Bayesian Multiple Linear Regression

```
knitr::opts_chunk$set(message = F, warning = F, fig.width = 5, fig.height = 2)
library(mvtnorm) # install if necessary
library(tidyverse)
library(kableExtra)
```

```
hosp_dat <- readRDS("hosp_infection.Rda")
```

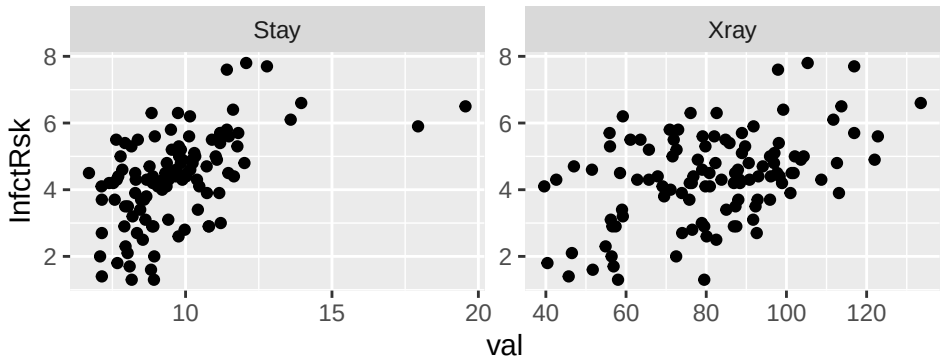
In the 1970s, the Centers for Disease Control conducted a nationwide study of hospitals to assess the current status of hospital based infections. The purpose of the study was to evaluate what hospitals were doing to prevent the spread of infection and whether those steps were helpful.The project was started out of a concern of whether the steps taken by individual infection control specialists in hospitals were actually reducing infections.

The data come from the Study on the Efficacy of Nosocomial Infection Control. It consists of a random sample of 113 hospitals from the original 338 hospitals surveyed. We are looking at only three variables from the study:

- `InfctRsk`: the average estimated probability of acquiring infection in hospital
- `Stay`: the average length of stay of all patients in the hospital (in days)
- `Xray`: ratio of number of X-rays performed:number of patients without signs/symptoms of pneumonia, times 100

Here's a snapshot of the data:

InfctRsk	Stay	Xray
4.1	7.13	39.6
1.6	8.82	51.7
2.7	8.34	74.0
5.6	8.95	122.8



Perhaps we'd like to understand how the infection risk is related to the length of stay and number of x-rays given in a hospital, so we fit the following regression model:

$$\text{InfctRisk}_i = \beta_0 + \beta_1 \text{Stay}_i + \beta_2 \text{Xray}_i + \epsilon_i, \qquad i = 1, \dots, n$$

where the $\epsilon_i \overset{iid}{\sim} N(0, \sigma^2)$ are iid.

Letting $Y_i = \text{InfctRisk}_i$, $\mathbf{x}_i = (1, \text{Stay}_i, \text{Xray}_i)'$, and $\vec{\beta} = (\beta_0, \beta_1, \beta_2)'$, this translates to the following sampling model:

$$y_i | \mathbf{x}_i, \vec{\beta}, \sigma^2 \overset{ind}{\sim} N(\mathbf{x}_i' \vec{\beta}, \sigma^2), \qquad i = 1, \dots, n$$

Prior option 1

We might choose the following (independent) priors:

$$\vec{\beta} \sim MVN_3(\vec{\mu}_0, \Sigma_0)$$
$$\frac{1}{\sigma^2} \sim \text{Gamma}(a, b)$$

And in particular, we might choose $\vec{\mu}_0 = (0, 0, 0)'$ (the scale of the Y_i are small, and *a priori* assume no relationship). We might also assume $\Sigma_0 = 100^2 \mathbf{I}_3$ (i.e. the coefficients are independent and have a s.d. of 100). Then, I will choose $a = b = 1$.

We can approximate the posteriors of $\vec{\beta}$ and σ^2 using a Gibbs sampler.

