

STAT 578 - Fall 2019 - Assignment 5

Frederick (Eric) Ellwanger - fre2

November 23, 2019

Exercise 1

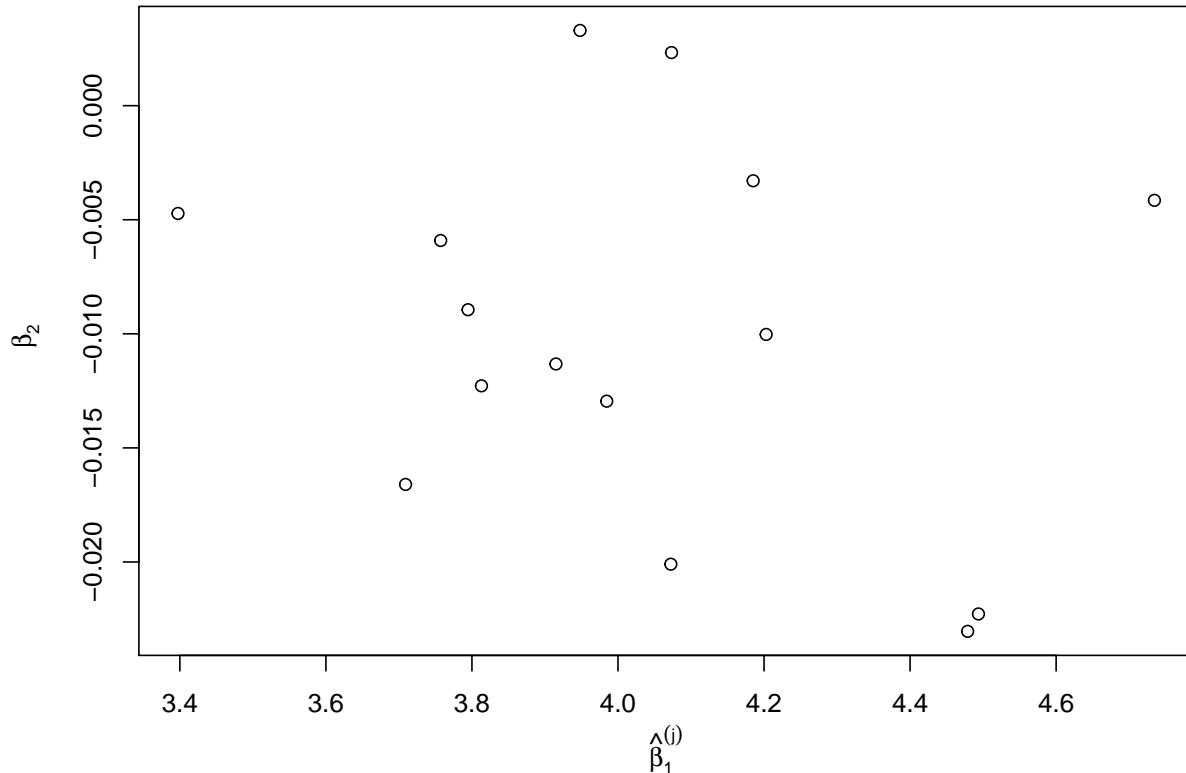
```
ozoneAQI <- read.table("ozoneAQIaug.txt", header=TRUE)
```

(a) Let $\hat{\beta}_1^{(j)}$ and $\hat{\beta}_2^{(j)}$ be the *ordinary least squares* estimates of $\beta_1^{(j)}$ and $\beta_2^{(j)}$, estimated for city j. (The coefficients are estimated completely separately for each city.)

(a)(i) Produce a scatterplot of the pairs $(\hat{\beta}_1^{(j)}, \hat{\beta}_2^{(j)})$, $j = 1, \dots, 15$.

```
betahat <- matrix(NA, nrow(ozoneAQI), 2)
for(j in 1:nrow(ozoneAQI)){
  betahat[j,] <- lsfit(cbind(1:31) - mean(1:31), t(log(ozoneAQI[j,])))$coef
}
plot(betahat[,2] ~ betahat[,1],
  main = expression(paste("Scatterplot of ", hat(beta)[1]^(j), "and ", hat(beta)[2]^(j))),
  xlab = expression(hat(beta)[1]^(j)), ylab = expression(hat(beta)[2]^(j)))
```

Scatterplot of $\hat{\beta}_1^{(j)}$ and $\hat{\beta}_2^{(j)}$



(a)(ii) Compute the average (sample mean) of $\hat{\beta}_1^{(j)}$ and also of $\hat{\beta}_2^{(j)}$.

```
(bmean = apply(betahat, 2, mean))
```

```
## [1] 4.04 -0.01
```

mean of $\hat{\beta}_1^{(j)}$ ≈ 4.037

mean of $\hat{\beta}_2^{(j)}$ ≈ -0.01

(a)(iii) Compute the sample variance of $\hat{\beta}_1^{(j)}$ and also of $\hat{\beta}_2^{(j)}$.

```
(vmean = var(betahat))
```

```
## [,1]      [,2]
```

```
## [1,] 0.119406 -0.0006552
```

```
## [2,] -0.000655  0.0000668
```

variance of $\hat{\beta}_1^{(j)}$ ≈ 0.119

variance of $\hat{\beta}_2^{(j)}$ ≈ 0

(a)(iv) Compute the sample correlation between $\hat{\beta}_1^{(j)}$ and $\hat{\beta}_2^{(j)}$.

```
(bcor = cor(betahat[,1], betaht[,2]))
```

```
## [1] -0.232
```

sample correlation between $\hat{\beta}_1^{(j)}$ and $\hat{\beta}_2^{(j)}$ ≈ -0.232

