Assignment 11.a for **STATGR6103**

submitted to Professor Andrew Gelman

Advait Rajagopal

21 November 2016

1 Question 1

Consider a model for the crop yields data in Table 8.4 with a mean level, row effects, column effects, and treatment effects. Assign a hierarchical model with mean 0 and unknown variance to each of the three batches of effects.

1.1 Part A

Write the model in statistical notation, specifying all aspects of the model unambiguously. (Write the model directly, not as a regression.)

The question requires us to account for all aspects of the model. That is, we are to include mean level, row level effects, column level effects and treatment effects. There are 5 possible rows, columns and treatments and the data is organized into a Latin Square Design in this manner. There are 25 data points in total showing the crop yield for a single treatment applied to a row - column combination. My model estimates crop yield as a function of the mean level of crop yield, row effects, column effects and the treatment which is 'spacing' between plots of land in inches.

The priors are given by the following equations;

$$p(\alpha) \sim \mathcal{N}(250, 100^2) = \frac{1}{100\sqrt{2\pi}} \exp\Big(\frac{-(\alpha - 250)^2}{2 * 100^2}\Big) \tag{1}$$

$$p(\beta_t) \sim \mathcal{N}(0, \sigma^2_{\beta_t}) = \prod_{k=1}^{5} \frac{1}{\sigma_{\beta_t}\sqrt{2\pi}} \exp\Big(\frac{-\beta^2_{t_k}}{2\sigma^2_{\beta_t}}\Big) \tag{2}$$

$$p(\beta_r) \sim \mathcal{N}(0, \sigma^2_{\beta_r}) = \prod_{m=1}^{5} \frac{1}{\sigma_{\beta_r}\sqrt{2\pi}} \exp\Big(\frac{-\beta^2_{r_m}}{2\sigma^2_{\beta_r}}\Big) \tag{3}$$

$$p(\beta_c) \sim \mathcal{N}(0, \sigma^2_{\beta_c}) = \prod_{n=1}^{5} \frac{1}{\sigma_{\beta_c}\sqrt{2\pi}} \exp\Big(\frac{-\beta^2_{c_n}}{2\sigma^2_{\beta_c}}\Big) \tag{4}$$

We assume noninformative distributions for the variance hyperparameters. The priors on the hyperparameters are given by;

$$p(\sigma_t) \propto 1$$

$$p(\sigma_{\beta_t}) \propto 1$$

$$p(\sigma_{\beta_r}) \propto 1$$

$$p(\sigma_{\beta_c}) \propto 1$$

The joint likelihood is given by the equation;

$$\begin{aligned} p(y|\alpha,\beta_t,\beta_r,\beta_c,\sigma_{\beta_t},\sigma_{\beta_r},\sigma_{\beta_c}) &\sim \mathcal{N}(\alpha+\beta_t+\beta_r+\beta_c,\sigma) \\ &= \prod_{i=1}^{25}\frac{1}{\sigma\sqrt{2\pi}}\exp\Big(\frac{-(y_i-\alpha-\beta_t-\beta_r-\beta_c)^2}{2\sigma^2}\Big) \end{aligned} \tag{5}$$

1.2 Part B

Write the joint posterior density (up to an arbitrary multiplicative constant).
From equations 1, 2, 3, 4 and 5 we obtain the posterior density as described by the equation below,

