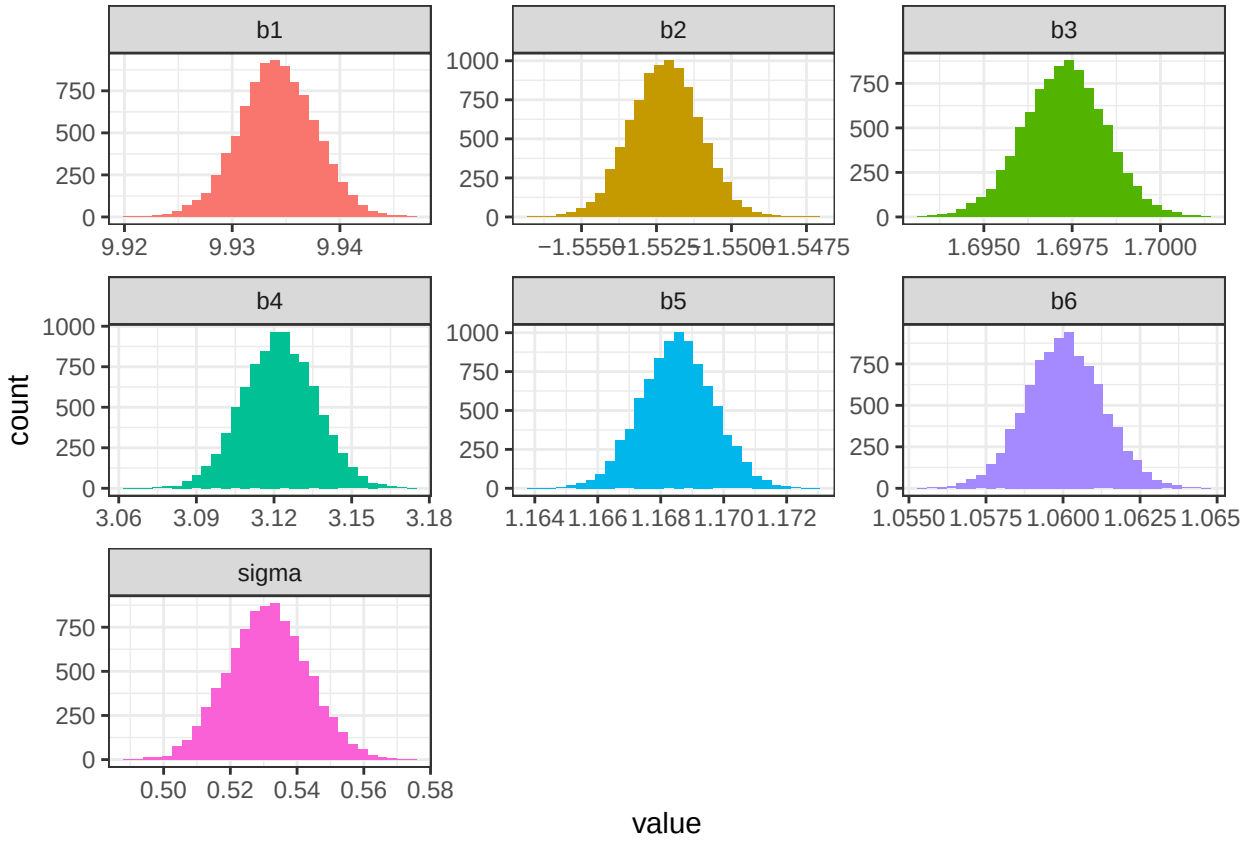
```
    theme(legend.position = "None") +
    facet_wrap("variable ~ .", scales = "free_y")
```




```
library(dplyr)
library(tidyr)
library(ggplot2)
library(forcats)


results %>%
  as_tibble() %>%
  mutate(iteration = 1:n()) %>%
  gather(variable, value, -iteration) %>%
  mutate(variable = factor(variable, labels = c(paste0("b", 1:6), "sigma"))) %>%
  ggplot(aes(x=value, group=variable, fill=variable)) +
    geom_histogram() +
    theme_bw() +
```

```
    theme(legend.position = "None") +
    facet_wrap("variable ~ .", scales = "free")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




It appears that both the linear regression and the blm under-estimate the sigma term. Why?

