Linear & Generalized Linear Models/ Advanced Regression for Independent Data

Andrew J. Spieker, Ph.D.

Associate Professor of Biostatistics
Vanderbilt University

Set 18 (Supplement 4): Bayesian methods for linear regression

Version: 07/03/2026

Table of Contents

Conjugate priors for linear regression

Linear regression:

- Suppose we're in the setting of linear regression.
- Model: $\mathrm{E}[\mathbf{y}|\mathbf{X}] = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}$, with $\mathrm{E}[\boldsymbol{\epsilon}] = \mathbf{0}$ and $\mathrm{Cov}[\boldsymbol{\epsilon}] = \sigma^2\mathbf{I}$.
- $\boldsymbol{\beta}$ is our target of inference, but σ^2 is a parameter of the data generating mechanism that needs attention. In the example that follows, it is easier to parameterize the model in terms of $\tau = 1/\sigma^2$.
- Note: In these slides, we're thinking of the models parametrically.

Conjugate priors for linear regression

Example 18.1: Conjugate prior

- The following is a Bayesian linear regression model:

$$\begin{aligned} \mathbf{y}|\boldsymbol{\beta}, \tau &\sim \mathcal{N}(\mathbf{X}\boldsymbol{\beta}, (1/\tau)\mathbf{I}), \\ \boldsymbol{\beta}|\tau &\sim \mathcal{N}(\boldsymbol{\phi}, (1/\tau)\mathbf{V}), \\ \tau &\sim \text{Gamma}(\alpha, \delta). \end{aligned}$$

- The first specification is a likelihood, a parametric model that might be correctly or incorrectly specified.
- The second two are prior specifications; conceptualizing "correct specification" doesn't work quite the same way. The family needs to be flexible enough such that you can choose parameters that adequately represent your prior belief about $\boldsymbol{\beta}$ and τ.
- We assume, though we don't *need* to, that $\boldsymbol{\phi}$, $\mathbf{V}$, α, and δ are fixed.

Conjugate priors for linear regression

Prior specification:

- Choosing $\mathbf{V}$ to be a diagonal matrix with very large diagonal elements and a value of δ very close to zero would be an example of a weakly informative prior (on the scale of $\boldsymbol{\beta}$).
  - Choosing a prior with the goal of being "non-informative" may not always be a good idea, particularly when taking scale into account.

Example 18.2: Assay data

- Goal: Evaluate the relationship between an immunofluorescence assay (X) and a thermal assay (Y) to evaluate an (vaccine-induced) immunologic response to SARS-CoV-2.
- File: `assay.csv`.

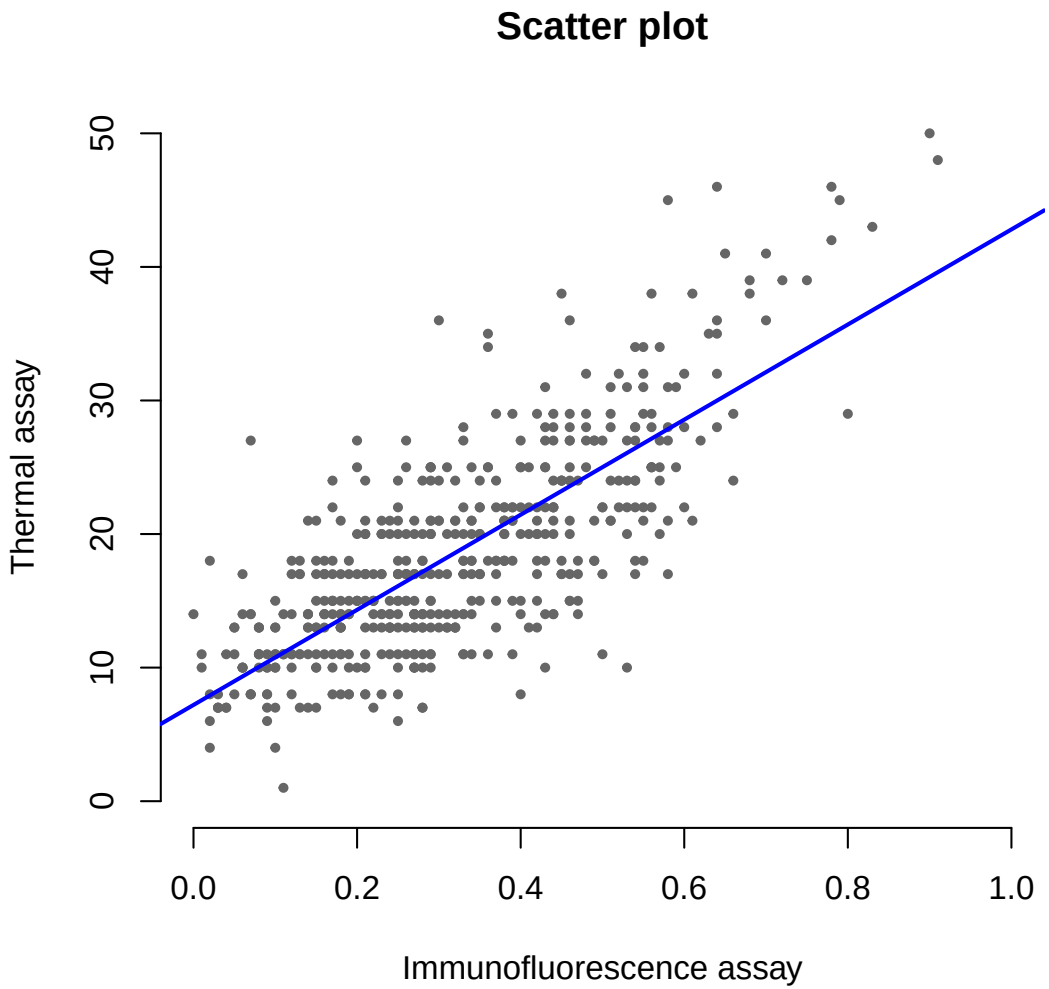

Figure S4.1: Assay data (linearity is approximated).

Example 18.2: Assay data

- Let's try a very basic prior that does not truly incorporate much of our prior knowledge about the data:
  - $\boldsymbol{\phi} = (20, 0)$.
  - $\mathbf{V} = 10\mathbf{I}$.
  - $\alpha = 2$.
  - $\delta = 100$.
- It is hard to understand just by looking at specifications for a prior what the data might look like under this combination of parameters. It is a good idea to check draws from the prior predictive distribution in order to make that assessment.
- Suppose we know in advance that the thermal assay is bounded between 0 and 50. We may want to check that our prior does not provide draws outside of this range too often (some such draws are okay).

Example 18.2: Prior predictive (computational)

```
## Read in data
dat <- read.csv("assay.csv")
x <- dat$immuno
y <- dat$therm
n <- dim(dat)[1]

## Specify prior
phi <- c(20, 0)
V <- matrix(c(10, 0, 0, 10), nrow = 2)
alpha <- 2
delta <- 100

## Set seed for reproducibility
set.seed(7345)

## Prior predictive
taus <- matrix(0, nrow = n, ncol = 1)
betas <- matrix(0, nrow = n, ncol = 2)
for (m in 1:n)
{
  tau.sample <- rgamma(1, shape = alpha, rate = delta)
  beta.sample <- rmvnorm(1, mean = phi, sigma = (1/tau.sample)*V)
  betas[m,] <- beta.sample
  taus[m] <- tau.sample
}
```

Example 18.2: Prior predictive (computational)

```r
## Generate random draws
y.tildes <- rnorm(n, mean = betas[,1] + betas[,2]*x, sd = 1/sqrt(taus[,1]))

## Plot densities repeatedly
plot(density(y.tildes), col = "gray90", ylim = c(0, 0.06), frame.plot = FALSE,
     xlab = expression(tilde(y)), ylab = "Density", main = "Prior predictive: Poor choice",
     xlim = c(-100, 100))
for (j in 1:500)
{
  y.tildes <- rnorm(n, mean = betas[,1] + betas[,2]*x, sd = 1/sqrt(taus[,1]))
  lines(density(y.tildes), col = "gray90")
}

## Real data overlaid
lines(density(y), col = "blue", lwd = 2)
## But we'd generally do this before we have the actual data
```

