Assignment 3 - Question 1

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.

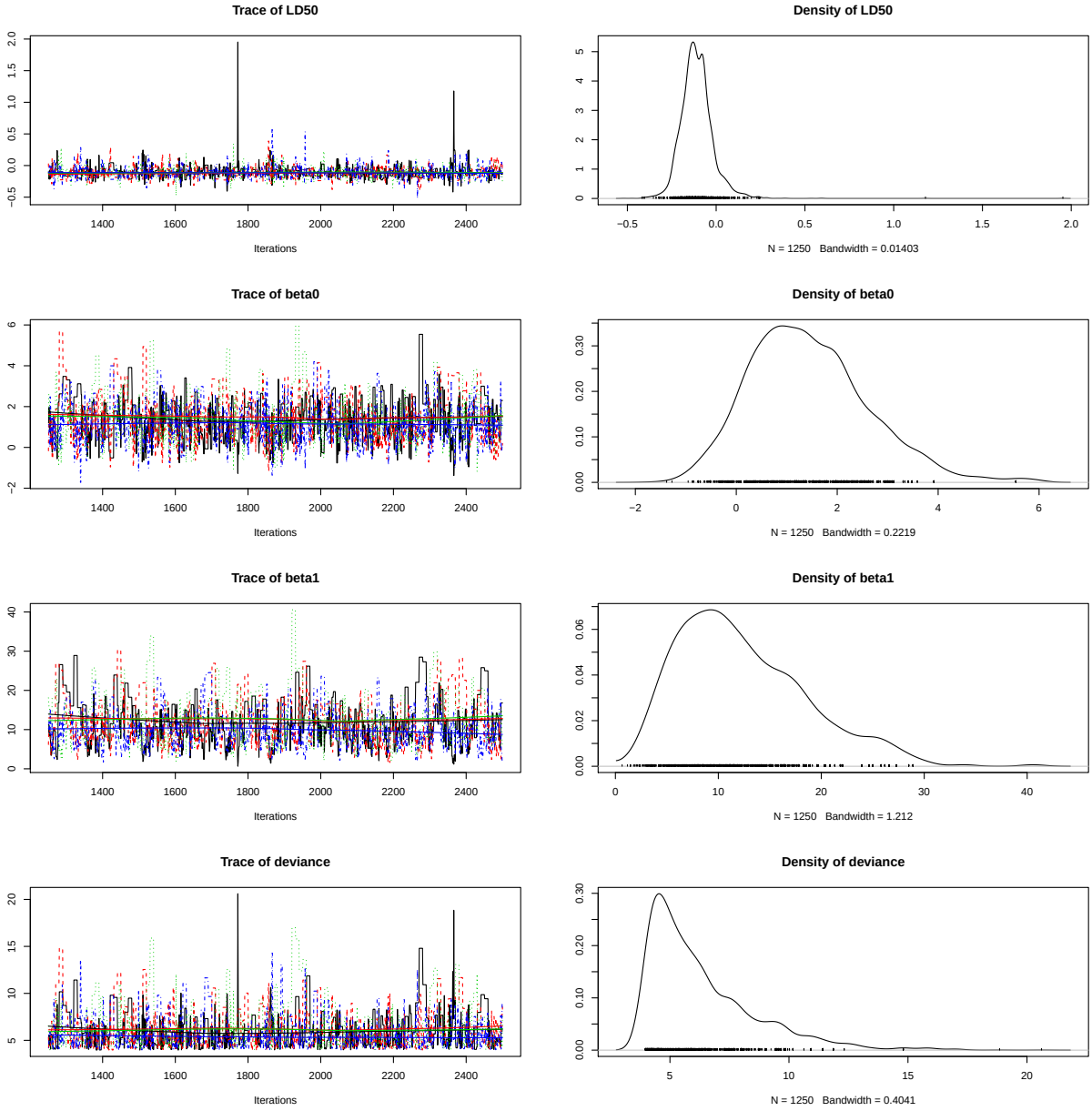
I'll be using the package *R2OpenBUGS* to send data from *R* into *WinBUGS* (or in my case *OpenBUGS*). The model files will be attached at the end of the assignment.

```
#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 ~ dflat()
  beta1 ~ dnorm(0,0.00001)
  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="/")
coda.file = paste(getwd(),"Q1Coda", sep="/")


#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=paste0("/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 = T,
                     working.directory = coda.file,
                     debug = FALSE)
```

```
samples <- read.openbugs(paste0(coda.file,"/"))
```

```
plot(samples)
```




```
summary(samples)
##
## Iterations = 1251:2500
## Thinning interval = 1
## Number of chains = 4
## Sample size per chain = 1250
##
## 1. Empirical mean and standard deviation for each variable,
##    plus standard error of the mean:
##
##             Mean      SD Naive SE Time-series SE
## LD50     -0.1085 0.09463 0.001338       0.002427
## beta0     1.4433 1.17963 0.016682       0.053000
## beta1    12.3777 6.28173 0.088837       0.369140
```

