365

```
#LingXin Li(Cynthia Li) sta365 final exam
###If you are running my rmd code for grading, and the image(jpg) cannot be loaded, that is because the path to image is in my own computer(problem3 parta), and so does the data(eg.the code of loading data is also to the path of my own computer). Please see the knitted file I hand in in quercus.

#I installed tinytex to this, but when I knit it to pdf, it came with error said that I have not installed it, so that I knit to the html/word document, and then convert word to pdf.
library(R2jags)
```

```
## Loading required package: rjags
```

```
## Loading required package: coda
```

```
## Linked to JAGS 4.3.1
```

```
## Loaded modules: basemod,bugs
```

```
##
## Attaching package: 'R2jags'
```

```
## The following object is masked from 'package:coda':
##
##     traceplot
```

```
library(rjags)
library(runjags)
library(MCMCpack)
```

```
## Loading required package: MASS
```

```
## ##
## ## Markov Chain Monte Carlo Package (MCMCpack)
```

```
## ## Copyright (C) 2003-2023 Andrew D. Martin, Kevin M. Quinn, and Jong Hee Park
```

```
## ##
## ## Support provided by the U.S. National Science Foundation
```

```
## ## (Grants SES-0350646 and SES-0350613)
## ##
```

```
library(lattice)
library(MASS)
library(tidyverse)
```

```
## ── Attaching core tidyverse packages ──────────────────────────────── tidyverse 2.0.0 ──
## ✔ dplyr     1.1.1     ✔ readr     2.1.4
## ✔ forcats   1.0.0     ✔ stringr   1.5.0
## ✔ ggplot2   3.4.1     ✔ tibble    3.2.1
## ✔ lubridate 1.9.2     ✔ tidyr     1.3.0
## ✔ purrr     1.0.1
```

```
## ── Conflicts ──────────────────────────────────────────────────────────────────── tidyverse_conflicts() ──
## ✖ tidyr::extract() masks runjags::extract()
## ✖ dplyr::filter()  masks stats::filter()
## ✖ dplyr::lag()     masks stats::lag()
## ✖ dplyr::select()  masks MASS::select()
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(dplyr)
library(tinytex)
```

```
#Problem1 handwriting, please see the other file.
```

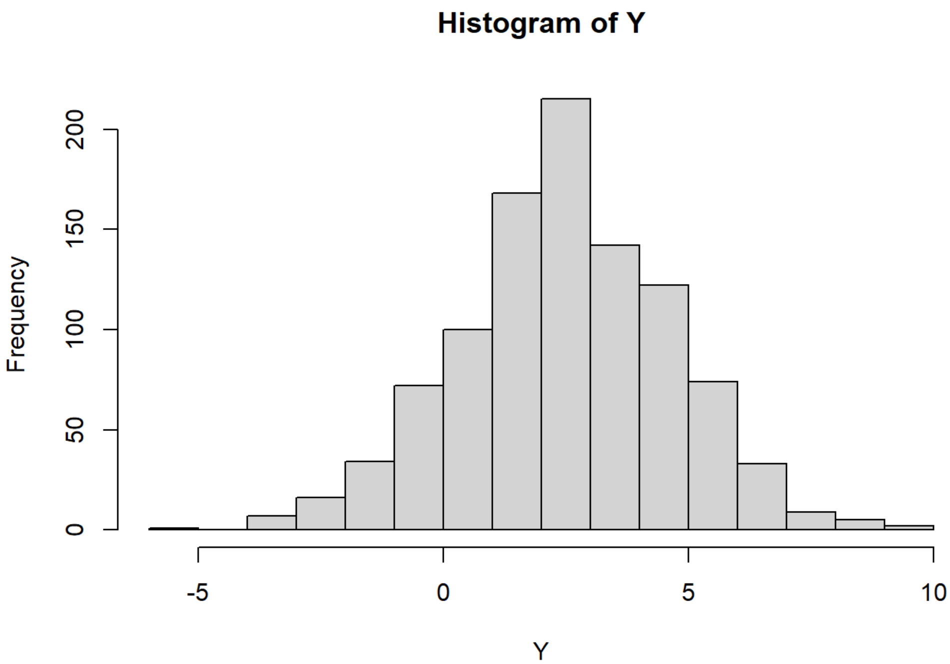
##Problem2

a. Write your choices of the parameters clearly.

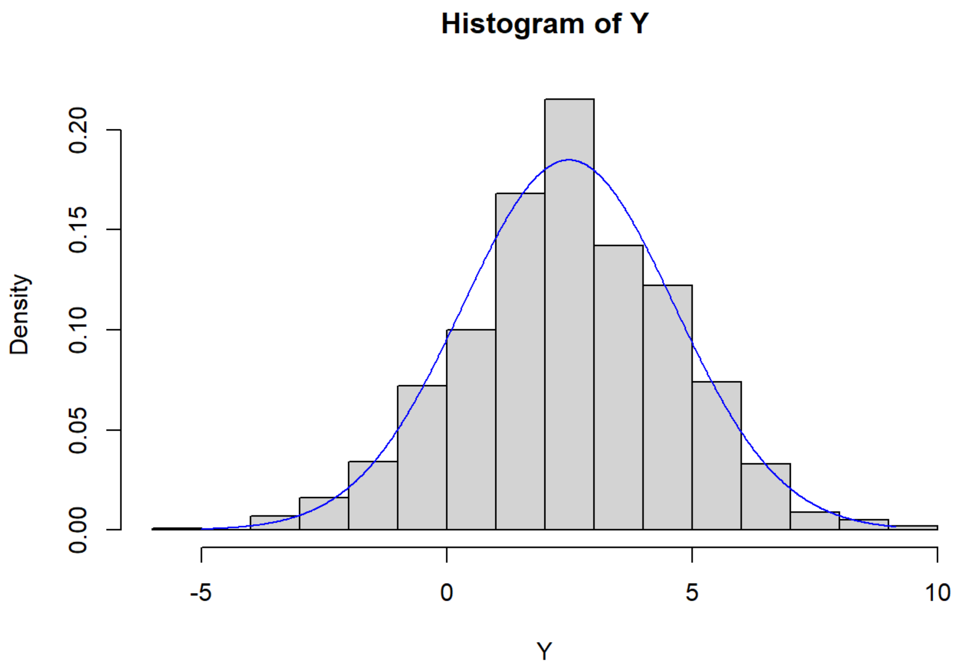
$\mu_1 = 1$ $\mu_2 = 4$ $\delta = 0.5$ $\sigma^2 = 9$

```
###Problem2
#b. Produce the code used to generate the simulations.
set.seed(1000)
N <- 1000
mu1 <- 1
mu2 <- 4
delta <- 0.5
sigma2 <- 9
#rnorm(n, mean, sd)
Z1 <- rnorm(N, mu1, sqrt(sigma2))
Z2 <- rnorm(N, mu2, sqrt(sigma2))
Y <- delta * Z1 + (1 - delta) * Z2
```