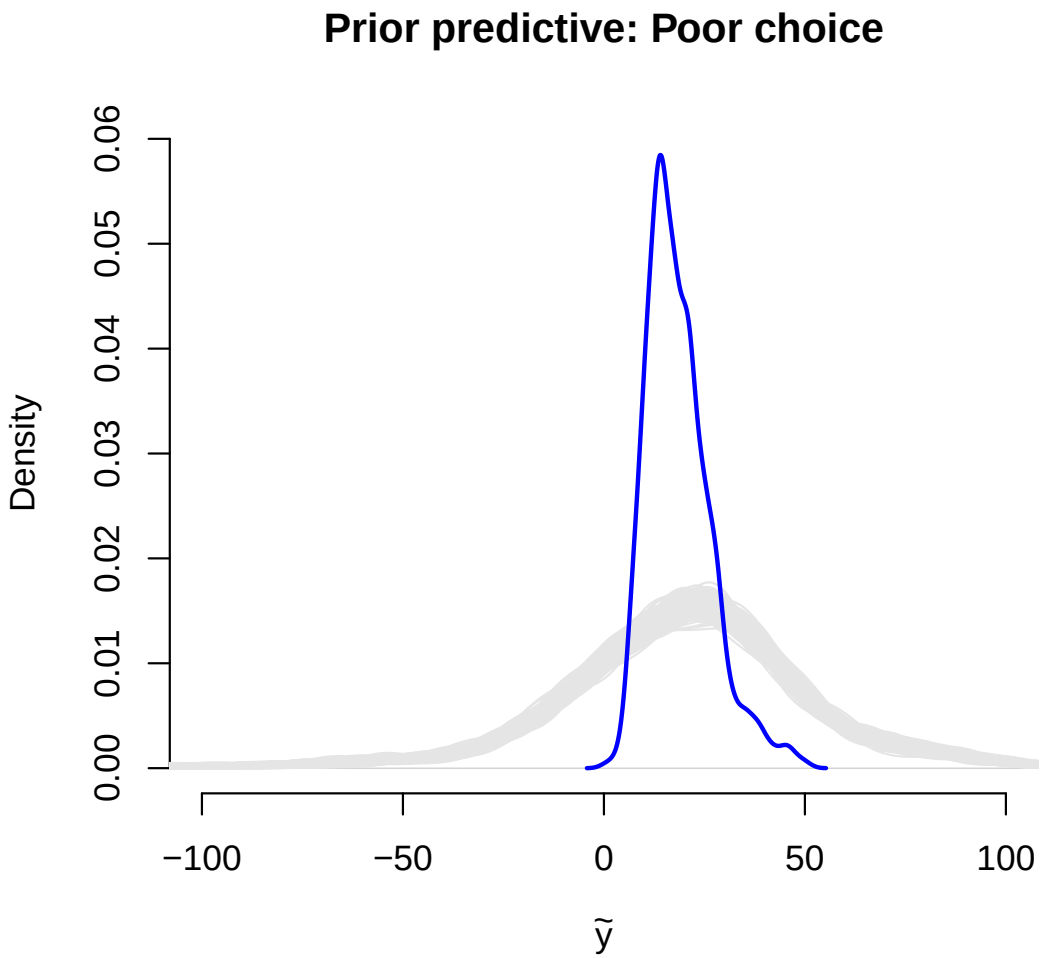

Figure S4.2: Prior predictive draws (poor choice).

Example 18.2: Assay data

- This is an opportunity to better understand prior specification.
- Here may be some knowledge that we want to exploit:
  - Mean thermal assay ≈ 0 when immunofluorescence $= 0$.
  - Mean thermal assay ≈ 50 when immunofluorescence $= 1$.
  - There should be a strictly positive association.
  - If $\beta_0 > 0$, β_1 "less positive," and vice versa.
  - Conditional SD of Y (given X) between 5-10 thermal units.
- Let's update out prior to reflect this.
  - $\boldsymbol{\phi} = (0, 50)^\intercal$.
  - $\mathbf{V} = \begin{bmatrix} 0.6 & -0.05 \\ -0.05 & 0.1 \end{bmatrix}$.
  - $\alpha = 5$.
  - $\delta = 500$.

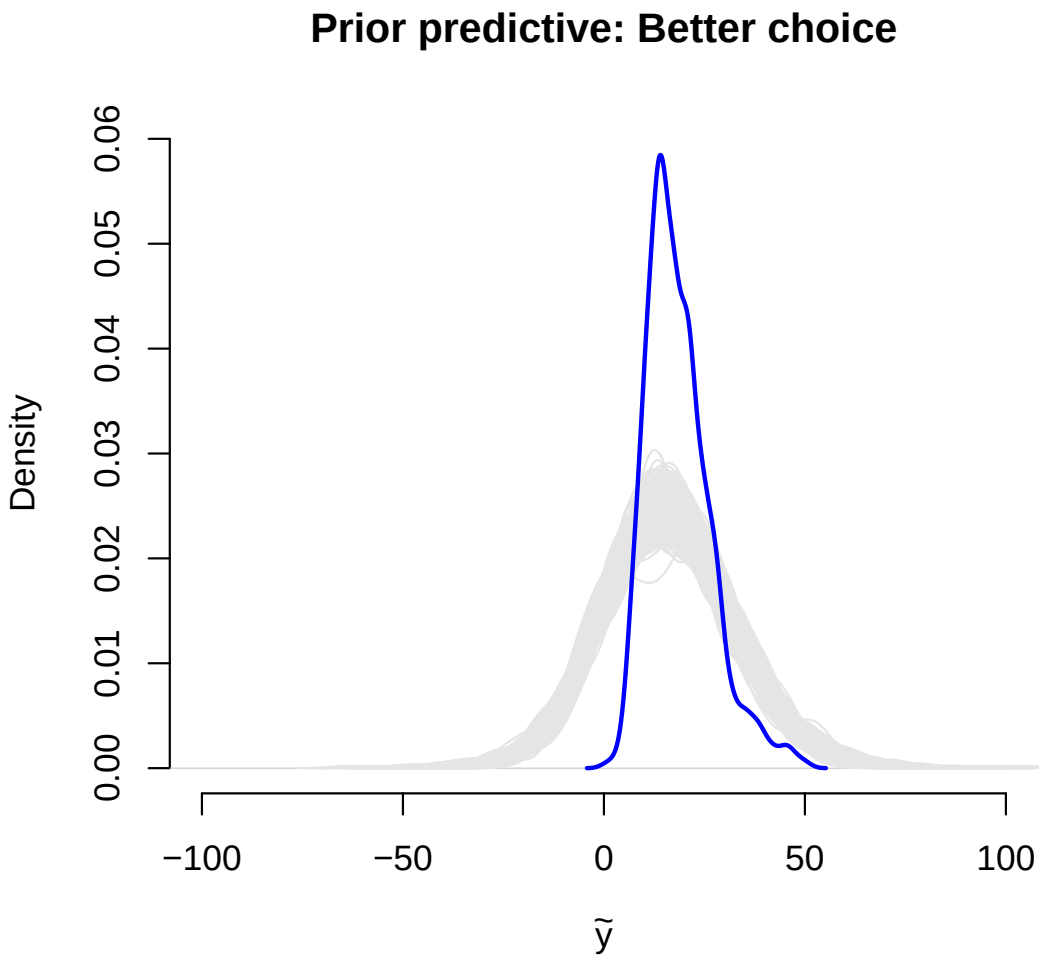

Figure S4.3: Prior predictive draws (better cohice).

Posterior: Conjugate relationship

- It can be shown (not-so-light algebra) that $\pi(\boldsymbol{\beta}, \tau|\mathbf{y})$, is of the same family (normal/Gamma) as the joint prior, $\pi(\boldsymbol{\beta}, \tau)$, which is to say the prior is conjugate.
- The *marginal* posterior $\pi(\boldsymbol{\beta}|\mathbf{y})$, takes the form (not-so-light algebra) of a multivariate t-distribution, $t_{N+2\alpha}(\boldsymbol{\phi}_*, \boldsymbol{\Sigma}_*)$, where:

$$\begin{aligned}
\boldsymbol{\phi}_* &= (\mathbf{V}^{-1} + \mathbf{X}^\intercal\mathbf{X})^{-1}(\mathbf{V}^{-1}\boldsymbol{\phi} + \mathbf{X}^\intercal\mathbf{y}), \\
\boldsymbol{\Sigma}_* &= \left[\frac{(\mathbf{y} - \mathbf{X}\boldsymbol{\phi})^\intercal(\mathbf{I} + \mathbf{X}\mathbf{V}\mathbf{X}^\intercal)^{-1}(\mathbf{y} - \mathbf{X}\boldsymbol{\phi}) + 2\delta}{N + 2\alpha}\right](\mathbf{V}^{-1} + \mathbf{X}^\intercal\mathbf{X})^{-1}
\end{aligned}$$

- $\mathrm{E}[\boldsymbol{\beta}|\mathbf{y}] = \boldsymbol{\phi}_*$ and $\mathrm{Cov}[\boldsymbol{\beta}|\mathbf{y}] = [(N + 2\alpha)/(N + 2\alpha - 2)]\boldsymbol{\Sigma}_*$.
- Let's think about these formulas in the context of the last statement of the previous slide.

Point estimation:

- In our simple example, $\boldsymbol{\phi}_*$ marks the posterior mean, posterior median, and posterior mode. This will not always be the case.

