

Assignment3

Benjamin Moran

28 October 2016

Question 1

The bioassay data below was presented and analysed at various times during the course.

Log dosage	No. of animals	No. of deaths
-0.863	5	0
-0.296	5	1
-0.053	5	3
0.727	5	5

This was an experiment done on 20 animals to estimate the toxicity of a certain drug. The data is the number of deaths (out of n) corresponding to a different dosage levels of the drug. We modeled this data as a binomial regression model as follows:

$$y_t \mid p_t \sim Bin(5, p_t); logit(p_t) = \log\left(\frac{p_t}{1 - p_t}\right) = \alpha + \beta x_t$$

- (a) Perform inference for this model under a flat prior and find point estimates and 95% intervals for each parameter using WinBUGS.

```
require(R2OpenBUGS)
require(coda)
## Loading required package: coda
#schools data in the R2OpenBUGS library

#define the model
bioassaymodel <- function(){
  for (i in 1:n) {
    logit(theta[i]) <- beta0 + beta1*xi[i]
    yi[i] ~ dbin(theta[i],ni[i])
  }
  beta0 ~ dnorm(0,0.001)
  beta1 ~ dnorm(0,0.001)
  LD50 <- (logit(0.50)-beta0)/beta1
}
# write the model code out to a file
write.model(bioassaymodel, "bioassaymodel.txt")
model.file1 = paste(getwd(),"bioassaymodel.txt", sep="/")
## and let's take a look:
file.show("bioassaymodel.txt")

#prepare the data for input into OpenBUGS
xi <- c(-0.863,-0.296,-0.053,0.727)
ni <- c(5,5,5,5)
yi <- c(0,1,3,5)
```

```

n <- 4
data <- list ("xi", "yi", "ni","n")

#initialization of variables
inits <- function(){
  list(beta0=0,beta1=1)
}

WINE="/opt/local/bin/wine"
WINEPATH="/opt/local/bin/winepath"
OpenBUGS.pgm="/Users/benjamin/Applications/wine/drive_c/ProgramFiles/OpenBUGS/OpenBUGS323/OpenBUGS.exe"

#these are the parameters to save
parameters = c("beta0", "beta1","LD50")

#run the model
bioassay.sim <- bugs(data,
                      inits,
                      model.file = model.file1,
                      parameters=parameters,
                      n.chains = 4,
                      n.iter = 2500,
                      OpenBUGS.pgm=OpenBUGS.pgm,
                      WINE=WINE,
                      WINEPATH=WINEPATH,
                      useWINE=T,
                      codaPkg = FALSE)

#R will pause. When model is complete a prompt will reappear
attach(bioassay.sim,warn.conflicts = FALSE)

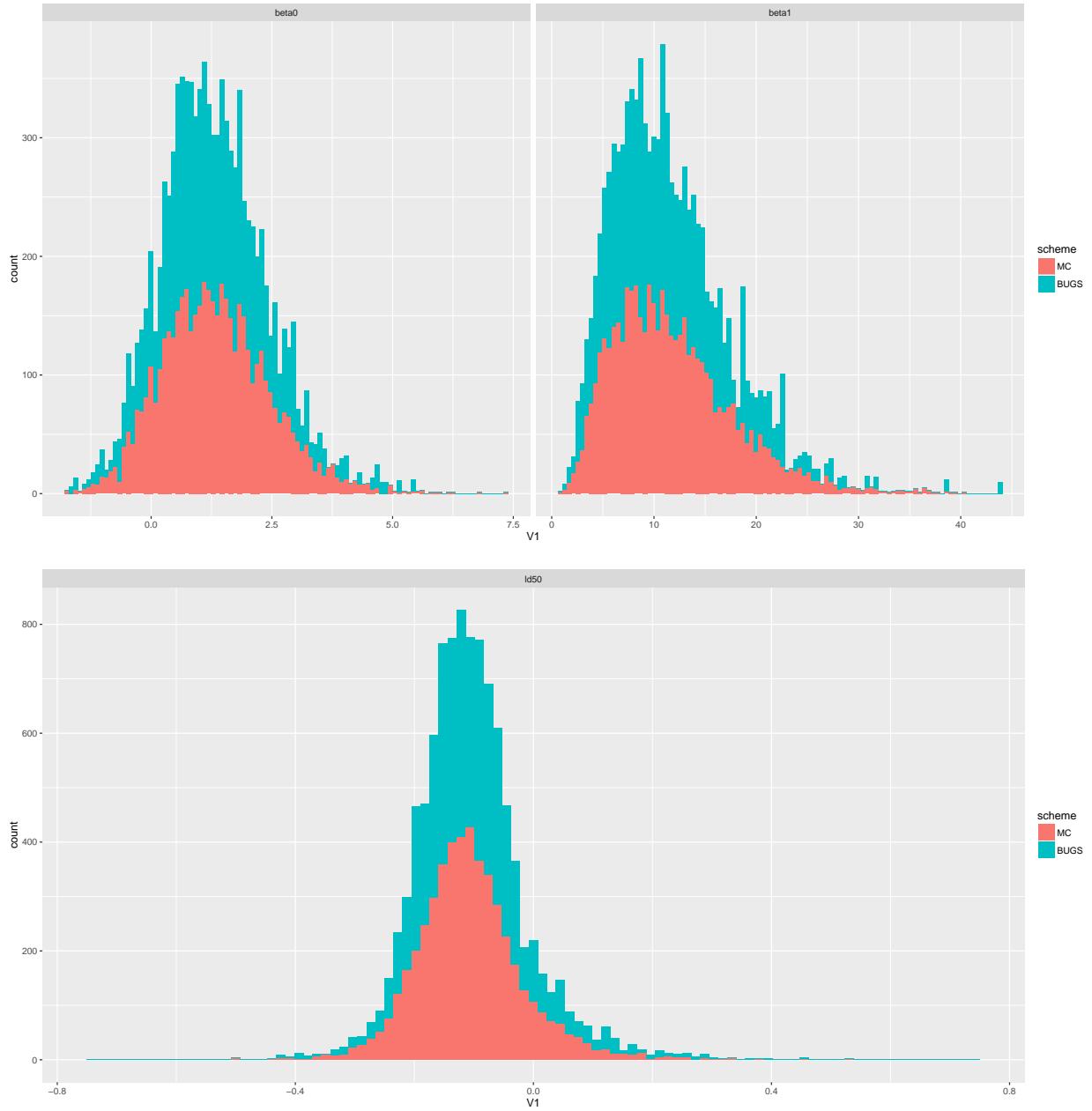
```

```

bioassay.sim$summary
##              mean        sd      2.5%     25%      50%      75%
## beta0    1.2911892 1.11169413 -0.640020  0.547900  1.1675  1.956500
## beta1    11.5425473 5.83361721  3.191925  7.328750 10.6550 14.620000
## LD50     -0.1031535 0.09640336 -0.262300 -0.159325 -0.1118 -0.060675
## deviance 6.0963606 2.17120146  4.005975  4.538000  5.4050  6.952000
##             97.5%   Rhat n.eff
## beta0    3.8870000 1.003205  1800
## beta1    24.8000000 1.002198  1700
## LD50     0.1289575 1.006328  1400
## deviance 11.8719935 1.006837  650

```

(b) Compare the results obtained by WinBUGS and the MC sampling scheme (Lab 8 solutions). Comments on the results.