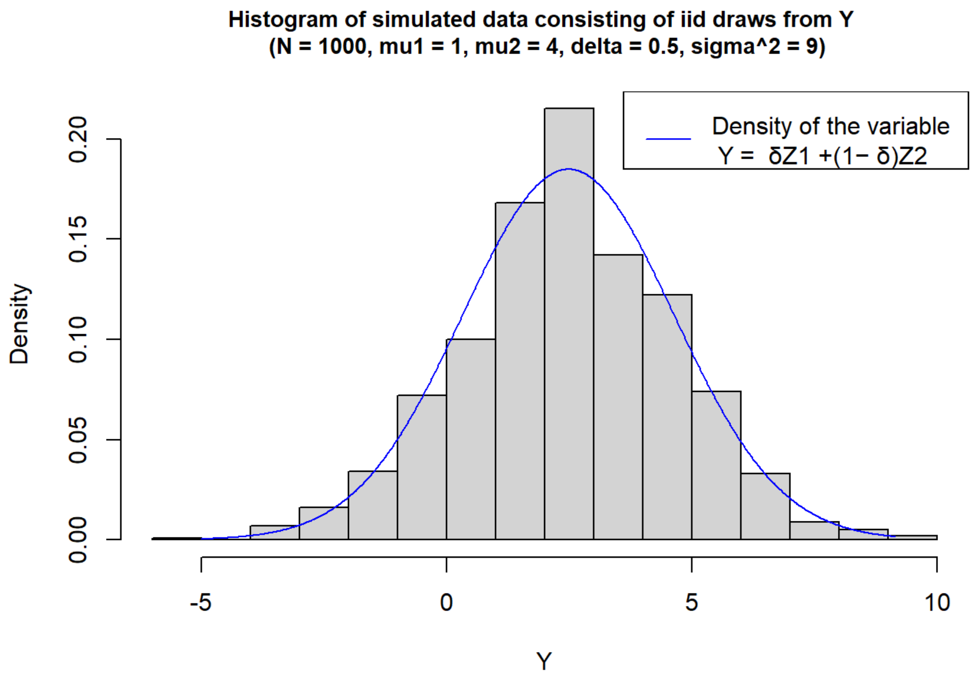
```
###Problem2
#c. Plot a histogram of your simulated data.
hist(Y)
```




```
###Problem2
#d. Overlay a plot of the density of the variable Y on the histogram.

#I used the following reference(the section of "Density" and "Random Variates" of this references link)to help me understand better about how to rnorm() and dnorm() in order to simulated data and calculate the pdf for the above and below code for problem 2, and also the function of seq:

#Probability Distributions in R (Stat 5101, Geyer). (n.d.). https://www.stat.umn.edu/geyer/old/5101/rlook.html#:~:text=dnorm%20is%20the%20R%20function,standard%20deviation%20of%20the%20distribution.



x <- seq(min(Y), max(Y), length.out = 1000)
hist(Y,  probability = TRUE)
lines(x, dnorm(x, mean = mean(Y), sd = sd(Y)), col = "blue" )
```




```
###Problem2


#e. Label the graph clearly, using captions or titles that mention your parameter choices.

hist(Y,
     main = "Histogram of simulated data consisting of iid draws from Y \n (N = 1000, mu1 = 1, mu2 = 4, delta = 0.5, sigma^2 =
9)",
     cex.main = 0.91,
     xlab = "Y", probability = TRUE)
lines(x, dnorm(x, mean = mean(Y), sd = sd(Y)), col = "blue")
legend("topright", legend = c("Density of the variable \n Y =  δZ1 +(1- δ)Z2"), col = "blue", lty = 1)
```



##Problem3

```
##Problem3
#Part(a) if you cannot see the picture of answer here if you are running the rmd(since the path to the image file is in my own computer, I save the image and rmd in the same path of my computer), please my knitted file hand in quercus
###HANDWRITING IMAGE FOR MY ANSWER IS IN MY KNITTED FILE IN QUERCUS

knitr::include_graphics("p3pa.jpg")
```

Problem 3

Part (a)

$$P((x_1,y_1),\dots,(x_n,y_n),F_1,\dots,F_n,\beta,\sigma^2 \mid \sigma_\beta^2, v_0,\sigma_0^2)$$

$$= P((x_1,y_1),\dots,(x_n,y_n) \mid F_1,\dots,F_n,\beta,\sigma^2)\, P(F_1,\dots F_n \mid \sigma^2)\, P(\beta \mid \sigma_\beta^2)\, P(\sigma^2 \mid v_0,\sigma_0^2)$$

$$= \underbrace{\prod_{i=1}^n P(y_i \mid N_i,\theta_i)}_{\text{Note: } y_i \sim Bin(N_i,\theta_i)} \underbrace{\prod_{i=1}^n P(F_i \mid x_i,\beta,\sigma^2)}_{\text{Note: } F_i \sim N(x_i\beta,\sigma^2)} \underbrace{\prod_{i=1}^n P(\beta_i \mid 0,\sigma_\beta^2 I)}_{\text{Note: } \beta \sim N(0,\sigma_\beta^2 I)} \underbrace{\prod_{i=1}^n P(\sigma_i^2 \mid v_0,\sigma_0^2)}_{\text{Note } \sigma^2 \sim \text{Inverse-Gamma}\left(\frac{v_0}{2},\frac{v_0\sigma_0^2}{2}\right)}$$

$$= \prod_{i=1}^n \left(P(y_i \mid N_i,\theta_i)\, P(F_i \mid x_i,\beta,\sigma^2)\, P(\beta_i \mid 0,\sigma_\beta^2, I)\, P(\sigma_i^2 \mid v_0,\sigma_0^2)\right)$$

```
##Problem3
#source of data of problem 3
#Gelman, & Hill. (n.d.). wells.dat. http://www.stat.columbia.edu/~gelman/arm/examples/arsenic/wells.dat. Retrieved April 13, 2023, from http://www.stat.columbia.edu/~gelman/arm/examples/arsenic/wells.dat

#Part (b)

wells <- read.table("wells.dat.txt", header = TRUE, col.names = c("nth_obs", "switch", "arsenic", "dist", "assoc", "educ"))

#arsenic levels and distance is Xi. Center all X-variables.
#The following code let each xi minus xbar
mean_arsenic <- mean(wells$arsenic)
mean_distance <- mean(wells$dist)
wells$arsenic_center <- wells$arsenic - mean_arsenic
wells$dist_center <- wells$dist - mean_distance
```

```
##Problem3
#Part (b)
#write a JAGS model

#The following references is
#Hu, J. a. a. J. (2020, July 30). Chapter 12 Bayesian Multiple Regression and Logistic Models | Probability and Bayesian Modeling. https://bayesball.github.io/BOOK/bayesian-multiple-regression-and-logistic-models.html#bayesian-logistic-regression

#2nd reference is lecture materials

#according to my references and the given information of the question, it said that Fi is a linear function of the predictors xi, theta i can also be expressed in p since Yi~BIN(n,theta(is also p of success)). Then Fi is equals to log(theta i/(1-theta i)) which is given by the function, and it is equals to beta0+beta1xi1+beta2xi2(because we have 2 xi in this question which is arsenic levels and distance from the wells). After we simiplied, theta i = exp(beta0+beta1xi1+beta2xi2) / (1 + exp(beta0+beta1xi1+beta2xi2) where theta lies between 0 and 1.


set.seed(1000)
JAGS_logistic = function() {
  # Likelihood
  for (i in 1:n) {
    Y[i] ~ dbin(theta[i],1) #yi ~ Binomial(Ni,theta i )
    f[i] ~ dnorm(X[i,] %*% beta, tau2_logistic) #This is Fi~ N(Xibeta,sigma^2 ), and according to week lecture I use tau2 for sigma2 position
    theta[i] <- exp(f[i]) / (1 + exp(f[i])) #interpreted according to my references, also we can use the function ilogit(x) refer to Rdocumentation
  }

  # Prior for beta
  for (j in 1:p) {
    beta[j] ~ dnorm(0.0, tau2_logistic_beta) #Given that beta ~ N(0,(sigma_beta)^2*I), and I use tau2 for sigma_beta square according to the professor's lecture materials.

  }

  tau2_logistic <- 1.0/sigma2 #refer to professor's lecture week8B sigma2 <- 1.0/tau2, and I convert it
  sigma2 ~ dgamma(nu0/2, nu0*sigma0_square/2) #Given that σ^2 ~ Inverse-Gamma(ν0/2,ν0*σ0^2/2), but I failed to run sigma2 ~dinvgamma(nu0/2, nu0*sigma0_sq/2), so that I think it should be dgamma

}
```

```
##Problem3
#Part (b)
Y <- wells$switch #well switching as the yi
X <- model.matrix(switch ~ arsenic_center + dist_center, data = wells) #function refer to RDocumentation
n <- nrow(wells)
nu0 <- 1
sigma0_square <- 1
sigma2 <- 1
tau2_logistic_beta <- 0.0001
p <- 3 #intercept and predictors: beta0, beta1, beta2
set.seed(1000)
data.JAGS = list(Y = Y,
                 X = X,
                 n = n,
                 p = p,
                 tau2_logistic_beta = tau2_logistic_beta,
                 nu0 = nu0,
                 sigma0_square = sigma0_square)

inits.JAGS = list(list(beta = rnorm(p, 0, 1), sigma2 = sigma2))

para.JAGS = c("beta", "sigma2")
```

```
##Problem3
#Part (b)
#References: lecture materials
set.seed(1000)
fit.JAGS.logistic = jags(data=data.JAGS,
                inits=inits.JAGS,
                parameters.to.save = para.JAGS,
                n.chains=1,
                n.iter=10000,
                n.burnin=1000,
                model.file=JAGS_logistic)
```

```
## module glm loaded
```

```
## Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
## Graph information:
##    Observed stochastic nodes: 3020
##    Unobserved stochastic nodes: 3024
##    Total graph size: 30215
##
## Initializing model
```

```
##Problem3
#Part (b)
#References: lecture materials
print(fit.JAGS.logistic)
```

```
## Inference for Bugs model at "C:/Users/CYNTHI~1/AppData/Local/Temp/RtmpmuNAdL/model5980765d4944.txt", fit using jags,
##  1 chains, each with 10000 iterations (first 1000 discarded), n.thin = 9
##  n.sims = 1000 iterations saved
##          mu.vect sd.vect     2.5%      25%      50%      75%    97.5%
## beta[1]     0.347   0.039    0.272    0.322    0.345    0.372    0.425
## beta[2]     0.482   0.046    0.394    0.452    0.486    0.510    0.571
## beta[3]    -0.009   0.001   -0.012   -0.010   -0.009   -0.009   -0.007
## sigma2      0.214   0.157    0.050    0.106    0.158    0.291    0.617
## deviance 3799.380  90.576 3576.964 3753.201 3827.568 3864.870 3906.479
##
## DIC info (using the rule, pD = var(deviance)/2)
## pD = 4102.0 and DIC = 7901.4
## DIC is an estimate of expected predictive error (lower deviance is better).
```