(b)(i) List an appropriate JAGS model. Make sure to create nodes for \sum_β , ρ , and σ_y^2 . Remember that the ozone AQI values are to be analyzed on the log scale. Now run your model using rjags. Make sure to use multiple chains with overdispersed starting points, check convergence, and monitor μ_β , \sum_β , σ_y^2 , ρ (after convergence) long enough to obtain effective sample sizes of at least 4000 for each parameter.

```
data {  
  dimY <- dim(log ozone)  
  daycent <- day - mean(day)  
}  
  
model {  
  for (j in 1:dimY[1]) {  
    for (i in 1:dimY[2]) {  
      logozone[j,i] ~ dnorm(beta[1,j] + beta[2,j]*daycent[i], sigmasqyinv)  
    }  
    beta[1:2,j] ~ dmnorm(mubeta, Sigmabetainv)  
  }  
  
  mubeta ~ dmnorm(mubeta0, Sigmamubetainv)  
  Sigmabetainv ~ dwish(2*Sigma0, 2)  
  sigmasqyinv ~ dgamma(0.0001, 0.0001)  
  Sigmabeta <- inverse(Sigmabetainv)  
  rho <- Sigmabeta[1,2] / sqrt(Sigmabeta[1,1] * Sigmabeta[2,2])  
  sigmasqy <- 1/sigmasqyinv  
}  
  
ozone = read.table("ozoneAQIaug.txt", header = TRUE)
```

```
library(rjags)
```

```
d1 <- list(logozone = log(ozone[,c(1:31)]), day = c(1:31), mubeta0 = c(0, 0),  
          Sigmamubetainv = rbind(c(1/1000^2, 0), c(0, 1/1000^2)),
```