$$\begin{aligned} p(\alpha,\beta_t,\beta_r,\beta_c,\sigma_{\beta_t},\sigma_{\beta_r},\sigma_{\beta_c}|y) &\propto p(\alpha)p(\beta_t)p(\beta_r)p(\beta_c)p(y|\alpha,\beta_t,\beta_r,\beta_c,\sigma_{\beta_t},\sigma_{\beta_r},\sigma_{\beta_c}) \\ &= 0.004*\exp\Big(\frac{-(\alpha-250)^2}{2*100^2}\Big) \\ &\times\prod_{k=1}^{5}\frac{1}{\sigma_{\beta_t}\sqrt{2\pi}}\exp\Big(\frac{-\beta_{t_k}^2}{2\sigma_{\beta_t}^2}\Big) \\ &\times\prod_{m=1}^{5}\frac{1}{\sigma_{\beta_r}\sqrt{2\pi}}\exp\Big(\frac{-\beta_{r_m}^2}{2\sigma_{\beta_r}^2}\Big) \\ &\times\prod_{n=1}^{5}\frac{1}{\sigma_{\beta_c}\sqrt{2\pi}}\exp\Big(\frac{-\beta_{c_n}^2}{2\sigma_{\beta_c}^2}\Big) \\ &\times\prod_{i=1}^{25}\frac{1}{\sigma\sqrt{2\pi}}\exp\Big(\frac{-(y_i-\alpha-\beta_t-\beta_r-\beta_c)^2}{2\sigma^2}\Big) \\ &= (\sqrt{2\pi})^{-40}\sigma_{\beta_t}^{-5}\sigma_{\beta_r}^{-5}\sigma_{\beta_c}^{-5}\sigma^{-25} \\ &\times 0.004*\exp\Big(\frac{-(\alpha-250)^2}{2*100^2}\Big) \\ &\times\prod_{k=1}^{5}\exp\Big(\frac{-\beta_{t_k}^2}{2\sigma_{\beta_t}^2}\Big) \\ &\times\prod_{m=1}^{5}\exp\Big(\frac{-\beta_{r_m}^2}{2\sigma_{\beta_r}^2}\Big) \\ &\times\prod_{n=1}^{5}\exp\Big(\frac{-\beta_{c_n}^2}{2\sigma_{\beta_c}^2}\Big) \\ &\times\prod_{i=1}^{25}\exp\Big(\frac{-(y_i-\alpha-\beta_t-\beta_r-\beta_c)^2}{2\sigma^2}\Big) \end{aligned}$$

1.3 Part C

Fit the model in Stan.
I use equation 5 to fit the regression in Stan. However to estimate the posterior densities I use the logarithm of the yield and regress it on an additive linear model of the mean level, treatment, row and column effects. In order to achieve this I use the following transformation;

$$p(y) \sim \mathcal{N}(\mu,\sigma)$$

Let z = log(y), then using Taylor expansions,

$$p(z) \sim \mathcal{N}(\log(\mu)-\sigma^2/2\mu^2,\sigma^2/\mu^2)$$

I also make sure to restrict the mean 'μ' to be positive so that the logarithm of the mean is defined and the HMC algorithm does not evaluate the mean at 0.

1.4 Part D

Display the inferences from the model.

Table 1 shows the posterior estimates of the parameters and hyperparameters from Stan. Figure 1 shows the posterior distributions of the individual level treatment, row and column effects.

Table 1: Posterior Distributions of Parameters and Hyperparameters

	mean	se.mean	sd	2.5%	25%	50%	75%	97.5%	n.eff	Rhat
α	5.52	0.02	0.09	5.35	5.49	5.52	5.59	5.17	148	1.04
β_A	0.03	0.00	0.06	-0.05	0.00	0.02	0.05	0.16	291	1.01
β_B	0.02	0.00	0.05	-0.07	-0.01	0.01	0.04	0.15	328	1.01
β_C	0.00	0.00	0.05	-0.10	-0.02	0.00	0.02	0.11	276	1.01
β_D	-0.02	0.00	0.05	-0.15	-0.04	-0.01	0.00	0.06	320	1.01
β_E	-0.03	0.00	0.06	-0.17	-0.06	-0.02	0.00	0.06	348	1.01
σ_{β_t}	0.07	0.00	0.06	0.01	0.02	0.05	0.08	0.23	204	1.02
σ_{β_r}	0.08	0.00	0.06	0.01	0.04	0.06	0.10	0.23	254	1.00
σ_{β_c}	0.13	0.01	0.09	0.01	0.07	0.11	0.16	0.36	143	1.02