Both approaches seem comparable, with the WINBUGS scheme centering more around the posterior mean values for each parameter compared the the MC method, but otherwise both methods produce similar results.

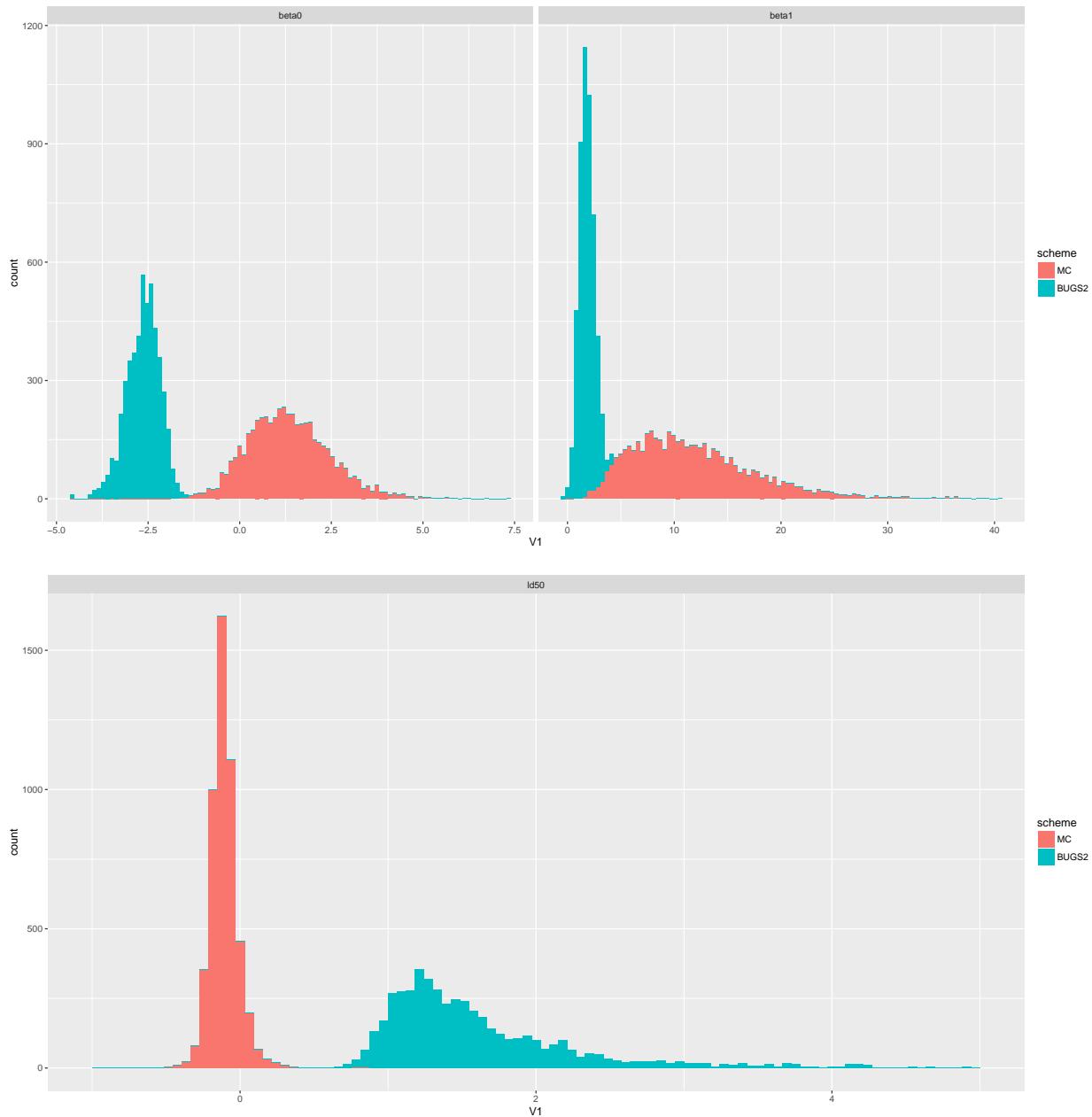
(c) Repeat part (a) after multiplying each n above by 5 (i.e. $n = 25$).

```
bioassay.sim$summary
##               mean        sd      2.5%     25%     50%     75%    97.5%
## beta0     -2.640670  0.4636925 -3.6320000 -2.9380  -2.6040  -2.316  -1.847975
## beta1      1.830160  0.7181135  0.5449425  1.3490   1.7840   2.300   3.274000
## LD50       2.097365 12.8540256  0.8786000  1.1815   1.4425   1.887   4.109000
## deviance  11.968405  1.9994917  9.9660000 10.5100  11.3700  12.810  17.031248
##                  Rhat n.eff
## beta0     1.010697    330
```

```

## beta1      1.006423   520
## LD50       1.183779   1800
## deviance  1.006671   710

```



The multiplication in this example has changed the underlying problem, and we can see the change when comparing the histograms for the MC (num) method and the BUGS method. By multiplying each n by 5 we have altered proportion of rats that die for each log-dosage, thereby reducing our estimates of toxicity.