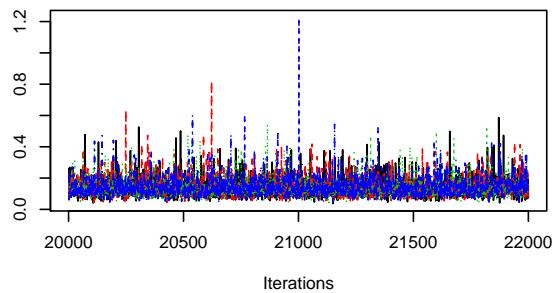
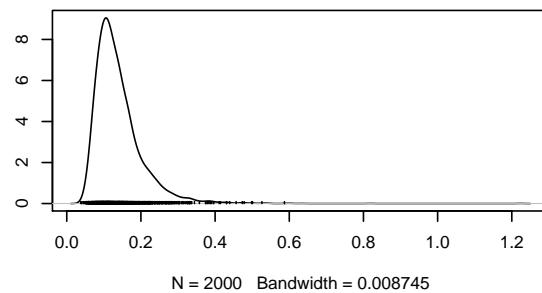
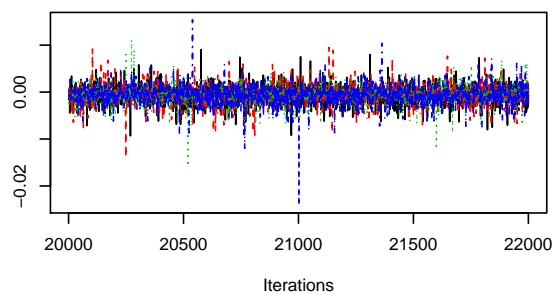
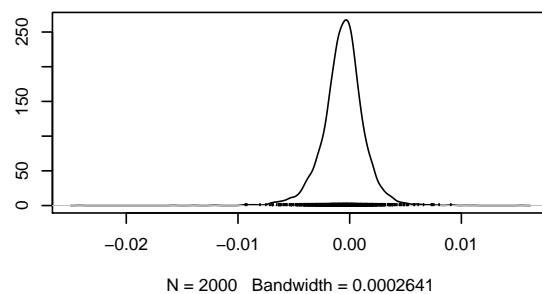
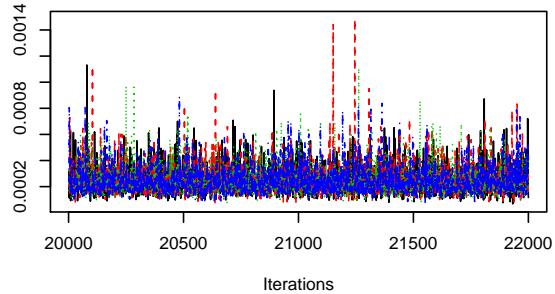
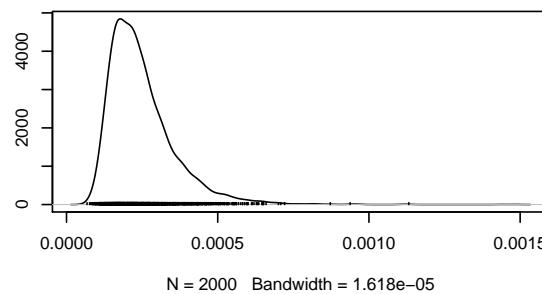
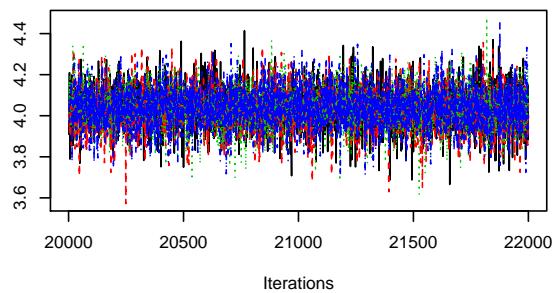
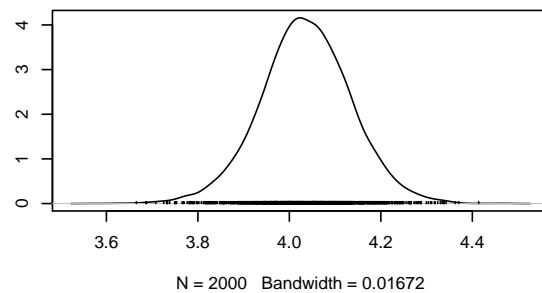
```

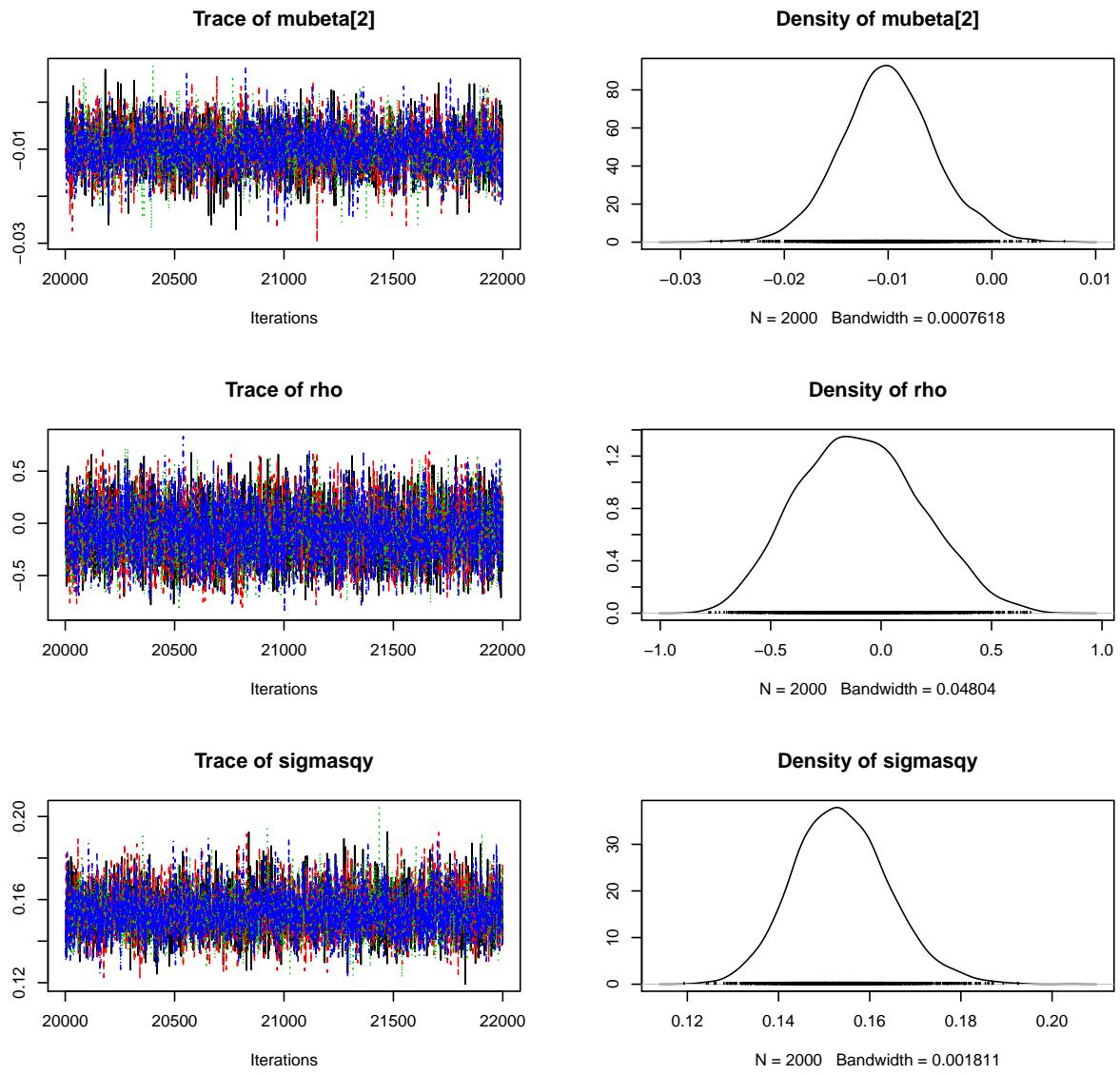
Sigma0 = rbind(c(0.1, 0), c(0, 0.001)))
#Set overdispersed starting points
inits1 <- list(list(sigmasqyinv = 100, mubeta = c(100, 1),
                    Sigmabeta1inv = rbind(c(100, 0), c(0, 100))),
                  list(sigmasqyinv = 0.001, mubeta = c(-100, 1),
                    Sigmabeta1inv = rbind(c(100, 0), c(0, 100))),
                  list(sigmasqyinv = 100, mubeta = c(100, -1),
                    Sigmabeta1inv = rbind(c(0.001, 0), c(0, 0.001))),
                  list(sigmasqyinv = 0.001, mubeta = c(-100, -1),
                    Sigmabeta1inv = rbind(c(0.001, 0), c(0, 0.001))))
#Setup model
m1 <- jags.model("ozone1.bug", d1, inits1, n.chains=4, n.adapt=2000)

## Compiling data graph
## Resolving undeclared variables
## Allocating nodes
## Initializing
## Reading data back into data table
## Compiling model graph
## Resolving undeclared variables
## Allocating nodes
## Graph information:
## Observed stochastic nodes: 465
## Unobserved stochastic nodes: 18
## Total graph size: 1532
##
## Initializing model

#Burn-in
update(m1, 20000)
#Run model
x1 <- coda.samples(m1, c("mubeta[1]", "mubeta[2]", "Sigmabeta[1,1]", "Sigmabeta[1,2]",
                           "Sigmabeta[2,2]", "sigmasqy", "rho"), n.iter=2000)
#Check for convergence
plot(x1, smooth = FALSE)