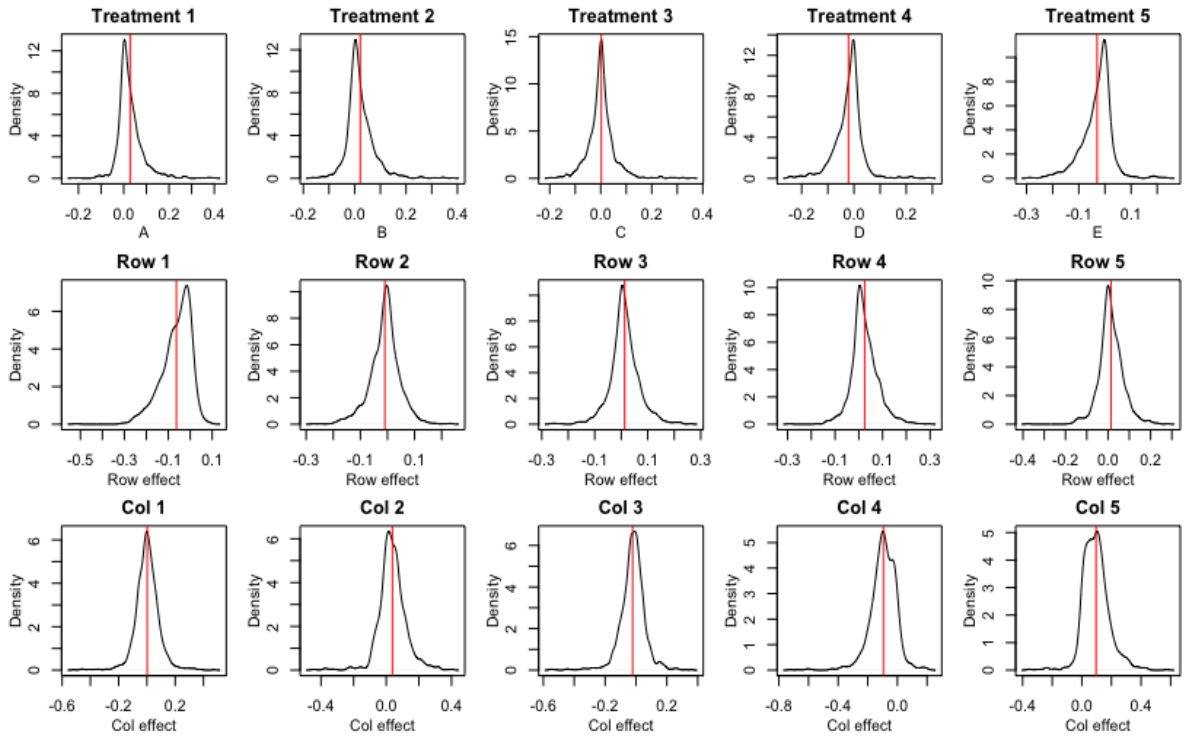

Figure 1: Marginal posterior densities of each of the treatment, row and column level effects, with the mean of the distribution in red.

1.5 Part E

Make a graph showing the data and fitted model.

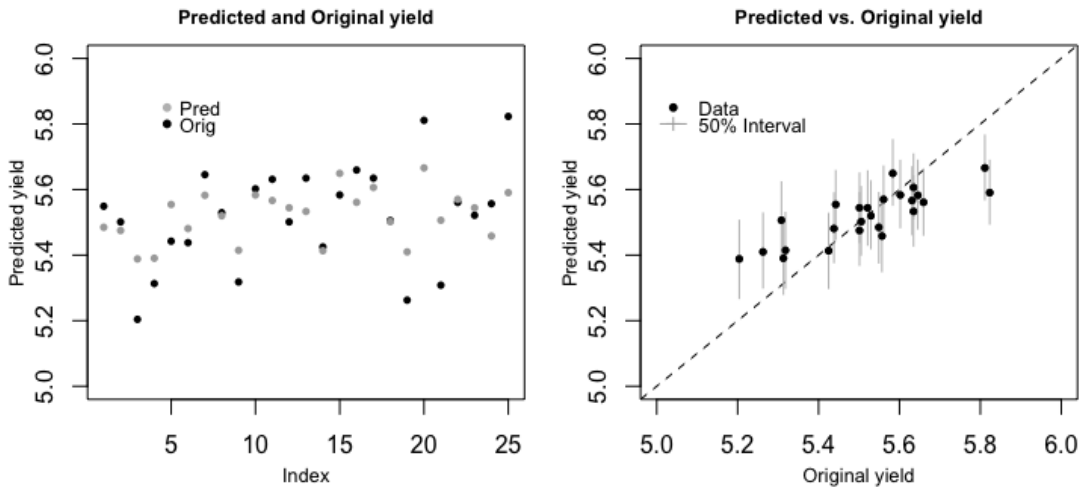

Figure 2: (a) Shows the logarithm of the original crop yields in black and the predicted yields in gray, (b) Shows that the model predicts the crop yields quite accurately and the gray lines show the 50% posterior interval of the predicted values.