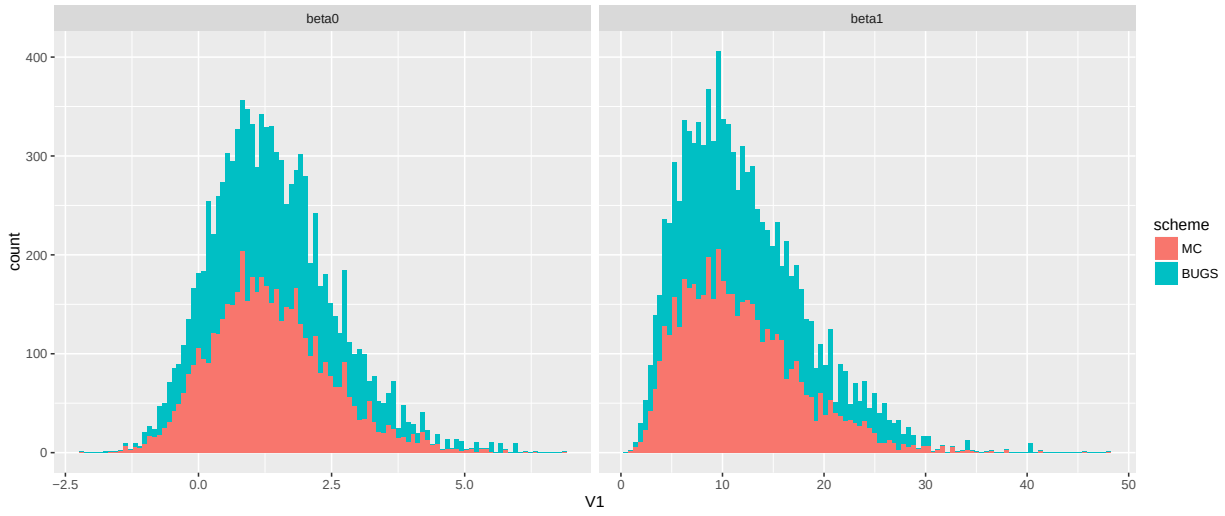
```
## deviance  6.3183 2.25823 0.031936       0.121857
##
## 2. Quantiles for each variable:
##
##              2.5%     25%     50%      75%    97.5%
## LD50     -0.2594 -0.1620 -0.1155 -0.06457  0.08667
## beta0    -0.5156  0.5931  1.3260  2.13400  3.99800
## beta1     3.3130  7.5970 11.1900 16.29000 26.87000
## deviance  4.0190  4.6190  5.6495  7.42525 12.21000
```