(d) Repeat part (a) after multiplying each n AND each y above by 5.

```

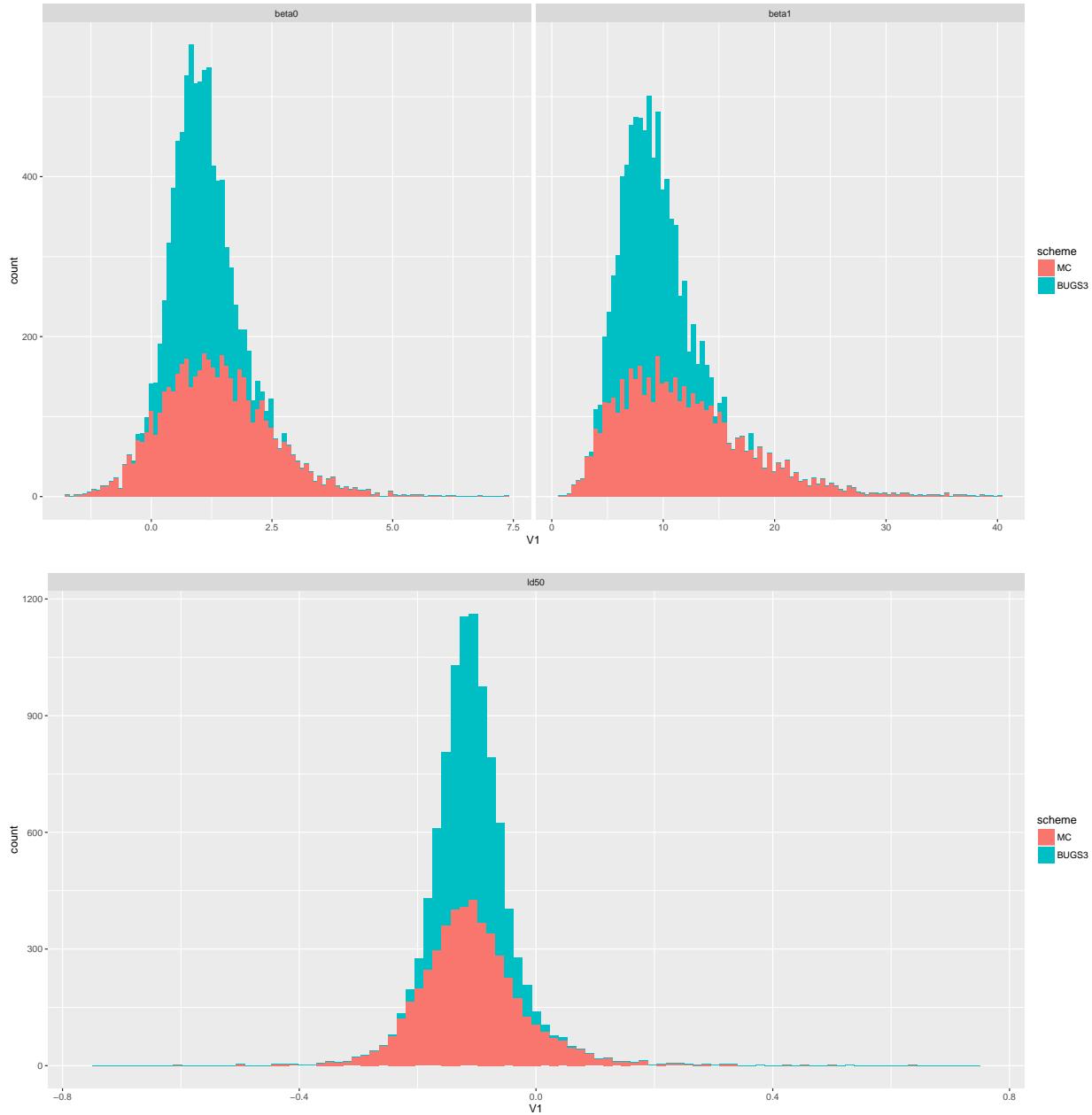
##               mean        sd     2.5%    25%    50%    75%
## beta0      1.0003774 0.49309227  0.1194950  0.652625  0.965650  1.3060000
## beta1      8.8344718 2.37124139  4.8947998  7.098500  8.633999 10.3100000

```

```

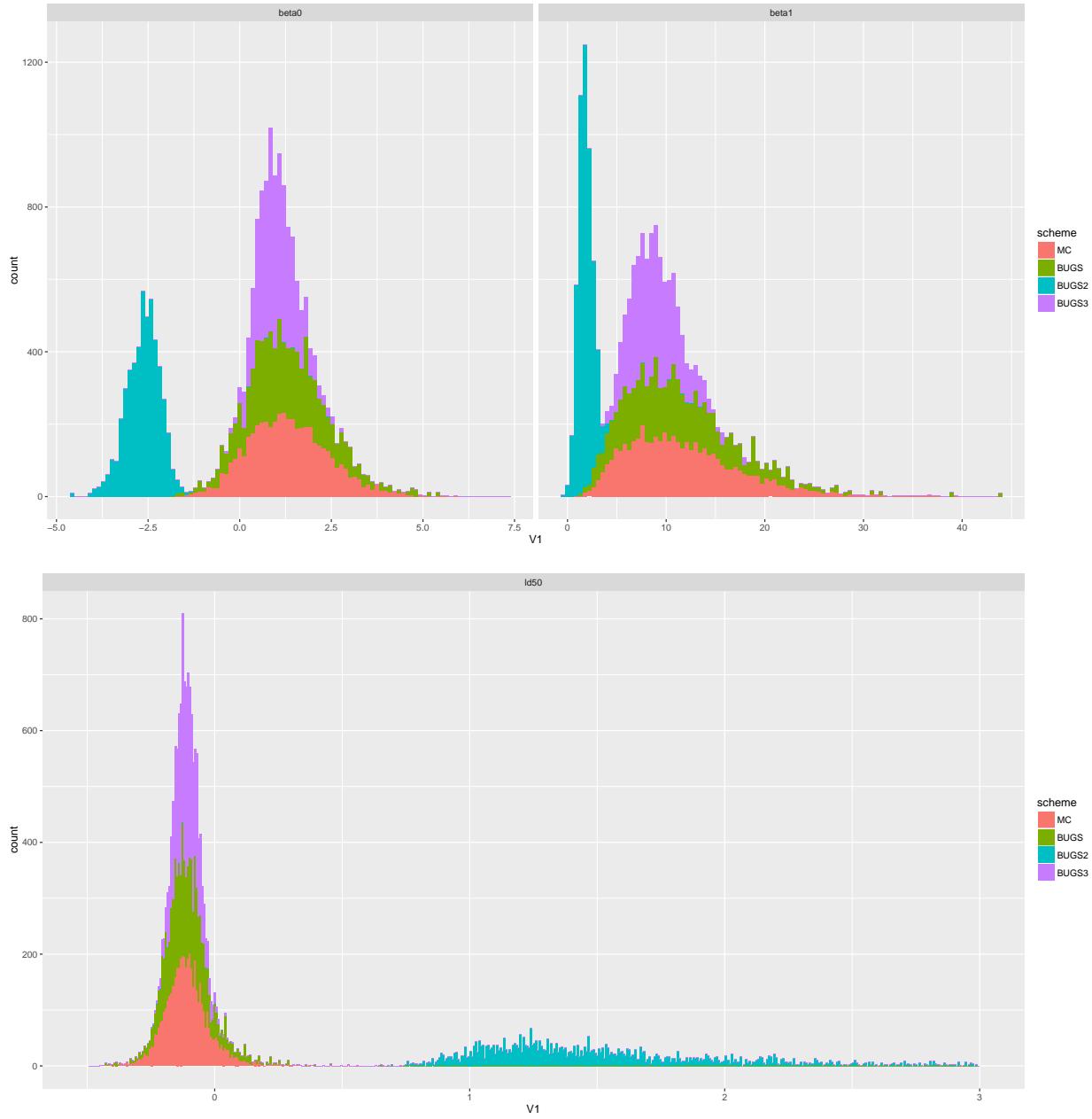
## LD50      -0.1106316 0.04279528 -0.1885125 -0.139200 -0.112800 -0.0840325
## deviance 9.3412886 2.27267273 7.1950000 7.764750 8.603000 10.1900000
##          97.5%     Rhat n.eff
## beta0     2.0900000 1.003882   830
## beta1     13.9700000 1.006861   400
## LD50     -0.0175865 1.001403  3100
## deviance 16.1200000 1.002372  1400

```



We again have a much better estimate compared to part (c) because. by multiplying both the n and y value for each dosage level, we have not fundamentally changed the underlying toxicity estimate from the problem as it was originally stated.

