Homework6\_TylerPoelking

Tyler Poelking

3/6/2018

salaries <- read.table("salaries.txt",header=T)  
#get salaries column  
sal <- salaries$Salary  
#n is total number entries  
n <- length(sal)  
  
#get unique number of departments. should be 8  
depts <- as.factor(salaries$Dept)  
deptsUniqCount <- length(unique(depts))  
  
#convert to numeric and extract counts of each dept  
deptsNum <- as.numeric(depts)  
deptsCounts <- as.vector(table(deptsNum))  
  
#create empty matrix to hold our data. this is in a format such that each col corresponds to a dept. Note: due to varying  
#salary amounts for each dept, not all data will be filled in. Regardless, this makes it easier for jags to process properly  
sal.Matrix <- matrix(c(NA), ncol=deptsUniqCount,nrow=max(deptsCounts))  
for (i in 1:deptsUniqCount){  
 temp <- sal[which(deptsNum == i)]  
 col <- c(temp, rep(NA, nrow(sal.Matrix)-length(temp)))  
 sal.Matrix[,i] <- col  
}  
  
# create a data list  
dataList = list('sal.Matrix' = sal.Matrix, 'deptsCounts' = deptsCounts, 'deptsUniqCount' =deptsUniqCount)  
  
  
# list of parameters to be monitored   
parameters <- c("theta", "mu", "prec.within", "prec.btw")  
  
  
  
# set initial values  
initsValues <- list("theta" = rep(80,deptsUniqCount), "mu"=80, "prec.within" = 1/100, "prec.btw"=1/100)  
  
# number of iteration for "tuning"   
adaptSteps <- 5000   
  
# number of iterations for "burn-in"   
burnInSteps <- 5000   
  
# number of chains to run  
nChains <- 2   
  
# total number of iterations to save  
numSavedSteps <- 5000   
  
# "thinning" (1 = keep every interation)  
thinSteps <- 1   
  
# iterations per chain  
ITER <- ceiling((numSavedSteps \* thinSteps )/ nChains)   
  
# -------------  
# Run JAGS  
# -------------  
  
# create, initialize, and adapt the model  
jagsModel <- jags.model("salaries\_model.txt", data = dataList, inits = initsValues, n.chains = nChains, n.adapt = adaptSteps)

## Compiling model graph  
## Resolving undeclared variables  
## Allocating nodes  
## Graph information:  
## Observed stochastic nodes: 297  
## Unobserved stochastic nodes: 11  
## Total graph size: 323  
##   
## Initializing model

# burn-in the algorithm  
update(jagsModel, n.iter = burnInSteps)  
  
# run algorithm to get interations for inference  
codaSamples <- coda.samples(jagsModel, variable.names = parameters, n.iter = ITER, thin = thinSteps)

Model Initial Values

initsValues

## $theta  
## [1] 80 80 80 80 80 80 80 80  
##   
## $mu  
## [1] 80  
##   
## $prec.within  
## [1] 0.01  
##   
## $prec.btw  
## [1] 0.01

Model Initial Values and the number of steps for tuning, burn-in, and for each chain were:

initsValues

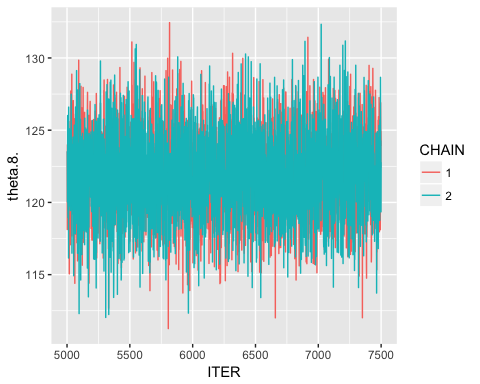
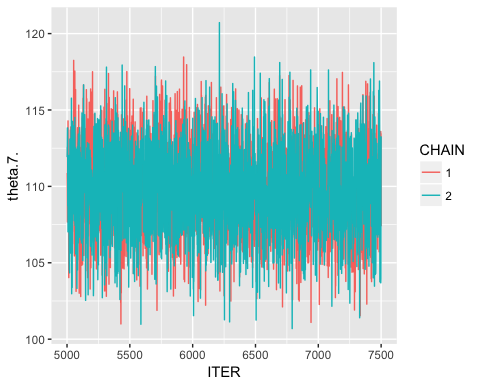
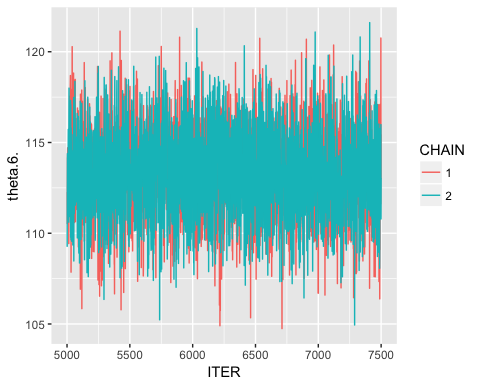
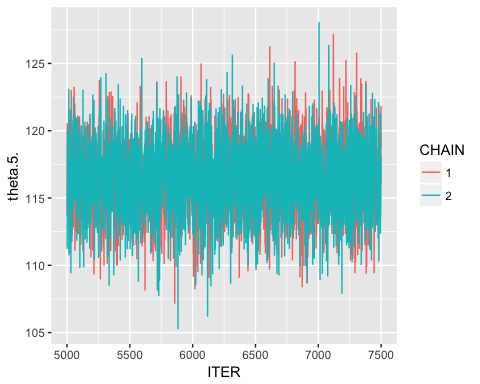
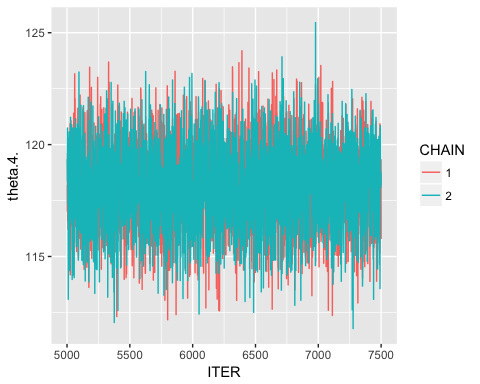
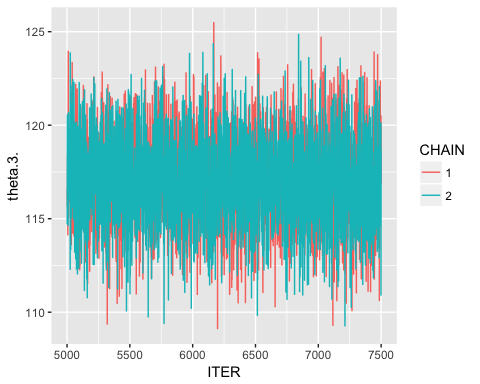
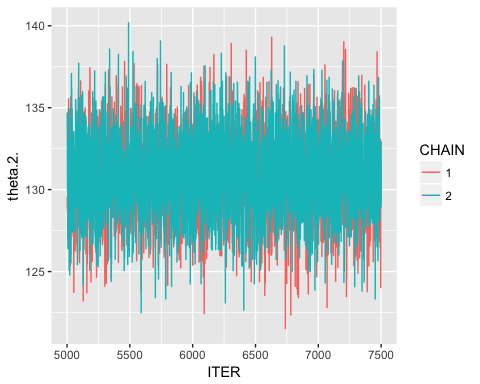
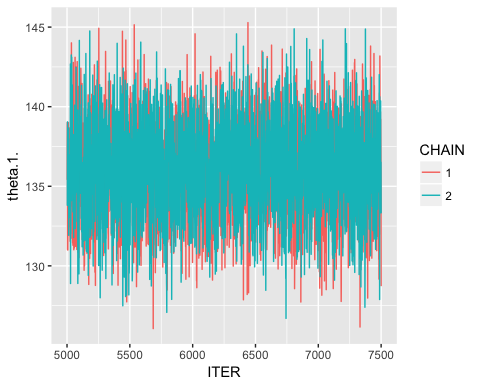
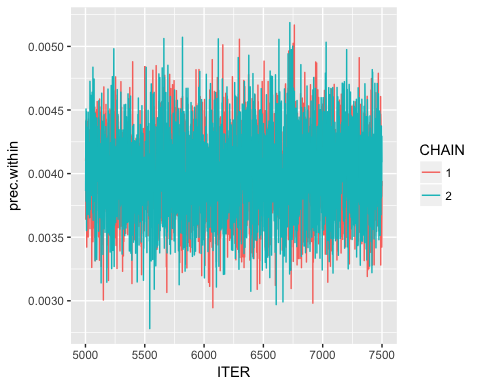
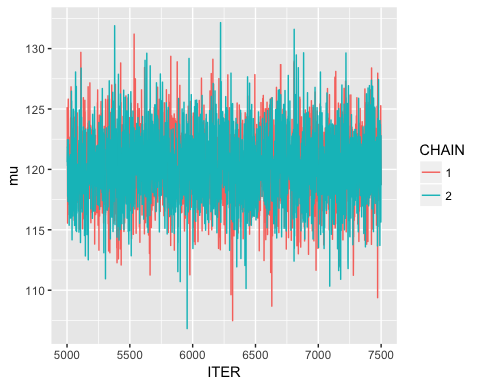
## $theta  
## [1] 80 80 80 80 80 80 80 80  
##   
## $mu  
## [1] 80  
##   
## $prec.within  
## [1] 0.01  
##   
## $prec.btw  
## [1] 0.01