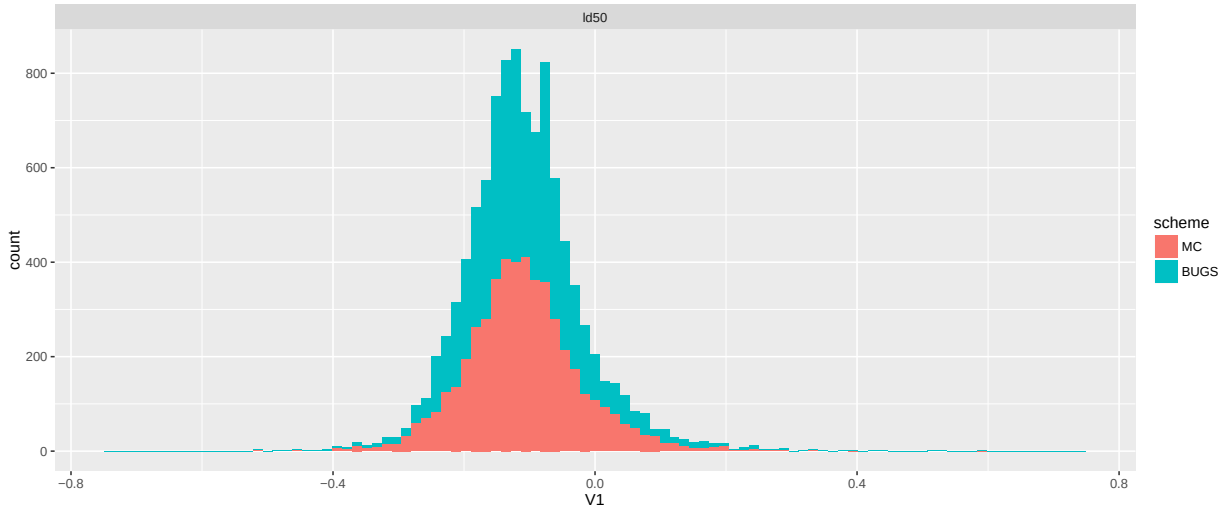
(b) Compare the results obtained by WinBUGS and the MC sampling scheme (Lab 8 solutions). Comments on the results.

I have excluded the code from lab 8 for brevity. Running the code gives the following results for the coefficients of regression beta0 and beta1.

```
require(ggplot2)
mean(beta0)
## [1] 1.356288
quantile(beta0,c(0.025,0.975))
##       2.5%      97.5%
## -0.5555556  3.9493243
mean(beta1)
## [1] 11.7862
quantile(beta1,c(0.025,0.975))
##      2.5%     97.5%
##  3.419419 25.118118

ld50=-beta0/beta1
mean(ld50)
## [1] -0.1088642
quantile(ld50,c(0.025,0.975))
##        2.5%       97.5%
## -0.27769712  0.09256529
```

By plotting histograms obtained from each sampling scheme together we can easily see how the estimates compare.