(e) Compare inferences for α , β and $LD50$: how have they changed from (a) to (c) and (d)? Do these changes make logical sense?



Yes, for the reasons stated above. The estimate from part (c) should look much different from the others because the data used to estimate it is skewed compared to the others: more trials for the same number of deaths. Additionally, we can see that the best estimate is the one obtained in part (d), which makes sense because it has the largest sample and therefore the least variability.

Question 2

Taken from Albert (2007). Normal linear regression.

Dobson (2001) describes a birthweight regression study. One is interested in predicting a baby's birthweight (in grams) based on the gestational age (in weeks) and the gender of the baby. The data is presented in the

file *birthweight.txt*. In the standard linear regression model, we assume that

$$\text{Birthweight}(i) = \beta_0 + \beta_1 \text{AGE}(i) + \beta_2 \text{GENDER}(i) + e(i)$$

where the $e(i)$'s are independent and Normally distributed random variables with mean $\mathbf{0}$ and variance σ^2 .

- (a) Use the R function lm to fit this model by Least-Squares. From the output, assess if the effects AGE and GENDER are significant, and if they are significant, describe the effects of each covariate on birthweight.

```
fit <- lm(weight ~ age + gender, data = birthweight, model = TRUE, x=TRUE,y=TRUE)
summary(fit)
##
## Call:
## lm(formula = weight ~ age + gender, data = birthweight, model = TRUE,
##     x = TRUE, y = TRUE)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -258.65 -155.48  -36.17  164.00  463.62
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -924.94     896.09  -1.032  0.31373
## age          103.02     23.32    4.417  0.00024 ***
## genderFemale -121.25     84.97  -1.427  0.16827
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 208.1 on 21 degrees of freedom
## Multiple R-squared:  0.5032, Adjusted R-squared:  0.4558
## F-statistic: 10.63 on 2 and 21 DF,  p-value: 0.0006461
```

From the output of the linear model we can see that age is the only covariate that significantly effects birthweight ($\alpha = 0.05$). It has a positive effect on birthweight - i.e. as age increases, so does birthweight. Gender - which is not found to have a significant impact on birthweight - negatively effects birthweight if the child is female; it effects bw positively if the child is male. The value of the intercept is given as ≈ -924.94 .

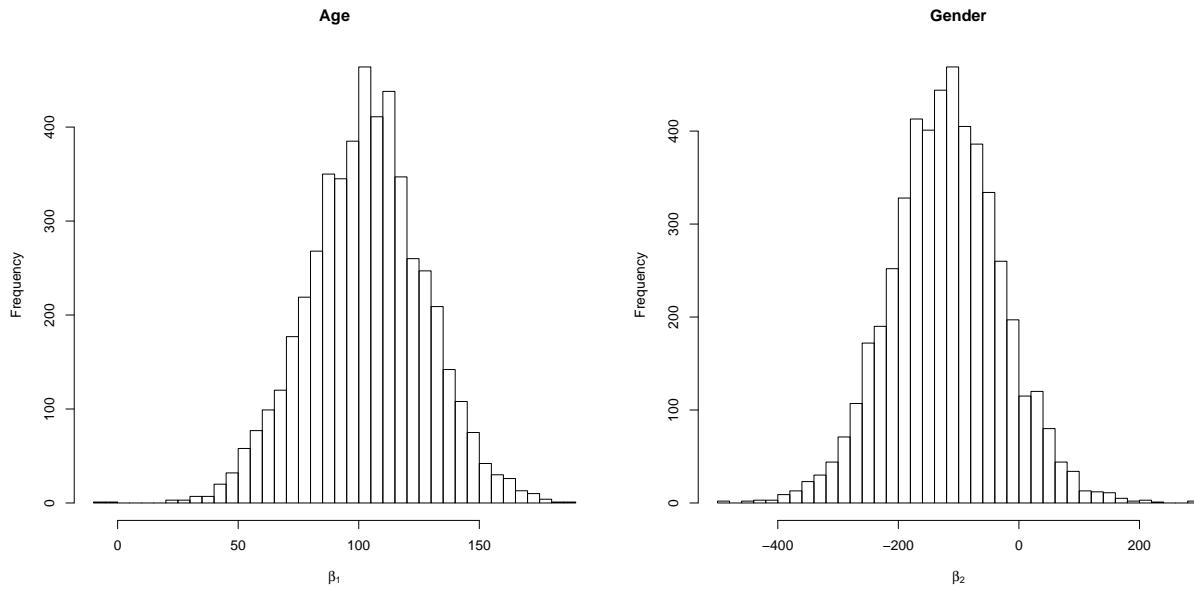
- (b) Suppose a uniform prior is placed on the regression parameters $\beta = (\beta_0, \beta_1, \beta_2)$. Use the LearnBayes package in R (blinreg command) to simulate an MC sample of 5000 draws from the joint posterior distribution of (β, σ^2) . From the simulated sample, compute posterior means and standard deviations of β_1 and β_2 . Compare the results with the LS estimates obtained in (a).

```
theta.sample <- blinreg(fit$y,fit$x,5000)
sumresids <- sum(fit$residual^2)
shape <- fit$df.residual/2
rate <- sumresids/2
sigma2 <- rigamma(1,shape,rate)
MSE <- sum(fit$residuals^2)/fit$df.residual
vbeta <- vcov(fit)/MSE
beta <- rmnorm(1,mean=fit$coef,varcov=vbeta*sigma2)
```

```

par(mfrow=c(1,2))
hist(theta.sample$beta[,2],main="Age",xlab=expression(beta[1]), 30)
hist(theta.sample$beta[,3],main="Gender", xlab=expression(beta[2])),30)

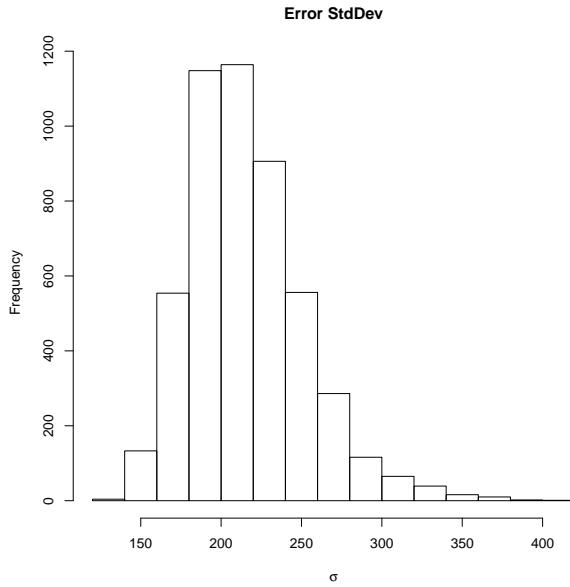
```



```

hist(theta.sample$sigma,main="Error StdDev", xlab=expression(sigma))
# CIs for the Beta coefficients
apply(theta.sample$beta,2,quantile,c(.05,.5,.95))
##      X(Intercept)      Xage XgenderFemale
## 5%   -2443.7899   62.32447   -269.51991
## 50%   -946.9443  103.39443   -120.19082
## 95%    628.0205  142.62286    30.83404
# CIs for sigma
quantile(theta.sample$sigma,c(.05,.5,.95))
##      5%      50%      95%
## 167.1212 211.4745 279.8939

