Homework 2

Gaurav Kandlikar

October 5, 2016

```r
library(R2jags)
```

```
## Loading required package: rjags

## Loading required package: coda

## Linked to JAGS 4.2.0

## Loaded modules: basemod,bugs

##
## Attaching package: 'R2jags'

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

```r
library(lattice) # Needed for scatterplot matrix
# Set working directory
setwd("/home/gsk/grad/courses/UCLA/biost234/homework/lab2")
getwd()
```

```
## [1] "/home/gsk/grad/courses/UCLA/biost234/homework/lab2"
```

```r
# Save the data file to your working directory.

#READ IN DATA
housing=read.table("housingdata2.txt")
head(housing)   ## Always look at your data
```

```
##       V1   V2   V3 V4 V5
## 1 15.783 3.00 2.00  2  2
## 2 12.570 1.66 2.33  3  2
## 3 19.600 3.33 2.33  2  2
## 4  8.206 1.66 1.66  2  2
## 5 15.333 2.33 2.33  5  2
## 6 14.955 5.00 3.00  2  2
```

```r
## Anything funny about any of the columns?
## There are no column names!
str(housing)
```

```
## 'data.frame':    21 obs. of  5 variables:
##  $ V1: num  15.78 12.57 19.6 8.21 15.33 ...
##  $ V2: num  3 1.66 3.33 1.66 2.33 5 4.33 2.33 1.33 3 ...
##  $ V3: num  2 2.33 2.33 1.66 2.33 3 3 2.33 1.66 2.66 ...
##  $ V4: int  2 3 2 2 5 2 2 3 2 2 ...
##  $ V5: int  2 2 2 2 2 2 2 2 2 2 ...
```

```r
colnames(housing) <- c("cost", "eaves", "windows", "yard", "roof")

#SEPARATE X & Y
y <- housing[,1]
x <- as.matrix(housing[,2:5])

## Remember: look at your data.
y
```

```
##  [1] 15.783 12.570 19.600  8.206 15.333 14.955 13.710 11.388  4.802 12.547
## [11] 13.677  9.683 16.798 25.615 15.734 13.510 13.855  3.986  5.997  9.778
## [21] 10.152
```

```r
head(x)
```

```
##      eaves windows yard roof
## [1,]  3.00    2.00    2    2
## [2,]  1.66    2.33    3    2
## [3,]  3.33    2.33    2    2
## [4,]  1.66    1.66    2    2
## [5,]  2.33    2.33    5    2
## [6,]  5.00    3.00    2    2
```

Classical regression

Before beginning our Bayesian analysis, we can conduct classical multiple linear regression:

```r
reg = lm(y~x)
summary(reg) # classical regression
```

```
##
## Call:
## lm(formula = y ~ x)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.8827 -1.8219 -0.9953  1.3467  7.4674
##
## Coefficients: (1 not defined because of singularities)
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -7.5055     3.9087  -1.920  0.07177 .
## xeaves        1.8981     0.9868   1.923  0.07133 .
## xwindows      3.5310     1.6703   2.114  0.04960 *
## xyard         2.5450     0.8721   2.918  0.00958 **
```

```
## xroof              NA         NA      NA       NA
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.275 on 17 degrees of freedom
## Multiple R-squared:  0.6318, Adjusted R-squared:  0.5668
## F-statistic: 9.722 on 3 and 17 DF,  p-value: 0.0005765
```

The yard and window quality seem to correlate with the contractor's estimates. We see a row of NAs for coefficients of roof, since all of the roofs in this dataset had a quality of 2- so, there is no variation in that measurement.

Bayesian approach

First we write out our model

```
# sink("housingmodel.txt")
cat("
model
{
   for(i in 1:N) {
         y[i] ~ dnorm(mu[i] , tau )
         mu[i] <- beta0 + inprod(x[i,] , beta[] )
        }

     beta0 ~ dnorm( mbeta0 , precbeta0) # mean and precision of beta0 (intercept) defined in R

for (j in 1:K) {
     beta[j] ~ dnorm( m[j] , prec[j] ) # the prior for the four different betas will be provided throug
                                      # in all models we will use mean beta values estimated in an old a

