STAT 578 - Advanced Bayesian Modeling - Fall 2019 Assignment 5

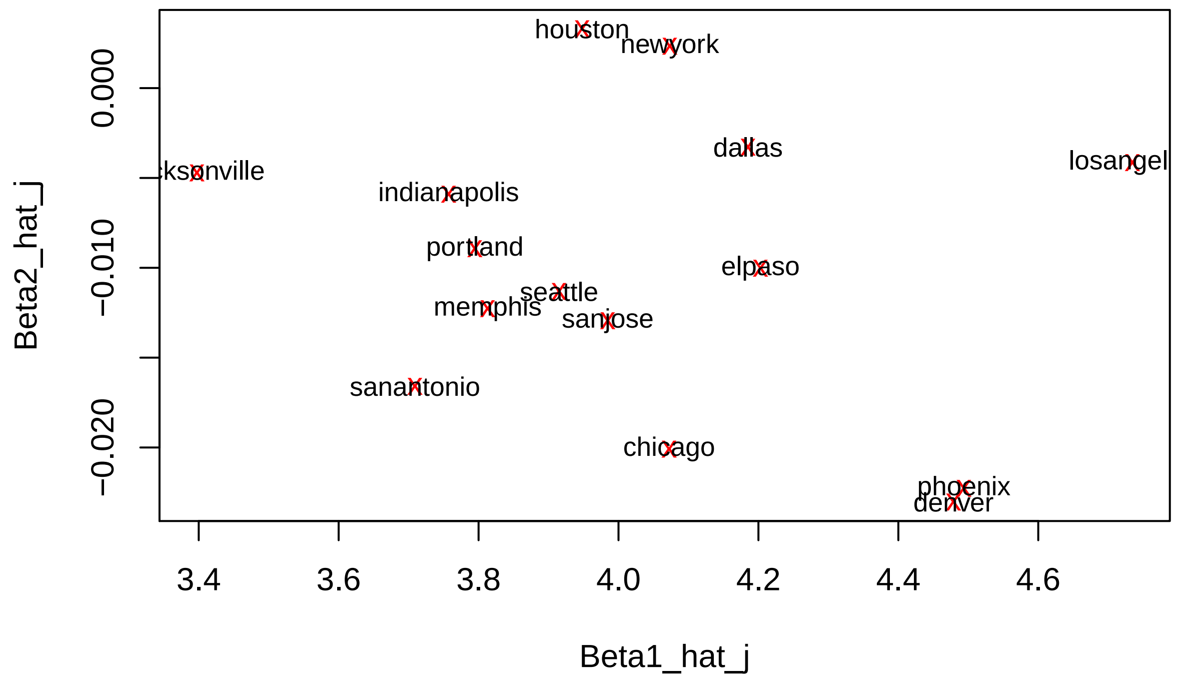
Xiaoming Ji

Solution for Problem (a)

```
AQI = log(as.matrix(read.csv("ozoneAQIaug.txt", sep="", header=TRUE)))

city_count = dim(AQI)[1]
day_count = dim(AQI)[2]
beta_j_hat = array(0, c(city_count,2))
day_cent = (1:day_count) - mean(1:day_count)
X = as.matrix(data.frame(x1=1, x2=day_cent))

X_hat = solve(t(X) %*% X) %*% t(X)

for (j in 1:city_count) {
  beta_j_hat[j,] = X_hat %*% AQI[j,]
}
```

(i)

```
plot(beta_j_hat, xlab="Beta1_hat_j", ylab="Beta2_hat_j", col="red", pch="x")
for (i in 1:dim(beta_j_hat)[1]){
  text(beta_j_hat[i,1], beta_j_hat[i,2], rownames(AQI)[i], cex = .8)
}
```



(ii)

```
apply(beta_j_hat,2,mean)
```

```
## [1]  4.037318857 -0.009999473
```

(iii)

```
apply(beta_j_hat,2,var)
```

```
## [1] 1.194059e-01 6.683493e-05
```

(iv)

```
cor(beta_j_hat[,1],beta_j_hat[,2])
```

```
## [1] -0.2319396
```

Solution for Problem (b)