Conjugate priors for linear regression

Posterior: Conjugate relationship

- The *marginal* posterior $\pi(\tau|\mathbf{y})$, takes the form (not-so-light algebra) of a Gamma(α_*, δ_*) distribution, where:

$$\begin{aligned} \alpha_* &= \alpha + N/2, \text{ and} \\ \delta_* &= (-\boldsymbol{\phi}_*^\intercal(\mathbf{V}^{-1} + \mathbf{X}^\intercal\mathbf{X})\boldsymbol{\phi}_* + \boldsymbol{\phi}^\intercal\mathbf{V}^{-1}\boldsymbol{\phi} + \mathbf{y}^\intercal\mathbf{y} + 2\delta)/2. \end{aligned}$$

- Intuitive explanation a bit more challenging.
- This knowledge makes it easier to determine posterior predictive distribution.

Interval estimation:

- Conveniently, it turns out in our conjugate prior example that

$$\frac{\mathbf{c}^\mathsf{T}\boldsymbol{\phi}_* - \mathbf{c}^\mathsf{T}\boldsymbol{\beta}}{\sqrt{\mathbf{c}^\mathsf{T}\boldsymbol{\Sigma}_*\mathbf{c}}} \sim t_{N+2\alpha}.$$

- More generally, a suitably centered and scaled a linear combination of a multivariate t-distributed random vector follows a univariate t-distribution.
- Choosing $\mathbf{c} = (0, \ldots, 0, 1, 0, \ldots, 0)^\mathsf{T}$, zeros everywhere except the k^{th} position, informs us about the marginal posterior for $\beta_k|\mathbf{y}$:

$$\frac{\beta_k - \phi_{*;k}}{\sigma_{*;ii}} \sim t_{N+2\alpha}.$$

- Sensible 95% credible interval: $\phi_{*;k} \pm t_{0.975,N+2\alpha}\sigma_{*;ii}$.

Posterior: Posterior mean and variance for β_1

```
posterior <- function(phi0, phi1, v1, v2, v3, alpha, delta, x, Y)
{
  N <- length(Y)
  X <- cbind(1, x)
  phi <- matrix(c(phi0, phi1), nrow = 2)
  V <- matrix(c(v1, v2, v2, v3), nrow = 2)
  post.mean <- as.numeric(solve(solve(V) + t(X) %*% X) %*% (solve(V) %*% phi + t(X) %*% Y))
  post.var <- matrix(as.numeric(((t(Y - X %*% phi) %*%
                                  solve((diag(N)+ X %*% V %*% t(X))) %*%
                                  (Y - X %*% phi) + 2*delta)/(N + 2*alpha - 2))) *
                       solve((solve(V) + t(X) %*% X)), nrow = 2)
  out <- list(mean = post.mean, var = post.var)
  return(out)
}
```

Example 18.2: 95% credible interval for β_1

```
## Posterior
pst <- posterior(0, 50, 0.6, -0.05, 0.1, 5, 500, x, y)

## Parameters of marginal posterior
phistar <- as.numeric(pst$mean)
Sigmastar <- pst$var

## Interested in beta1
c <- matrix(c(0,1), ncol = 1)

## Define sigma11
sigma11 <- sqrt((t(c) %*% Sigmastar %*% c))

## Point estimate
> t(c) %*% phistar
       [,1]
[1,] 41.442

## 95% credible interval
crt <- qt(0.975, df = n + 2*alpha)
> c(t(c) %*% phistar -  crt * sigma11,
    t(c) %*% phistar + crt * sigma11)
[1] 39.259 43.626
```

Example 18.2: 95% confidence interval for β_1

```
## OLS
zz <- lm(y ~ x)

## Coefficients
> summary(zz)

Call:
lm(formula = y ~ x)

Residuals:
    Min      1Q  Median      3Q     Max
-16.074  -3.173  -0.037   3.123  18.115

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)    7.205      0.489    14.7   <2e-16 ***
x             35.601      1.350    26.4   <2e-16 ***
---
...
Residual standard error: 5.16 on 498 degrees of freedom
Multiple R-squared:  0.583, Adjusted R-squared:  0.582
F-statistic:  696 on 1 and 498 DF,  p-value: <2e-16

## 95% confidence interval
> confint(zz)
              2.5 %  97.5 %
(Intercept)  6.244  8.1655
x           32.949 38.2536
```

Posterior predictive:

- In this example, we have a nice result for the posterior predictive distribution:

$$\widetilde{\mathbf{y}}|\mathbf{X} \sim t_{N+2\alpha}\left(\mathbf{X}\boldsymbol{\phi}_*, \frac{\delta_*}{\alpha_*}(\mathbf{I}_N + \mathbf{X}(\mathbf{V}^{-1} + \mathbf{X}^\intercal\mathbf{X})^{-1}\mathbf{X}^\intercal)\right).$$

- Each draw, $\widetilde{y}$, will follow a (shifted/scaled) univariate t-distribution.
- Suppose that we didn't know this form for the posterior predictive distribution. Could we generalize our previous trick to generate draws computationally?
  - You bet!

Example 18.2: Posterior predictive

```r
## Design matrix and outcome
X <- cbind(1,x)
y <- cbind(y)

## Posterior hyperparameters
phistar <- solve(solve(V) + t(X) %*% X) %*% (solve(V) %*% phi + t(X) %*% y)
alphastar <- alpha + n/2
deltastar <- (-t(phistar) %*% (solve(V) + t(X) %*% X) %*% phistar
              + t(phi) %*% solve(V) %*% phi + t(y) %*% y + 2*delta)/2

## Storage for posterior draws
taus <- matrix(0, nrow = n, ncol = 1)
sigmas <- matrix(0, nrow = n, ncol = 1)
betas <- matrix(0, nrow = n, ncol = 2)

for (m in 1:n)
{
  tau.sample <- rgamma(1, alphastar, deltastar)
  beta.sample <- rmvnorm(1, mean = phistar,
                         sigma = (1/tau.sample)*solve(solve(V) + t(X) %*% X))
  taus[m,1] <- tau.sample
  betas[m,] <- beta.sample
}

## Draw from posterior predictive
y.tildes <- rnorm(n, mean = betas[,1] + x*betas[,2], sd = 1/sqrt(taus[,1]))
```

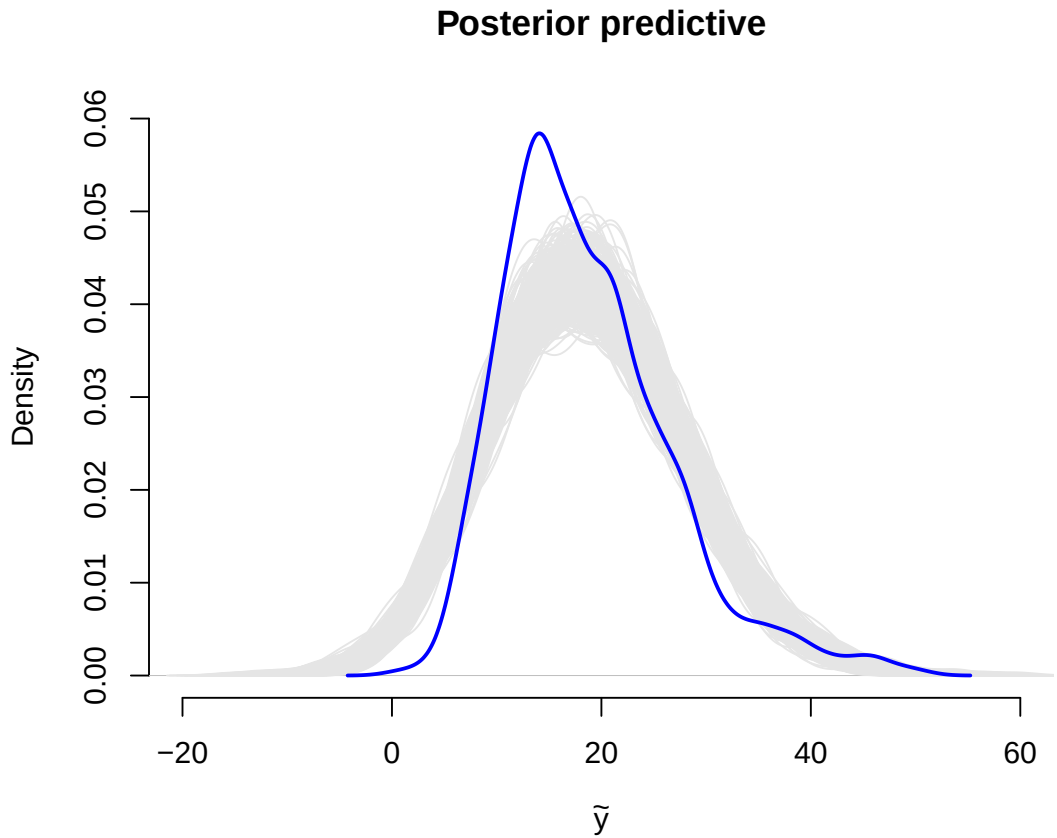

Figure S4.4: Posterior predictive draws.