        }
       tau ~ dgamma( tau.a , tau.b )  # we tend to use gamma for tau because it is always non-negative
                                    # we will weigh the priors differently in a few different models by
                                    # fidgeting with the precision in prior
       sigma <- 1 / sqrt(tau)         # note that we use <- here since it's a simple calculation
    }
  ",fill = TRUE)
```

```
##
## model
## {
##    for(i in 1:N) {
##       y[i] ~ dnorm(mu[i] , tau )
##       mu[i] <- beta0 + inprod(x[i,] , beta[] )
##      }
##
##   beta0 ~ dnorm( mbeta0 , precbeta0) # mean and precision of beta0 (intercept) defined in R
##
## for (j in 1:K) {
##   beta[j] ~ dnorm( m[j] , prec[j] ) # the prior for the four different betas will be provided throug
##                                       # in all models we will use mean beta values estimated in an old
##
```

```
##      }
##     tau ~ dgamma( tau.a , tau.b )  # we tend to use gamma for tau because it is always non-negative
##                                     # we will weigh the priors differently in a few different models
##                                     # fidgeting with the precision in prior
##     sigma <- 1 / sqrt(tau)         # note that we use <- here since it's a simple calculation
##   }
##
```

```
# sink()
```

Now, we set up three different models:

```
# define some variables
N = nrow(housing)
K = ncol(x) # number of coefficients to estimate betas for
m = c(1.6053, 1.2556, 2.3413, 3.6771)# mean betas from previous analysis

dataA <- list(N=N, K=4, m = m,
              prec = c(.2164, .1105, .2061, .1337), tau.a=17,
              tau.b = 1128, mbeta0= -5.682, precbeta0=.05464, x=x, y=y)

dataB <- list(N=N, K=4, m=m,
            prec=c(.02774, .014160, .02642, .01714), tau.a=2.1795,
            tau.b=144.6, mbeta0= -5.682, precbeta0=.007005, x=x, y=y)

dataC <- list(N=N, K=4, m=m,
            prec=c(.005549, .002832, .005284, .003428), tau.a=.4359,
            tau.b=28.92, mbeta0= -5.682, precbeta0=.00140, x=x, y=y)


inits <- rep(list(list(beta0=0, beta=c(1,1,1,1),tau=1)),5) # 5 equal to the n.chains in jags call

#DEFINE PARAMETERS TO MONITOR
parameters <- c("beta0", "beta" , "tau", "sigma")
```

Then we run jags:

```
#RUN THE JAGS PROGRAM, SAVING DATA TO LAB2.SIM
lab2.simA <- jags (dataA, inits, parameters, "housingmodel.txt", n.chains=5,
    n.iter=5100, n.burnin=100, n.thin=1, DIC=FALSE) # DIC = F - deviance not calcuated
```

```
## module glm loaded

## module dic loaded

## Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
## Graph information:
##    Observed stochastic nodes: 21
##    Unobserved stochastic nodes: 6
##    Total graph size: 191
##
## Initializing model
```

```r
lab2.simB <- jags (dataB, inits, parameters, "housingmodel.txt", n.chains=5,
    n.iter=5100, n.burnin=100, n.thin=1, DIC=FALSE)
```

```
## Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
## Graph information:
##    Observed stochastic nodes: 21
##    Unobserved stochastic nodes: 6
##    Total graph size: 191
##
## Initializing model
```

```r
lab2.simC <- jags (dataC, inits, parameters, "housingmodel.txt", n.chains=5,
    n.iter=5100, n.burnin=100, n.thin=1, DIC=FALSE)
```

```
## Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
## Graph information:
##    Observed stochastic nodes: 21
##    Unobserved stochastic nodes: 6
##    Total graph size: 191
##
## Initializing model
```

```r
lab2.sims = list(lab2.simA, lab2.simB, lab2.simC)

knitr::kable(lab2.simA$BUGSoutput$summary[,c("mean", "sd", "2.5%", "97.5%")], digits = 3)
```

	mean	sd	2.5%	97.5%
beta[1]	1.794	1.374	-0.897	4.498
beta[2]	1.769	1.994	-2.115	5.670
beta[3]	2.212	1.308	-0.342	4.770
beta[4]	2.627	2.243	-1.764	6.982
beta0	-6.987	3.851	-14.514	0.531
sigma	6.940	0.687	5.752	8.434
tau	0.021	0.004	0.014	0.030