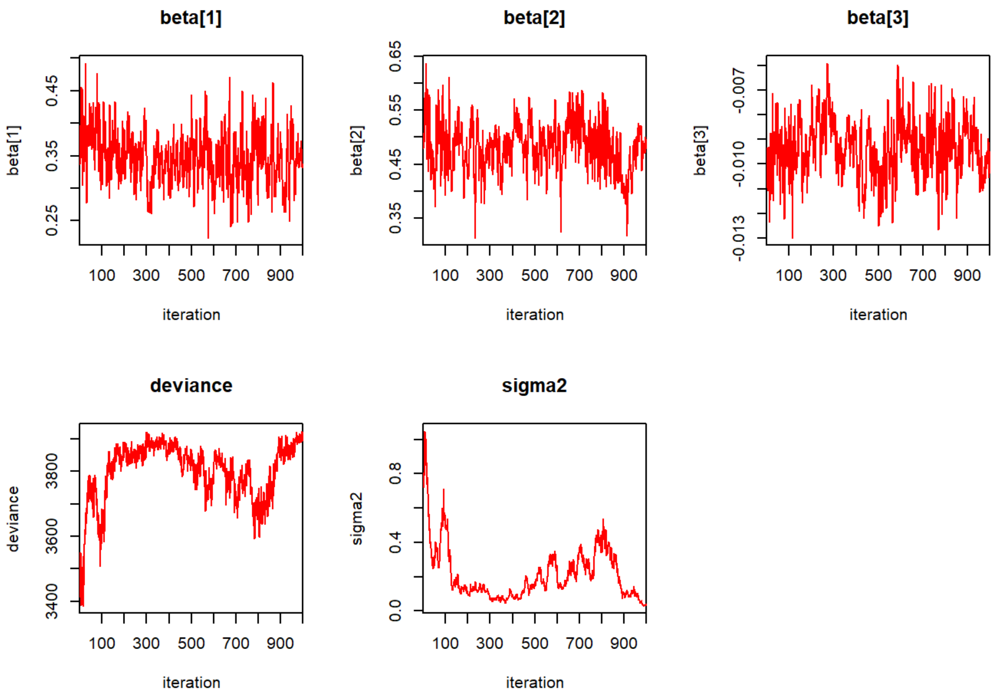
beta[1] here represent beta0 which is our intercept.

beta[2] here actually represent beta1 which is the first predictor arsenic levels(which I already centered). 95% credible intervals for beta 1 is we looked at the section“beta[2]”, look at the 2.5% and 97.5 quantiles. So that the 95% credible intervals for beta 1 is [0.394, 0.571]

beta[3] here actually represent beta2 which is the second predictor distance from the wells(which I already centered by using xi-xbar). 95% credible intervals for beta 1 is we looked at the section“beta[2]”, look at the 2.5% and 97.5 quantiles. So that the 95% credible intervals for beta 1 is [-0.012, -0.007]

look at the sigma2 section, 95% credible intervals for sigma square is [0.050,0.617] which is also according to 2.5% and 97.5 quantiles.

```
#References: lecture materials
##Problem3
#Part (c)
traceplot(fit.JAGS.logistic,mfrow=c(2,3),ask=FALSE)
```



Note that beta[1] here represent beta0 which is our intercept, beta[2] here actually represent beta1 which is the first predictor arsenic levels(which I already centered), beta[3] here actually represent beta2 which is the second predictor distance from the wells(which I already centered by using xi-xbar).

(I looked at their examples of traceplots to help me understand better about some criteria/good example/bad example for traceplots: I looked into the website of bad examples here Evaluation of MCMC samples. (n.d.). Cross Validated. https://stats.stackexchange.com/questions/311151/evaluation-of-mcmc-samples (https://stats.stackexchange.com/questions/311151/evaluation-of-mcmc-samples) The above user of the website led me to here: Evaluating Markov Chain Monte Carlo (MCMC) Algorithms. (n.d.). https://link.springer.com/content/pdf/10.1007/978-0-387-71265-9_6.pdf (https://link.springer.com/content/pdf/10.1007/978-0-387-71265-9_6.pdf)

and I have also looked at Bakker, R. (n.d.). Florida State University Bayesian Workshop. https://spia.uga.edu/faculty_pages/rbakker/bayes/Day2/Day2_Convergence.pdf (https://spia.uga.edu/faculty_pages/rbakker/bayes/Day2/Day2_Convergence.pdf))

I generated traceplots. I believe MCMC moderately converge to a posterior, and their mixing is fine. All of the traceplot for beta showed above tend to have moderate number of flutuations, where partial pattern and partial trends looks deviate from the majority pattern. In other words, some trends look unstable here, but overall, I would say it did provide moderately strong evidence of convergence.