```

Trace of Sigmabeta[1,1]**Density of Sigmabeta[1,1]****Trace of Sigmabeta[1,2]****Density of Sigmabeta[1,2]****Trace of Sigmabeta[2,2]****Density of Sigmabeta[2,2]****Trace of mubeta[1]****Density of mubeta[1]**



```
#Another check for convergence
gelman.diag(x1, autoburnin=FALSE, multivariate=FALSE)
```

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

```

##          Point est. Upper C.I.
## Sigmabeta[1,1]      1     1.01
## Sigmabeta[1,2]      1     1.00
## Sigmabeta[2,2]      1     1.00
## mubeta[1]           1     1.00
## mubeta[2]           1     1.00
## rho                 1     1.00
## sigmasqy           1     1.00

#Another check for convergence and Neff > 4000/parameter
effectiveSize(x1[,c("mubeta[1]", "mubeta[2]", "Sigmabeta[1,1]", "Sigmabeta[1,2]",
                  "Sigmabeta[2,2]", "sigmasqy", "rho")])

```

```

##      mubeta[1]    mubeta[2]  Sigmabeta[1,1]  Sigmabeta[1,2]  Sigmabeta[2,2]
##      8762        5035       6557         4896       5956
##      sigmasqy      rho
##      7367        4613

```

There does not appear to be any issues with convergence and all monitored parameters exceed an effective sample size of 4000

(b)(ii) Display the coda summary of the results for the monitored parameters.

```
(x1sum = summary(x1))
```

```

##
## Iterations = 20001:22000
## 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
## Sigmabeta[1,1] 0.139114 0.06362 0.00071125 0.00078584
## Sigmabeta[1,2] -0.000581 0.00191 0.00002132 0.00002732
## Sigmabeta[2,2] 0.000249 0.00011 0.00000123 0.00000143
## mubeta[1]      4.035674 0.09924 0.00110958 0.00107714
## mubeta[2]      -0.010030 0.00452 0.00005051 0.00006414
## rho            -0.096258 0.27346 0.00305737 0.00402937
## sigmasqy      0.153827 0.01045 0.00011681 0.00012205
##
## 2. Quantiles for each variable:
##
##          2.5%      25%      50%      75%     97.5%
## Sigmabeta[1,1] 0.062183 0.097237 0.125004 0.163944 0.299954
## Sigmabeta[1,2] -0.004566 -0.001559 -0.000511 0.000456 0.003058
## Sigmabeta[2,2] 0.000114 0.000173 0.000225 0.000296 0.000528
## mubeta[1]      3.833078 3.972478 4.034888 4.100022 4.231249
## mubeta[2]      -0.018837 -0.012951 -0.010088 -0.007140 -0.000825
## rho            -0.595794 -0.296380 -0.104066 0.090030 0.446641
## sigmasqy      0.134365 0.146608 0.153363 0.160421 0.176050

```

(b)(iii) Give an approximate 95% central posterior credible interval for the correlation parameter ρ , and also produce a graph of its (estimated) posterior density.

```

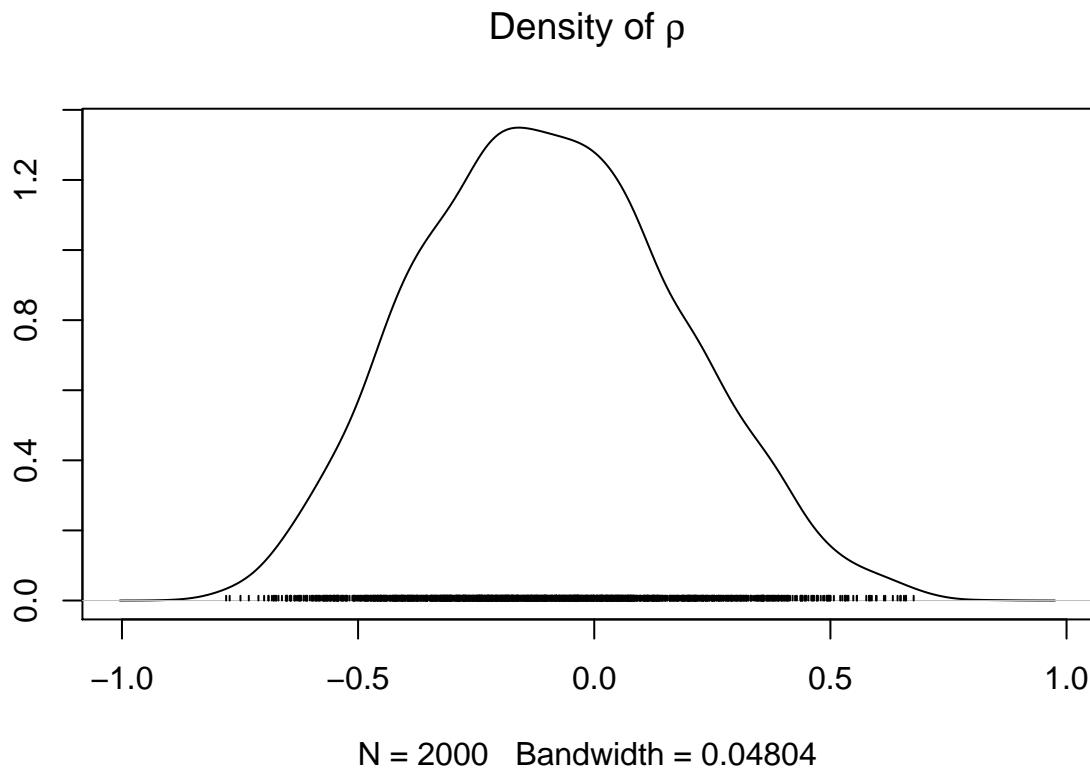
(rhostats = x1sum$quantiles["rho", c("2.5%", "97.5%")])

##   2.5% 97.5%
## -0.596 0.447

The 95% central posterior credible interval, for  $\rho$  is (-0.596, 0.447)

densplot(x1[, "rho"], main = expression(paste("Density of ", rho)))