```r
knitr::kable(lab2.simB$BUGSoutput$summary[,c("mean", "sd", "2.5%", "97.5%")], digits = 3)
```

	mean	sd	2.5%	97.5%
beta[1]	1.906	1.412	-0.840	4.702
beta[2]	3.190	2.354	-1.470	7.811
beta[3]	2.505	1.258	0.031	5.002
beta[4]	1.173	5.036	-8.690	11.075
beta0	-8.890	9.713	-28.066	10.179
sigma	4.837	0.772	3.609	6.629

	mean	sd	2.5%	97.5%
tau	0.046	0.014	0.023	0.077

```
knitr::kable(lab2.simC$BUGSoutput$summary[,c("mean", "sd", "2.5%", "97.5%")], digits = 3)
```

	mean	sd	2.5%	97.5%
beta[1]	1.904	1.162	-0.432	4.227
beta[2]	3.486	1.957	-0.385	7.310
beta[3]	2.538	1.026	0.504	4.567
beta[4]	0.741	10.665	-20.209	21.939
beta0	-8.873	21.175	-51.062	32.751
sigma	3.823	0.679	2.763	5.391
tau	0.075	0.025	0.034	0.131

Question 1

1. Summarize briefly the effects on all parameters of changing from prior A to B to C. For all priors, changing from A to B to C brought the point estimate (mean of posterior) closer to the likelihood estimates reported at the beginning of this document and increased the magnitude of the credibility intervals.

Question 2

2. Give a table of inferences for the coefficient of roofs for the three priors. Briefly explain why it comes out as it does.

```
temp3 <- t(sapply(lab2.sims, function(x) x$BUGSoutput$summary["beta[4]",
                                                              c("mean", "sd", "2.5%", "97.5%")]))
rownames(temp3) <- c("Model A", "Model B", "Model C")
knitr::kable(temp3, caption = "Estimates of Roof coefficient under three priors")
```

Table 4: Estimates of Roof coefficient under three priors

	mean	sd	2.5%	97.5%
Model A	2.6270344	2.242930	-1.763813	6.982076
Model B	1.1729911	5.036322	-8.689873	11.074854
Model C	0.7409143	10.664672	-20.209143	21.939270

In our dataset there is no variation in the roof parameter across samples, so as we place less and less emphasis on our prior (moving from models A to B to C), our estimate of β_{roof} gets wider and more centered on zero.

Question 3

3. For one of the three priors:

b. Which house is in the worst condition? Calculate the three futurefit, futureobs and futuretail variables for this house and provide a formatted table.

First we write a new model:

```
# sink("housingmodelq3.txt")
cat("
model
{
   for(i in 1:N) {
         y[i] ~ dnorm(mu[i] , tau )
         mu[i] <- beta0 + inprod(x[i,] , beta[] )
        }

     beta0 ~ dnorm( mbeta0 , precbeta0) # mean and precision of beta0 (intercept) defined in R

for (j in 1:K) {
```

```
    beta[j] ~ dnorm( m[j] , prec[j] ) # the prior for the four different betas will be provided throug
                                   # in all models we will use mean beta values estimated in an old a

       }
      tau ~ dgamma( tau.a , tau.b )  # we tend to use gamma for tau because it is always non-negative
                                  # we will weigh the priors differently in a few different models by
                                  # fidgeting with the precision in prior
      sigma <- 1 / sqrt(tau)         # note that we use <- here since it's a simple calculation
    futurefit <- beta0 + beta[1]*fut[1] + beta[2]*fut[2] + beta[3]*fut[3] + beta[4]*fut[4]
        # note that I made fut[] be an input vector so we don't need to rewrite
        # new models every time we want to fit future for a different house
    futureobs ~ dnorm(futurefit, tau)
    futuretail <- beta0 + beta[1]*fut[1] + beta[2]*fut[2] + beta[3]*fut[3] + beta[4]*fut[4] + 1.645*sig
    }
  ",fill = TRUE)
```

