Homework #3

Analysis of Arsenic in Rice Products

Antoine Baldassari

November 17, 2015

1. Standard conditionally-conjugate specification of the hierarchical model

1.1 Model specification

In this model specification, the i^{th} arsenic reading of group j, y_{ij} is normally-distributed, so that

$$y_{ij} \sim \mathcal{N}\left(\theta_j, \sigma^2\right)$$

Where θ_j is the mean arsenic reading for the rice products indexed by j. θ_j is normally distributed, centered at the population mean μ, with between-group variance τ^2:

$$\theta_j \sim \mathcal{N}\left(\mu, \tau^2\right)$$

We use conditionally-conjugate Normal and Inverse-Gamma priors on the hyperparameters:

$$\begin{aligned} 1/\sigma^2 &\sim \text{gamma } (\nu_0/2, \nu_0\sigma_0^2/2) \\ 1/\tau^2 &\sim \text{gamma } (\eta_0/2, \eta_0\tau_0^2/2) \\ \mu &\sim \text{normal } (\mu_0, \gamma_0^2) \end{aligned}$$

The full conditional distribution of the parameters can be found to be (from the book):

$$\begin{aligned} \left\{\theta_j \,|\, \sigma^2, y_{j,1}, \ldots, y_{j,n}\right\} &\sim \mathcal{N}\left(\frac{n_j\bar{y}_j/\sigma^2 + \mu/\tau^2}{n_j/\sigma^2 + 1/\tau^2}, \left[n_j/\sigma^2 + 1/\tau^2\right]^{-1}\right) \\ \left\{\mu \,|\, \theta_1, \ldots, \theta_m, \tau\right\} &\sim \mathcal{N}\left(\frac{m\bar{\theta}/\tau^2 + \mu_0/\gamma_0^2}{m/\tau^2 + 1/\gamma_0^2}, \left[m/\tau^2 + 1/\gamma_0^2\right]^{-1}\right) \\ \left\{1/\tau^2 \,|\, \theta_1, \ldots, \theta_m, \mu\right\} &\sim \mathcal{N}\left(\frac{\eta_0 + m}{2}, \frac{\eta_o\tau_0^2 + \sum\left(\theta_j - \mu\right)^2}{2}\right) \\ \left\{1/\sigma^2 \,|\, \boldsymbol{\theta}, y_1, \ldots, y_n\right\} &\sim \mathcal{N}\left(\frac{1}{2}\left[\nu_0 + \sum_{j=1}^{m} n_j\right], \frac{1}{2}\nu_0\sigma_0^2 + \sum_{j=1}^{m}\sum_{i=1}^{n_j}\left(y_{i,j} - \theta_j\right)^2\right) \end{aligned}$$

1.2 Main analyses

We pick relatively uninformative priors, centering μ around 1 with somewhat large within and between sample variances: $\sigma_0^2 = 10, \nu_0 = 1, \tau_0^2 = 10, \eta_0 = 1, \gamma_0^2 = 10$. The marginal distributions of $\theta_1, \ldots, \theta_m, \mu, \sigma^2$ and τ^2 can be obtained from the full condition distributions using a Monte-Carlo Markov-Chain algorithm, Gibbs sampling, which we implement in R as follows::

First, we input the dataset downloaded from Sakai, modified in Stata to have numeric codes for rice products categories.

```
library(foreign)
Y <- read.dta(file="arsenicrice2.dta")
```

We set the weakly informative prior values

```
n <-  nrow(Y)
nu0 <- 1; eta0 <- 1;
t20 <- 10;
mu0 <- 1;
g20 <- s20 <- var(Y$arsenic)
```

We set initial values for algorithm

```
m <- length(unique(Y$food_num)) #number of groups
n <- sv <- ybar <- rep(NA,m)
for (i in 1:m)
{
  n[i] <- sum(Y$food_num==i)
  sv[i] <- var(Y$arsenic[which(Y$food_num==i)])
  ybar[i] <- mean(Y$arsenic[which(Y$food_num==i)])
}
theta <- ybar; s2 <- mean(sv)
mu <- mean(theta); tau2 <- var(theta)
```

We create a Markov chain for each parameter by sequentially sampling from their posterior over 10,000 iterations. Elements are stored in the chain at the end of each iteration.