```
# Results
coef <- apply(results, 2, median)
se <- apply(results, 2, sd)


# For each ...
for(i in 1:7) {


  msg <- paste0(
    ifelse(i == 7, "sigma: ", paste0("coef: b",i-1)),
    "\n",
```

```
    "BLM Est. ", round(ifelse(i == 7, coef[i], coef[i]), digits=4), "\n",
    "LR Est. ", round(ifelse(i == 7, 1/sqrt(var(residuals(linreg))),
                        unname(coefficients(linreg)[i])), digits=4), "\n",
    "Actual: ", round(real[[i]], digits=4), "\n",
    "SE: ", round(se[i], digits=5)
  )

  # Print
  cat(msg)
  cat("\n\n")

}
```

```
## coef: b0
## BLM Est. 9.9342
## LR Est. 9.767
## Actual: 3.4
## SE: 0.00352
##
## coef: b1
## BLM Est. -1.5522
## LR Est. -1.5615
## Actual: -1.5
## SE: 0.00116
##
## coef: b2
## BLM Est. 1.6973
## LR Est. 1.6597
## Actual: 1.7
## SE: 0.00116
##
## coef: b3
## BLM Est. 3.122
```

```
## LR Est. 3.1819
## Actual: 3
## SE: 0.01422
##
## coef: b4
## BLM Est. 1.1685
## LR Est. 1.1177
## Actual: 1.2
## SE: 0.00115
##
## coef: b5
## BLM Est. 1.06
## LR Est. 0.0583
## Actual: 1.1
## SE: 0.00126
##
## sigma:
## BLM Est. 0.5311
## LR Est. 0.3853
## Actual: 0.3
## SE: 0.01193
```

Calculate common point estimates

```
# Get point estimates
# See https://stattrek.com/matrix-algebra/sums-of-squares.aspx

# Posterior mean/median (EAP)
Exp_a_post_mean <- apply(results,2,mean)
Exp_a_post_med <- apply(results,2,median)

# Posterior standard deviation
post_sd <- apply(results,2,sd)
```

```r
# Correlation matrix
post_cm <- cor(results)

# Var-covar matrix
h <- matrix(rep(1, nrow(results)), ncol=1)
v <- (results - ((h %*% t(h) %*% results) * (1/nrow(results))))
post_vc_matrix <- (t(v) %*% v) * (1/nrow(results))

# Calculate 95% credible interval
post_credint <- apply(results,2, function(x) quantile(x, c(0.025, 0.975)))

# MC error
post_mcerror <- post_sd / sqrt(k)

# Print
cat(Exp_a_post_mean)
```

```
## 9.934164 -1.55221 1.697279 3.122026 1.168525 1.060051 0.5311906
```

```r
cat(Exp_a_post_med)
```

```
## 9.934173 -1.552207 1.697292 3.122019 1.168526 1.06004 0.5311228
```

```r
print(post_credint)
```

```
##           [,1]      [,2]     [,3]     [,4]     [,5]     [,6]      [,7]
## 2.5%  9.92713 -1.554474 1.694979 3.094087 1.166282 1.057607 0.5082244
## 97.5% 9.94111 -1.549957 1.699532 3.149691 1.170800 1.062548 0.5545902
```

```r
cat(post_sd)
```

```
## 0.003520261 0.001159745 0.001157047 0.0142158 0.001152701 0.001259156 0.01192946
```

```r
cat(post_mcerror)
```

```
## 3.520261e-05 1.159745e-05 1.157047e-05 0.000142158 1.152701e-05 1.259156e-05 0.0001192946
```

Look at autocorrelation

```r
# Make a helper function
autocor <- function(x, n=10) {

  # Results
  res <- rep(0, n)

  # Lag for each n and calculate correlation
  for(i in 1:n) {
    res[i] <- cor(x, c(rep(NA, i), x[1:(length(x)-i)]), use="complete.obs")
  }

  # Return
  return(
    data.frame(
      "lag" = 1:n,
      "correlation" = res
    )
  )

}

# Plot autocorrelation per variable
library(purrr)
library(scales)
```

```
##
## Attaching package: 'scales'
```

```
## The following object is masked from 'package:purrr':
##
##     discard
```

```
results %>%
  as.data.frame() %>%
  map(., function(x) autocor(x, n=80)) %>%
  dplyr::bind_rows(.id="id") %>%
  ggplot(., aes(x=lag, y=correlation, group=id)) +
    geom_bar(stat="identity") +
    scale_x_continuous(breaks= pretty_breaks()) +
    scale_y_continuous(limits = c(-1,1)) +
    facet_wrap(id ~ .)
```