```



Here we are given values for the Intercept, Age coefficient and Gender coefficient of approximately -907.2544 , 102.49836 and -119.5783 respectively, which are pretty close to the values obtained using the method in part (a).

(c) Repeat part (b) using WinBUGS. Compare the results with the LS estimates obtained in (a) and MC sampling in part (b).

```

require(R2OpenBUGS)
require(coda)

#define the model
birthweightmodel <- function(){
  for (i in 1:n) {
    mu[i] <- beta0 + beta1*x1[i] + beta2*x2[i]
    yi[i] ~ dnorm(mu[i],tau)
  }
  beta0 ~ dflat()
  sigma ~ dunif(0.01, 100)
  tau <- 1/(sigma*sigma)
  beta1 ~ dflat()
  beta2 ~ dflat()
}
# write the model code out to a file
write.model(birthweightmodel, "birthweightmodel.txt")
model.file1 = paste(getwd(),"birthweightmodel.txt", sep="/")

#prepare the data for input into OpenBUGS
x1 <- fit$model[,2]
x2 <- as.integer(unclass(fit$model[,3]))
yi <- fit$model[,1]
n <- 24
data <- list("x1","x2","yi","n")

```

```

#initialization of variables
inits <- function(){
  list(beta0=0, beta1=1,beta2=1,sigma=1)
}

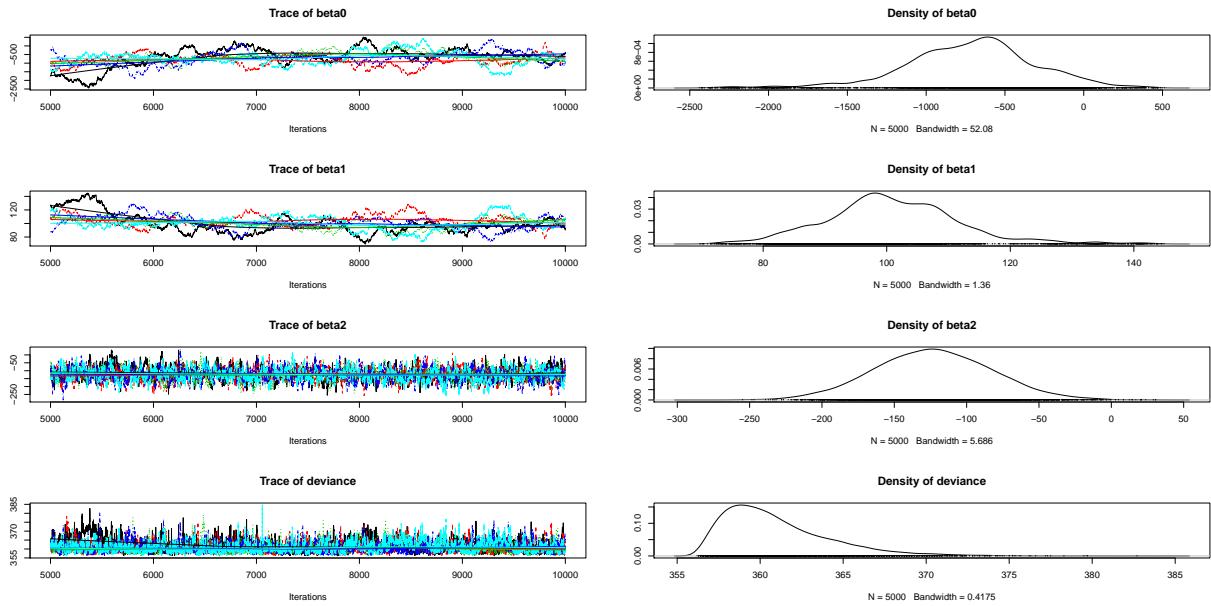
WINE="/opt/local/bin/wine"
WINEPATH="/opt/local/bin/winepath"
OpenBUGS.pgm="/Users/benjamin/Applications/wine/drive_c/ProgramFiles/OpenBUGS/OpenBUGS323/OpenBUGS.exe"

#these are the parameters to save
parameters = c("beta0", "beta1","beta2")

#run the model
birthweight.sim <- bugs(data,
                         inits,
                         model.file = model.file1,
                         parameters=parameters,
                         n.chains = 5,
                         n.iter = 10000,
                         OpenBUGS.pgm=OpenBUGS.pgm,
                         WINE=WINE,
                         WINEPATH=WINEPATH,
                         useWINE=T,
                         codaPkg = T,
                         working.directory = getwd(),
                         debug = F)

samples <- read.openbugs()
## Abstracting beta0 ... 5000 valid values
## Abstracting beta1 ... 5000 valid values
## Abstracting beta2 ... 5000 valid values
## Abstracting deviance ... 5000 valid values
## Abstracting beta0 ... 5000 valid values
## Abstracting beta1 ... 5000 valid values
## Abstracting beta2 ... 5000 valid values
## Abstracting deviance ... 5000 valid values
## Abstracting beta0 ... 5000 valid values
## Abstracting beta1 ... 5000 valid values
## Abstracting beta2 ... 5000 valid values
## Abstracting deviance ... 5000 valid values
## Abstracting beta0 ... 5000 valid values
## Abstracting beta1 ... 5000 valid values
## Abstracting beta2 ... 5000 valid values
## Abstracting deviance ... 5000 valid values
## Abstracting beta0 ... 5000 valid values
## Abstracting beta1 ... 5000 valid values
## Abstracting beta2 ... 5000 valid values
## Abstracting deviance ... 5000 valid values
plot(samples)

```



```
summary(samples)
##
## Iterations = 5001:10000
## Thinning interval = 1
## Number of chains = 5
## Sample size per chain = 5000
##
## 1. Empirical mean and standard deviation for each variable,
##    plus standard error of the mean:
##
##           Mean        SD Naive SE Time-series SE
## beta0     -718.8  418.164   2.64470      75.9269
## beta1      100.9   10.816   0.06841      1.9390
## beta2     -123.1   40.655   0.25712      1.1250
## deviance  361.0    3.191   0.02018      0.1361
##
## 2. Quantiles for each variable:
##
##          2.5%       25%       50%       75%   97.5%
## beta0    -1660.00 -966.12  -692.70 -467.17  61.33
## beta1      80.98   94.27  100.30 107.30 125.40
## beta2     -202.60 -150.80 -123.50 -95.72 -42.15
## deviance  356.80  358.60  360.30 362.60 369.00
```

WINBUGS reports values for the Intercept, Age coefficient and Gender coefficient of approximately -718.8 , 100.9 and -123.1 respectively. The value given to the intercept is much lower than the values reported in parts (a) and (b), however the other coefficients are roughly the same as those calculated using the other methods.