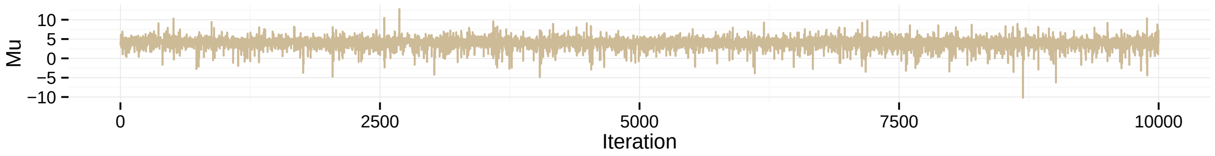
```
#Setup MCMC
set.seed(0808)
S <- 10000
THETA <- matrix(nrow=S, ncol=m)
OTH <- matrix(nrow=S, ncol=3)
ALL <- matrix(nrow=S, ncol=3+m)

#Run algorithm
for(i in 1:S)
{
  #Get new values for parameters
  for(j in 1:m) theta[j] <- newTheta(n[j], ybar[j], s2, tau2, mu)
  s2 <- newSigma2(m, n, nu0, s20, theta, Y)
  mu <- newMu(m, theta, tau2, g20)
  tau2 <- newTau2(m, eta0, t20, theta, mu)

  #Store in chain
  THETA[i,] <- theta
  OTH[i,] <- c(mu,s2,tau2)
  ALL[i,] <- c(theta,mu,s2,tau2)
}
```

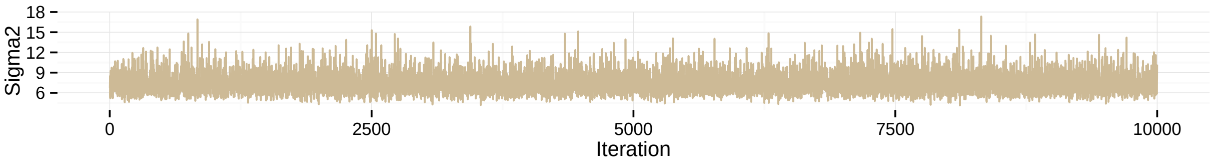
Where the functions updating the parameters follow the equations listed above:

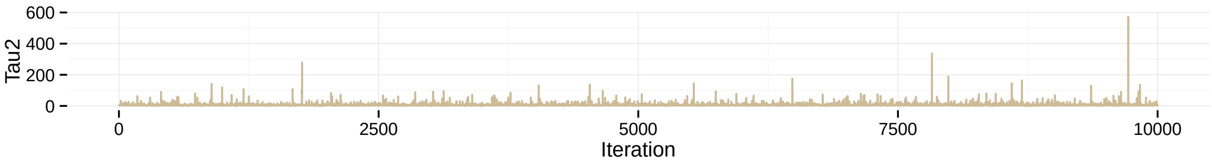
```r
newTheta <- function(n, ybar, s2, tau2, mu)
{
  v = 1/(n/s2 +1/tau2)
  e = v * (ybar*n/s2 +mu/tau2)
  new <- rnorm(1, e, sqrt(v))
  return(new)
}
newSigma2 <- function(m, n, nu0, s20, theta, Y)
{
  nun = nu0 + sum(n)
  ss <- nu0 * s20
  for(i in 1:m) ss = ss+sum((Y$arsenic[which(Y$food_num==i)] - theta[j])^2)
  sigma2 <- 1/rgamma(1, nun/2, ss/2)
  return(sigma2)
}
newMu <- function(m, theta, tau2, g20)
{
  v = 1/(m/tau2 + 1/g20)
  e = v *(m*mean(theta)/tau2 + mu0/g20)
  mu <- rnorm(1, e, v)
  return(mu)
}
newTau2 <- function(m, eta0, t20, theta, mu)
{
  etam = eta0 + m
  ss <- eta0*t20 + sum( (theta-mu) ^2 )
  tau2 <- 1/rgamma(1, etam/2, ss/2)
  return(tau2)
}
```