It is clear from Figure 2 that the regression model fits the data pretty well and that the predictions are close to the original values. The model under predicts extreme values slightly and over predicts very low values as is expected because we use the mean level effect in the intercept and moreover we use priors on the treatment, row and column level effects with mean 0 and unknown variance. This centers these three batches of effects around zero. It is worth relaxing this and estimating a full hierarchical model allowing the level effects to have their own mean and variance and estimate these using Stan.

2 Code

2.1 R code

Listing 1: R code

```
rm(list = ls())
setwd("/Users/Advait/Desktop/New_School/Fall16/BDA/Class20")
library(rstan)
rstan_options(auto_write = TRUE)
options(mc.cores = parallel::detectCores())
install.packages("dplyr")
library(dplyr)
#
y <- c(257, 245,182,203,231,
       230,283,252,204,271,
       279,245,280,227,266,
       287,280,246,193,334,
       202,260,250,259,338)

treat <- c(2,4,5,1,3,
           5,1,2,3,4,
           1,5,3,4,2,
           3,2,4,5,1,
           4,3,1,2,5)

row_level <- c(rep(1,5),rep(2,5),
                    rep(3,5),rep(4,5),rep(5,5))
col_level <- (rep(seq(1:5),5))

df <- cbind(y,treat,row_level,col_level)
df <- as.data.frame(df)
class(df)
ylog <- log(y)
stanc("11a_trial.stan")
fit <- stan("11a_trial.stan",
            data = list("ylog", "treat", "row_level", "col_level"),
            iter = 1000, chains = 3)
print(fit)
ext <- extract(fit)

y_rep <- ext$y_rep_log
str(y_rep)
class(y_rep)
```

Listing 2: R code contd.

```
upper <- NULL
lower <- NULL
for(i in 1:25){
  upper[i] = quantile(y_rep[,i], probs = 0.75)
  lower[i] = quantile(y_rep[,i], probs = 0.25)
}
par(mfrow = c(1, 2),
    mar=c(3, 3, 2, 1),
    oma = c(0.5, 0.5, 0.5, 0.5), mgp=c(2, 1, 0))

plot(c(1:25), ylog, pch = 16, cex = 0.7, ylab = "Predicted␣yield",
     xlab = "Index", cex.lab = 0.8,
     main = "Predicted␣and␣Original␣yield",
     cex.main = 0.8,ylim = c(5,6))
points(c(1:25),colMeans(ext$y_rep_log),
pch = 16, col = "darkgray", cex = 0.7 )
legend(4,5.9, legend = c("Pred", "Orig"),
       col = c("gray", "black"),
       bty = "n",
       cex = 0.8, pch = 16)

plot(ylog, colMeans(ext$y_rep_log),
     pch = 16,
     ylim = c(5, 6),
     xlim = c(5, 6),
     cex = 0.7, ylab = "Predicted␣yield",
     xlab = "Original␣yield", cex.lab = 0.8,
     main = "Predicted␣vs.␣Original␣yield", cex.main = 0.8)
abline(0, 1, lty = 2)
arrows(ylog, colMeans(ext$y_rep_log),
ylog, upper, col = "gray",length = 0 )
arrows(ylog, colMeans(ext$y_rep_log),
ylog, lower, col = "gray",length = 0 )
points(ylog, colMeans(ext$y_rep_log),
       pch = 16, cex = 0.7)
legend(5, 5.9, legend = c("Data", "50\%␣Interval"),
       col = c("black" , "darkgrey") , pch=c(16, 3),
       lty = c(0, 1),
       bty = 'n',
       cex = 0.8)
```

