STAT 578 - Advanced Bayesian Modeling - Fall 2019 Assignment 5

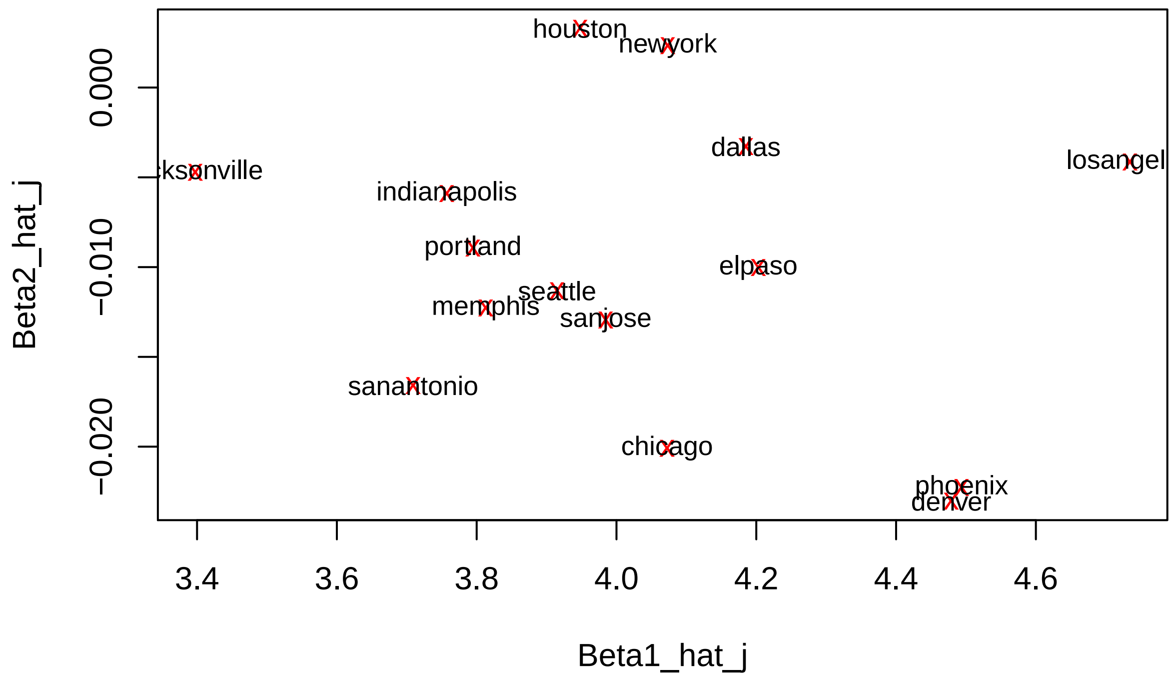
Xiaoming Ji

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)

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

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

(iii)

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

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

(iv)

```r
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
}
```

```r
library(rjags)

df_jags <- 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 <- 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, initial_vals, 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)
```

```r
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)

```r
summary(coef_sample_1)
```

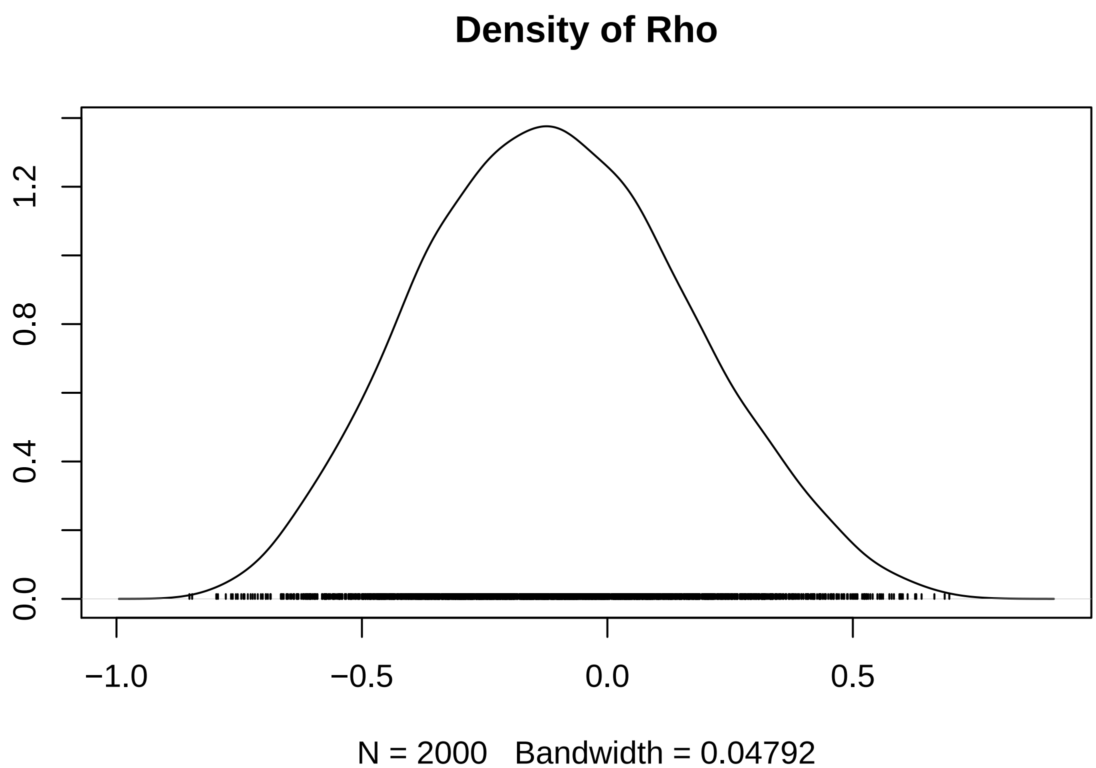
```
##
## Iterations = 10001:12000
## Thinning interval = 1
## Number of chains = 4
## Sample size per chain = 2000
##
## 1. Empirical mean and standard deviation for each variable,
##    plus standard error of the mean:
##
##                       Mean        SD  Naive SE Time-series SE
## mu_beta[1]       4.0356468 0.0989349 1.106e-03      1.134e-03
## mu_beta[2]      -0.0100588 0.0045094 5.042e-05      6.294e-05
## rho             -0.1053954 0.2727672 3.050e-03      3.937e-03
## sigma_beta[1,1]  0.1394865 0.0624308 6.980e-04      7.857e-04
## sigma_beta[2,1] -0.0006491 0.0019021 2.127e-05      2.667e-05
## sigma_beta[1,2] -0.0006491 0.0019021 2.127e-05      2.667e-05
## sigma_beta[2,2]  0.0002485 0.0001119 1.251e-06      1.458e-06
## sigma_sq_y       0.1539469 0.0103462 1.157e-04      1.176e-04
##
## 2. Quantiles for each variable:
##
##                       2.5%        25%        50%        75%      97.5%
## mu_beta[1]       3.8419397  3.9721410  4.0354782  4.0989405  4.2352900
## mu_beta[2]      -0.0188769 -0.0130285 -0.0100628 -0.0071577 -0.0010091
## rho             -0.6107540 -0.3017590 -0.1127901  0.0819587  0.4444140
## sigma_beta[1,1]  0.0623631  0.0974568  0.1255717  0.1654197  0.2971477
## sigma_beta[2,1] -0.0047245 -0.0016142 -0.0005563  0.0004104  0.0028816
## sigma_beta[1,2] -0.0047245 -0.0016142 -0.0005563  0.0004104  0.0028816
## sigma_beta[2,2]  0.0001111  0.0001729  0.0002222  0.0002949  0.0005363
## sigma_sq_y       0.1351546  0.1468468  0.1534342  0.1606881  0.1757525
```

(iii)

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