Question 3

Variance components model, taken from Box & Tiao (1973).

Read the complete WinBUGS example “Dyes: variance components model” (Examples Vol. 1). To show the

advantages of the Bayesian approach, Box & Tiao also simulated data from a model with a known between batch variance of 4 and a within batch variance of 16, presented in *dyestuff.txt*.

(a) Use WinBUGS to plot estimated posterior densities of the two variance components, including the joint posterior (by scatterplot), and give appropriate 95% intervals.

```

require(R2OpenBUGS)
require(coda)
## Loading required package: coda

#define the model
dyesmodel <- function(){
  for(i in 1 : batches) {
    m[i] ~ dnorm(theta, tau.btw)
    for(j in 1 : samples) {
      y[j , i] ~ dnorm(m[i], tau.with)
    }
  }
  sigma2.with <- 1 / tau.with
  sigma2.btw <- 1 / tau.btw
  tau.with ~ dgamma(0.001, 0.001)
  tau.btw ~ dgamma(0.001, 0.001)
  theta ~ dflat()
}
# write the model code out to a file
write.model(dyesmodel, "dyesmodel.txt")
model.file1 = paste(getwd(),"dyesmodel.txt", sep="/")
coda.file = paste(getwd(),"Q3Coda", sep="/")
## and let's take a look:
# file.show("dyesmodel.txt")

#prepare the data for input into OpenBUGS
batches <- 6
samples <- 5
y <- as.matrix(dyedata)
data <- list ("batches", "samples", "y")

#initialization of variables
inits <- function(){
  list(theta=0, tau.with=1, tau.btw=1)
}

WINE="/opt/local/bin/wine"
WINEPATH="/opt/local/bin/winepath"
OpenBUGS.pgm="/Users/benjamin/Applications/wine/drive_c/ProgramFiles/OpenBUGS/OpenBUGS323/OpenBUGS.exe"

#these are the parameters to save
parameters = c("sigma2.with", "sigma2.btw", "theta")

#run the model
dyes.sim <- bugs(data,
                  inits,
                  model.file = model.file1,
                  parameters=parameters,

```

```

n.burnin = 25000,
n.chains = 3,
n.iter = 125000,
OpenBUGS.pgm=OpenBUGS.pgm,
WINE=WINE,
WINEPATH=WINEPATH,
useWINE=T,
codaPkg = T,
working.directory = coda.file,
debug = F)

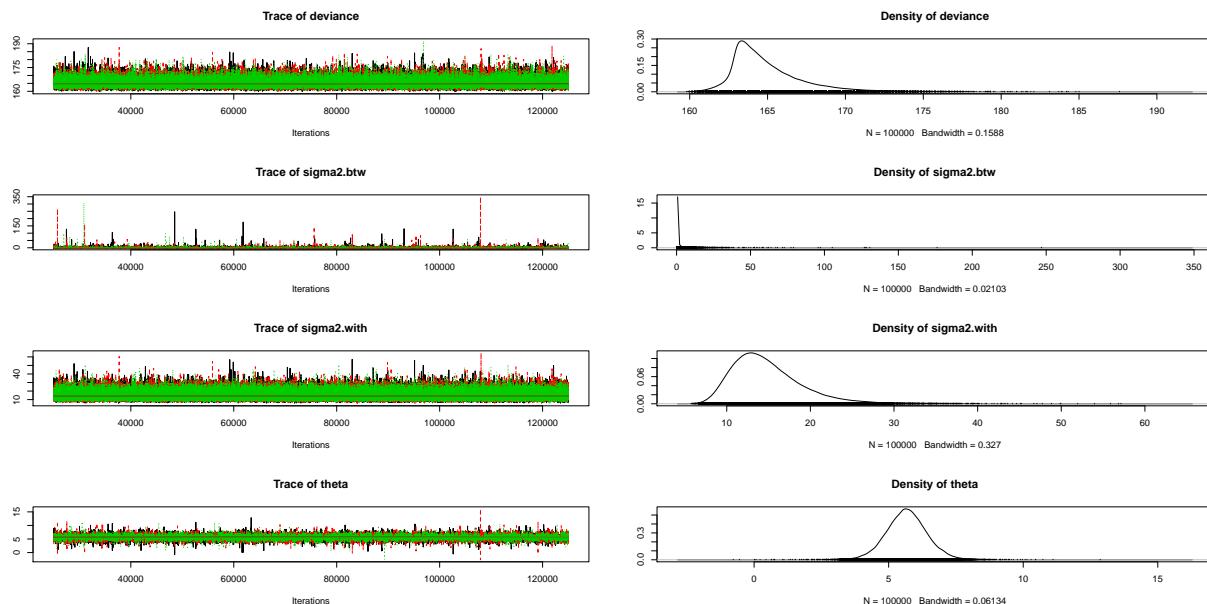
```

```

samples <- read.openbugs(paste0(coda.file,"/"))
## Abstracting deviance ... 100000 valid values
## Abstracting sigma2.btw ... 100000 valid values
## Abstracting sigma2.with ... 100000 valid values
## Abstracting theta ... 100000 valid values
## Abstracting deviance ... 100000 valid values
## Abstracting sigma2.btw ... 100000 valid values
## Abstracting sigma2.with ... 100000 valid values
## Abstracting theta ... 100000 valid values
## Abstracting deviance ... 100000 valid values
## Abstracting sigma2.btw ... 100000 valid values
## Abstracting sigma2.with ... 100000 valid values
## Abstracting theta ... 100000 valid values
samples2 <- as.matrix(samples[1])