Data set-up

We need to do just a small amount of work to specify the data matrix $\mathbf{X}$, where each row corresponds to each observation i's predictors, and we have a column of 1's for the intercept:

```
# option 1 (manally add column of 1s for intercept)
X <- cbind(1, hosp_dat$Stay, hosp_dat$Xray)
head(X)
```

```
     [,1]  [,2]  [,3]
[1,]    1  7.13  39.6
[2,]    1  8.82  51.7
[3,]    1  8.34  74.0
[4,]    1  8.95 122.8
[5,]    1 11.20  88.9
[6,]    1  9.76  97.0
```

```
# option 2 (uses syntax like lm())
X <- model.matrix(~ Stay + Xray, hosp_dat)
head(X)
```

```
  (Intercept)  Stay  Xray
1           1  7.13  39.6
2           1  8.82  51.7
3           1  8.34  74.0
4           1  8.95 122.8
5           1 11.20  88.9
6           1  9.76  97.0
```

Then we set-up the rest of the necessary data components:

```
y <- hosp_dat$InfctRsk
n <- length(y)
p <- ncol(X) # sorry, this is p + 1 in our notes, but it's easier to call it p!
```

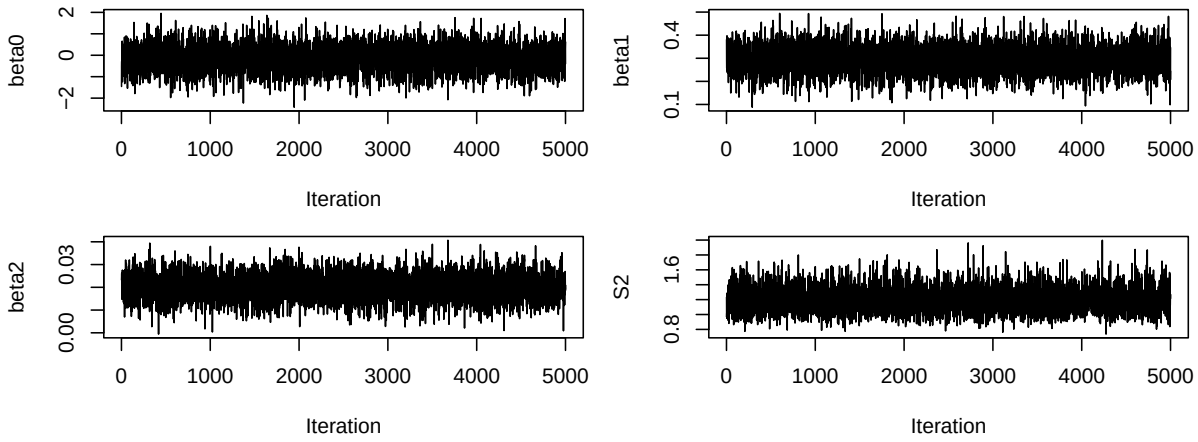
Running Gibbs sampler

```
set.seed(12)
# prior mean vector mu0
mu0 <- rep(0, p)

# prior mean covariance Sigma0
Sigma0 <- 100^2 * diag(p)
## more flexible coding-wise: Sigma0 <- c(100^2, 100^2 , 100^2) * diag(p)

## terrible but works:
# Sigma0 <- matrix(c(100^2, 0,0, 0, 100^2, 0, 0,0,100^2), nrow = p, ncol = p, byrow = T)

a <- 1; b <- 1
G <- 10000
BETAS <- matrix(NA, nrow = G, ncol = p)
S2 <- rep(NA, G)

# initialize sigma2
s2 <- a/b
for(g in 1:G){
  ## update beta

  # solve() takes inverse of matrix, %*% does matrix multiplication, t() takes transpose
  V <- solve(t(X)%*%X/s2 + solve(Sigma0))
  m <- V %*% (t(X) %*% y/s2 + solve(Sigma0) %*% mu0)
  beta_vec <- t(rmvnorm(1, m, V))
    # Syntax: rmvnorm(n, mean vector, covariance matrix)
  # rmvnorm() defaults to returning a row vector, so we take t() to get column

  ## update sigma2
  ssr <- t(y - X %*% beta_vec) %*% (y - X %*% beta_vec) # ssr = sum of squared residuals
  s2 <- 1/rgamma(1, a + 0.5*n, b + 0.5*ssr)

  # store
  BETAS[g,] <- beta_vec
  S2[g] <- s2
}
# burn
BETAS <- BETAS[-(1:G/2),]
S2 <- S2[-(1:G/2)]
```