##Problem4

```
###Problem4
load("C:/Bayes/swim_time.RData")
swim_time <- get(load('C:/Bayes/swim_time.RData'))
library(reshape2)
```

```
##
## Attaching package: 'reshape2'
```

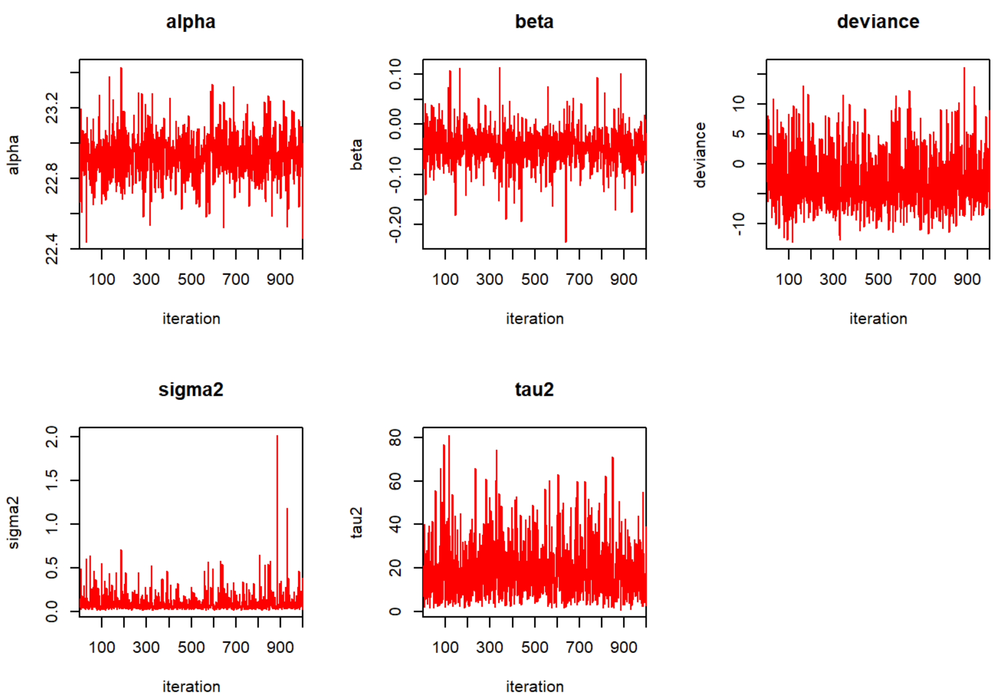
```
## The following object is masked from 'package:tidyr':
##
##     smiths
```

```
###Problem4
#use of melt in reshape2 package citation is according to the reference of R. (n.d.). and reference Zach. (2022) which you can see in the references page.

Y$Swimmer <- factor(1:4)

swim_time1 <- melt(Y,
                   id.vars = "Swimmer",
                   variable.name = "Week",
                   value.name = "Time")

swim_time1$Week <- parse_number(as.character(swim_time1$Week))


#mean of range 22 to 24 is 23, variance of range 22 to 24 is ((22 - 23)^2 + (23 - 23)^2 + (24 - 23)^2) / 2 = 1
#many the following code is initially from the professor lecture materials with my modification

para.JAGS <- c("alpha", "beta", "tau2", "sigma2")
set.seed(1000)
linear.model.JAGS = function(){
  for(i in 1:n){
    y[i] ~ dnorm(mu[i],tau2)
    mu[i]<- alpha + beta*(x[i]-x.bar)
  }
  x.bar <- mean(x)
  alpha ~ dnorm(23, 1)
  beta ~ dnorm(0.0, 1.0E-4)
  sigma2 <- 1.0/tau2
  tau2 ~ dgamma(0.1,0.1)
}
```

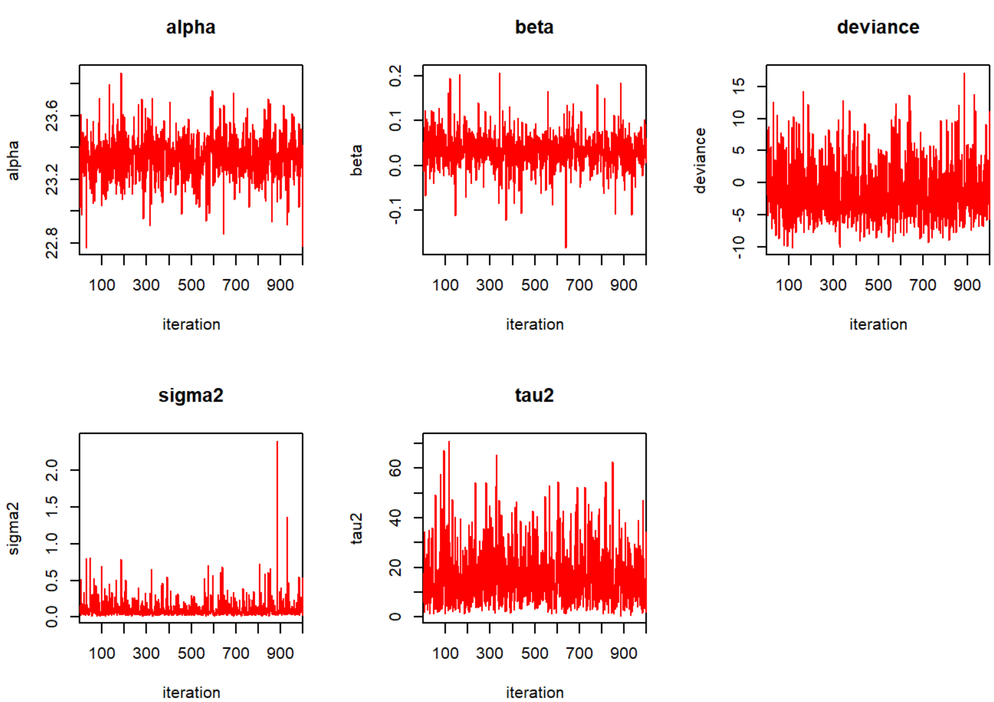
```
###Problem4
lst0 <- list()

swim_time2 <- swim_time1 %>% group_by(Swimmer)
set.seed(1000)
for (j in 1:4) {
  new_data <- swim_time2 %>% filter(Swimmer == j)
  y <- new_data$Time
  x <- new_data$Week
  n <- length(x)

#references: professor lecture materials
  data.JAGS = list(y = y, x = x, n = n)
  inits.JAGS = list(list(alpha = 23.0, beta = 0.0, tau2 = 1.0))
  set.seed(1000)
  fit.JAGS = jags(data = data.JAGS,
                  inits = inits.JAGS,
                  parameters.to.save = para.JAGS,
                  n.chains = 1,
                  n.iter = 9000,
                  n.burnin = 1000,
                  model.file = linear.model.JAGS)

  lst0[[j]] <- fit.JAGS
}
```

```
## Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
## Graph information:
##    Observed stochastic nodes: 6
##    Unobserved stochastic nodes: 3
##    Total graph size: 42
##
## Initializing model
##
## Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
## Graph information:
##    Observed stochastic nodes: 6
##    Unobserved stochastic nodes: 3
##    Total graph size: 42
##
## Initializing model
##
## Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
## Graph information:
##    Observed stochastic nodes: 6
##    Unobserved stochastic nodes: 3
##    Total graph size: 42
##
## Initializing model
##
## Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
## Graph information:
##    Observed stochastic nodes: 6
##    Unobserved stochastic nodes: 3
##    Total graph size: 42
##
## Initializing model
```