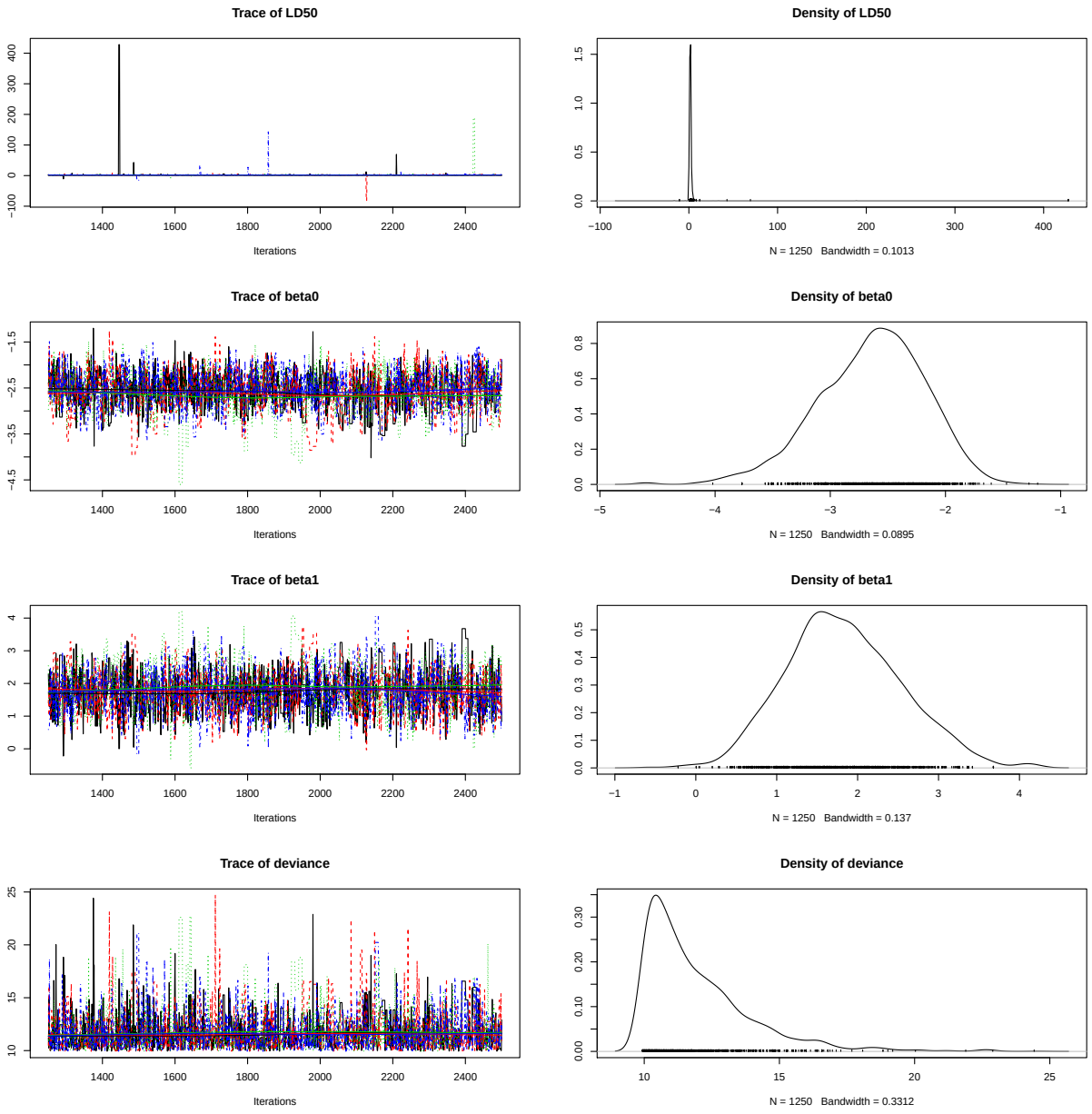


Both approaches seem comparable, with the WINBUGS scheme centering more around the posterior mean values for each parameter compared the the MC method, but otherwise both methods produce similar results.

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

Here again I have excluded the code for brevity. Running the same code given in part (a) with the adjusted values gives the following results.

