STA365_Assignment2

Ruike Xu

4/15/2022

Question 1

```
#install.packages("R2jags")
library(R2jags)
```

```
## Warning: package 'R2jags' was built under R version 4.1.3
```

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

```
## Warning: package 'rjags' was built under R version 4.1.3
```

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

```
## Warning: package 'coda' was built under R version 4.1.3
```

```
## Linked to JAGS 4.3.0
```

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

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

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

```
#raw_swim <- load(url("https://www.dropbox.com/s/dg17rzny0Owaenb/swim_time.RData"))
```

Question 2

Part a

```r
#install.packages("MASS")
library(MASS)
```

```
## Warning: package 'MASS' was built under R version 4.1.3
```

```r
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.2     v dplyr   1.0.6
## v tidyr   1.1.3     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.1

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## x dplyr::select() masks MASS::select()
```

```r
us_crime_data <- UScrime
```

```r
set.seed(1006562550)
# Using noninformative prior norm(0.001, 0.001)
library(R2jags)
library(xtable)
# Bayesian linear regression model with uninformative priors
JAGS_BLR_nonin = function(){
  # Likelihood
  for(i in 1:47){
    y[i] ~ dnorm(mu[i], tau)
    mu[i] <- beta0 + beta1*M[i] + beta2*So[i] + beta3*Ed[i] + beta4*Po1[i] +
      beta5*Po2[i] + beta6*LF[i] + beta7*M.F[i] + beta8*Pop[i] + beta9*NW[i] +
      beta10*U1[i] + beta11*U2[i] + beta12*GDP[i] + beta13*Ineq[i] +
      beta14*Prob[i] + beta15*Time[i]
    }
  beta0 ~ dnorm(0, 0.0001)
  beta1 ~ dnorm(0, 0.0001)
  beta2 ~ dnorm(0, 0.0001)
  beta3 ~ dnorm(0, 0.0001)
  beta4 ~ dnorm(0, 0.0001)
  beta5 ~ dnorm(0, 0.0001)
  beta6 ~ dnorm(0, 0.0001)
  beta7 ~ dnorm(0, 0.0001)
  beta8 ~ dnorm(0, 0.0001)
  beta9 ~ dnorm(0, 0.0001)
  beta10 ~ dnorm(0, 0.0001)
  beta11 ~ dnorm(0, 0.0001)
  beta12 ~ dnorm(0, 0.0001)
  beta13 ~ dnorm(0, 0.0001)
  beta14 ~ dnorm(0, 0.0001)
  beta15 ~ dnorm(0, 0.0001)
```

```
  tau ~ dgamma (0.0001, 0.0001)
  sigma2 <- 1/tau
}

non_inits <- list(list("beta0" = 0, "beta1" = rnorm(1), "beta2" = rnorm(1),
                       "beta3" = rnorm(1),"beta4" = rnorm(1),"beta5" = rnorm(1),
                       "beta6" = rnorm(1),"beta7" = rnorm(1),"beta8" = rnorm(1),
                       "beta9" = rnorm(1),"beta10" = rnorm(1),
                       "beta11" = rnorm(1),"beta12" = rnorm(1),
                       "beta13" = rnorm(1),"beta14" = rnorm(1),
                       "beta15" = rnorm(1)))

non_parameters <- c("beta0", "beta1", "beta2","beta3","beta4","beta5","beta6",
                    "beta7","beta8","beta9","beta10",
                    "beta11","beta12","beta13","beta14","beta15", "sigma2" )

fit_JAGS_nonin = jags(data = us_crime_data, inits = non_inits,
                      parameters.to.save = non_parameters, n.chains=1,
                      n.iter=10000, n.burnin=1000, model.file=JAGS_BLR_nonin )
```

```
## module glm loaded

## Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
## Graph information:
##    Observed stochastic nodes: 47
##    Unobserved stochastic nodes: 17
##    Total graph size: 1350
##
## Initializing model
```

```
#print(fit_JAGS_nonin)

mcmc_non = as.mcmc(fit_JAGS_nonin)

mcmc_non_summary <- summary(mcmc_non)
```

```
post_mean_non <- mcmc_non_summary$statistics[, "Mean"]
ci_lower_non <- mcmc_non_summary$quantiles[, "2.5%"]
ci_upper_non <- mcmc_non_summary$quantiles[, "97.5%"]
expansionary <- c("", "M", "U1", "U2", "GDP", "Ineq", "Prob", "Time", "So",
                  "Ed", "Po1", "Po2", "LF", "M.F", "Pop", "NW", "", "")
table_non <- data.frame("Expansionary variable" = expansionary,
                        "Marginal posterior mean" = post_mean_non,
                        "Credible interval lower bound" = ci_lower_non,
                        "Credible interval upper bound" = ci_upper_non)
#xtable(table_non)
```