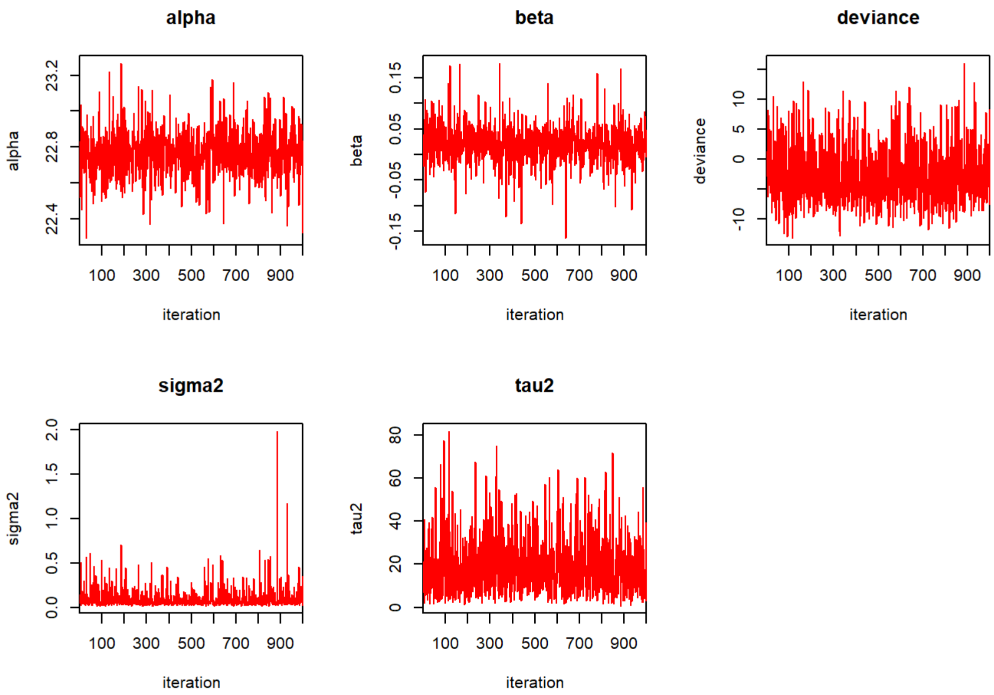
```
#references: professor lecture materials
#traceplot for swimmer1
traceplot(lst0[[1]],mfrow=c(2,3),ask=FALSE)
```



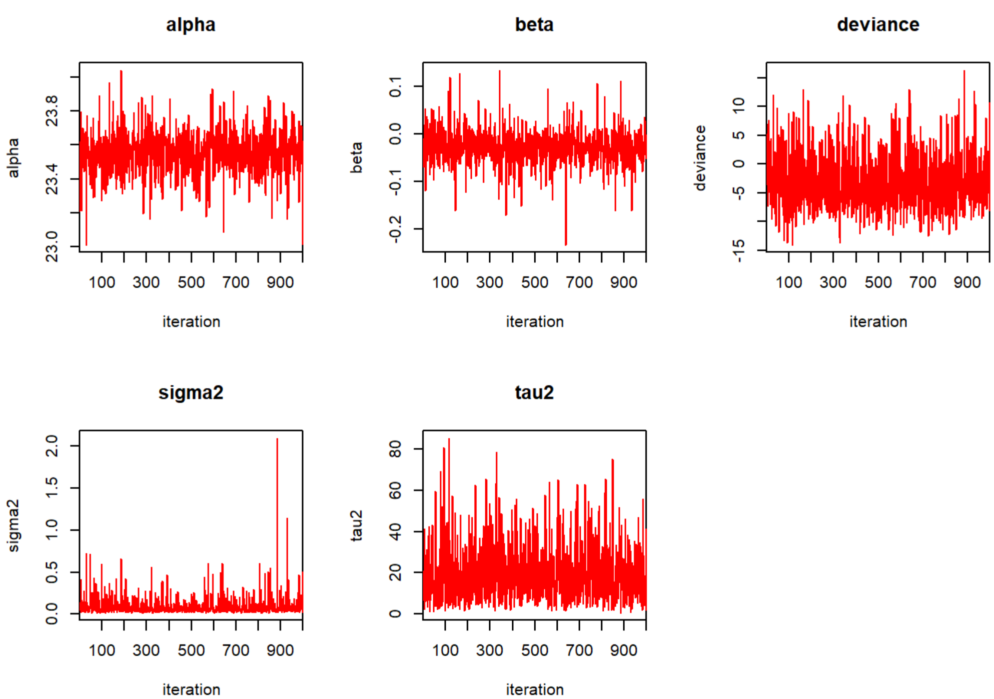

```
#traceplot for swimmer2
#references: professor lecture materials
traceplot(lst0[[2]],mfrow=c(2,3),ask=FALSE)
```




```
#traceplot for swimmer3
#references: professor lecture materials
traceplot(lst0[[3]],mfrow=c(2,3),ask=FALSE)
```




```
#traceplot for swimmer4
#references: professor lecture materials
traceplot(lst0[[4]],mfrow=c(2,3),ask=FALSE)
```



##Problem4 continued 1. comment on the suitability of the resulting model and Whether we have reached MCMC convergence to a posterior

(I looked at their examples of traceplots to help me understand better about some criteria/good example/bad example for traceplots: I looked into the website of bad examples here Evaluation of MCMC samples. (n.d.). Cross Validated. https://stats.stackexchange.com/questions/311151/evaluation-of-mcmc-samples (https://stats.stackexchange.com/questions/311151/evaluation-of-mcmc-samples) The above user of the website led me to here: Evaluating Markov Chain Monte Carlo (MCMC) Algorithms. (n.d.). https://link.springer.com/content/pdf/10.1007/978-0-387-71265-9_6.pdf (https://link.springer.com/content/pdf/10.1007/978-0-387-71265-9_6.pdf)

and I have also looked at Bakker, R. (n.d.). Florida State University Bayesian Workshop. https://spia.uga.edu/faculty_pages/rbakker/bayes/Day2/Day2_Convergence.pdf (https://spia.uga.edu/faculty_pages/rbakker/bayes/Day2/Day2_Convergence.pdf))

my answer: A traceplot indicated how the value of each parameter has changed across iterations of the chain. First, ignored all graph of deviance since it is not one of our parameters. Second, for all four swimmers, all of the traceplots based on each parameter roughly reached MCMC converged to a posterior. Because I observed rare big fluctuations in the pattern. However, if we look at it in a more strict way, I would say that there are more big fluctuations in all traceplots of all 4 swimmers in sigma square, but I think it should still be considered as a stable pattern which values converging around a certain point. The non-existence of big fluctuations indicates that most of the values are within similar range, each values are not going so far from the average(or other statistical values) of the distribution which means they are moving around similar points without much deviation.

Overall, all of the traceplots for all swimmers based on each parameter can be considered as good convergence(and or good mixing), one obvious thing is that there is no special trends that deviate from the majority.

2. Whether my priors are reasonable.

my answer: I think my prior is reasonable since it is based on the information that already existed which is 22 to 24 seconds is the competitive times range for this age group. mean of range 22 to 24 is 23, variance of range 22 to 24 is ((22 - 23)^2 + (23 - 23)^2 + (24 - 23)^2) / 2 = 1, so standard deviation is the square root of variance which is one. The prior is only on the alpha which is on the intercept since I do not have the information on beta, tau and sigma. Therefore the prior for beta, tau and sigma are noninformative priors which their posterior is heavily rely on the data.

3. comment on how I would revising the model, and how I would evaluate if this revised model is better than the current version

my answer:

First, according to Evaluating Markov Chain Monte Carlo (MCMC) Algorithms. (n.d.). https://link.springer.com/content/pdf/10.1007/978-0-387-71265-9_6.pdf (https://link.springer.com/content/pdf/10.1007/978-0-387-71265-9_6.pdf), covergence could be influenced by many factors, for example, the initial values for the parameters. With that being said, I think we could improve our priors for each parameter, but it required more information which we do not have this time. Second, we use swimming time as the response variable and week as the explanatory variable, we could use other predictors to predict y next time, there should more things that are related to a swimmer’s swimming time. According to week11A lecture of the professor, we learned many methods of model assessment. We could use Bayes factors, cross validation, Deviance information criteria(DIC) to compared our initial and revised model so that we could select the best one.