(i)

```
data {
    dim_y <- dim(AQI)
    day_cent <- day - mean(day)
}
model {
    for (j in 1:dim_y[1]) {
        for (i in 1:dim_y[2]) {
            AQI[j,i] ~ dnorm(beta[1,j] + beta[2,j] * day_cent[i], sigma_sq_y_inv)
        }
        beta[1:2,j] ~ dmnorm(mu_beta, sigma_beta_inv)
    }
    mu_beta ~ dmnorm(mu_beta_0, sigma_mu_beta_inv)
    sigma_beta_inv ~ dwish(2 * sigma_0, 2)
    sigma_sq_y_inv ~ dgamma(0.0001, 0.0001)
    sigma_beta <- inverse(sigma_beta_inv)
    rho <- sigma_beta[1,2] / sqrt(sigma_beta[1,1] * sigma_beta[2,2])
    sigma_sq_y <- 1 / sigma_sq_y_inv
}
```

```
library(rjags)

df_jags_1 <- list(AQI = AQI,
                day = 1:31,
                mu_beta_0 = c(0, 0),
                sigma_mu_beta_inv = rbind(c(1/(1000^2), 0),
                                          c(0, 1/(1000^2))),
                sigma_0 = rbind(c(0.1, 0),
                                c(0, 0.001)))

initial_vals_1 <- list(list(sigma_sq_y_inv = 1000, mu_beta = c(1000, 10),
                            sigma_beta_inv = rbind(c(100, 0),
                                                   c(0, 100))),
                       list(sigma_sq_y_inv = 0.001, mu_beta = c(-1000, 10),
                            sigma_beta_inv = rbind(c(100, 0),
                                                   c(0, 100))),
                       list(sigma_sq_y_inv = 1000, mu_beta = c(1000, -10),
                            sigma_beta_inv = rbind(c(0.001, 0),
                                                   c(0, 0.001))),
                       list(sigma_sq_y_inv = 0.001, mu_beta = c(-1000, -10),
                            sigma_beta_inv = rbind(c(0.001, 0),
                                                   c(0, 0.001))))
model_1 <- jags.model("aqi_1.bug", df_jags_1, initial_vals_1, n.chains = 4,
                      n.adapt = 1000)
update(model_1, 10000)
coef_sample_1 <- coda.samples(model_1, c("mu_beta","sigma_beta","sigma_sq_y","rho"),
                              n.iter = 2000)
```

```
gelman.diag(coef_sample_1, autoburnin=FALSE, multivariate=FALSE)
```

```
## Potential scale reduction factors:
```

```
##
##                  Point est. Upper C.I.
## mu_beta[1]               1          1
## mu_beta[2]               1          1
## rho                      1          1
## sigma_beta[1,1]          1          1
## sigma_beta[2,1]          1          1
## sigma_beta[1,2]          1          1
## sigma_beta[2,2]          1          1
## sigma_sq_y               1          1
```