```



(b)(iv) Approximate the posterior probability that $\rho > 0$. Also, compute the Bayes factor favoring $\rho > 0$ versus $\rho < 0$. (You may use the fact that $\rho > 0$ and $\rho < 0$ have equal prior probability.) Describe the level of data evidence that $\rho > 0$.

```

rho.sim = as.matrix(x1)[, "rho"]
mean(rho.sim > 0)

## [1] 0.359

```

The approximate posterior probability that $\rho > 0 \approx 0.36$

```

#Since PR_prior_h1 = PR_prior_h2 we divide by 1
(BF = (mean(rho.sim > 0)/mean(rho.sim < 0))/1)

```

```
## [1] 0.561
```

The Bayes Factor is approximately 0.561. This is below 1 on the Bayes Factor scale, actually suggesting slight evidence to support that $\rho < 0$

(b)(v) Your model implies that, over the 30-day period from the first day of August to the last, the (population) median ozone AQI value should have changed by a factor of:

$$e^{30\mu_{\beta 2}}$$

Form an approximate 95% central posterior credible interval for this quantity.

```
(mu2q = exp(30*x1sum$quantiles["mubeta[2]", c("2.5%", "97.5%")]))
```

```
## 2.5% 97.5%
## 0.568 0.976
```

The 95% central posterior credible interval, for this quantity is (0.568, 0.976)

(b)(vi) Use the rjags function dic.samples to compute the effective number of parameters (“penalty”) and Plummer’s DIC (“Penalized deviance”). Use at least 100,000 iterations.

```
#Find Effective No Parameters and Plummrs Deviance
load.module("dic")
(dev1 = dic.samples(m1, 100000))
```

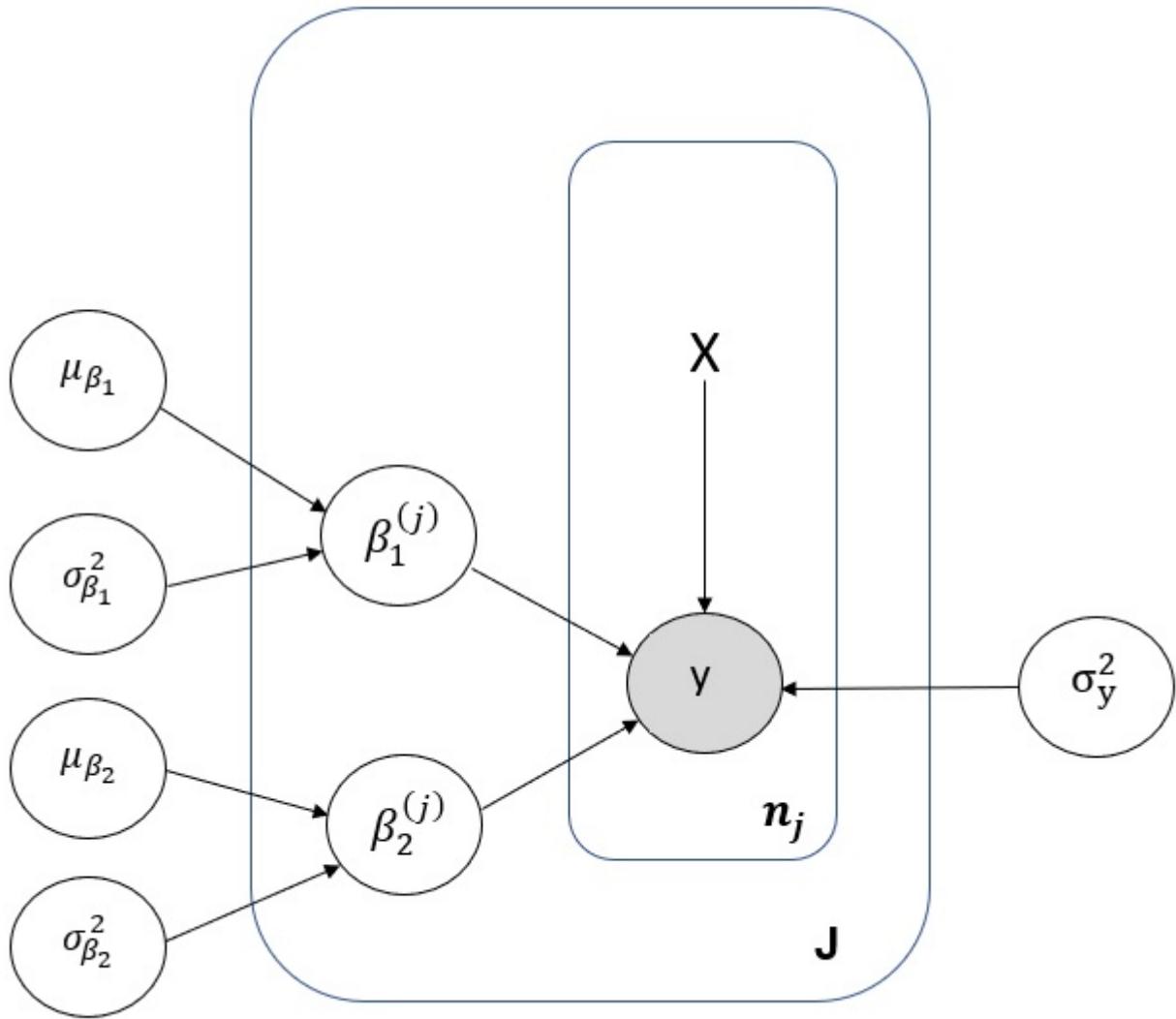
```
## Mean deviance: 448
## penalty 27.5
## Penalized deviance: 476
```