```
##
## model
## {
##    for(i in 1:N) {
##       y[i] ~ dnorm(mu[i] , tau )
##       mu[i] <- beta0 + inprod(x[i,] , beta[] )
##      }
##
##   beta0 ~ dnorm( mbeta0 , precbeta0) # mean and precision of beta0 (intercept) defined in R
##
## for (j in 1:K) {
##   beta[j] ~ dnorm( m[j] , prec[j] ) # the prior for the four different betas will be provided throug
##                                     # in all models we will use mean beta values estimated in an ol
##
##      }
##     tau ~ dgamma( tau.a , tau.b )  # we tend to use gamma for tau because it is always non-negative
##                                    # we will weigh the priors differently in a few different models
##                                    # fidgeting with the precision in prior
##     sigma <- 1 / sqrt(tau)         # note that we use <- here since it's a simple calculation
##     futurefit <- beta0 + beta[1]*fut[1] + beta[2]*fut[2] + beta[3]*fut[3] + beta[4]*fut[4]
##         # note that I made fut[] be an input vector so we don't need to rewrite
##         # new models every time we want to fit future for a different house
##     futureobs ~ dnorm(futurefit, tau)
##     futuretail <- beta0 + beta[1]*fut[1] + beta[2]*fut[2] + beta[3]*fut[3] + beta[4]*fut[4] + 1.645*
##  }
##
```

```
# sink()

inits <- rep(list(list(beta0=0, beta=c(1,1,1,1),tau=1, futureobs = 0)),5) # 5 equal to the n.chains in
parameters <- c("beta0", "beta" , "tau", "sigma", "futurefit", "futureobs", "futuretail")
```

I will run this with Model A (most informative prior)

```
dataA <- list(N=N, K=4, m = m,
              prec = c(.2164, .1105, .2061, .1337), tau.a=17,
```

```
             tau.b = 1128, mbeta0= -5.682, precbeta0=.05464, x=x, y=y,
             fut = c(1,1,2,2)) # added a futures line here for the model

lab2.simq3 <- jags (dataA, inits, parameters, "housingmodelq3.txt", n.chains=5,
    n.iter=5100, n.burnin=100, n.thin=1, DIC=FALSE) # DIC = F - deviance not calcuated
```

```
## Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
## Graph information:
##    Observed stochastic nodes: 21
##    Unobserved stochastic nodes: 7
##    Total graph size: 206
##
## Initializing model
```

a. Show summaries of the futurefit, futureobs, futuretail in a properly formatted table for the house in perfect condition.

```
futures <- c("futurefit", "futureobs", "futuretail")
for_table <- c("mean", "sd", "2.5%", "97.5%")
knitr::kable(lab2.simq3$BUGSoutput$summary[futures, for_table],
             caption = "Summary of predictions for a perfect house")
```

Table 5: Summary of predictions for a perfect house

	mean	sd	2.5%	97.5%
futurefit	6.235914	2.926471	0.478876	11.97177
futureobs	6.231245	7.525624	-8.473874	20.93633
futuretail	17.647019	3.221702	11.444381	24.11591

b. Which house is in the worst condition? Calculate the three futurefit, futureobs and futuretail variables for this house and provide a formatted table.
By one measure, house 14 is the "worst", as it has the highest average score. Its scores are:

```
x[14,]
```

```
##   eaves windows    yard    roof
##    3.00    3.33    4.00    2.00
```

```
dataA <- list(N=N, K=4, m = m,
             prec = c(.2164, .1105, .2061, .1337), tau.a=17,
             tau.b = 1128, mbeta0= -5.682, precbeta0=.05464, x=x, y=y,
             fut = x[14,]) # futures for house 14
lab2.simq3b <- jags (dataA, inits, parameters, "housingmodelq3.txt", n.chains=5,
    n.iter=5100, n.burnin=100, n.thin=1, DIC=FALSE) # DIC = F - deviance not calcuated
```

```
## Compiling model graph
##    Resolving undeclared variables
```

```
##    Allocating nodes
## Graph information:
##    Observed stochastic nodes: 21
##    Unobserved stochastic nodes: 7
##    Total graph size: 206
##
## Initializing model
```