The above table shows the marginal posterior mean and 95% credible interval for our constructed Bayesian linear regression with non-informative prior. We employ N(0, 0.0001) as the non-informative prior for our expansionary variables and intercept, the variance for response variable follows a Gamma(0.0001, 0.0001)

	Expansionary.variable	Marginal.posterior.mean	CI.lower.bound	CI.upper.bound
beta0		-21.89	-209.43	182.42
beta1	M	6.53	-3.77	17.31
beta10	U1	0.59	-9.20	10.52
beta11	U2	7.37	-13.52	27.42
beta12	GDP	0.20	-2.36	2.76
beta13	Ineq	5.67	0.11	11.40
beta14	Prob	-13.99	-207.35	187.89
beta15	Time	-0.74	-14.80	14.24
beta2	So	-4.11	-180.81	180.17
beta3	Ed	13.52	-2.95	29.47
beta4	Po1	25.28	-3.43	51.07
beta5	Po2	-13.24	-42.40	17.59
beta6	LF	0.99	-2.42	4.18
beta7	M.F	-4.51	-8.38	-0.44
beta8	Pop	-2.28	-5.54	0.87
beta9	NW	-0.21	-1.64	1.39
deviance		662.26	651.86	676.30
sigma2		78683.94	48071.26	127872.08

Table 1: Summary of marginal posterior means and 95% credible intervals for non-informative prior

distribution. Since we are constructing a linear regression, for instance, holding all else constant, when M (percentage of males aged 14-24) increased by one unit, the rate of crimes per head of population (y) would increase by 6.53. The other expansionary variables play the same role as the previous example. Under 0.05 level of significance, if 0 falls into the range of the 95% credible intervals, we wouldn't be able to reject the null hypothesis that this expansionary variable is not significant. We can observe that only expansionary variables Ineq (Income inequality) and M.F (Number of males per 1000 females) seems significant in our model under 0.05 level of significance. Thus, Ineq and M.F seem to be strongly predictive of crime rate.