##Problem5

```
###Problem5
data(UScrime)
#Part (a)
#references: professor lecture materials
set.seed(1000)
JAGS_BLR_flat = function(){
  # Likelihood
  for(i in 1:n){
    Y[i] ~ dnorm(mu[i],inv_sigma2)
    mu[i] <- beta_0 + inprod(X[i,],beta)
    # same as beta_0 + X[i,1]*beta[1] + ... + X[i,p]*beta[p]
  }
  # Prior for beta
  for(j in 1:p){
    beta[j] ~ dnorm(0,0.0001)
    #non-informative priors
  }
  # Prior for intercept
  beta_0 ~ dnorm(0, 0.0001)

  # Prior for the inverse variance
  inv_sigma2 ~ dgamma(0.0001, 0.0001)
  sigma2 <- 1.0/inv_sigma2
}
```

```
###Problem5
#Part (a)
set.seed(1000)
mydat <- setNames(list(
  UScrime$y,
  UScrime[,-16],
  nrow(UScrime),
  ncol(UScrime[,-16])
), c("Y", "X", "n", "p"))
p <- mydat$p
```

```
###Problem5
#Part (a)
#references: professor lecture materials
set.seed(1000)
fit_JAGS_flat = jags(data=mydat,
                inits=list(list(beta = rnorm(p),
                                beta_0 = 0,
                                inv_sigma2 = 1)),
                parameters.to.save = c("beta_0","beta","sigma2"),
                n.chains=1,
                n.iter=10000,
                n.burnin=1000,
                model.file=JAGS_BLR_flat)
```

```
## Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
## Graph information:
##    Observed stochastic nodes: 47
##    Unobserved stochastic nodes: 17
##    Total graph size: 917
##
## Initializing model
```

```
###Problem5
#Part (a)
#references: professor lecture materials
print(fit_JAGS_flat)
```

```
## Inference for Bugs model at "C:/Users/CYNTHI~1/AppData/Local/Temp/RtmpmuNAdL/model59801f412c6b.txt", fit using jags,
##  1 chains, each with 10000 iterations (first 1000 discarded), n.thin = 9
##  n.sims = 1000 iterations saved
##            mu.vect  sd.vect     2.5%       25%       50%       75%      97.5%
## beta[1]      6.026    5.728   -6.228     2.541     6.080     9.930     16.529
## beta[2]     -4.103   87.263 -182.821   -62.158    -3.972    52.995    166.040
## beta[3]     13.274    8.291   -2.741     7.233    13.340    19.164     28.765
## beta[4]     25.379   13.387    0.471    16.256    25.128    34.230     52.495
## beta[5]    -13.280   14.843  -44.401   -22.494   -13.017    -3.453     14.480
## beta[6]      0.940    1.707   -2.542    -0.185     0.932     2.089      4.290
## beta[7]     -4.326    2.068   -8.280    -5.718    -4.327    -2.981     -0.201
## beta[8]     -2.371    1.624   -5.513    -3.487    -2.349    -1.219      0.810
## beta[9]     -0.175    0.754   -1.659    -0.685    -0.155     0.336      1.318
## beta[10]     0.392    5.047   -9.337    -3.056     0.255     3.695     10.393
## beta[11]     7.248   10.441  -13.029     0.242     7.079    14.367     26.460
## beta[12]     0.167    1.354   -2.475    -0.741     0.182     1.033      2.976
## beta[13]     5.580    2.793    0.338     3.759     5.534     7.388     11.286
## beta[14]   -14.968  180.483 -195.821   -76.643   -11.166    52.785    186.231
## beta[15]    -0.602    7.232  -15.362    -5.604    -0.610     4.134     13.668
## beta_0     -25.110  212.761 -228.391   -83.827   -16.611    46.257    175.444
## sigma2   79153.866 20590.837 50199.992 64888.117 75707.275 89775.680 125815.958
## deviance   662.047    6.543  652.125   657.584   661.313   665.822    676.774
##
## DIC info (using the rule, pD = var(deviance)/2)
## pD = 21.4 and DIC = 683.5
## DIC is an estimate of expected predictive error (lower deviance is better).
```

```
###Problem5
#Part (a)
#references: professor lecture materials
fit_flat =as.mcmc(fit_JAGS_flat)
summary(fit_flat)
```

```
##
## Iterations = 1001:9992
## Thinning interval = 9
## Number of chains = 1
## Sample size per chain = 1000
##
## 1. Empirical mean and standard deviation for each variable,
##    plus standard error of the mean:
##
##                Mean        SD  Naive SE Time-series SE
## beta[1]      6.0265 5.728e+00   0.18112        0.20154
## beta[10]     0.3920 5.047e+00   0.15958        0.15958
## beta[11]     7.2479 1.044e+01   0.33018        0.33018
## beta[12]     0.1672 1.354e+00   0.04280        0.04280
## beta[13]     5.5796 2.793e+00   0.08832        0.08832
## beta[14]   -14.9684 1.805e+02   5.70737        5.70737
## beta[15]    -0.6020 7.232e+00   0.22870        0.22870
## beta[2]     -4.1029 8.726e+01   2.75951        2.33101
## beta[3]     13.2740 8.291e+00   0.26219        0.24286
## beta[4]     25.3794 1.339e+01   0.42334        0.42334
## beta[5]    -13.2804 1.484e+01   0.46937        0.46937
## beta[6]      0.9399 1.707e+00   0.05399        0.05399
## beta[7]     -4.3264 2.068e+00   0.06538        0.06538
## beta[8]     -2.3714 1.624e+00   0.05136        0.05136
## beta[9]     -0.1745 7.539e-01   0.02384        0.02384
## beta_0     -25.1096 2.128e+02   6.72809        6.72809
## deviance   662.0468 6.543e+00   0.20692        0.20692
## sigma2   79153.8659 2.059e+04 651.13944      651.13944
##
## 2. Quantiles for each variable:
##
##                2.5%        25%        50%       75%      97.5%
## beta[1]     -6.2285     2.5406     6.0797     9.930  1.653e+01
## beta[10]    -9.3372    -3.0559     0.2547     3.695  1.039e+01
## beta[11]   -13.0288     0.2420     7.0790    14.367  2.646e+01
## beta[12]    -2.4751    -0.7408     0.1823     1.033  2.976e+00
## beta[13]     0.3376     3.7592     5.5336     7.388  1.129e+01
## beta[14]  -195.8207   -76.6429   -11.1662    52.785  1.862e+02
## beta[15]   -15.3623    -5.6040    -0.6099     4.134  1.367e+01
## beta[2]   -182.8208   -62.1582    -3.9717    52.995  1.660e+02
## beta[3]     -2.7411     7.2325    13.3395    19.164  2.876e+01
## beta[4]      0.4713    16.2559    25.1280    34.230  5.250e+01
## beta[5]    -44.4015   -22.4939   -13.0170    -3.453  1.448e+01
## beta[6]     -2.5422    -0.1845     0.9317     2.089  4.290e+00
## beta[7]     -8.2798    -5.7180    -4.3275    -2.981 -2.011e-01
## beta[8]     -5.5134    -3.4871    -2.3490    -1.219  8.096e-01
## beta[9]     -1.6592    -0.6854    -0.1547     0.336  1.318e+00
## beta_0    -228.3915   -83.8270   -16.6110    46.257  1.754e+02
## deviance   652.1248   657.5845   661.3129   665.822  6.768e+02
## sigma2   50199.9916 64888.1170 75707.2750 89775.680  1.258e+05
```

1. (ignored the deviance, beta0, sigma2 since they are not our predictors, I keep them for more detailed information) Look at the output in “1. Empirical mean and standard deviation for each variable, plus standard error of the mean”,

   the column of “Mean” represent the marginal posterior mean for each of beta[i].