```
knitr::kable(lab2.simq3b$BUGSoutput$summary[futures, for_table],
             caption = "Summary of predictions for the worst house")
```

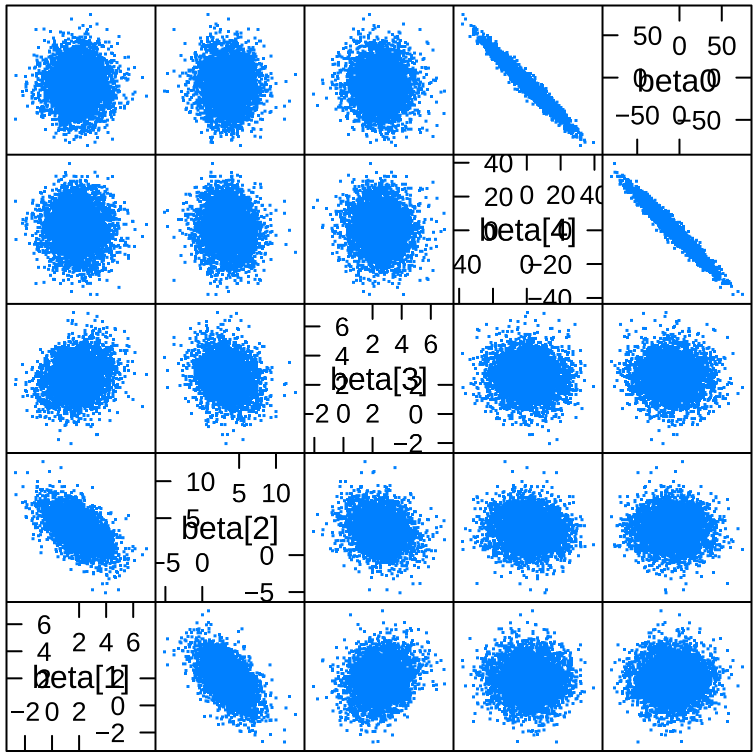
Table 6: Summary of predictions for the worst house

	mean	sd	2.5%	97.5%
futurefit	18.36006	2.727273	12.955898	23.59813
futureobs	18.35698	7.484771	3.647901	33.07167
futuretail	29.77082	2.914689	24.247941	35.59320

Question 4

4. For prior (C), what two coefficients (including the intercept) have the highest posterior correlation? Briefly explain why.

```
temp = apply(lab2.simC$BUGSoutput$sims.array, 3, unlist)
lattice::splom(temp[1:5000,1:5],pch=".")     #   Scatterplot matrix of correlation plots
```



Scatter Plot Matrix

```
# 25000 takes a long time. "." is a better character to plot with
```

We can see in the plot above that beta0 and beta[4] (i.e. the estimates for intercept and coefficient for roof, respectively) are highly correlated with Prior C. This is because priors for both were very weak in this model **SAY MORE HERE!**

Question 5

5. Briefly interpret the three variables futurefit, futureobs, futuretail in your own words.

`futurefit` is the mean of the distribution that is used to estimate the cost to fix some unknown house.
`futureobs` is a sample from the distribution of predictions for the cost estimate; the distribution has mean `futurefit` and precision τ.
`futuretail` is the upper limit of the credible interval around the value reported by `futureobs`; we can similarly calculate the lower bound of the interval.

Question 6

6. Suppose we pool the two data sets after the inflation correction. Also, the expert at the housing department told you he thought each unit increase in any rating scale ought to increase the cost by around \$1000. You're not sure that all coefficients should be positive. Suggest priors (all regression coefficients and for sigma^2) to use now. Write one or two sentences justifying your priors.

- Intercept β_0 Ño (-5.682, .007005)

- EAVES: β_1 Ño (1, .2164)
- WINDOWS: β_2 Ño (1, .1105)
- YARDS: β_3 Ño (1, .2061)
- ROOF: β_4 Ño (1, .1337)
- Tau: τ G̃amma(17, 1128)

Since the local housing expert thinks that each unit increase in any of the scales results in a ~$1000 increase in repair cost, I use 1 as the prior mean for all coefficients in this scenario. I also set the precision to 1 so that ~85% of my prior distribution is greater than zero, but there is still a fair amount of density in the priors for negative coefficients.