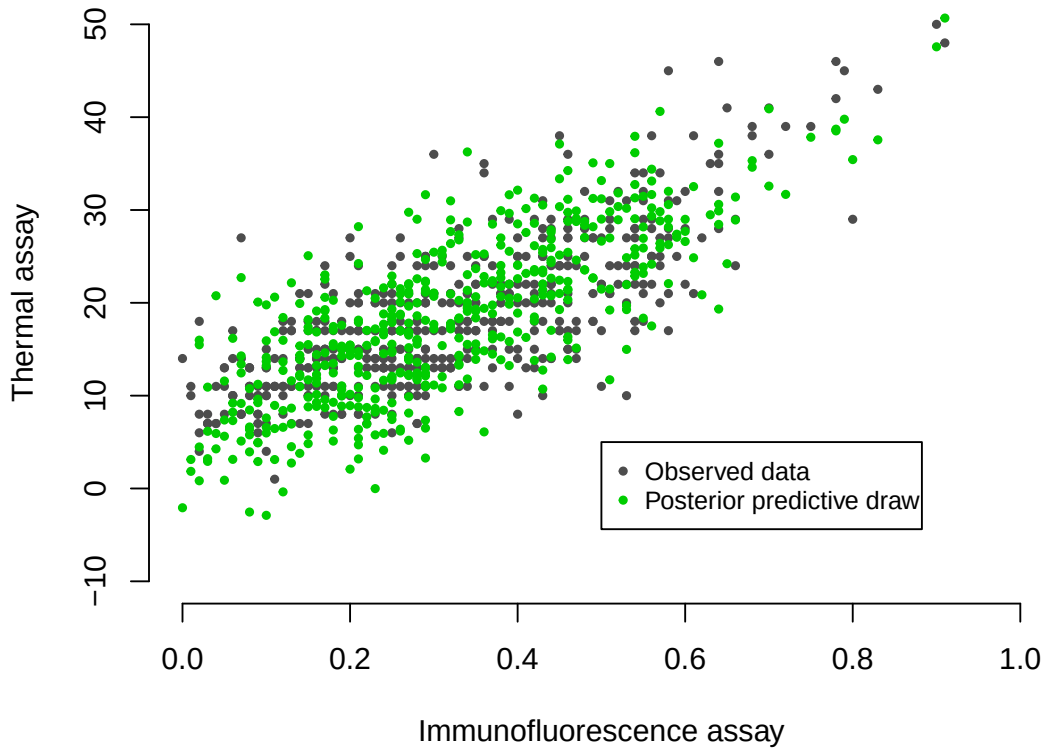

Figure S4.5: Posterior predictive draws (conditional on X).

Table of Contents

Metropolis-Hastings for linear regression

Bayesian linear regression model:

- The following is a Bayesian linear regression model:

$$\begin{aligned} \mathbf{y}|\boldsymbol{\beta}, \sigma &\sim \mathcal{N}(\mathbf{X}\boldsymbol{\beta}, \sigma^2\mathbf{I}), \\ \boldsymbol{\beta} &\sim \mathcal{N}(\boldsymbol{\phi}, \mathbf{V}), \\ \sigma &\sim \text{Gamma}(\alpha, \delta). \end{aligned}$$

- This example differs from the previous example in that:
  - We parameterize in terms of σ in place of $\tau = 1/\sigma^2$.
  - $\boldsymbol{\beta}$ and σ are independent.
- Again, assume $\boldsymbol{\phi}$, $\mathbf{V}$, α, and δ are fixed.
- We seek a computational tool to sample draws from $\pi(\boldsymbol{\beta}, \sigma|\mathbf{y})$.

Metropolis-Hastings for linear regression

Algorithm: Metropolis-Hastings (Big picture)

- Once you're satisfied with your model, the general concept behind the Metropolis-Hastings algorithm is as follows:
  1. Initialize a posterior draw as $\boldsymbol{\theta}^{(0)}$ (it doesn't have to be a great guess, but it does have to be "not awful").
  2. Starting from a posterior draw, $\boldsymbol{\theta}^{(j)}$, take a random (but structured, in a way we will discuss) step away to a "nearby" value of $\boldsymbol{\theta}$, call it $\boldsymbol{\theta}_{\text{prop}}^{(j+1)}$.
  3. With stochasticity (we will discuss), evaluate the plausibility of $\boldsymbol{\theta}_{\text{prop}}^{(j+1)}$ as a candidate to include as a valid posterior draw.
  4. If algorithm says "yes" in Step 3, that new draw can be added to your collection of posterior draws **and is your new basis to go back to Step 2 and repeat the process** (that is $\boldsymbol{\theta}^{(j+1)} = \boldsymbol{\theta}_{\text{prop}}^{(j+1)}$). If "no," go back to Step 2 and try again (i.e., from $\boldsymbol{\theta}^{(j)}$, the last accepted draw).
  5. Iterate Steps 2-4 until there are enough samples from the apparent stationary distribution.
- Under some assumptions, the stationary distribution is $\pi(\boldsymbol{\theta}|\mathbf{y})$.

Metropolis-Hastings for linear regression

Metropolis-Hastings: Elaboration on Step 2 in context of example

- Example:

$$\begin{aligned} \boldsymbol{\beta} &\sim \mathcal{N}(\boldsymbol{\phi}, \mathbf{V}), \\ \sigma &\sim \text{Gamma}(\alpha, \delta). \end{aligned}$$

- Starting from $\boldsymbol{\theta}^{(j)}$, consider taking a random draw, $\boldsymbol{\theta}_{\text{prop}}^{(j+1)}$ from the following proposal distribution:

$$\begin{aligned} \boldsymbol{\beta}_{\text{prop}}^{(j+1)} &\sim \mathcal{N}(\boldsymbol{\beta}^{(j)}, \omega_1 \mathbf{I}), \\ \sigma_{\text{prop}}^{(j+1)} &\sim \text{Gamma}(\sigma^{(j)}/\omega_2, 1/\omega_2). \end{aligned}$$

- The distribution for the draw is centered at the previous accepted proposal, $\boldsymbol{\theta}^{(j)}$; ω_1 and ω_2 control the search window (variance).
- Why are my proposals of the same family as my prior? Although they don't have to be, there is a convenience aspect in this example.

Metropolis-Hastings for linear regression

Metropolis-Hastings: Elaboration on Step 2 in context of example

- The values of ω_1 and ω_2 control how close to $\boldsymbol{\theta}^{(j)}$ you'd like your subsequent proposal to be, inviting the following trade-off:
  - Larger window: search for new draws in unexplored areas. Too large, and the algorithm may fail to converge to a stationary distribution.
    - ⋆ In our example, higher values of ω_1 and ω_2 cast a wider net as compared to smaller values.
  - Narrower window: refine search closer to the areas where good candidates have been identified. Too narrow, and the algorithm may not find the optimum in a reasonable amount of time.
- See the third previously possibility listed for "what can go wrong."

Metropolis-Hastings for linear regression

Metropolis-Hastings: Elaboration on Step 3 in context of example

- Once we have our candidate, $\boldsymbol{\theta}_{\text{prop}}^{(j+1)}$, we compare it to our previously accepted step, $\boldsymbol{\theta}^{(j)}$, in the following way:

$$Q = \frac{f(\mathbf{y}|\mathbf{X};\boldsymbol{\theta}_{\text{prop}}^{(j+1)})}{f(\mathbf{y}|\mathbf{X};\boldsymbol{\theta}^{(j)})} \cdot \frac{\pi(\boldsymbol{\theta}_{\text{prop}}^{(j+1)})}{\pi(\boldsymbol{\theta}^{(j)})} \cdot \frac{g_{\boldsymbol{\omega}}(\boldsymbol{\theta}^{(j)}|\boldsymbol{\theta}_{\text{prop}}^{(j+1)})}{g_{\boldsymbol{\omega}}(\boldsymbol{\theta}_{\text{prop}}^{(j+1)}|\boldsymbol{\theta}^{(j)})},$$

  where, for instance, $g_{\boldsymbol{\omega}}(\boldsymbol{\theta}_{\text{prop}}^{(j+1)}|\boldsymbol{\theta}^{(j)})$ evaluates the proposal density of $\boldsymbol{\theta}_{\text{prop}}^{(j+1)}$ assuming parameters $\boldsymbol{\theta}^{(j)}$.
- The value of $\boldsymbol{\omega} = (\omega_1, \omega_2)$ does not update across iterations.
- Let $U \sim \text{Uniform}(0, 1)$ denote an iteration-specific random draw. If $Q > U$, we accept the candidate, so that $\boldsymbol{\theta}^{(j+1)} = \boldsymbol{\theta}_{\text{prop}}^{(j+1)}$. Otherwise, go back to Step 2 and try generating another proposal.
- If the proposal distributions are symmetric, the last fraction should be one (Metropolis-Hastings $\rightarrow$ Metropolis).