```
##      2.5%     97.5%
## -0.610754  0.444414
```

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



(iv)

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

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

```
## [1] 0.3495
```

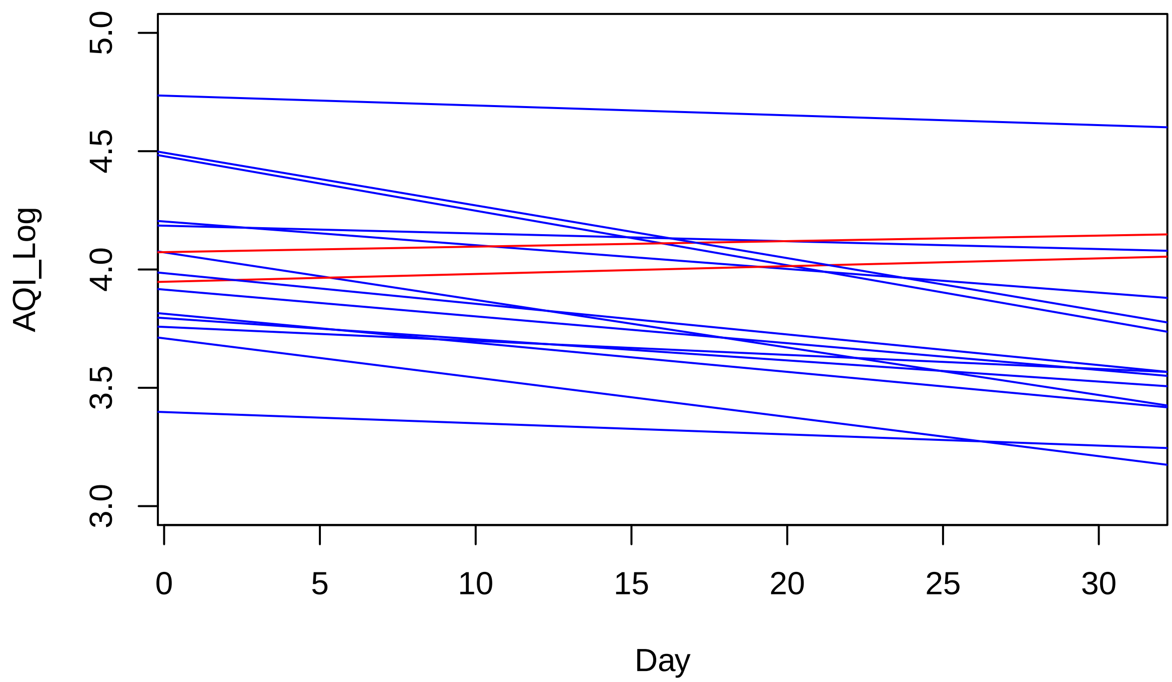
Bayes factor favoring $\rho > 0$ versus $\rho < 0$.

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

```
## [1] 0.537279
```

From the 15 fitted (ordinary least squares) models in (a), we do see some 2 cities have both positive intercept and slope values (as shown below). Thus we can see this as evidence from data that ρ>0.

```r
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]){
  if (beta_j_hat[i,2] > 0) abline(beta_j_hat[i,], col="red")
  else abline(beta_j_hat[i,], col="blue")
}
```



(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%
## 0.5676179 0.9701818
```

(vi)

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

```
## Mean deviance:  448.3
## penalty 27.56
## 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)

Solution for Problem (d)

(i)