Part b

```
# Create training and testing data from random sampling
set.seed(1006562550)
us_crime_data$ID <- c(1:47)
train <- us_crime_data[sample(seq_len(nrow(us_crime_data)), size = 24), ]
test <- us_crime_data[!us_crime_data$ID %in% train$ID, ]
```

```
# Bayesian linear regression model with uninformative priors using training data
JAGS_BLR_nonin_train = function(){
  # Likelihood
  for(i in 1:24){
    y[i] ~ dnorm(mu[i], tau)
    mu[i] <- beta0 + beta1*M[i] + beta2*So[i] + beta3*Ed[i] + beta4*Po1[i] +
      beta5*Po2[i] + beta6*LF[i] + beta7*M.F[i] + beta8*Pop[i] + beta9*NW[i] +
      beta10*U1[i] + beta11*U2[i] + beta12*GDP[i] + beta13*Ineq[i] +
      beta14*Prob[i] + beta15*Time[i]
    }
  beta0 ~ dnorm(0, 0.0001)
  beta1 ~ dnorm(0, 0.0001)
  beta2 ~ dnorm(0, 0.0001)
```

```
  beta3 ~ dnorm(0, 0.0001)
  beta4 ~ dnorm(0, 0.0001)
  beta5 ~ dnorm(0, 0.0001)
  beta6 ~ dnorm(0, 0.0001)
  beta7 ~ dnorm(0, 0.0001)
  beta8 ~ dnorm(0, 0.0001)
  beta9 ~ dnorm(0, 0.0001)
  beta10 ~ dnorm(0, 0.0001)
  beta11 ~ dnorm(0, 0.0001)
  beta12 ~ dnorm(0, 0.0001)
  beta13 ~ dnorm(0, 0.0001)
  beta14 ~ dnorm(0, 0.0001)
  beta15 ~ dnorm(0, 0.0001)
  tau ~ dgamma (0.0001, 0.0001)
  sigma2 <- 1/tau
}

non_inits <- list(list("beta0" = 0, "beta1" = rnorm(1), "beta2" = rnorm(1),
                       "beta3" = rnorm(1),"beta4" = rnorm(1),"beta5" = rnorm(1),
                       "beta6" = rnorm(1),"beta7" = rnorm(1),"beta8" = rnorm(1),
                       "beta9" = rnorm(1),"beta10" = rnorm(1),
                       "beta11" = rnorm(1),"beta12" = rnorm(1),
                       "beta13" = rnorm(1),"beta14" = rnorm(1),
                       "beta15" = rnorm(1)))

non_parameters <- c("beta0", "beta1", "beta2","beta3","beta4","beta5","beta6",
                    "beta7","beta8","beta9","beta10",
                    "beta11","beta12","beta13","beta14","beta15", "sigma2" )

fit_JAGS_nonin_train = jags(data = train, inits = non_inits,
                            parameters.to.save = non_parameters, n.chains=1,
                            n.iter=1100, n.burnin=100,
                            model.file=JAGS_BLR_nonin_train )
```

```
## Warning in jags.model(model.file, data = data, inits = init.values, n.chains =
## n.chains, : Unused variable "ID" in data

## Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
## Graph information:
##    Observed stochastic nodes: 24
##    Unobserved stochastic nodes: 17
##    Total graph size: 726
##
## Initializing model
```

```
#print(fit_JAGS_nonin_train)

mcmc_non_train = as.mcmc(fit_JAGS_nonin_train)
```

```
ma <- as.matrix(mcmc_non_train)
# Using for loop to calculate 1000 iteration medians
overall_median <- c()
for (i in 1:1000){
  y = c()
  for (j in 1:23){
    y[j] = ma[i,1] + ma[i,2]*test[j,1] + ma[i, 3]*test[j,10] +
      ma[i,4]*test[j,11] + ma[i,5]*test[j,12] + ma[i,6]*test[j,13] +
      ma[i,7]*test[j,14] + ma[i,8]*test[j,15] + ma[i,9]*test[j,2] +
      ma[i,10]*test[j,3] + ma[i,11]*test[j,4] + ma[i,12]*test[j,5] +
      ma[i,13]*test[j,6] + ma[i,14]*test[j,7] + ma[i,15]*test[j,8] +
      ma[i,16]*test[j,9]
  overall_median[i] = median(y)
  }
}

# Find posterior predictive median of 1000 iterations median
posterior_non_median <- median(overall_median)

# Find actual crime rate median in test data
actual_median <- median(test$y)

posterior_non_median
```

```
## [1] 722.4128
```

```
actual_median
```

```
## [1] 856
```

We employ a for loop to calculate the median of response variable of 23 test observations based on predictors in test data, and we repeat the procedure for 1000 iterations. The posterior predictive median (722.41) is a bit smaller than the actual crime rate medium (856) in the test dataset.