(ii)

```
summary(coef_sample_1)
```

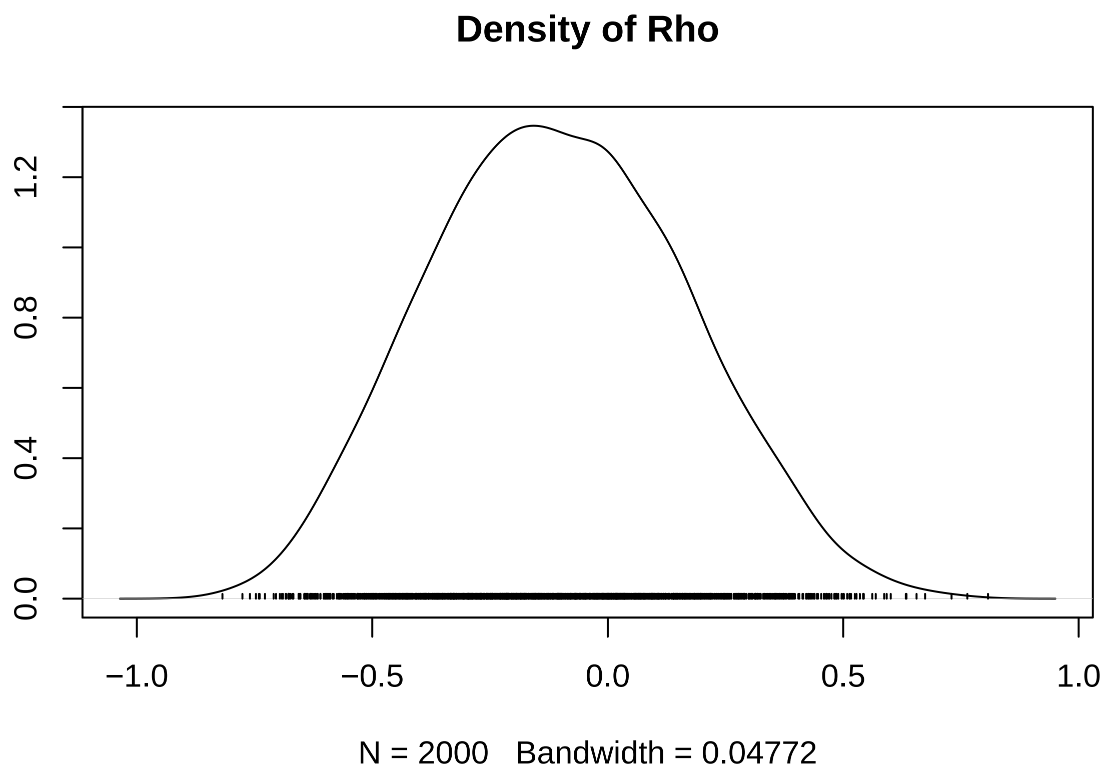
```
##
## Iterations = 10001:12000
## Thinning interval = 1
## Number of chains = 4
## Sample size per chain = 2000
##
## 1. Empirical mean and standard deviation for each variable,
##    plus standard error of the mean:
##
##                        Mean        SD  Naive SE Time-series SE
## mu_beta[1]       4.0377776 0.0987196 1.104e-03      1.122e-03
## mu_beta[2]      -0.0100186 0.0045636 5.102e-05      6.152e-05
## rho             -0.1038987 0.2716568 3.037e-03      4.111e-03
## sigma_beta[1,1]  0.1390760 0.0627538 7.016e-04      8.036e-04
## sigma_beta[2,1] -0.0006428 0.0019195 2.146e-05      2.823e-05
## sigma_beta[1,2] -0.0006428 0.0019195 2.146e-05      2.823e-05
## sigma_beta[2,2]  0.0002489 0.0001132 1.265e-06      1.474e-06
## sigma_sq_y       0.1540264 0.0105878 1.184e-04      1.251e-04
##
## 2. Quantiles for each variable:
##
##                       2.5%        25%        50%        75%      97.5%
## mu_beta[1]       3.8449796  3.9735753  4.0357352  4.1021252  4.2358035
## mu_beta[2]      -0.0191144 -0.0129811 -0.0100603 -0.0071147 -0.0008791
## rho             -0.6030241 -0.2992694 -0.1108556  0.0886591  0.4303472
## sigma_beta[1,1]  0.0624312  0.0976068  0.1252654  0.1634047  0.2946494
## sigma_beta[2,1] -0.0047200 -0.0016206 -0.0005410  0.0004359  0.0027757
## sigma_beta[1,2] -0.0047200 -0.0016206 -0.0005410  0.0004359  0.0027757
## sigma_beta[2,2]  0.0001119  0.0001736  0.0002243  0.0002944  0.0005342
## sigma_sq_y       0.1343948  0.1465854  0.1535978  0.1611273  0.1759372
```

(iii)

```
rho_sample = as.matrix(coef_sample_1)[,"rho"]
quantile(rho_sample, c(0.025, 0.975))
```

```
##       2.5%      97.5%
## -0.6030241  0.4303472
```

```
densplot(coef_sample_1[, c("rho")], main = "Density of Rho")
```