Metropolis-Hastings for linear regression

Metropolis-Hastings: That's it!

- That's the Metropolis-Hastings algorithm!
- It is not in any way "obvious" that this algorithm would have the property of converging to the posterior as a stationary distribution (again, such technical details are not in the scope of this course).
- However, we can gain some intuition for the value of Q.
  - First note that—all else being equal—higher values of Q are more likely to be accepted as compared to lower values. Q is higher when...
    - $\boldsymbol{\theta}_{\text{prop}}^{(j+1)}$ is more consistent with the likelihood as compared to $\boldsymbol{\theta}^{(j)}$.
    - $\boldsymbol{\theta}_{\text{prop}}^{(j+1)}$ is more consistent with the prior as compared to $\boldsymbol{\theta}^{(j)}$.
    - $\boldsymbol{\theta}_{\text{prop}}^{(j+1)}$ is *less* consistent with the proposal as compared to $\boldsymbol{\theta}^{(j)}$.
- Intuitively, this procedure encourages the acceptance of draws that are more consistent with your model and prior belief, but can also be designed to encourage exploration of under-explored areas.

Metropolis-Hastings for linear regression

More notes: Assessment of stationarity

- To assess whether a stationary distribution has been achieved, you'd be nervous about *autocorrelation* (tendency for nearby iterations to be closer to one another as compared to to other further-away iterations).
  - Of course, because this is a Markov chain, we're not expecting to see the random draws completely uncorrelated.
  - High autocorrelation reduces the effective number of iterations.
  - When there is high autocorrelation, we sometimes benefit from reparameterizing the model.
- Thinning is the process by which certain draws are discarded to reduce autocorrelation. This is generally not considered a good practice.
- If chains appear not to be approximately stationary after a moderate number of iterations, a good strategy—other than searching for and fixing bugs in the code—may be to tweak the model (as opposed to just running more iterations and hoping for the best).

Metropolis-Hastings for linear regression

More notes: Strategies

- One typically runs MCMC with different initial parameters to evaluate whether they converge to the same stationary distribution.
  - When this appears to be the case, we say there is *good mixing*.
  - *Slow mixing* signifies challenges.
- If after many iterations there does not appear to be convergence to a stationary distribution, there are many things that could have gone wrong. Possibilities include:
  - A prior that is insufficiently informative.
  - A poor initializer.
  - A poor chose of a proposal "window" (I will elaborate on this).
  - Parameters that are too tightly correlated.

Table of Contents

Metropolis-Hastings from scratch

Example 18.3: Toy data

- Let's return to the immune assay data set and implement a Bayesian analysis using Metropolis-Hastings from scratch.

Example 18.3: Setting the prior

- Prior for $\boldsymbol{\beta}$:

$$\boldsymbol{\beta} \sim \mathcal{N}(\boldsymbol{\phi}, \mathbf{V}).$$

- This time, let's try the following prior hyperparameters:
  - $\boldsymbol{\phi} = (0, 50)^\intercal$.
  - $\mathbf{V} = \begin{bmatrix} 50 & 0 \\ 0 & 200 \end{bmatrix}$.

Example 18.3: Setting the prior

- Recall the form of the prior for σ:

$$\sigma \sim \text{Gamma}(\alpha, \delta).$$

- This time, let's try the following prior hyperparameters:
  - $\alpha = 3$ and $\delta = 3/5$.

Example 18.3: Coding the prior

```
## Prior
pr <- function(theta, alpha = 3, delta = 3/5, v1 = 50, v2 = 200, phi = c(0,50)) {
  d.beta <- dmvnorm(theta[1:2], mean = phi, sigma = matrix(c(v1, 0, 0, v2), nrow = 2))
  d.sigma <- dgamma(theta[3], shape = alpha, rate = delta)
  out <- log(d.beta) + log(d.sigma)
  return(out)
}
```

Please note that use of the log-density has better computational stability.

Example 18.3: Setting the proposal

- Now, starting from $\boldsymbol{\theta}^{(j)}$, we want to have a suitable window in which to propose our next draw:

$$\begin{aligned} \boldsymbol{\beta}_{\text{prop}}^{(j+1)} &\sim \mathcal{N}(\boldsymbol{\beta}^{(j)}, \omega_1 \mathbf{I}), \\ \sigma_{\text{prop}}^{(j+1)} &\sim \text{Gamma}(\sigma^{(j)}/\omega_2, 1/\omega_2). \end{aligned}$$

- Consider $\omega_1 = 0.5$ and $\omega_2 = 1/100$.
  - Modifying window will affect the properties of the algorithm.
  - Consider an exploration of this sort to be a good use of time.
- The most essential part is to center the draws to have mean $\boldsymbol{\theta}^{(j)}$.
- That ω_1 and ω_2 are directly proportional to the variances is nice, but not always straightforward to achieve (nor is it essential). However, you must understand how the windows "control" the variance.
  - Example: Upcoming homework problem with a Weibull prior.

Metropolis-Hastings from scratch

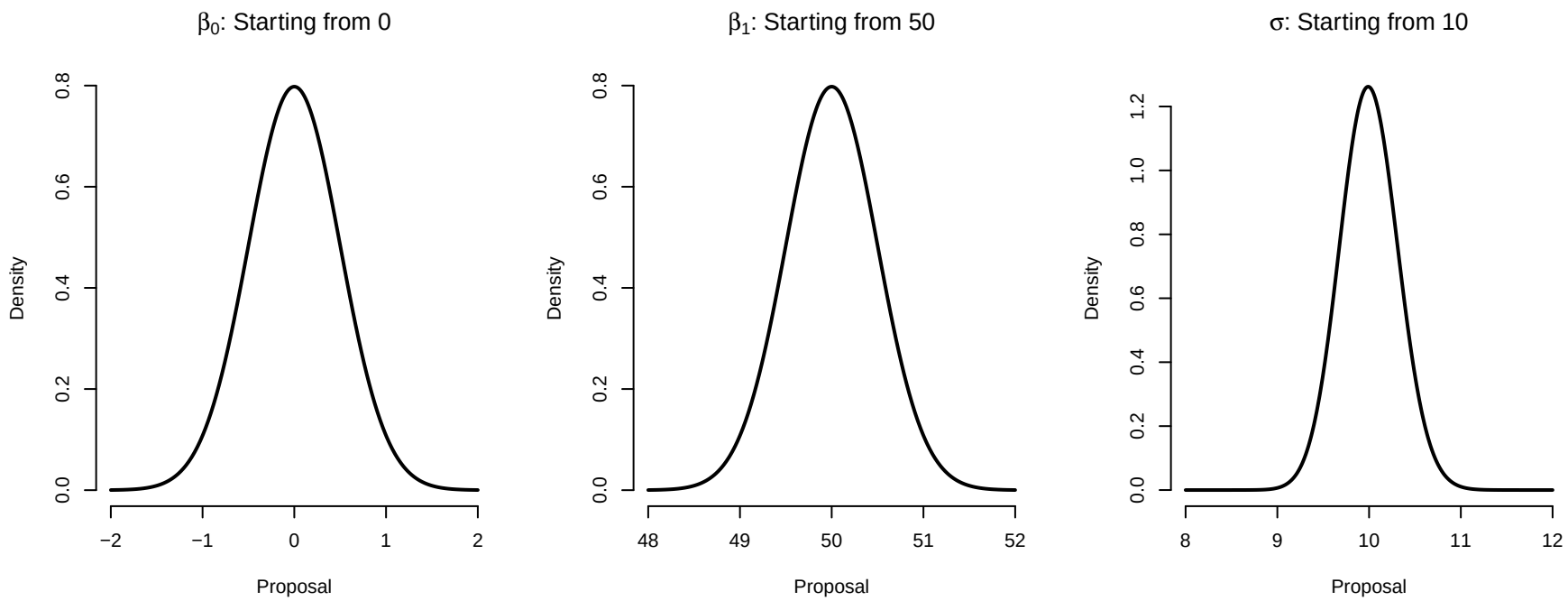

Figure S4.6: Proposal densities.

