Michael Dann  
2014/10/13  
GE509

# Lab 5: OpenBUGS

Introduction & Model

The purpose of this lab serves as an introduction to the OpenBUGS software. We are utilizing the loblolly pine data (from Duke) to estimate the mean and the variance of a dataset using Bayesian MCMC methods.  
  
For our two tasks we will be estimating the mean, and mean & variance respectively. The models are specified in OpenBUGS as follows:

model{

# Task/Question 1

# Unknown Mean, Known Variance 27.373

mean ~ dnorm(20, 0.01) # A prior being assigned to "mean".

prec <- 1/27.373 # known var of 27.373

for (i in 1:297){

Y[i] ~ dnorm(mean, prec) # Data model

}

}

model{

# Task/Question 2

# Unkown Mean/ Unknown variance

mean ~ dnorm(20,0.01)

prec ~ dgamma (0.1, 0.1)

sd <- sqrt(1/prec)

for (i in 1:297){

Y[i] ~ dnorm(mean, prec)

}

}

## initial conditions, task 1 (data omitted for brevity)

list(mean=5)

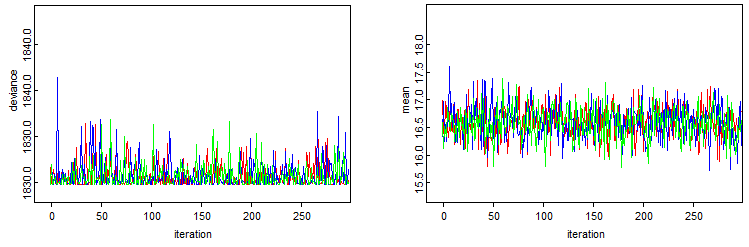
list(mean=15)

list(mean=25)  
  
## initial conditions, task 2

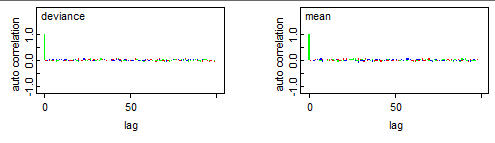
# Random

Task One:

The posterior estimate of the mean of the loblolly pine distribution may be estimated from the loblolly pine data using the above code (first model), where the variance is assumed to be 27.373. Upon loading in the data, deviance and mean chains for the first 1000 iterations are as follows:  
chains for the first 1000 iterations are as follows:



From the charts no burn in appears to be required, which agrees with our intuition regarding the prior used for a static distribution of data. As such, **beg** **of 1** is appropriate for summary statistics. Similarly there is no expectation of autocorrelation in the estimate. An autocorrelogram for our variables is as follows:



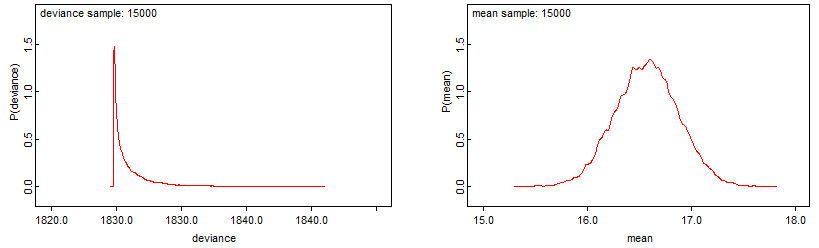
…yielding no obvious autocorrelation. Therefore a **thin of 1** is appropriate for summary statistics in sampling.  
  
At this point the sampling is iterated through 5000 times. At this point I realized that 5000 iterations on three chains is in fact 15000 samples. Mea culpa.

**mean sd MC\_err val2.5pc median val97.5pc start sample**

dev 1826.0 1.399 0.01057 1825.0 1825.0 1830.0 1 15000

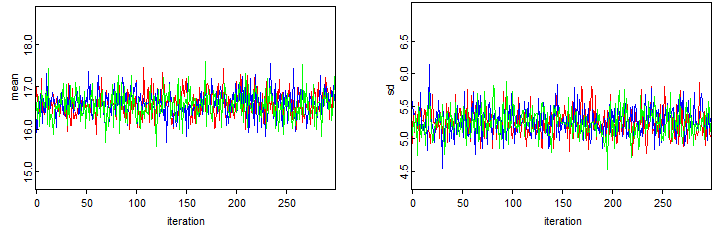
mean 16.57 0.3033 0.002415 15.98 16.58 17.17 1 15000

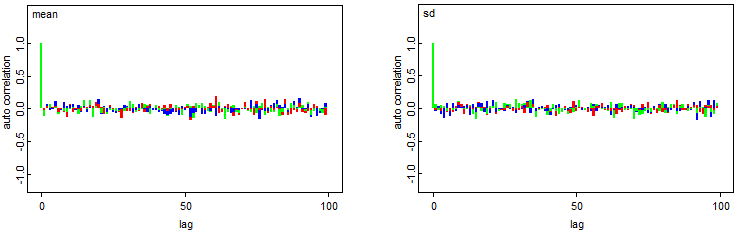
The posterior density is sketched out by the mean, median, 2.5th and 97.5th percentiles:



Task two

The same procedure is used in task two, however now the variance is estimated as well. This necessitates another prior (see code block).





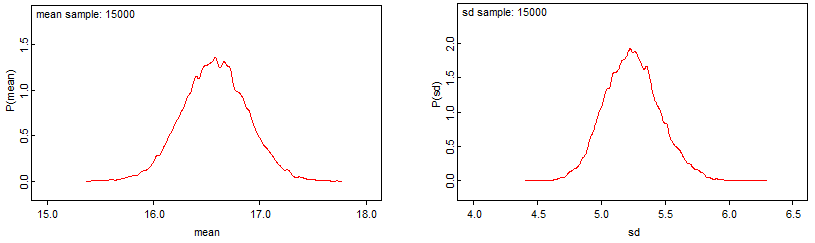
Again there is no burn in and no autocorrelation. **Thin** and **beg** of 1 are both appropriate.

5000 iterations are run and 15000 samples generated:

**mean sd MC\_err val2.5pc median val97.5pc start sample**

mean 16.58 0.3038 0.002248 15.98 16.58 17.18 1 15000

sd 5.245 0.2156 0.00178 4.847 5.238 5.693 1 15000



Conclusion

One straightforward way to assess the results is to compare the posteriors of the mean and standard deviation to frequentist estimates of the mean and standard deviation.

R says:   
mean of 16.57138  
sd of 5.23191

While I do not have the tools yet to graph these lines on the posterior plots, I can compare the values to the table values reported above. Both the mean and the standard deviation of the data are within 0.2% of the mean of the posteriors. Though I’m not certain I can articulate under what scenarios it would be otherwise, the MCMC method has converged to the mean and standard deviation of the data.

There are a few asterisks on this, we assumed a normal data model. Here is a density plot of the actual data. I do not yet understand what effect the specification of a normal data model for a non-normal data set will have, but it does seem like it should have an effect.