Part c

```
# Construct Bayesian linear regression based on spike-and-slab priors
JAGS_BLR_SpikeSlab = function(){
  # Likelihood
  for(i in 1:47){
    y[i] ~ dnorm(mu[i],inv_sigma2)
    mu[i] <- beta0 + inprod(X[i,],beta)
  }
  # Prior for beta
  for(j in 1:15){
    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
```

```
  beta0 ~ 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
}
```

```
# Fit the Bayesian spike-and-slab linear model
us_crime_predictor = as.matrix(UScrime%>%select(-16))

fit_JAGS_SpikeSlab = jags(data=list(X = us_crime_predictor, y = UScrime$y),
                          inits=list(list(beta = c(rnorm(15)),
                                          beta0 = 0,
                                          inv_sigma2 = 1,
                                          gamma = rep(1,length=15))),
                          parameters.to.save = c("beta0", "beta"),
                          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: 995
##
## Initializing model
```

```
# Transform fitted JAGs model to mcmc object

mcmc_spike = as.mcmc(fit_JAGS_SpikeSlab)

mcmc_spike_summary <- summary(mcmc_spike)
```

```
# Extract summary statistics of Spike-slab model and construct summary table

post_mean_spike <- mcmc_spike_summary$statistics[, "Mean"]
ci_lower_spike <- mcmc_spike_summary$quantiles[, "2.5%"]
ci_upper_spike <- mcmc_spike_summary$quantiles[, "97.5%"]
expansionary_spike <- c("M", "U1", "U2", "GDP", "Ineq", "Prob", "Time", "So",
                        "Ed", "Po1", "Po2", "LF", "M.F", "Pop", "NW", "", "")
table_spike <- data.frame("Expansionary variable" = expansionary_spike,
                          "Marginal posterior mean" = post_mean_spike,
                          "Credible interval lower bound" = ci_lower_spike,
                          "Credible interval upper bound" = ci_upper_spike)
#xtable(table_spike)
```

The above table shows the marginal posterior mean and 95% credible interval for our constructed Bayesian linear regression with Spike-and-slab prior. Since we are constructing a linear regression, for instance, holding

	Expansionary.variable	Marginal.posterior.mean	CI.lower.bound	CI.upper.bound
beta[1]	M	0.40	-4.58	6.01
beta[10]	U1	-0.28	-3.69	1.08
beta[11]	U2	0.45	-5.31	8.00
beta[12]	GDP	-0.10	-1.54	0.07
beta[13]	Ineq	0.96	-0.07	4.77
beta[14]	Prob	-1.92	-15.87	17.11
beta[15]	Time	0.25	-7.37	8.88
beta[2]	So	0.32	-15.55	17.10
beta[3]	Ed	0.18	-4.25	7.46
beta[4]	Po1	8.42	-0.05	18.11
beta[5]	Po2	1.30	-9.54	12.03
beta[6]	LF	-0.01	-0.65	0.18
beta[7]	M.F	-0.09	-1.17	0.07
beta[8]	Pop	-0.11	-2.08	0.21
beta[9]	NW	0.06	-0.07	0.95
beta0		-23.31	-220.32	171.54
deviance		663.31	656.80	670.80

Table 2: Summary of marginal posterior means and 95% credible intervals with Spike-and-slab prior