Diagnostics + Model assessment

Check traceplots to assess convergence:

```
par(mfrow = c(2,2), mar = c(4,4,1,1))
for(i in 1:p){
  plot(BETAS[,i], type = "l", ylab = paste0("beta", i-1), xlab = "Iteration")
}
plot(S2, type = "l", xlab = "Iteration")
```




```
library(coda)
apply(BETAS, 2, effectiveSize)
```

```
[1] 5000.000 5000.000 4780.791
```

```
effectiveSize(S2)
```

```
    var1
4727.705
```

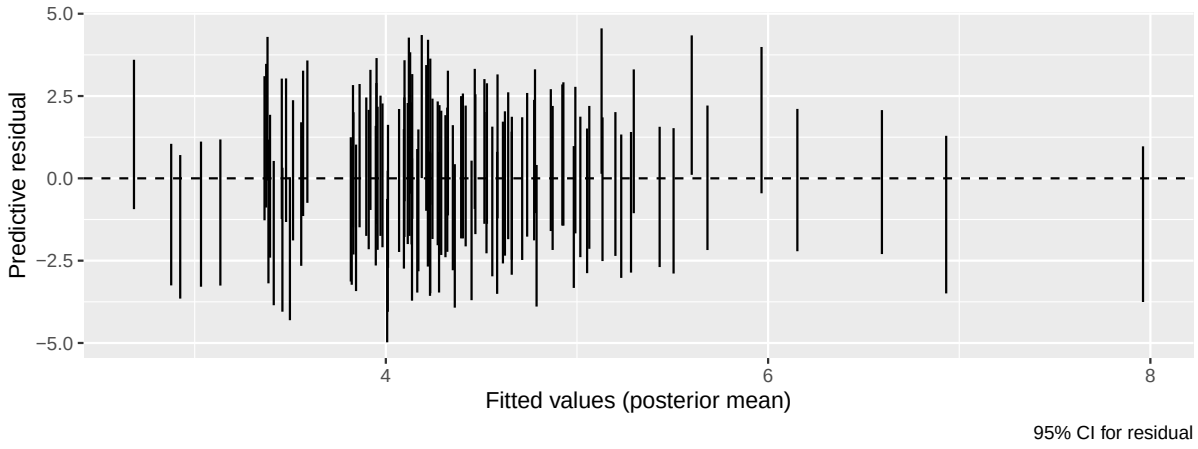
How does this compare to ESS from SLR? Why do we think that is?

Check sample lag-2 autocorrelation:

```
          lag-0         lag-1         lag-2
Intercept     1  0.0147251768  0.0005496842
stay          1 -0.0005619647 -0.0087501308
xray          1  0.0223121661 -0.0079189967
s2            1  0.0278917424 -0.0066139794
```

Time to check the residuals. When we had SLR, we plotted residuals against the explanatory variable. However, in the case of MLR we have more the one predictor. So here, we plot the posterior mean of the fitted/estimated values $\hat{y}_i$ on the x-axis (you're welcome for the code!):

```
set.seed(1)
resids_ls <- list()
for(i in 1:n){
  fitted <- mean(BETAS %*% X[i,]) # posterior mean fitted
  y_preds_i <- rnorm(G/2, BETAS %*% X[i,], sqrt(S2)) # post_pred
  resids <- y[i] - y_preds_i # residual
  resids_ls[[i]] <- data.frame(resid = resids, obs = i, fitted = fitted)
}
resids_df_all <- do.call(rbind, resids_ls)
resids_df_all |>
  group_by(obs, fitted) |> # grouping by obs, and keep fitted for later
  summarise(lb = quantile(resid, 0.025), ub = quantile(resid, 0.975)) |>
  ungroup() |>
  ggplot() +
  geom_segment(aes(x = fitted, y = lb, xend = fitted, yend = ub),
               position = position_jitter(height = 0, width = 0.1)) +
  labs(x = "Fitted values (posterior mean)", y = "Predictive residual",
       caption = "95% CI for residual") +
  geom_hline(yintercept = 0, linetype = "dashed")
```



What do we think about our choice of model?