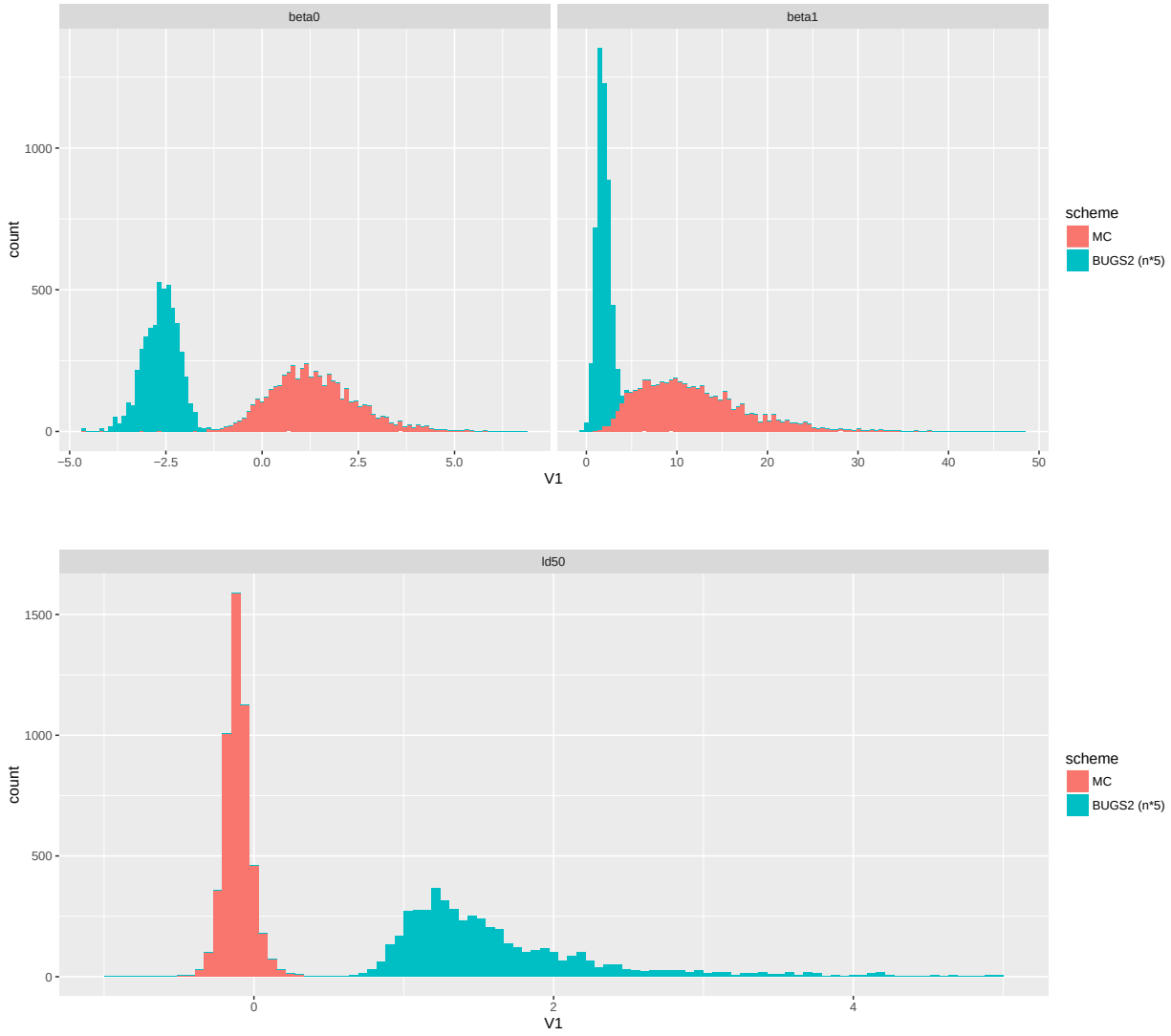
```
plot(samples)
```




```
summary(samples)
##
## Iterations = 1251:2500
## Thinning interval = 1
## Number of chains = 4
## Sample size per chain = 1250
##
## 1. Empirical mean and standard deviation for each variable,
##    plus standard error of the mean:
##
##            Mean      SD Naive SE Time-series SE
## LD50      2.083 12.1209 0.171416        0.30829
## beta0    -2.642  0.4638 0.006559        0.01689
## beta1     1.832  0.7175 0.010147        0.02385
```

```
## deviance 11.966  1.9964 0.028234        0.06220
##
## 2. Quantiles for each variable:
##
##              2.5%    25%    50%    75%  97.5%
## LD50       0.8824  1.182  1.442  1.885  4.105
## beta0     -3.6340 -2.939 -2.605 -2.317 -1.848
## beta1      0.5479  1.352  1.787  2.303  3.275
## deviance   9.9660 10.510 11.370 12.810 16.961
```

Over plotting the new results with the MC results, we get.