Example 18.3: Coding the proposal

```
## Proposal
proposal <- function(theta, omega1, omega2) {
  prop <- rep(0,3)
  prop[1:2] <- rmvnorm(1, mean = theta[1:2], sigma = diag(2)*omega1)
  prop[3] <- rgamma(1, shape = theta[3]/omega2, rate = 1/omega2)
  return(prop)
}
```

Example 18.3: Coding the log-likelihood

```
## Log-likelihood
loglik <- function(theta, x, y) {
  out <- sum(log(dnorm(y, mean=theta[1] + theta[2]*x, sd=theta[3])))
  return(out)
}
```

Example 18.3: Coding the proposal density ratio

```
## Proposal ratio
prop.ratio <- function(theta.j, theta.proposed, omega1, omega2) {
  d1 <- dmvnorm(theta.j[1:2], mean = theta.proposed[1:2], sigma = diag(2)*omega1)
  d2 <- dgamma(theta.j[3], shape = theta.proposed[3]/omega2, rate = 1/omega2)
  d3 <- dmvnorm(theta.proposed[1:2], mean = theta.j[1:2], sigma = diag(2)*omega1)
  d4 <- dgamma(theta.proposed[3], shape = theta.j[3]/omega2, rate = 1/omega2)
  out <- log(d1) + log(d2) - log(d3) - log(d4)
  return(out)
}
```

Example 18.3: Setting up MCMC

```
## Search window width
omega1 <- 0.5
omega2 <- 1/100

## Track number accepted/rejected
accepted <- 0
rejected <- 0

## Initialize based on a random draw from prior
theta.j <- as.numeric(c(rnorm(1, mean = 0, sd = sqrt(50)),
                        rnorm(1, mean = 50, sd = sqrt(200)),
                        rgamma(1, shape = 3, rate = 3/5)))

## Number of MCMC iterations
M = 500000

## Store accepted thetas
accepted.thetas <- matrix(0, nrow = M, ncol = 3)
```

Example 18.3: Running MCMC

```
## Set seed for reproducibility
set.seed(7345)

## Run MCMC
for (j in 1:M) {
  theta.proposed <- proposal(theta.j, omega1, omega2)
  ll.th.proposed <- loglik(theta.proposed, x, y)
  ll.th.j <- loglik(theta.j, x, y)
  prior.proposed <- pr(theta.proposed)
  prior.theta.j <- pr(theta.j)
  p.ratio <- prop.ratio(theta.j, theta.proposed, omega1, omega2)
  logQ <- ll.th.proposed - ll.th.j + prior.proposed - prior.theta.j + p.ratio
  U <- runif(1)
  if (exp(logQ) > U) {
    accepted <- accepted + 1
    accepted.thetas[accepted,] <- theta.proposed
    theta.j <- theta.proposed
  }
  if (exp(logQ) <= U) {rejected <- rejected + 1}
  if (round(j/5000) == (j/5000)) {
    print(paste(j, "iterations complete!"))
  }
}
```

Exercise: Go through code to understand how it synthesizes the functions.

Example 18.3:

```
## Report accepted/rejected draws
> accepted
[1] 23745

> rejected
[1] 76255

## Don't forget that you have more storage than necessary
accepted.thetas <- accepted.thetas[1:accepted,]
```

More samples are rejected than accepted.

Metropolis-Hastings from scratch

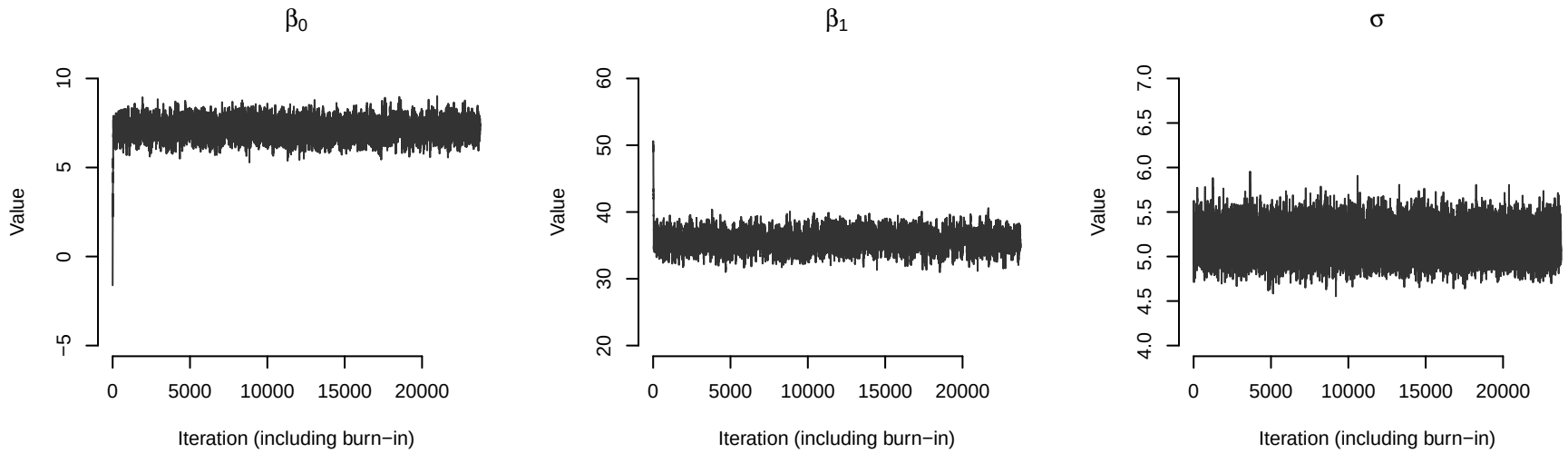

Figure S4.7: Trace plot (depicting burn-in).

Metropolis-Hastings from scratch

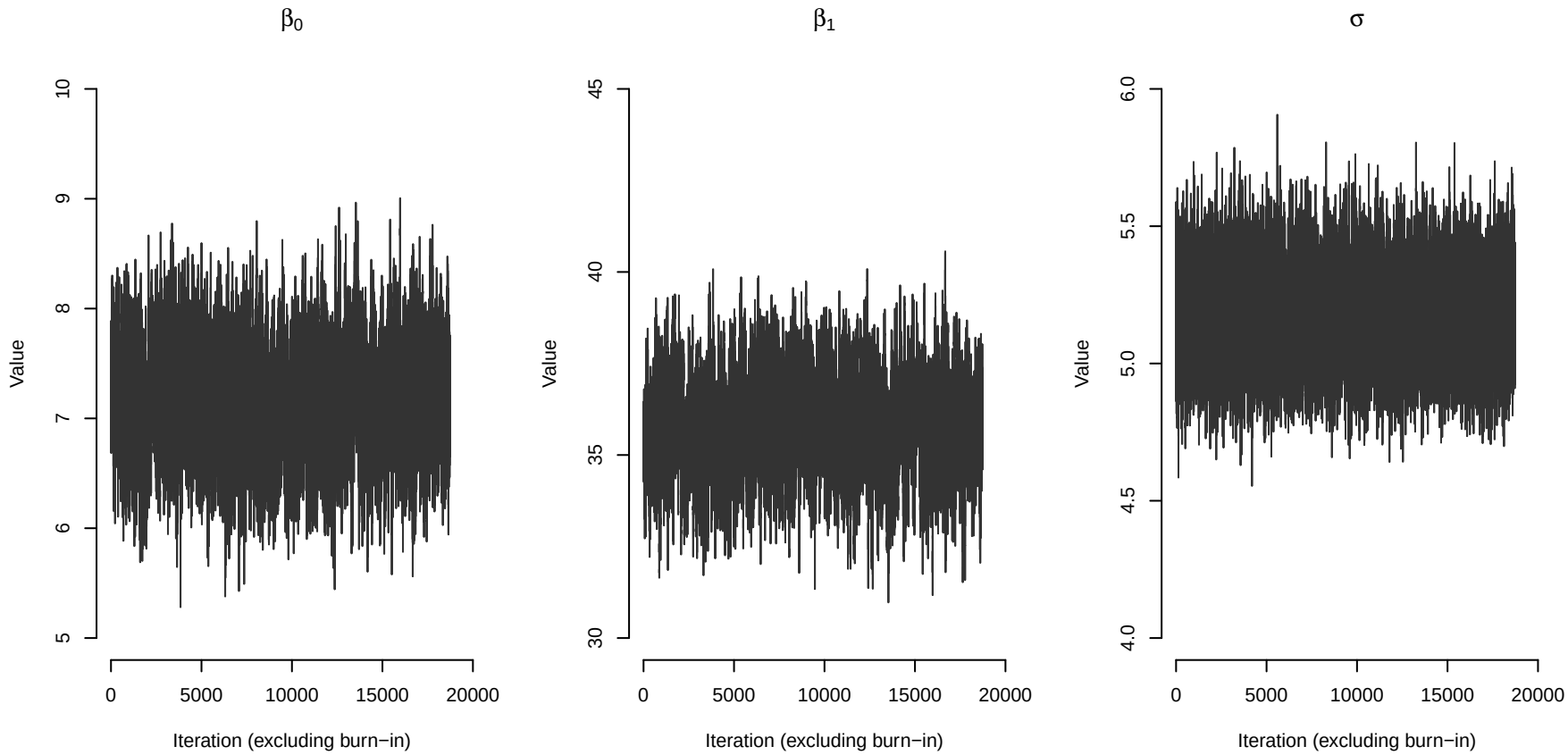

Figure S4.8: Trace plot (excluding 1:5000).