```

```
plot(samples)
```



```

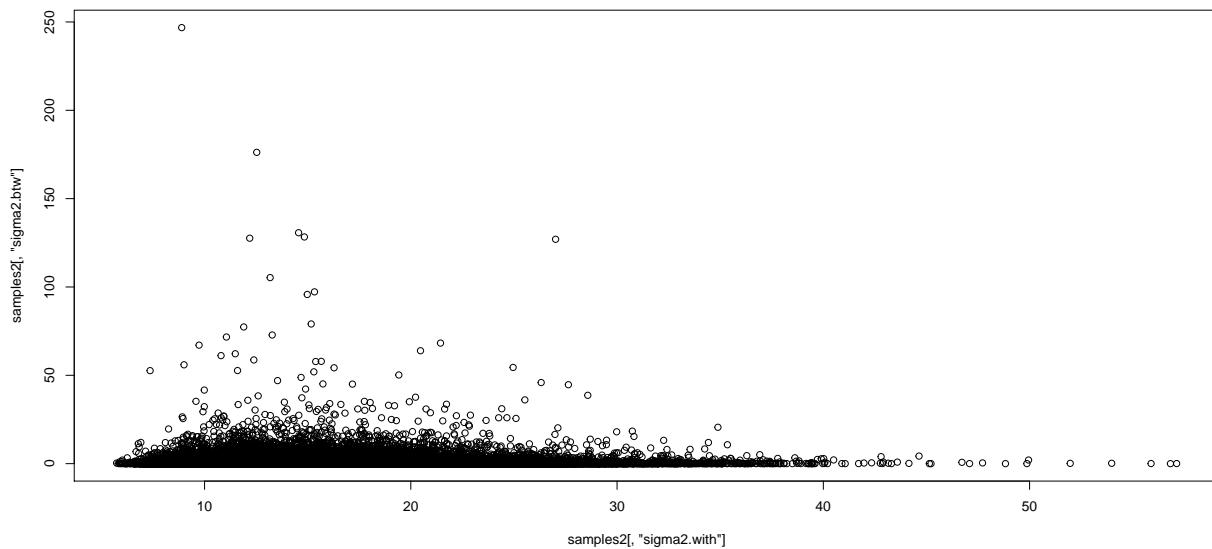
summary(samples)
##
## Iterations = 25001:125000
## Thinning interval = 1
## Number of chains = 3
## Sample size per chain = 1e+05

```

```

## 
## 1. Empirical mean and standard deviation for each variable,
##     plus standard error of the mean:
## 
##           Mean      SD Naive SE Time-series SE
## deviance    164.9326 2.2897 0.004180      0.017834
## sigma2.btw   0.5398 2.2694 0.004143      0.009786
## sigma2.with  14.8296 4.2291 0.007721      0.011738
## theta        5.6640 0.7609 0.001389      0.009316
## 
## 2. Quantiles for each variable:
## 
##           2.5%      25%      50%      75%     97.5%
## deviance    1.619e+02 1.634e+02 164.40000 165.900 170.900
## sigma2.btw  8.594e-04 8.788e-03  0.05492   0.340   4.143
## sigma2.with 8.714e+00 1.186e+01 14.12000 17.010 25.070
## theta        4.167e+00 5.178e+00  5.66200  6.144   7.173
plot(samples2[, "sigma2.with"], samples2[, "sigma2.btw"])

```



Question 4

The observed breaking strengths (in grams) of 20 pieces of yarn randomly sampled from spinning machines in a certain production area are as follows:

46 58 40 47 47 53 43 48 50 55 49 50 52 56 49 54 51 50 52 50

Assume that breaking strengths are known to follow a Normal distribution, and that the mean breaking strength of yarn produced by machines in this area is known to be 51 grams. We are trying to estimate the variance, which we will now call θ .

- (a) What is the conjugate family of prior distributions for a Normal variance (not precision) when the mean is known?

For a Gaussian distribution with a known mean but unknown variance we use an Inverse Gamma. So, in full we combine $IG(\alpha, \beta)$

$$p(\theta) \propto (\theta)^{-(\alpha+1)} e^{-\beta/\theta},$$

with the likelihood of the Gaussian $\sim N(\bar{y} = 51, \theta)$

$$p(y | \theta) \propto (\theta)^{-n/2} e^{\sum_{i=1}^n (y_i - \bar{y})^2 / 2\theta}$$

to get a posterior Inverse Gamma $\sim IG(n + 2\alpha/2, \beta + \sum_{i=1}^n (y_i - \bar{y})^2 / 2)$

$$p(\bar{y}, \theta | y) \propto (\theta)^{-\left(\frac{n+2\alpha}{2} + 1\right)} e^{-\left(\frac{\beta}{\theta} + \frac{\sum_{i=1}^n (y_i - \bar{y})^2}{2\theta}\right)}$$

(b) Suppose previous experience suggests that the expected value of θ is 12 and the variance of θ is 4. What parameter values are needed for the prior distribution to match these moments?

Because θ follows an Inverse Gamma such that $p(\theta) \sim IG(\alpha, \beta)$, we have

$$E[\theta] = \frac{\beta}{\alpha - 1}, \quad \alpha > 1 \\ E[\theta^2] = \frac{\beta^2}{(\alpha - 1)^2(\alpha - 2)}, \quad \alpha > 2$$

Assuming the derived values $E[\theta] = 12$ and $E[\theta^2] = 4$, we solve simultaneously to get

$$\begin{aligned} 12 &= \frac{\beta}{\alpha - 1} \\ \implies \beta &= 12(\alpha - 1) \\ \implies 4 &= \frac{12^2(\alpha - 1)^2}{(\alpha - 1)^2(\alpha - 2)} \\ 144 &= 4(\alpha - 2) \\ 36 &= \alpha - 2 \\ \implies \alpha &= 38 \\ \implies \beta &= 444 \end{aligned}$$

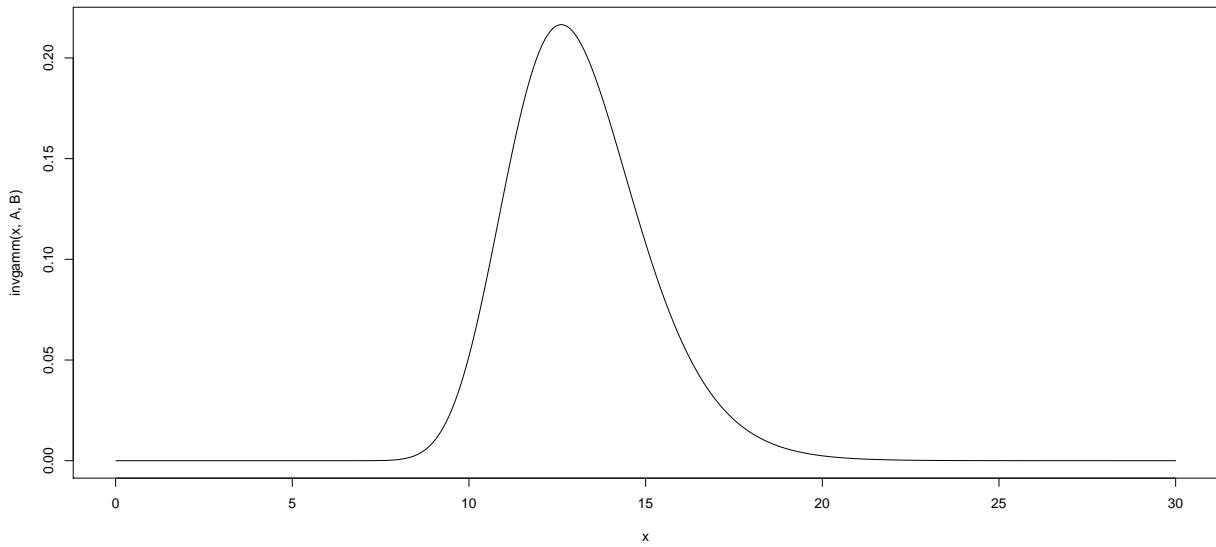
(c) What is the posterior distribution $p(\theta | y)$ for this data under the prior from the previous step?