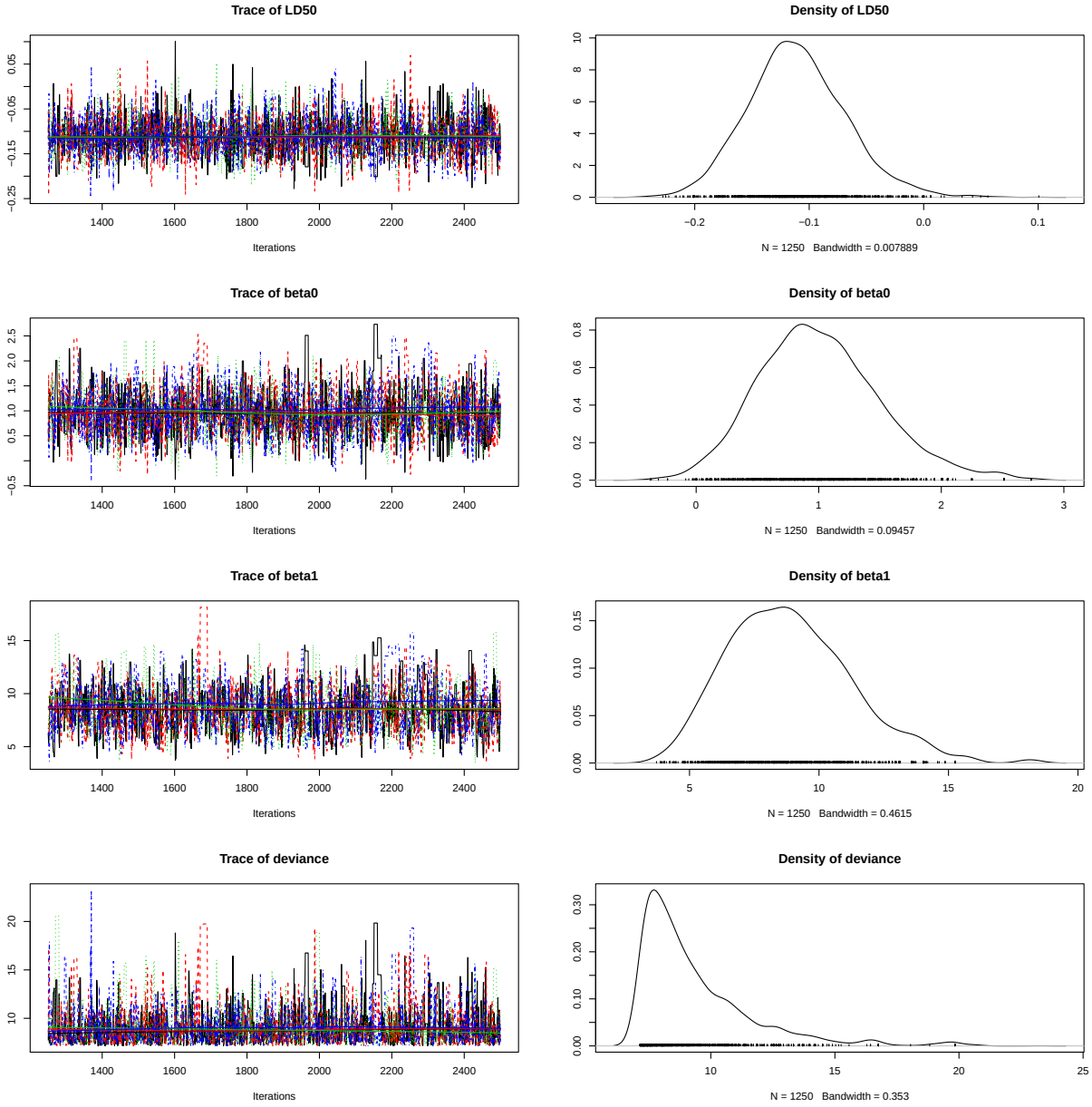
The multiplication in this example has changed the underlying problem, and we can see the change when comparing the histograms for the MC method and the BUGS method. By multiplying each n by 5 we have altered proportion of rats that die for each log-dosage, thereby reducing our estimates of toxicity.

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

Again, code excluded for brevity.

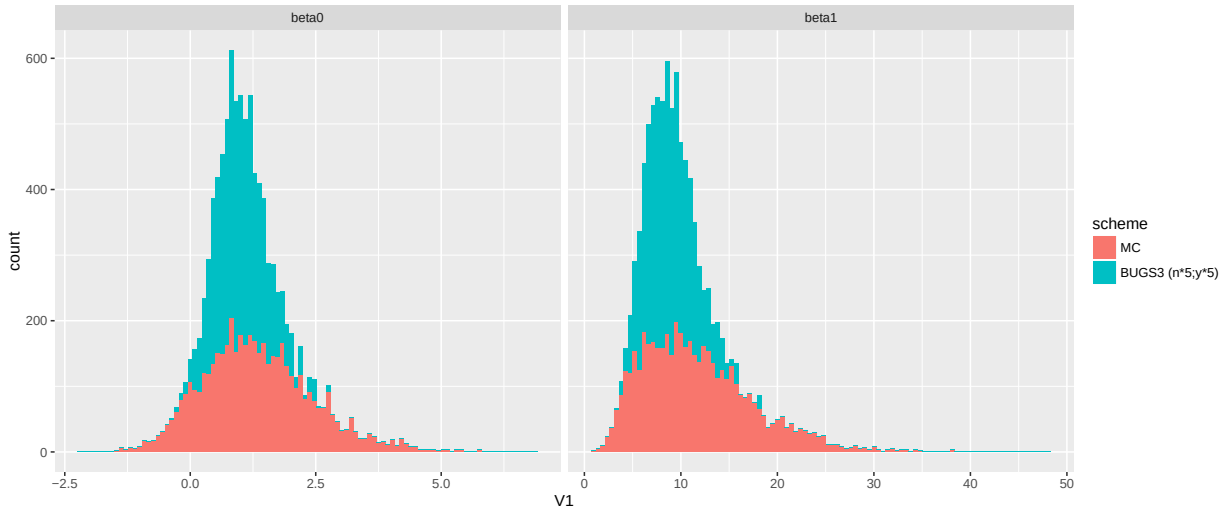
```
samples <- read.openbugs(paste0(coda.file,"/"))
```