Metropolis-Hastings from scratch

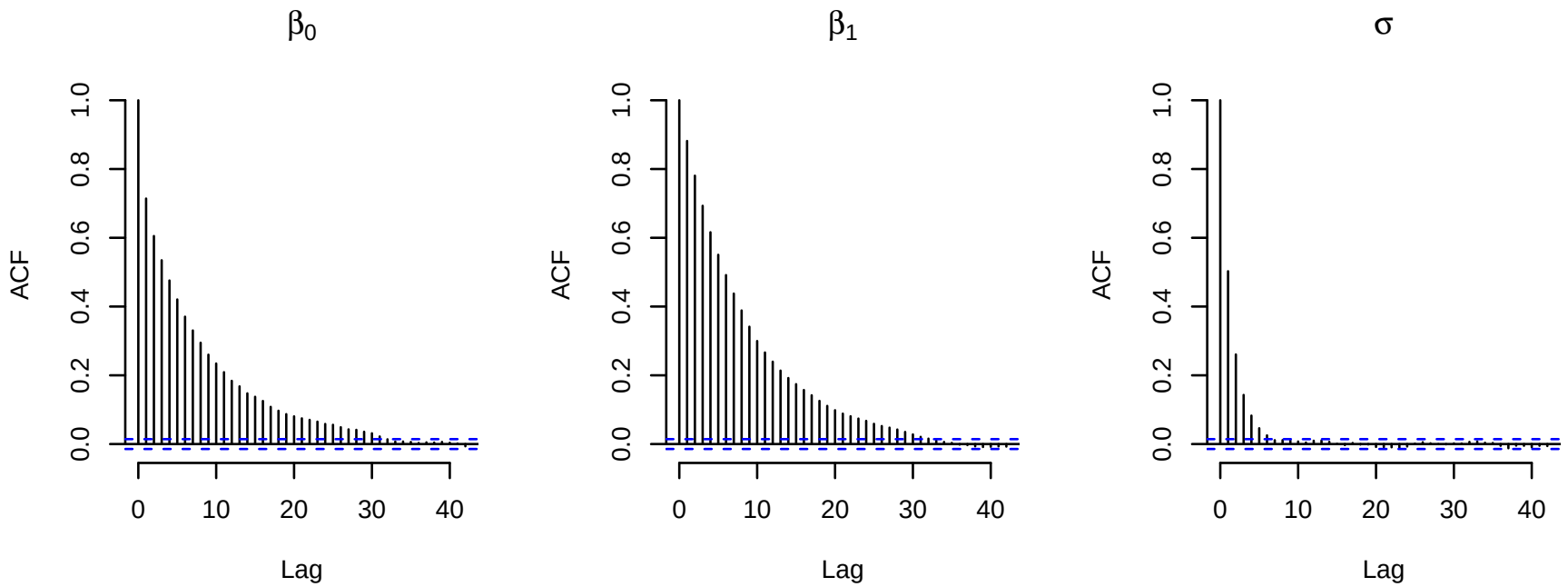

Figure S4.9: Assessing autocorrelation (`acf()` function in R).

Metropolis-Hastings from scratch

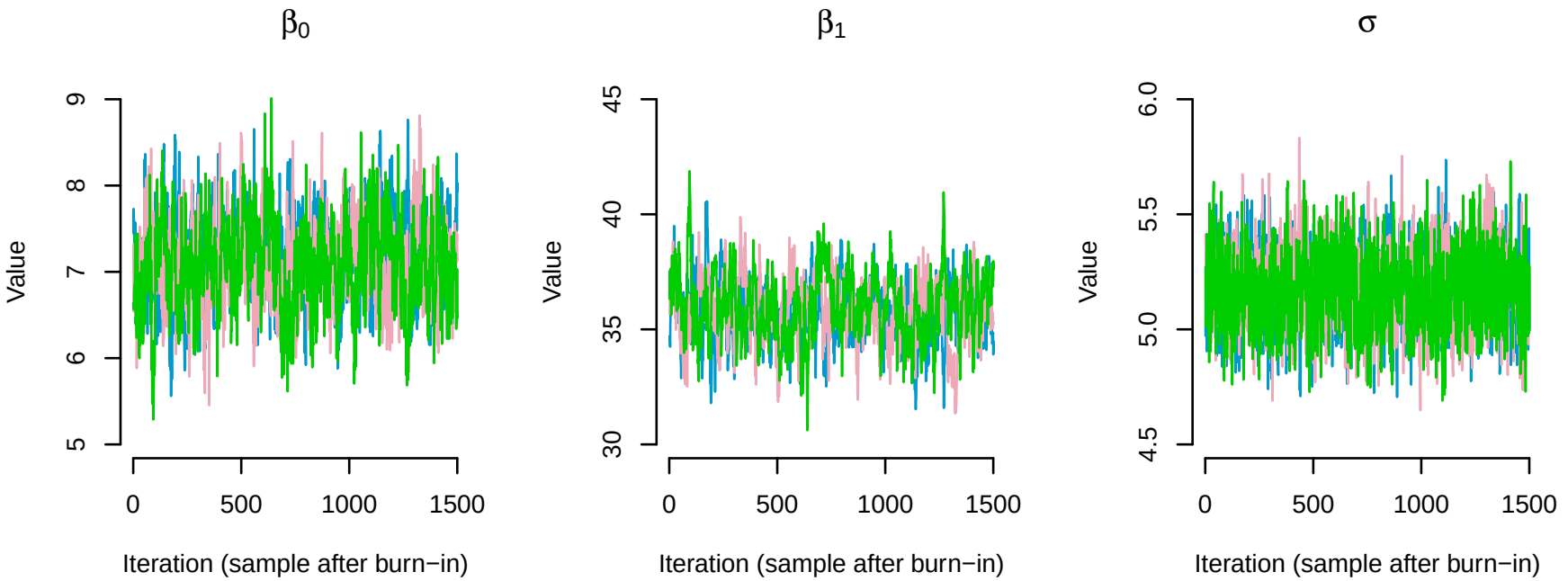

Figure S4.10: Multiple chains (done in practice; code not shown).

Example 18.3: Draws represent estimate of joint posterior

```
## Combine three chains (code for three chains run but not shown):
final.thetas <- rbind(accepted.thetas, accepted.thetas2, accepted.thetas3)

> head(final.thetas)
         [,1]     [,2]     [,3]
[1,] 6.747033 35.87393 5.264335
[2,] 7.342367 34.90827 5.588177
[3,] 7.083618 35.64153 5.348211
[4,] 6.907126 36.34663 5.192740
[5,] 7.067542 35.27238 5.159027
[6,] 7.309792 36.18202 5.129287

## (etc.)

## Column represents draws from marginal posterior.

## Draws are correlated within an iteration
## (not apparent from looking at marginal posteriors).
```

Metropolis-Hastings from scratch

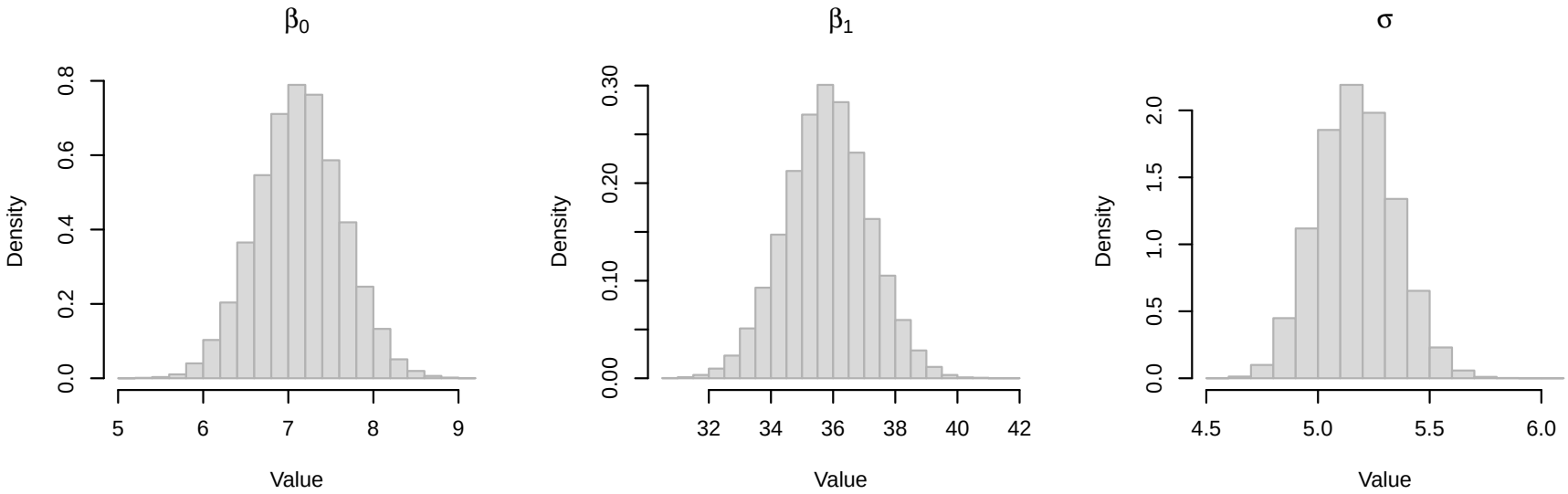

Figure S4.11: Marginal posteriors.