print(c(adaptSteps, burnInSteps, ITER))

## [1] 5000 5000 2500

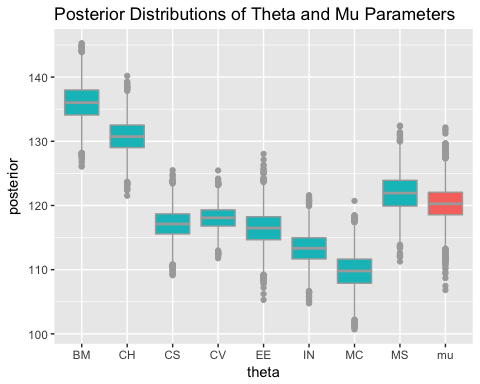
Checking for model convergence and trace plot creation

# -------------  
# Look at posterior samples  
# -------------  
  
# make a dataframe with the posterior samples  
mcmcChainDF <- data.frame(as.matrix(codaSamples, iters = T, chains = T))  
  
# create a vector with the variable names  
varNames <- names(mcmcChainDF)[3:(dim(mcmcChainDF)[2])]  
  
# number of variables  
nVars <- length(varNames)  
  
mcmcChainDF$CHAIN <- as.factor(mcmcChainDF$CHAIN)  
# construct trace plots  
par(ask = T)  
 for( i in 1:nVars ){  
 print(ggplot(mcmcChainDF, aes( x = ITER, y = mcmcChainDF[ ,varNames[i]])) +  
 geom\_line(aes(color = CHAIN)) +   
 labs(y = varNames[i]))  
 flush.console()  
 }



The trace plots above shows no evidence against converge because the trace plots are consitent around one value and no pattern seems to be present that has not repeated.

postDFreshape <- melt( mcmcChainDF, id.vars = "ITER", measure.vars = c("theta.1.", "theta.2.", "theta.3.", "theta.4.", "theta.5.", "theta.6.", "theta.7.", "theta.8.", "mu"))  
  
cols <- hue\_pal()(4)  
  
plot = ggplot(postDFreshape,   
 aes(x = variable, y = value )) +  
 geom\_boxplot( fill = c(rep(cols[3], deptsUniqCount),   
 cols[1]),  
 color = "darkgrey") +  
 ggtitle("Posterior Distributions of Theta and Mu Parameters") +   
 ylab( "posterior" ) +  
 xlab( "theta" ) +  
 scale\_x\_discrete( labels = c("BM","CH", "CS", "CV", "EE", "IN", "MC", "MS", "mu" ))   
  
plot

 At around 136k, the Biomedical department has the largest average salary of the 8 departments. Chemical comes in a fairly close second at just above 130k. Mechanical is the lowest at around 110k. The Average is about 120k dollars per year.