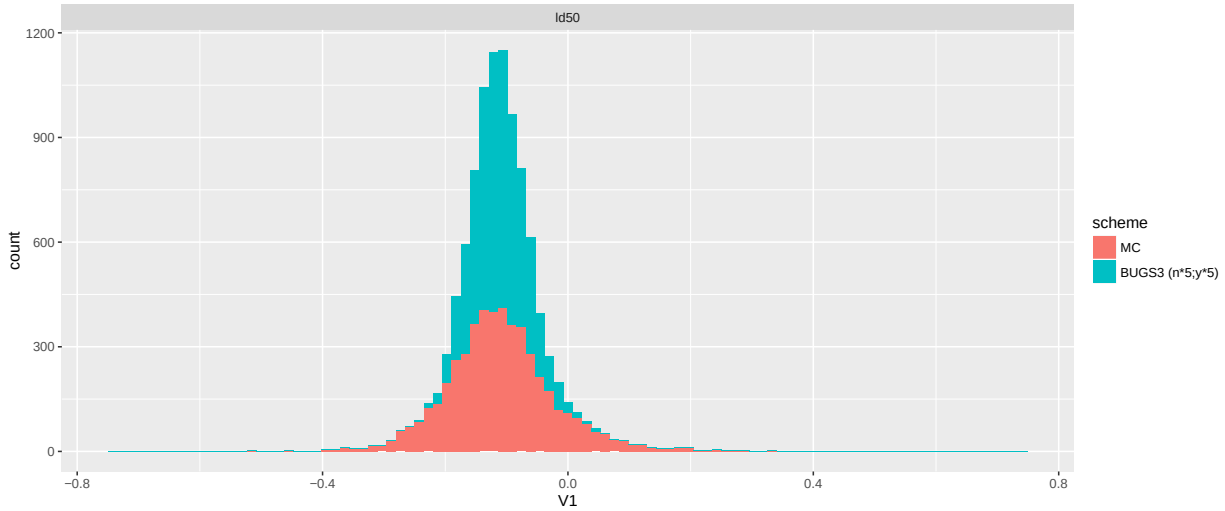
```
plot(samples)
```




```
summary(samples)
##
## Iterations = 1251:2500
## Thinning interval = 1
## Number of chains = 4
## Sample size per chain = 1250
##
## 1. Empirical mean and standard deviation for each variable,
```

```
##    plus standard error of the mean:
##
##              Mean      SD  Naive SE Time-series SE
## LD50     -0.1109 0.04265 0.0006031      0.0009799
## beta0     1.0081 0.49527 0.0070042      0.0159553
## beta1     8.8859 2.39158 0.0338220      0.0916343
## deviance  9.3666 2.29798 0.0324983      0.0935154
##
## 2. Quantiles for each variable:
##
##              2.5%     25%     50%      75%    97.5%
## LD50     -0.1902 -0.1391 -0.1131 -0.08432 -0.01862
## beta0     0.1243  0.6593  0.9772  1.31600  2.10200
## beta1     4.9097  7.1397  8.6875 10.37250 14.06000
## deviance  7.1980  7.7790  8.6240 10.23000 16.22000
```