Listing 3: R code contd.

```
##Part D
par(mfrow = c(3, 5),
    mar=c(3, 3, 2, 1),
    oma = c(0.5, 0.5, 0.5, 0.5), mgp=c(2, 1, 0))
plot(density(ext$bt[,1]), main = "Treatment 1",
     xlab = "A")
abline(v = mean(ext$bt[,1]), col = "red")
#
plot(density(ext$bt[,2]), main = "Treatment 2",
     xlab = "B")
abline(v = mean(ext$bt[,2]), col = "red")
#
plot(density(ext$bt[,3]), main = "Treatment 3",
     xlab = "C")
abline(v = mean(ext$bt[,3]), col = "red")
#
plot(density(ext$bt[,4]), main = "Treatment 4",
     xlab = "D")
abline(v = mean(ext$bt[,4]), col = "red")
#
plot(density(ext$bt[,5]), main = "Treatment 5",
     xlab = "E")
abline(v = mean(ext$bt[,5]), col = "red")
#######
plot(density(ext$br[,1]), main = "Row 1",
xlab = "Row effect")
abline(v = mean(ext$br[,1]), col = "red")
#
plot(density(ext$br[,2]), main = "Row 2",
xlab = "Row effect")
abline(v = mean(ext$br[,2]), col = "red")
#
plot(density(ext$br[,3]), main = "Row 3",
xlab = "Row effect")
abline(v = mean(ext$br[,3]), col = "red")
#
plot(density(ext$br[,4]), main = "Row 4",
xlab = "Row effect")
abline(v = mean(ext$br[,4]), col = "red")
#
plot(density(ext$br[,5]), main = "Row 5",
xlab = "Row effect")
abline(v = mean(ext$br[,5]), col = "red")
```

Listing 4: R code contd.

```
######
plot(density(ext$bc[,1]), main = "Col␣1",
xlab = "Col␣effect")
abline(v = mean(ext$bc[,1]), col = "red")
#
plot(density(ext$bc[,2]), main = "Col␣2",
xlab = "Col␣effect")
abline(v = mean(ext$bc[,2]), col = "red")
#
plot(density(ext$bc[,3]), main = "Col␣3",
xlab = "Col␣effect")
abline(v = mean(ext$bc[,3]), col = "red")
#
plot(density(ext$bc[,4]), main = "Col␣4",
xlab = "Col␣effect")
abline(v = mean(ext$bc[,4]), col = "red")
#
plot(density(ext$bc[,5]), main = "Col␣5",
xlab = "Col␣effect")
abline(v = mean(ext$bc[,5]), col = "red")
```

2.2 Stan Code

Listing 5: Stan code

```
data{
  vector[25] ylog;
  int treat[25];
  int row_level[25];
  int col_level[25];
}
parameters{
  real <lower = 1> a;
  real bt[5];
  real br[5];
  real bc[5];
  real <lower = 0> sigma;
  real <lower = 0> sig_bt;
  real <lower = 0> sig_br;
  real <lower = 0> sig_bc;
}
model{
  for(i in 1:25){
    ylog[i] ~ normal(log(a + bt[treat[i]]
                + br[row_level[i]]
                + bc[col_level[i]]) - (pow(sigma,2)/(a + bt[treat[i]]
                + br[row_level[i]]
                + bc[col_level[i]])), sigma/sqrt((a + bt[treat[i]]
                + br[row_level[i]]
                + bc[col_level[i]])));
}
a ~ normal(250,50);
bt ~ normal(0, sig_bt);
br ~ normal(0, sig_br);
bc ~ normal(0, sig_bc);
}
generated quantities{
  vector[25] y_rep_log;
  for(i in 1:25){
    y_rep_log[i] = normal_rng(log(a + bt[treat[i]]
                + br[row_level[i]]
                + bc[col_level[i]]) - (pow(sigma,2)/(a + bt[treat[i]]
                + br[row_level[i]]
                + bc[col_level[i]])), sigma/sqrt((a + bt[treat[i]]
                + br[row_level[i]]
                + bc[col_level[i]])));
  }
}
```