If we sub $\alpha = 38$ and $\beta = 444$ into the posterior derived in part (a), we get

$$p(\theta | y) \propto (\theta)^{-\left(\frac{444}{\theta} + \frac{\sum_{i=1}^n (y_i - \bar{y})^2}{2\theta}\right)}$$

which is $\sim IG\left(\frac{n+76}{2}, 444 + \frac{\sum_{i=1}^n (y_i - \bar{y})^2}{2}\right)$

```
inv gamm<-function(x,a,b){  
lg = a*log(b)-lgamma(a)-(a+1)*log(x)-b/x  
return(exp(lg))}  
n <- length(strdata)  
A=(n + 76)/2  
B= 444 + ((n-1)*var(strdata))/2  
x=seq(0,30,by=0.01)  
plot(x,inv gamm(x,A,B),type='l')
```



(d) Find the posterior mean and variance of θ .

```

g <- rgamma(10000,A,B)
CI <- quantile(1/g,c(0.05,0.95))
# Confidence Intervale for theta
CI
##      5%      95%
## 10.35222 16.52717
# Posterior Mean for theta
mean(1/g)
## [1] 13.12164
# Posterior Variance for theta
var(1/g)
## [1] 3.630992

```

(e) Now use WinBUGS to carry out the analysis. You will have to specify the model in terms of the precision.

I've made a mistake somewhere here. I'm not sure where. The

```

require(R2openBUGS)
require(coda)

strdata <- c(46,58,40,47,47,53,43,48,50,55,49,50,52,56,49,54,51,50,52,50)
#define the model
strmodel <- function() {
  for (i in 1:n) {
    x[i] ~ dnorm(mu,tau)
  }

  sigma ~ dunif(0.1,100)
  tau <- 1/(sigma*sigma)
}

```

```

# write the model code out to a file
write.model(strmodel, "strmodel.txt")
model.file1 = paste(getwd(),"strmodel.txt", sep="/")
coda.file = paste(getwd(),"Q4Coda", sep="/")
file.show("strmodel.txt")

#prepare the data for input into OpenBUGS
x <- strdata
mu <- 51
n <- length(strdata)
data <- list("x","mu","n")

#initialization of variables
inits <- function(){
  list(sigma=2)
}

WINE="/opt/local/bin/wine"
WINEPATH="/opt/local/bin/winepath"
OpenBUGS.pgm="/Users/benjamin/Applications/wine/drive_c/ProgramFiles/OpenBUGS/OpenBUGS323/OpenBUGS.exe"

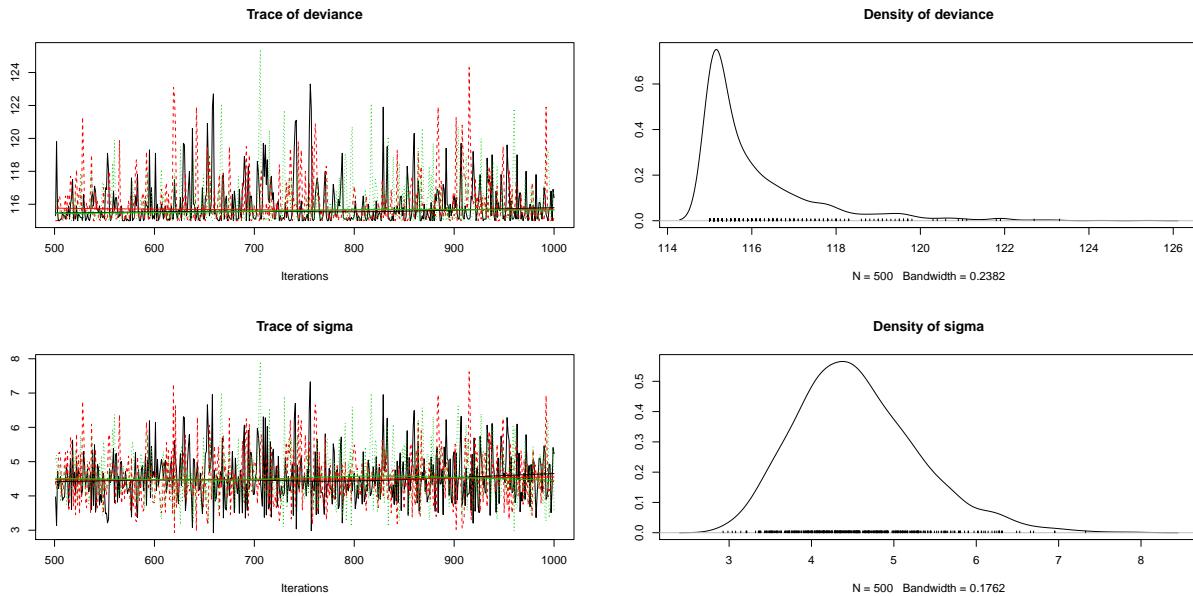
#these are the parameters to save
parameters = c("sigma")

#run the model
str.sim <- bugs(data,
                  inits,
                  model.file = model.file1,
                  parameters=parameters,
                  n.chains = 3,
                  n.iter = 1000,
                  OpenBUGS.pgm=OpenBUGS.pgm,
                  WINE=WINE,
                  WINEPATH=WINEPATH,
                  useWINE=T,
                  codaPkg = T,
                  working.directory = coda.file,
                  debug = F)

samples <- read.openbugs(paste0(coda.file,"/"))
## Abstracting deviance ... 500 valid values
## Abstracting sigma ... 500 valid values
## Abstracting deviance ... 500 valid values
## Abstracting sigma ... 500 valid values
## Abstracting deviance ... 500 valid values
## Abstracting sigma ... 500 valid values

plot(samples)

```



```
summary(samples)
##
## Iterations = 501:1000
## Thinning interval = 1
## Number of chains = 3
## Sample size per chain = 500
##
## 1. Empirical mean and standard deviation for each variable,
##    plus standard error of the mean:
##
##           Mean      SD Naive SE Time-series SE
## deviance 116.030 1.3945  0.03600      0.04644
## sigma     4.559  0.7471  0.01929      0.02050
##
## 2. Quantiles for each variable:
##
##        2.5%    25%    50%    75%   97.5%
## deviance 115.000 115.100 115.500 116.4 119.95
## sigma     3.356   4.039   4.475   5.0   6.28
```

End of Assignment 3