Effective # of parameters ≈ 27.542

Penalized deviance is ≈ 448.278

(c) Now consider a different model with “univariate” hyperpriors for the model coefficients, which do not allow for a coefficient correlation parameter.

(c)(i) Draw a complete DAG for this new model.



(ii) List an appropriate JAGS model. Make sure that there are nodes for $\sigma_{\beta_1}^2$, $\sigma_{\beta_2}^2$, and σ_y^2 .

```

data {
  dimY <- dim(log ozone)
  daycent <- day - mean(day)
}

model {
  for (j in 1:dimY[1]) {
    for (i in 1:dimY[2]) {
      log ozone[j,i] ~ dnorm(beta[1,j] + beta[2,j]*daycent[i], sigmasqyinv)
    }
    beta[1,j] ~ dnorm(mubeta1, 1/(Sigmabeta1^2))
    beta[2,j] ~ dnorm(mubeta2, 1/(Sigmabeta2^2))
  }

  mubeta1 ~ dnorm(0, 1/(1000^2))
}

```

```

mubeta2 ~ dnorm(0, 1/(1000^2))
Sigmabeta1 ~ dunif(0, 1000)
Sigmabeta2 ~ dunif(0, 1000)
sigmasqyinv ~ dgamma(0.0001, 0.0001)

Sigmasqbeta1 <- Sigmabeta1^2
Sigmasqbeta2 <- Sigmabeta2^2
sigmasqy <- 1/sigmasqyinv
}

```

Remember that the ozone AQI values are to be analyzed on the log scale.

```
d2 <- list(log ozone = log(ozone[,c(1:31)]), day = c(1:31))
```