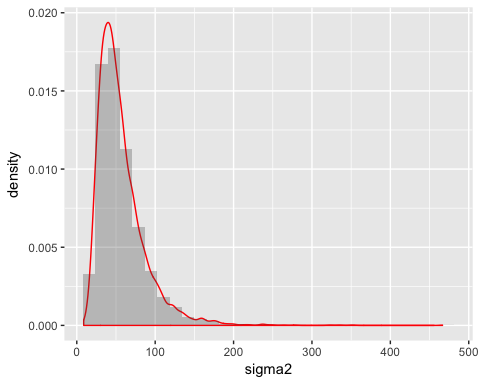
Numerically summarize how the variation in salaries within discipline compares to the total variation in salaries. Include a numerical summary of the uncertainty associated with your inference about the sources of variation in salaries

#use prec.within and prec.btw to calculate:  
#sigma^2 - variance within samples   
#tau^2 - variance between a sample:  
mcmcChainDF$sigma2 <- 1/mcmcChainDF$prec.btw  
mcmcChainDF$tau2 <- 1/mcmcChainDF$prec.within  
  
mcmcChainDF$propVar <- mcmcChainDF$sigma2 /  
 (mcmcChainDF$sigma2 +  
 mcmcChainDF$tau2)

We can find distributions of sigma^2 and tau^2, as well as the proportion of variance in the posterior distributions that is due to between-batch variance:

par(ask = F)  
 ggplot( mcmcChainDF, aes( x = sigma2 )) +  
 geom\_density( color = "red") +  
 geom\_histogram( aes(y = ..density.. ),  
 alpha = .3)

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.



mean( mcmcChainDF$sigma2 )

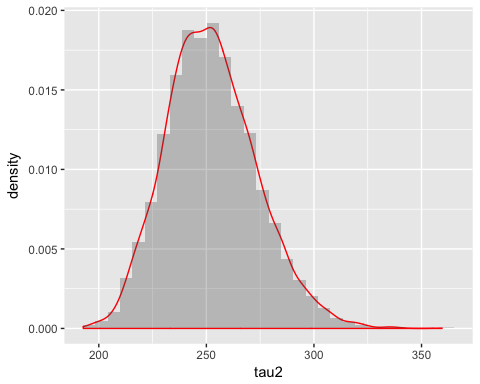
## [1] 56.44471

sd( mcmcChainDF$sigma2 )

## [1] 30.97915

par(ask = F)  
 ggplot( mcmcChainDF, aes( x = tau2 )) +  
 geom\_density( color = "red") +  
 geom\_histogram( aes(y = ..density.. ),  
 alpha = .3)

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.



mean( mcmcChainDF$tau2 )

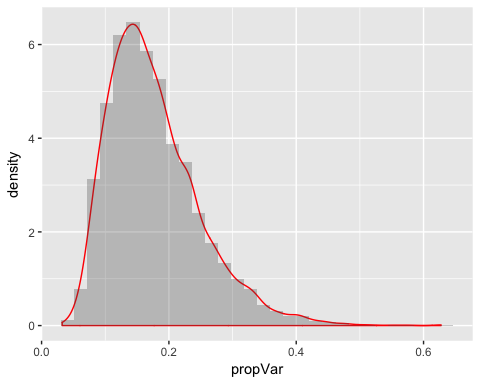
## [1] 252.4497

sd( mcmcChainDF$tau2 )

## [1] 21.07496

par(ask = F)  
 ggplot( mcmcChainDF, aes( x = propVar )) +  
 geom\_density( color = "red") +  
 geom\_histogram( aes(y = ..density.. ),  
 alpha = .3)

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.



mean( mcmcChainDF$propVar)

## [1] 0.1766795

sd( mcmcChainDF$propVar)

## [1] 0.07213793

The expected proportion of variation explained by salaries within discipline is .177. This means that on average 17.7% of the variance is explained by salaries within discipline while the rest of the variance is explained by the salaries between disciplines. The standard deviation of the proportions is 0.072, which provides an idea on the variablility we expect to see in the proportion of variability within discipline salary is covers (lower sd implying higher confidence).

What is the posterior probability that a new biomedical engineering faculty member makes more than $140K?

bm.post <- mcmcChainDF$theta.1.  
num\_bm = length(bm.post)  
probability <- length(which(bm.post > 140))/num\_bm  
print(probability)

## [1] 0.0882