all else constant, when M (percentage of males aged 14-24) increased by one unit, the rate of crimes per head of population (y) would increase by 0.40. The other expansionary variables can be interpret in a similar way. Under 0.05 level of significance, if 0 falls into the range of the 95% credible intervals, we wouldn't be able to reject the null hypothesis that this expansionary variable is not significant. We can observe that all expansionary variables' credible intervals capture 0, which means that all variables seems insignificant in our model under 0.05 level of significance.

```
# Construct Bayesian linear regression based on spike-and-slab priors
# and train model using training data
JAGS_BLR_SpikeSlab_train = function(){
  # Likelihood
  for(i in 1:24){
    y[i] ~ dnorm(mu[i],inv_sigma2)
    mu[i] <- beta0 + inprod(X[i,],beta)
  }
  # Prior for beta
  for(j in 1:15){
    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
  beta0 ~ 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
}
```

```
# Fit the Bayesian spike-and-slab linear model using training data
us_crime_predictor_spike = as.matrix(train%>%select(-c(16, 17)))
```

```
fit_JAGS_SpikeSlab_train = jags(data=list(X = us_crime_predictor_spike,
                                          y = train$y),
                        inits=list(list(beta = c(rnorm(15)),
                                        beta0 = 0,
                                        inv_sigma2 = 1,
                                        gamma = rep(1,length=15))),
                        parameters.to.save = c("beta0", "beta"),
                        n.chains=1,
                        n.iter=10000,
                        n.burnin=1000,
                        model.file=JAGS_BLR_SpikeSlab_train)
```

```
## Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
## Graph information:
##    Observed stochastic nodes: 24
##    Unobserved stochastic nodes: 32
##    Total graph size: 558
##
## Initializing model
```

```
mcmc_spike_train = as.mcmc(fit_JAGS_SpikeSlab_train)

ma2 <- as.matrix(mcmc_spike_train)

overall_median_spike <- c()

for (i in 1:1000){
  y = c()
  for (j in 1:23){
    y[j] = ma2[i,1] + ma2[i,2]*test[j,1] + ma2[i, 3]*test[j,10] +
      ma2[i,4]*test[j,11] + ma2[i,5]*test[j,12] + ma2[i,6]*test[j,13] +
      ma2[i,7]*test[j,14] + ma2[i,8]*test[j,15] + ma2[i,9]*test[j,2] +
      ma2[i,10]*test[j,3] + ma2[i,11]*test[j,4] + ma2[i,12]*test[j,5] +
      ma2[i,13]*test[j,6] + ma2[i,14]*test[j,7] + ma2[i,15]*test[j,8] +
      ma2[i,16]*test[j,9]
  overall_median_spike[i] = median(y)
  }
}

# Find posterior predictive median of 1000 iterations median
posterior_spike_median <- median(overall_median_spike)

# Find actual crime rate median in test data
actual_median <- median(test$y)

posterior_spike_median
```

```
## [1] 978.1336
```

```
actual_median
```

```
## [1] 856
```

We employ a for loop to calculate the median of response variable of 23 test observations based on predictors in test data, and we repeat the procedure for 1000 iterations. The posterior predictive median for spike-and-slab prior (978.1336) is higher than the actual crime rate medium (856) in the test dataset.

Question 3

Part a

```
#install.packages("geoR")
library(geoR)
```

```
## Warning: package 'geoR' was built under R version 4.1.3
```

```
## ----------------------------------------------------------------
##  Analysis of Geostatistical Data
##  For an Introduction to geoR go to http://www.leg.ufpr.br/geoR
##  geoR version 1.8-1 (built on 2020-02-08) is now loaded
## ----------------------------------------------------------------
```

```
v_loc = unique(gambia[,"x"])
v = match(gambia[,"x"],v_loc)
```

```
# Constructing a logistic
JAGS_gambia_model = function() {
  # Likelihood
  for (i in 1:n) {
    Y[i] ~ dbern(prob[i])
    logit(prob[i]) <-  alpha0 + inprod(X[i,],beta)
  }
  #prior
  alpha0 ~ dnorm(mu_a, sigma_a2)
  mu_a ~ dnorm(0, 0.001)
  sigma_a2 ~ dnorm(0, 0.001)

  for(j in 1:p){
    beta[j] ~ dnorm(mu_b,sigma_b2)
    mu_b ~ dnorm(0, 0.001)
    sigma_b2 ~ dnorm(0, 0.001)
  }
}
```

```
# # Fit the logistic regression model with gambia data
# gambia_predictor <- as.matrix(gambia%>%select(-8))
#
```

```
# fit_JAGS_gambia_model = jags(data = list(X = gambia_predictor, Y = gambia$y,
#                                          n = 2035, p = 7),
#   inits = list(list(beta=c(rep(0, 7)), mu_a = 0, sigma_a2 = 0, mu_b = 0,
#                     sigma_b2 = 0)),
#   parameters.to.save = c("phc"),
#   n.chains = 1,
#   n.iter = 1000,
#   n.burnin = 200,
#   model.file = JAGS_gambia_model
# )
```