2. (ignored the deviance, beta0, sigma2 also) Look at the output in “2. Quantiles for each variable:”,

95% credible intervals only need the quantiles of 2.5% and 97.5%. We look at the column of 2.5% and 97.5% for each beta[i]. In confidence interval in the frequent statistics where it is more likely to find no relationship of variables after you run the experiment one more time if the confidence interval includes zero. I think Bayesian could be similar to this criteria which indicates that if my 95% credible interval excludes zero, then we reject the null hypothesis assuming that there is no linear relationship between crimes and a certain explanatory variable beta[i]. In other words, there is a linear relationship between crimes and that certain explanatory variable if 95% credible interval excludes zero. According to the output, the following variables seem strongly predictive of crime rates:

beta[13] 95% credible interval is [0.3376,1.129e+01] beta[4] 95% credible interval is [0.4713,5.250e+01]

(If the number is different when running the rmd, that may due to R studio problem, because I have set seed for my simulations, it should be the same number)

beta[4] is police expenditure in 1960, so that police expenditure in 1960 seems strongly predictive of crime rates.

beta[13] is income inequality, so that income inequality seems strongly predictive of crime rates.

##Problem5 Partb

```
#references: professor lecture materials
set.seed(1000)
JAGS_BLR_SpikeSlab = function(){
  # Likelihood
  for(i in 1:n){
    Y[i] ~ dnorm(mu[i],inv_sigma2)
    mu[i] <- beta_0 + inprod(X[i,],beta)
  }
  # Prior for beta
  for(j in 1:p){
    beta[j] ~ dnorm(0,inv_tau2[j])
    inv_tau2[j] <- (1-gamma[j])*1000+gamma[j]*0.01
    gamma[j] ~ dbern(0.5)
  }
  # Prior for intercept
  beta_0 ~ dnorm(0, 0.0001)

  # Prior for the inverse variance
  inv_sigma2 ~ dgamma(0.0001, 0.0001)
  sigma2 <- 1.0/inv_sigma2
  tau2 <- 1.0/inv_tau2
}
```

```
###Problem5
#Part (b)
set.seed(1000)
mydat1 <- setNames(list(
  UScrime$y,
  UScrime[,-16],
  nrow(UScrime),
  ncol(UScrime[,-16])
), c("Y", "X", "n", "p"))
p <- mydat1$p
```

```
###Problem5
#references: professor lecture materials
#Part (b)
set.seed(1000)
fit_JAGS_SpikeSlab = jags(data=mydat1,
                       inits=list(list(beta = rnorm(p),
                                       beta_0 = 0,
                                       inv_sigma2 = 1,
                                       gamma = rep(1,length=p))),
                       parameters.to.save = c("beta","gamma"),
                       n.chains=1,
                       n.iter=10000,
                       n.burnin=1000,
                       model.file=JAGS_BLR_SpikeSlab)
```

```
## Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
## Graph information:
##    Observed stochastic nodes: 47
##    Unobserved stochastic nodes: 32
##    Total graph size: 997
##
## Initializing model
```

```
###Problem5
#Part (b)
#references: professor lecture materials
print(fit_JAGS_SpikeSlab)
```

```
## Inference for Bugs model at "C:/Users/CYNTHI~1/AppData/Local/Temp/RtmpmuNAdL/model5980769e4caa.txt", fit using jags,
##  1 chains, each with 10000 iterations (first 1000 discarded), n.thin = 9
##  n.sims = 1000 iterations saved
##            mu.vect sd.vect    2.5%     25%     50%     75%   97.5%
## beta[1]      0.481   2.158  -4.169  -0.026   0.009   0.070   6.191
## beta[2]      0.342   6.217 -14.391  -0.041   0.005   0.064  15.046
## beta[3]      0.243   2.862  -5.224  -0.033  -0.001   0.033   8.308
## beta[4]      8.518   4.764  -0.048   7.015   9.114  10.823  17.726
## beta[5]      1.132   4.725  -8.137  -0.035   0.009   1.680  11.382
## beta[6]      0.005   0.282  -0.331  -0.027  -0.003   0.022   0.584
## beta[7]     -0.087   0.455  -1.412  -0.027  -0.003   0.021   0.100
## beta[8]     -0.074   0.600  -1.909  -0.027  -0.002   0.021   0.632
## beta[9]      0.049   0.225  -0.069  -0.018   0.004   0.029   0.849
## beta[10]    -0.358   1.518  -4.890  -0.034  -0.004   0.024   1.899
## beta[11]     0.585   3.034  -4.908  -0.025   0.004   0.041   9.612
## beta[12]    -0.115   0.427  -1.526  -0.032  -0.009   0.015   0.066
## beta[13]     0.864   1.375  -0.064  -0.005   0.032   1.554   4.709
## beta[14]    -2.623  70.916 -18.263  -0.742   0.001   0.068  15.007
## beta[15]     0.538   3.138  -5.612  -0.031   0.004   0.046   9.744
## gamma[1]     0.359   0.480   0.000   0.000   0.000   1.000   1.000
## gamma[2]     0.426   0.495   0.000   0.000   0.000   1.000   1.000
## gamma[3]     0.279   0.449   0.000   0.000   0.000   1.000   1.000
## gamma[4]     0.895   0.307   0.000   1.000   1.000   1.000   1.000
## gamma[5]     0.451   0.498   0.000   0.000   0.000   1.000   1.000
## gamma[6]     0.073   0.260   0.000   0.000   0.000   0.000   1.000
## gamma[7]     0.116   0.320   0.000   0.000   0.000   0.000   1.000
## gamma[8]     0.123   0.329   0.000   0.000   0.000   0.000   1.000
## gamma[9]     0.077   0.267   0.000   0.000   0.000   0.000   1.000
## gamma[10]    0.255   0.436   0.000   0.000   0.000   1.000   1.000
## gamma[11]    0.274   0.446   0.000   0.000   0.000   1.000   1.000
## gamma[12]    0.120   0.325   0.000   0.000   0.000   0.000   1.000
## gamma[13]    0.408   0.492   0.000   0.000   0.000   1.000   1.000
## gamma[14]    0.513   0.500   0.000   0.000   1.000   1.000   1.000
## gamma[15]    0.325   0.469   0.000   0.000   0.000   1.000   1.000
## deviance   663.497   4.165 656.758 661.191 663.173 665.634 672.965
##
## DIC info (using the rule, pD = var(deviance)/2)
## pD = 8.7 and DIC = 672.2
## DIC is an estimate of expected predictive error (lower deviance is better).
```

```
###Problem5
#Part (b)
#references: professor lecture materials
fit_SpikeSlab =as.mcmc(fit_JAGS_SpikeSlab)
summary(fit_SpikeSlab)
```

```
##
## Iterations = 1001:9992
## Thinning interval = 9
## Number of chains = 1
## Sample size per chain = 1000
##
## 1. Empirical mean and standard deviation for each variable,
##    plus standard error of the mean:
##
##                 Mean      SD Naive SE Time-series SE
## beta[1]     0.481052  2.1579 0.068239       0.121244
## beta[10]   -0.357701  1.5183 0.048014       0.082061
## beta[11]    0.584588  3.0343 0.095953       0.123974
## beta[12]   -0.115143  0.4274 0.013515       0.027082
## beta[13]    0.864331  1.3746 0.043468       0.108544
## beta[14]   -2.623413 70.9165 2.242576       2.242576
## beta[15]    0.537837  3.1378 0.099225       0.162340
## beta[2]     0.342246  6.2172 0.196604       0.196604
## beta[3]     0.243275  2.8625 0.090519       0.145058
## beta[4]     8.518454  4.7636 0.150639       0.321993
## beta[5]     1.131657  4.7250 0.149418       0.309330
## beta[6]     0.005474  0.2817 0.008909       0.008909
## beta[7]    -0.087482  0.4552 0.014395       0.025838
## beta[8]    -0.074467  0.6004 0.018986       0.020745
## beta[9]     0.048569  0.2248 0.007109       0.011464
## deviance  663.497138  4.1649 0.131706       0.259535
## gamma[1]    0.359000  0.4799 0.015177       0.042653
## gamma[10]   0.255000  0.4361 0.013790       0.032569
## gamma[11]   0.274000  0.4462 0.014111       0.039142
## gamma[12]   0.120000  0.3251 0.010281       0.021218
## gamma[13]   0.408000  0.4917 0.015549       0.047004
## gamma[14]   0.513000  0.5001 0.015814       0.054962
## gamma[15]   0.325000  0.4686 0.014819       0.045700
## gamma[2]    0.426000  0.4947 0.015645       0.067796
## gamma[3]    0.279000  0.4487 0.014190       0.034276
## gamma[4]    0.895000  0.3067 0.009699       0.035926
## gamma[5]    0.451000  0.4978 0.015743       0.056852
## gamma[6]    0.073000  0.2603 0.008230       0.010636
## gamma[7]    0.116000  0.3204 0.010131       0.018363
## gamma[8]    0.123000  0.3286 0.010391       0.017582
## gamma[9]    0.077000  0.2667 0.008435       0.013472
##
## 2. Quantiles for each variable:
##
##               2.5%       25%       50%      75%    97.5%
## beta[1]    -4.16900 -0.026004  9.494e-03  0.07040  6.19131
## beta[10]   -4.88986 -0.034116 -4.281e-03  0.02447  1.89945
## beta[11]   -4.90763 -0.025291  4.396e-03  0.04070  9.61173
## beta[12]   -1.52649 -0.032119 -8.719e-03  0.01489  0.06587
## beta[13]   -0.06385 -0.005222  3.165e-02  1.55375  4.70921
## beta[14]  -18.26350 -0.742266  1.216e-03  0.06834 15.00703
## beta[15]   -5.61174 -0.031399  3.650e-03  0.04612  9.74411
## beta[2]   -14.39101 -0.041386  5.045e-03  0.06370 15.04622
## beta[3]    -5.22447 -0.032739 -9.728e-04  0.03320  8.30776
## beta[4]    -0.04832  7.015473  9.114e+00 10.82298 17.72602
## beta[5]    -8.13738 -0.034841  8.941e-03  1.68001 11.38192
## beta[6]    -0.33075 -0.026741 -2.867e-03  0.02160  0.58362
## beta[7]    -1.41182 -0.026551 -3.017e-03  0.02132  0.10027
## beta[8]    -1.90897 -0.026587 -1.724e-03  0.02117  0.63197
## beta[9]    -0.06916 -0.017907  4.350e-03  0.02924  0.84912
## deviance  656.75775 661.190886  6.632e+02 665.63422 672.96459
## gamma[1]    0.00000  0.000000  0.000e+00  1.00000  1.00000
## gamma[10]   0.00000  0.000000  0.000e+00  1.00000  1.00000
## gamma[11]   0.00000  0.000000  0.000e+00  1.00000  1.00000
## gamma[12]   0.00000  0.000000  0.000e+00  0.00000  1.00000
## gamma[13]   0.00000  0.000000  0.000e+00  1.00000  1.00000
## gamma[14]   0.00000  0.000000  1.000e+00  1.00000  1.00000
## gamma[15]   0.00000  0.000000  0.000e+00  1.00000  1.00000
```

```
## gamma[2]     0.00000   0.000000  0.000e+00   1.00000   1.00000
## gamma[3]     0.00000   0.000000  0.000e+00   1.00000   1.00000
## gamma[4]     0.00000   1.000000  1.000e+00   1.00000   1.00000
## gamma[5]     0.00000   0.000000  0.000e+00   1.00000   1.00000
## gamma[6]     0.00000   0.000000  0.000e+00   0.00000   1.00000
## gamma[7]     0.00000   0.000000  0.000e+00   0.00000   1.00000
## gamma[8]     0.00000   0.000000  0.000e+00   0.00000   1.00000
## gamma[9]     0.00000   0.000000  0.000e+00   0.00000   1.00000
```