(iv)

Approximate the posterior probability that $\rho > 0$.

```
mean(rho_sample > 0)
```

```
## [1] 0.354625
```

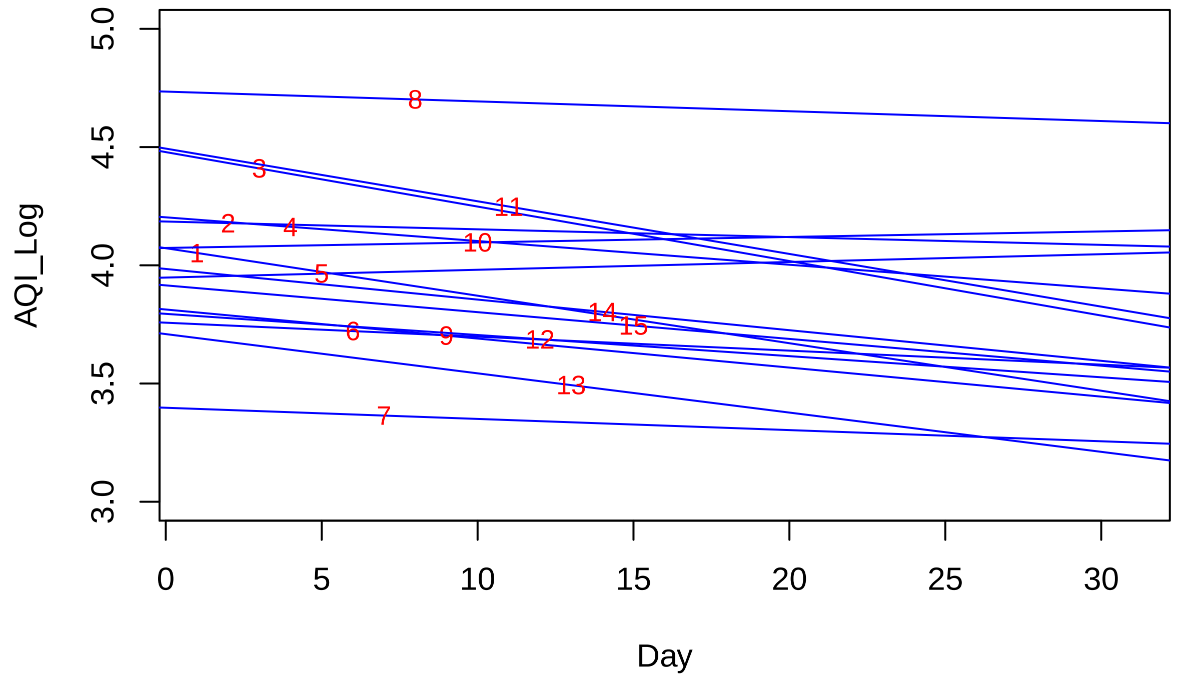
Bayes factor favoring $\rho > 0$ versus $\rho < 0$.

```
mean(rho_sample > 0) / mean(rho_sample < 0)
```

```
## [1] 0.5494867
```

From the 15 fitted (ordinary least squares) models in (a) and draw the fitted line as below. We can inspect some lines (for example:7) that start with lower intercepts also change more slowly than others (for example: 13), this means the correlation between intercept and slope would be positively related. Thus we can see this as evidence from data that ρ>0.

```
plot(1:31, AQI[2,], ylim=c(3,5),type="n", xlab = "Day", ylab = "AQI_Log")
for (i in 1:dim(beta_j_hat)[1]){
  abline(beta_j_hat[i,], col="blue")
  text(i, beta_j_hat[i,1] + beta_j_hat[i,2] * i, i, cex = 0.8, col = "red")
}
```



(v)

```
mu_beta_2 = as.matrix(coef_sample_1)[,"mu_beta[2]"]
quantile(exp(30 * mu_beta_2), c(0.025, 0.975))
```

```
##      2.5%     97.5%
## 0.5635877 0.9739732
```