Now run your model using rjags. Make sure to use multiple chains with overdispersed starting points, check convergence, and monitor μ_{β_1} , μ_{β_2} , $\sigma_{\beta_1}^2$, $\sigma_{\beta_2}^2$, and σ_y^2 (after convergence) long enough to obtain effective sample sizes of at least 4000 for each parameter.

```

#Set overdispersed starting points
inits2 <- list(list(sigmasqyinv = 100, mubeta1 = 100, mubeta2 = -1,
                     Sigmabeta1 = 100, Sigmabeta2 = 100),
                  list(sigmasqyinv = 0.001, mubeta1 = -100, mubeta2 = 1,
                     Sigmabeta1 = 100, Sigmabeta2 = 100),
                  list(sigmasqyinv = 100, mubeta1 = 100, mubeta2 = 1,
                     Sigmabeta1 = 0.001, Sigmabeta2 = 0.001),
                  list(sigmasqyinv = 0.001, mubeta1 = -100, mubeta2 = -1,
                     Sigmabeta1 = 0.001, Sigmabeta2 = 0.001))

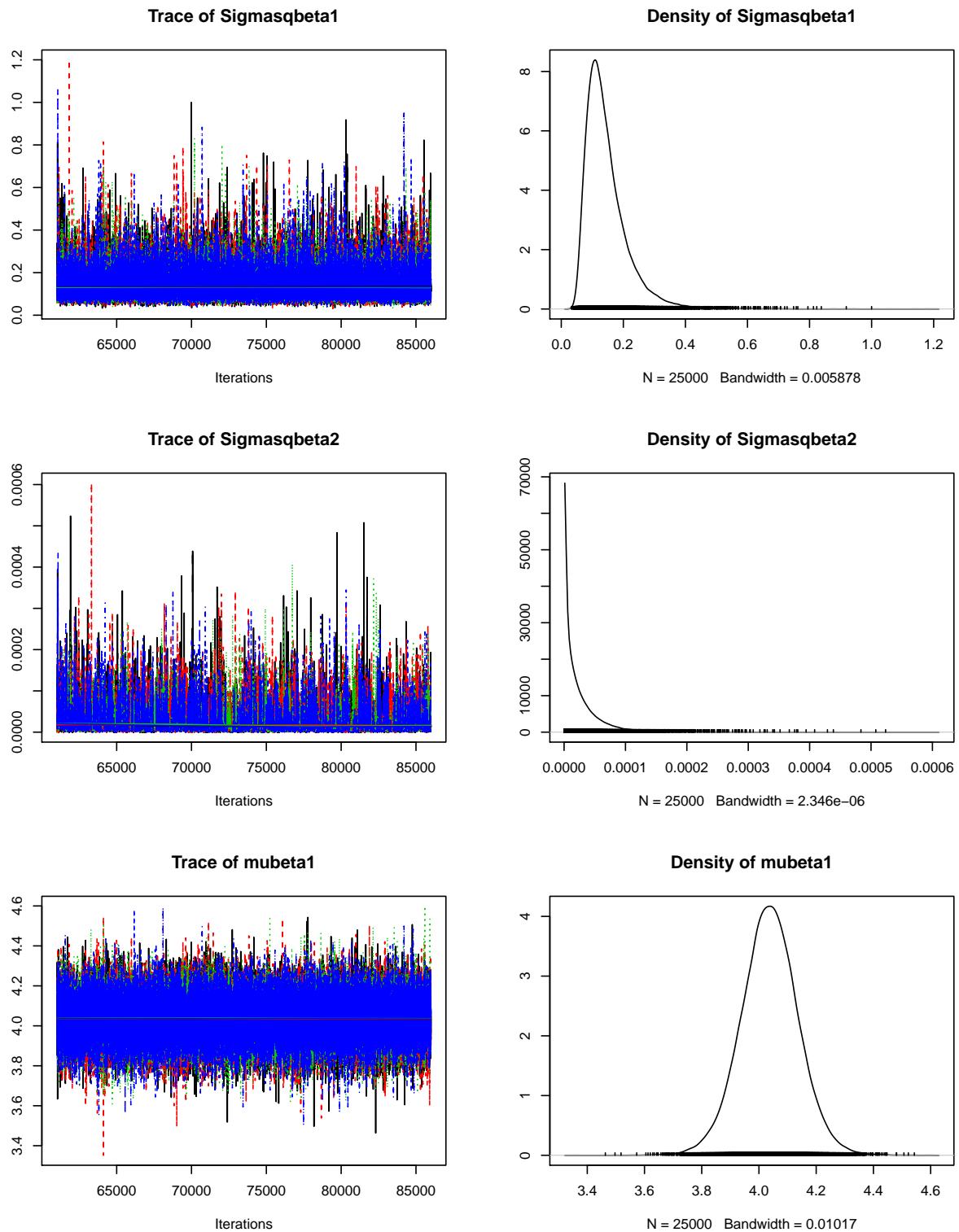
#Setup model
m2 <- jags.model("ozone2.bug", d2, inits2, n.chains=4, n.adapt=1000)

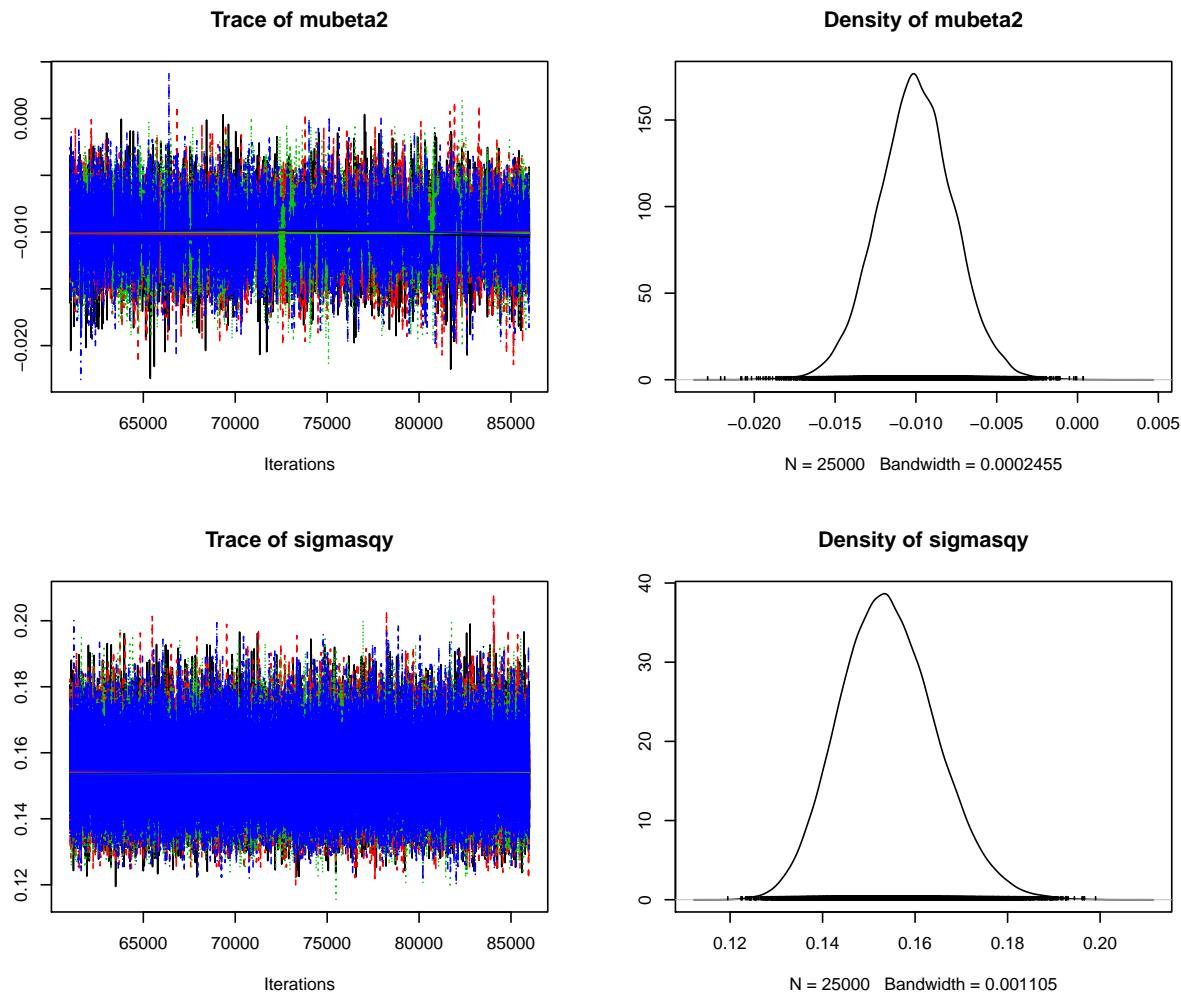
## Compiling data graph
## Resolving undeclared variables
## Allocating nodes
## Initializing
## Reading data back into data table
## Compiling model graph
## Resolving undeclared variables
## Allocating nodes
## Graph information:
## Observed stochastic nodes: 465
## Unobserved stochastic nodes: 35
## Total graph size: 1506
##
## Initializing model

#Burn-in
update(m2, 60000)
x2 <- coda.samples(m2, c("mubeta1", "mubeta2", "Sigmasqbeta1", "Sigmasqbeta2", "sigmasqy"),
                   n.iter=25000)

#Check convergence
plot(x2)