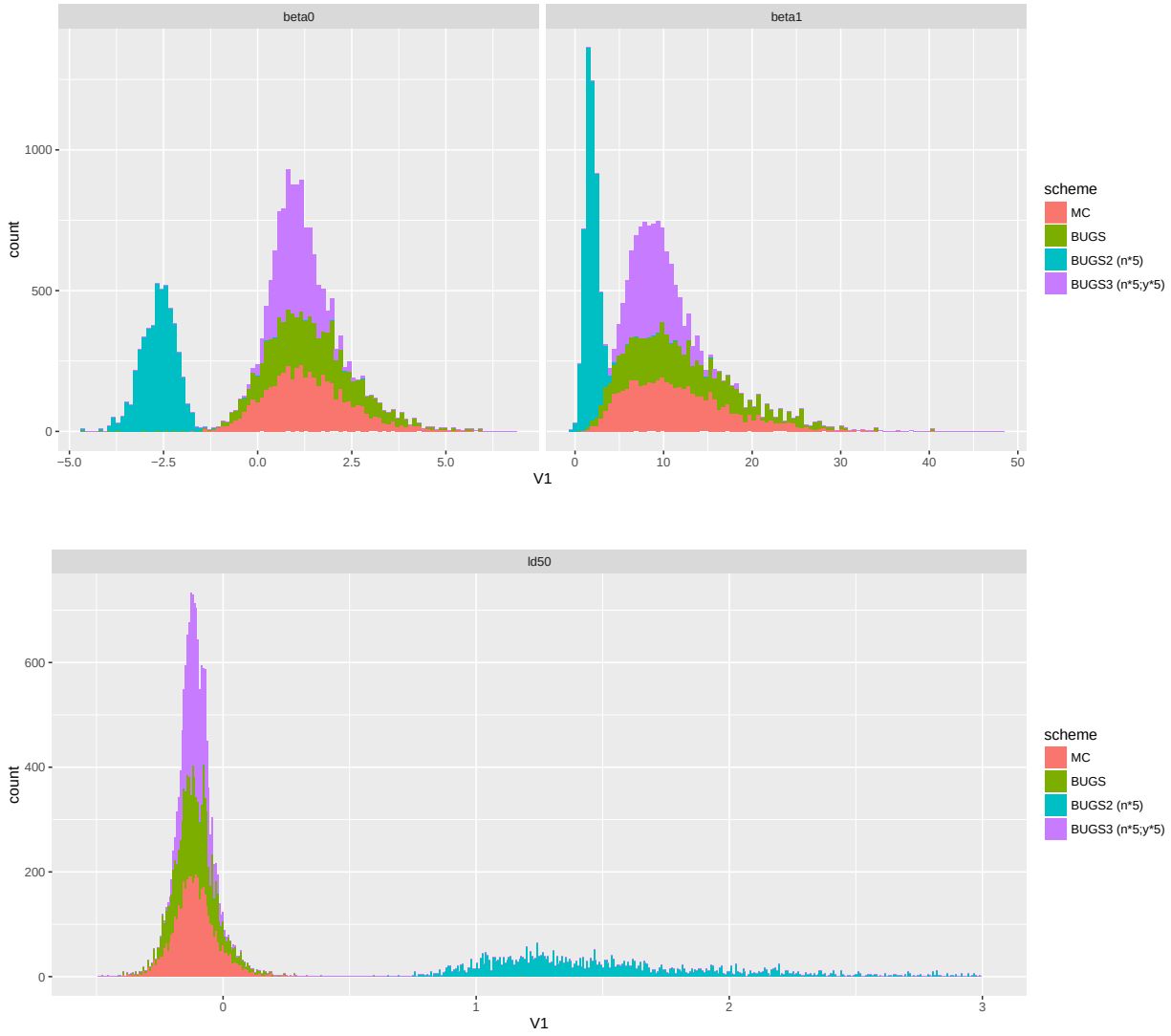


We again have a much better estimate compared to part (c) because, by multiplying both the n and y value for each dosage level, we have not fundamentally changed the underlying toxicity estimate from the problem

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?

First let's over plot the histograms of samples from each scheme for each variable.



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.