(vi)

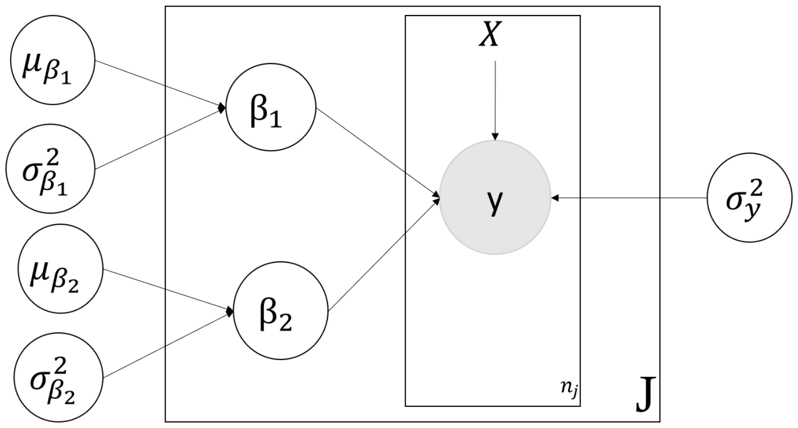
```
dic.samples(model_1, 100000)
```

```
## Mean deviance:  448.3
## penalty 27.57
## Penalized deviance: 475.9
```

Actual number of parameters is: 30 ($\beta^{(j)}s$) + 1 (σ_y^2) + 2 (μ_β) + 3 ($\sum_\beta$) = 36.

Solution for Problem (c)

(i)



(ii)

```
data {
    dim_y <- dim(AQI)
    day_cent <- day - mean(day)
}
model {
    for (j in 1:dim_y[1]) {
        for (i in 1:dim_y[2]) {
            AQI[j,i] ~ dnorm(beta[1,j] + beta[2,j] * day_cent[i], sigma_sq_y_inv)
        }
        beta[1,j] ~ dnorm(mu_beta[1], 1 / sigma_sq_beta[1])
        beta[2,j] ~ dnorm(mu_beta[2], 1 / sigma_sq_beta[2])
    }
    mu_beta[1] ~ dnorm(0, 1 / (1000 ^2))
    mu_beta[2] ~ dnorm(0, 1 / (1000 ^2))
    sigma_beta[1] ~ dunif(0, 1000)
    sigma_beta[2] ~ dunif(0, 1000)
    sigma_sq_beta[1] <- sigma_beta[1] ^ 2
    sigma_sq_beta[2] <- sigma_beta[2] ^ 2
    sigma_sq_y_inv ~ dgamma(0.0001, 0.0001)
    sigma_sq_y <- 1 / sigma_sq_y_inv
}
```

```
df_jags_2 <- list(AQI = AQI,
                  day = 1:31)

initial_vals_2 <- list(list(sigma_sq_y_inv = 1000, mu_beta = c(1000, 10),
                            sigma_beta = c(0.01, 0.01)),
                       list(sigma_sq_y_inv = 0.001, mu_beta = c(-1000, 10),
                            sigma_beta = c(0.01, 0.01)),
                       list(sigma_sq_y_inv = 1000, mu_beta = c(1000, -10),
                            sigma_beta = c(1000, 1000)),
                       list(sigma_sq_y_inv = 0.001, mu_beta = c(-1000, -10),
                           sigma_beta = c(1000, 1000)))
model_2 <- jags.model("aqi_2.bug", df_jags_2, initial_vals_2, n.chains = 4,
                      n.adapt = 1000)
update(model_2, 10000)
coef_sample_2 <- coda.samples(model_2, c("mu_beta","sigma_sq_beta","sigma_sq_y"),
                              n.iter = 2000)
```

```
gelman.diag(coef_sample_2, autoburnin=FALSE, multivariate=FALSE)
```

```
## Potential scale reduction factors:
##
##                  Point est. Upper C.I.
## mu_beta[1]             1.00       1.00
## mu_beta[2]             1.00       1.01
## sigma_sq_beta[1]       1.00       1.01
## sigma_sq_beta[2]       1.01       1.02
## sigma_sq_y             1.00       1.00
```