```





```
#Another check for convergence
gelman.diag(x2, autoburnin=FALSE, multivariate=FALSE)
```

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

```

##          Point est. Upper C.I.
## Sigmasqbeta1      1      1
## Sigmasqbeta2      1      1
## mubeta1           1      1
## mubeta2           1      1
## sigmasqy          1      1

#Another check for convergence and Neff > 4000/parameter
effectiveSize(x2[,c("mubeta1","mubeta2", "Sigmasqbeta1","Sigmasqbeta2", "sigmasqy")])

```

```

##      mubeta1      mubeta2 Sigmasqbeta1 Sigmasqbeta2      sigmasqy
##      94251       7247     31029        5059      91456

```

There does not appear to be any issues with convergence and all monitored parameters exceed an effective sample size of 4000

(c)(iii) Display the coda summary of the results for the monitored parameters.

```
(sum2 = summary(x2))
```

```

##
## Iterations = 61001:86000
## Thinning interval = 1
## Number of chains = 4
## Sample size per chain = 25000
##
## 1. Empirical mean and standard deviation for each variable,
##    plus standard error of the mean:
##
##              Mean        SD   Naive SE Time-series SE
## Sigmasqbeta1 0.1460869 0.0705162 0.000222992 0.000407162
## Sigmasqbeta2 0.0000244 0.0000322 0.000000102 0.000000463
## mubeta1      4.0372395 0.0998747 0.000315831 0.000325341
## mubeta2     -0.0100181 0.0023842 0.000007540 0.000029026
## sigmasqy     0.1542672 0.0104256 0.000032969 0.000034476
##
## 2. Quantiles for each variable:
##
##            2.5%       25%       50%       75%    97.5%
## Sigmasqbeta1 0.0622257018 0.09954381 0.1300611 0.1738566 0.323304
## Sigmasqbeta2 0.0000000405 0.00000345 0.0000133 0.0000331 0.000112
## mubeta1      3.8378507650 3.97313114 4.0374665 4.1016628 4.234648
## mubeta2     -0.0147173498 -0.01157223 -0.0100232 -0.0084688 -0.005307
## sigmasqy     0.1352191775 0.14700220 0.1538003 0.1610539 0.175947

```

(c)(iv) Recall the (population) median ozone AQI change factor $e^{30\mu_{\beta_2}}$ considered in the previous analysis. Form an approximate 95% central posterior credible interval for this quantity, and compare it with the previous results.

```
(q95 = exp(sum2$quantiles["mubeta2", c("2.5%", "97.5%")])))
```

```
## 2.5% 97.5%
## 0.985 0.995
```

The 95% central posterior credible interval for this qty is $\approx (0.985, 0.995)$

This interval is much tighter than the interval from the previous results, and it is slightly higher than the previous results.

(c)(v) Use the `rjags` function `dic.samples` to compute the effective number of parameters (“penalty”) and Plummer’s DIC (“Penalized deviance”). Use at least 100,000 iterations.

```
(dev2 = dic.samples(m2, 100000))
```

```
## Mean deviance: 449  
## penalty 20.2  
## Penalized deviance: 470
```

Effective # of parameters ≈ 20.23

Penalized deviance is ≈ 449.363

(c)(vi) Compare the (Plummer’s) DIC values for this model and the previous one. Which is preferred?

```
if((sum(dev1$deviance) - sum(dev2$deviance)) > 0){  
  winner = "Part C Model"  
}else{  
  winner = "Part B Model"  
}
```

The model with the lowest DIC is preffered, therefore Part B Model is preffered, but there is only a small difference between the 2 models.

(d)(i) It is possible that the variability in log-value depends on the city. How might you modify your model to account for this? Would your solution need more hyperparameters?

In order to have the model variance of the log-value depend on the city, σ_y would need to be on the j plate. We would need to give it’s distribution hyperparameters. We could give $\sigma_y^{(j)}$ an non-informative hyperprior with a normal distribution with mean 0 and uniform standard deviation. $\sigma_y^{(j)} \sim N(0, \sigma_{yj}^2)$ where $\sigma_{yj} \sim U(0, \infty)$

(d)(ii) 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. What specific model assumption would this violate? (Is i or j involved?)

This would violate the independance and exchangeability asumption of the response values within a city. If there was correlations not captured by the simple linear regression model, the individual data values will not be conditionally iid and they would also not be exchangeable. Since this effects the responses within a city, i would be involved

(d)(iii) 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 (because some cities are closer to each other than others). What specific model assumption would this violate? (Is i or j involved?)

This would violate the assumption that the β_j' s are iid and exchangeable. If there was some spatial correlation between the reponses between 2 cities that is not captured in the simple linear regression model, the β_j' s would no longer be iid or exchangeable. Since this effects the the responses between cities, j would be involved