Example 18.3: Draws represent estimate of joint posterior

```
## Posterior means
> colMeans(final.thetas)
[1]  7.131045 35.809624  5.170375

## Posterior covariance
> cov(final.thetas)
              [,1]          [,2]          [,3]
[1,]  2.495971e-01 -5.695972e-01 -8.547916e-05
[2,] -5.695972e-01  1.808847e+00  6.711686e-05
[3,] -8.547916e-05  6.711686e-05  2.984033e-02

## OLS (comparison)
zz <- lm(y ~ x)

> vcov(zz)
            (Intercept)        x
(Intercept)     0.23914 -0.58209
x              -0.58209  1.82210
```

Example 18.3: Results

```
## Posterior mean
> as.numeric(mean(final.thetas[,2]))
[1] 35.80962

## 2.5th and 97.5th percentiles
> as.numeric(quantile(final.thetas[,2], c(0.025, 0.975)))
[1] 33.14443 38.44808

## Recall HPDint function
> as.numeric(HPDint(final.thetas[,2]))
[1] 33.21951 38.51218

## 95% CI from OLS
> as.numeric(confint(zz)[2,])
[1] 32.949 38.254
```

Example 18.3: Discussion

- Indeed, in *this* particularly nice case, the interval estimates are quite concordant. But...
  - The sample size was not particularly small.
  - The normal likelihood was satisfied.
  - No interim analyses on the data (affects frequentist properties).
- Play around with these, and you'll start to see differences emerge.
  - As this is an introduction, we're starting with simple examples and they will behave accordingly.

Example 18.3: Posterior predictive

```
## Number of final accepted iterations
n.mc <- dim(final.thetas)[1]

## Sample draws (equal to sample size from data)
samp <- sample(c(1:n.mc), size = n, replace = TRUE)

## Posterior predictive draw
y.tildes <- rnorm(n, mean = final.thetas[samp,1] + final.thetas[samp,2]*x,
            sd=final.thetas[samp,3])
```

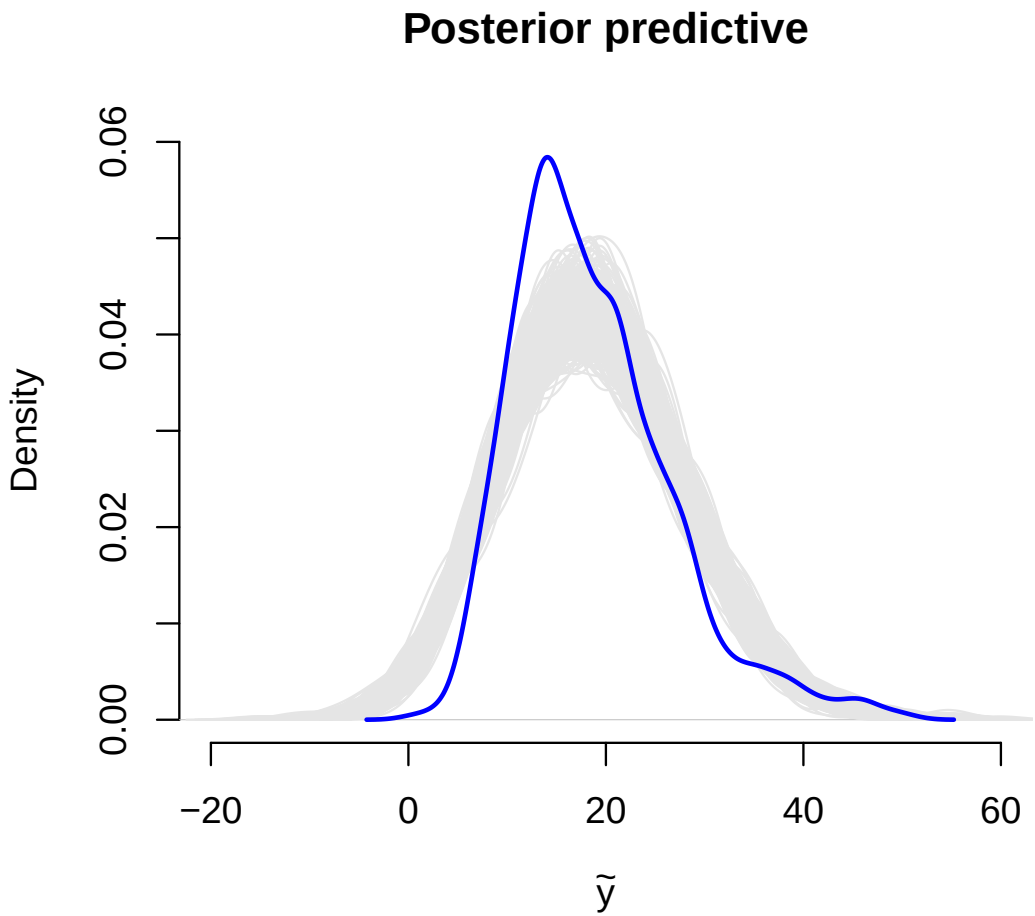

Figure S4.12: Posterior predictive.

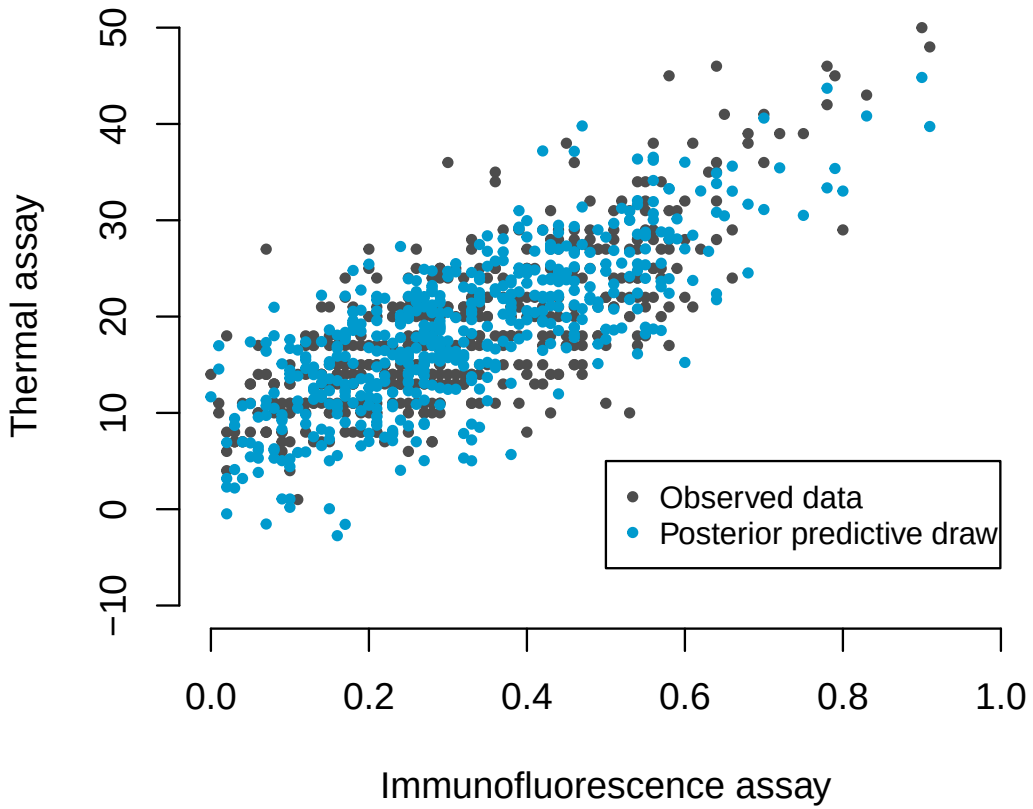

Figure S4.13: Posterior predictive draws (conditional on X).

Example 18.3: Sub-populations

- Having draws from the joint posterior distribution is quite helpful.
- Forming a credible interval for a subpopulation-specific mean (say, $X = 0.1$) is incomprehensibly easy.

```
## Credible interval for E[Y|X = 0.1]
submean.draws <- final.thetas[,1] + 0.1*final.thetas[,2]

> as.numeric(quantile(submean.draws, c(0.025, 0.975)))
[1] 9.948057 11.478941

## Analogous CI from OLS
C <- matrix(c(1, 0.1), nrow = 1)
submean.est <- C %*% coef(zz)
submean.se <- sqrt(C %*% vcov(zz) %*% t(C))

> c(submean.est - qnorm(0.975)*submean.se,
    submean.est + qnorm(0.975)*submean.se)
[1] 10.029 11.501
```

- This generalizes in the way you might expect. For instance, in logistic regression, you can get a posterior for an odds ratio by exponentiating the posterior draws for a difference in log-odds.