(iii)

```
summary(coef_sample_2)
```

```
##
## Iterations = 11001:13000
## Thinning interval = 1
## Number of chains = 4
## Sample size per chain = 2000
##
## 1. Empirical mean and standard deviation for each variable,
##    plus standard error of the mean:
##
##                        Mean        SD  Naive SE Time-series SE
## mu_beta[1]        4.034e+00 1.002e-01 1.121e-03      1.211e-03
## mu_beta[2]       -1.005e-02 2.410e-03 2.694e-05      1.012e-04
## sigma_sq_beta[1]  1.467e-01 7.201e-02 8.051e-04      1.239e-03
## sigma_sq_beta[2]  2.316e-05 3.039e-05 3.398e-07      1.424e-06
## sigma_sq_y        1.544e-01 1.046e-02 1.169e-04      1.218e-04
##
## 2. Quantiles for each variable:
##
##                        2.5%        25%        50%        75%      97.5%
## mu_beta[1]        3.833e+00  3.969e+00  4.033e+00  4.0999692  4.2299441
## mu_beta[2]       -1.506e-02 -1.168e-02 -1.008e-02 -0.0084105 -0.0052806
## sigma_sq_beta[1]  6.382e-02  9.895e-02  1.297e-01  0.1731151  0.3284009
## sigma_sq_beta[2]  9.784e-10  2.751e-06  1.232e-05  0.0000316  0.0001064
## sigma_sq_y        1.353e-01  1.471e-01  1.541e-01  0.1611769  0.1757813
```

(iv)

```
mu_beta_2 = as.matrix(coef_sample_2)[,"mu_beta[2]"]
quantile(exp(30 * mu_beta_2), c(0.025, 0.975))
```

```
##      2.5%     97.5%
## 0.6364768 0.8534939
```

Compared with previous results, this interval range is *smaller*.

(v)

```
dic.samples(model_2, 100000)
```

```
## Mean deviance:  449.4
## penalty 20.17
## Penalized deviance: 469.6
```

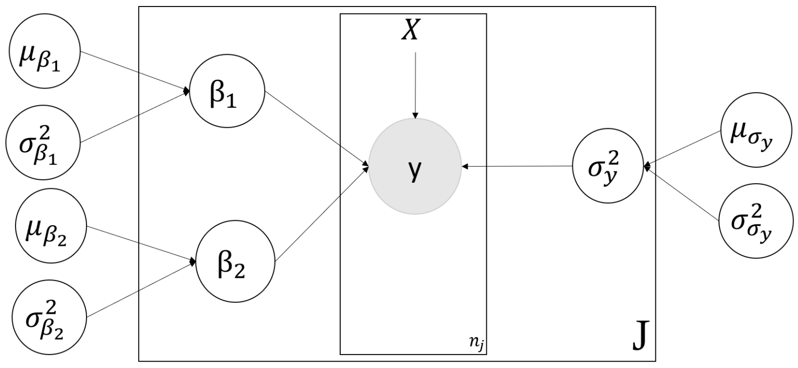
(vi)

This model gives us smaller DIC (Penalized deviance), thus we prefer this model.

Solution for Problem (d)

(i)

Yes. AQI is impacted by lot of factors (for example: industry type, population etc) that depend on city, it is very likely that the variability in log-value depends on the city. To better model this case, we could make $\sigma_y^{(j)}$ for each city and give it a normal distribution (as illustrated in the following DAG). This will add 16 parameters including 2 hyperparameters.



(ii)

Yes. It is possible that there are time-series correlations in the successive log-values of each city that are not captured by the simple linear regression model. Naturally, AQI of 2 consecutive day could be very close. This violate the assumption that responses from observations within a group should be exchangeable (i involved).

(iii)

Yes, It is possible that there are spatial correlations among the log-values on a given day that are not captured by the simple linear regression model. This violate the assumption that the response should be independent between groups (j involved).