1. (ignored the deviance and gamma since they are not our predictors, I keep them for more detailed model information) Look at the output in "1. Empirical mean and standard deviation for each variable, plus standard error of the mean:",

   the column of "Mean" represent the marginal posterior mean for each of beta[i].

2. (ignored the deviance and gamma also) Look at the output in "2. Quantiles for each variable:",

None of the 95% credible interval excludes zero, so that none of the explanatory variable beta[i] seem strongly predictive of crime rates.

References

Bakker, R. (n.d.). Florida State University Bayesian Workshop. https://spia.uga.edu/faculty_pages/rbakker/bayes/Day2/Day2_Convergence.pdf (https://spia.uga.edu/faculty_pages/rbakker/bayes/Day2/Day2_Convergence.pdf)

Evaluating Markov Chain Monte Carlo (MCMC) Algorithms. (n.d.). https://link.springer.com/content/pdf/10.1007/978-0-387-71265-9_6.pdf (https://link.springer.com/content/pdf/10.1007/978-0-387-71265-9_6.pdf)

Evaluation of MCMC samples. (n.d.). Cross Validated. https://stats.stackexchange.com/questions/311151/evaluation-of-mcmc-samples (https://stats.stackexchange.com/questions/311151/evaluation-of-mcmc-samples)

Gelman, & Hill. (n.d.). wells.dat. http://www.stat.columbia.edu/~gelman/arm/examples/arsenic/wells.dat (http://www.stat.columbia.edu/~gelman/arm/examples/arsenic/wells.dat). Retrieved April 13, 2023, from http://www.stat.columbia.edu/~gelman/arm/examples/arsenic/wells.dat (http://www.stat.columbia.edu/~gelman/arm/examples/arsenic/wells.dat)

Home - RDocumentation. (n.d.). https://www.rdocumentation.org/ (https://www.rdocumentation.org/)

Hu, J. a. a. J. (2020, July 30). Chapter 12 Bayesian Multiple Regression and Logistic Models | Probability and Bayesian Modeling. https://bayesball.github.io/BOOK/bayesian-multiple-regression-and-logistic-models.html#bayesian-logistic-regression (https://bayesball.github.io/BOOK/bayesian-multiple-regression-and-logistic-models.html#bayesian-logistic-regression)

Melting data | R. (n.d.). DataCamp. https://campus.datacamp.com/courses/abc-intro-2-r/data-wrangling?ex=2 (https://campus.datacamp.com/courses/abc-intro-2-r/data-wrangling?ex=2)

Probability Distributions in R (Stat 5101, Geyer). (n.d.). https://www.stat.umn.edu/geyer/old/5101/rlook.html# (https://www.stat.umn.edu/geyer/old/5101/rlook.html#):~:text=dnorm%20is%20the%20R%20function,standard%20deviation%20of%20the%20distributio

Zach. (2022). How to Use the melt() Function in R. Statology. https://www.statology.org/melt-in-r/ (https://www.statology.org/melt-in-r/)