Posterior inference

The following displays posterior mean and 95% credible intervals:

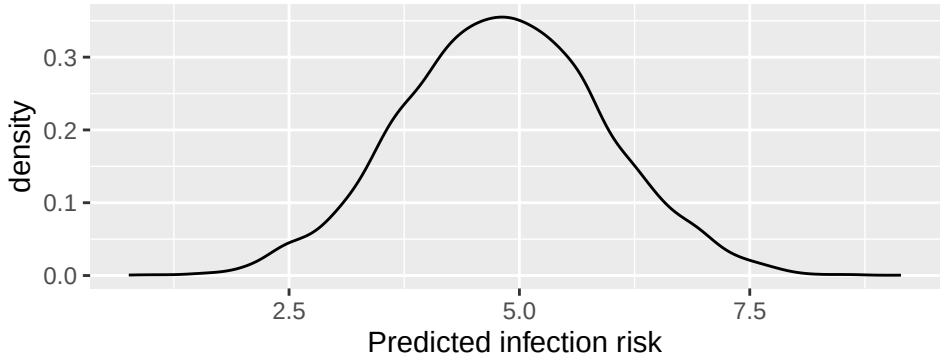
	Posterior mean	2.5%	97.5%
β_0	-0.148	-1.329	1.015
β_{stay}	0.296	0.177	0.408
β_{xray}	0.020	0.009	0.031
σ	1.089	0.953	1.244

Let's give some interpretation! Do you think the length of stay and x-ray ratio are helpful in explaining the probability of getting an infection while in the hospital?

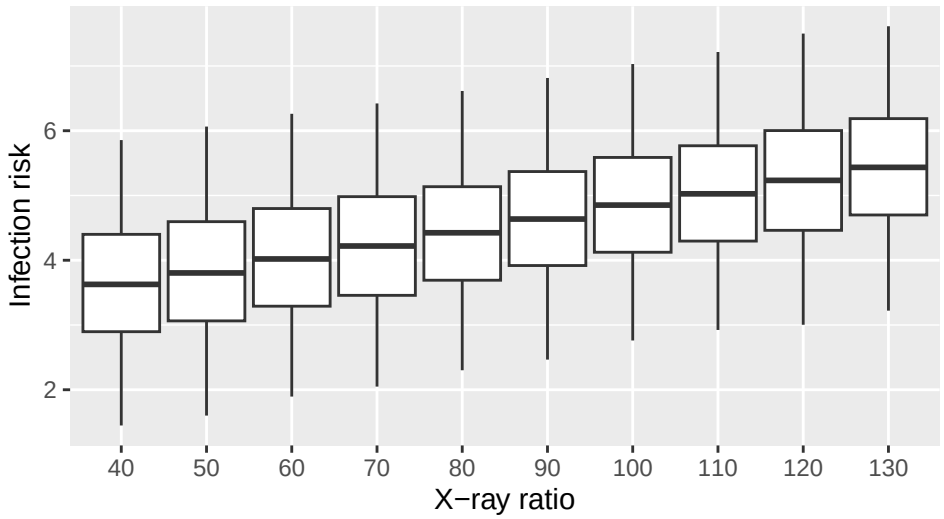
Posterior predictions

If we have another hospital whose average length of `Stay` is 10 days `Xray` ratio is 100, we can easily obtain predictions of the infection risk as we did in SLR. Repeat the following many times:

1. Use the posterior samples of $\beta_0, \beta_1, \beta_3$ to simulate the expected infection risk at a given level $\mathbf{x}^{(new)} = (1, 10, 100)$.
2. Randomly sample the infection risk at this mean using the Normal model and a posterior sample of σ^2



Sometimes, when we have multiple variables, we might want to visually inspect how changing one variable effects the predicted response, holding others constant. *This is especially usual when we have interactions, polynomial predictors, or a generalized linear regression model.* In the following, we display uncertainty in our posterior predicted infection risks at different levels of `Xray` for fixed `Stay` of 10 days:



0.025 − 0.25 − 0.5 − 0.75 − 0.975 posterior predictive percentiles