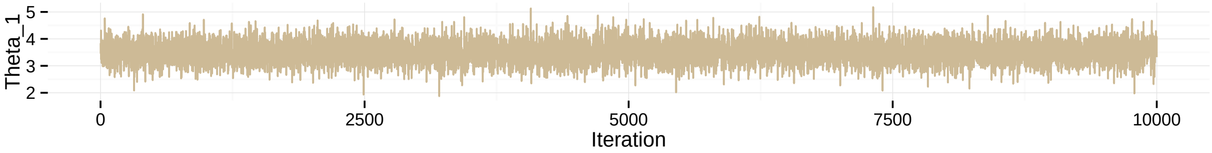
Before we go any further, we check that the MCMC model converged for all four statistics using ggplot2 (code used for μ repeated for other parameters):

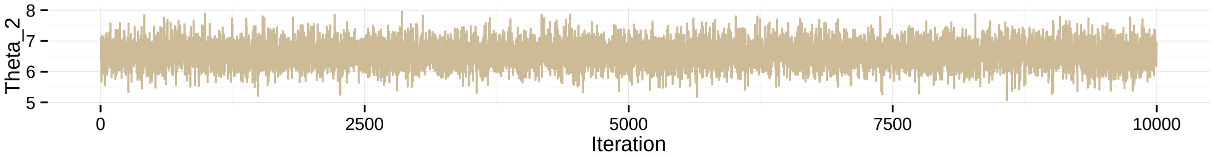
```r
library(ggplot2)
graphdata <- data.frame(
              "Iteration"=c(1:S), "Mu"=OTH[,1], "Sigma2"=OTH[,2], "Tau2"=OTH[,3],
              "Theta_1" = THETA[,1], "Theta_2" = THETA[,2], "Theta_3" = THETA[,3],
              "Theta_4" = THETA[,4], "Theta_5" = THETA[,5])

ggplot(graphdata,aes(x=Iteration,y=Mu)) +
theme_minimal(base_family = "") + geom_line(colour="wheat3")
```

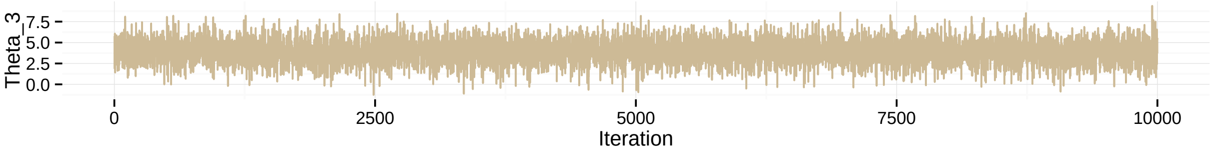


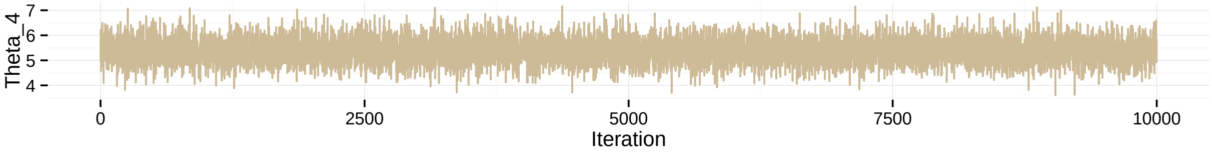


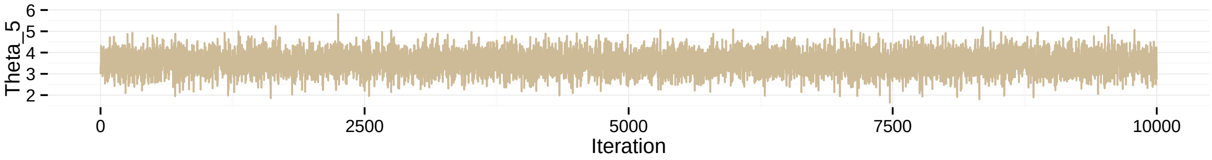












We conclude from the graphs that convergence was achieved for all parameters.

1.3 Algorithm output

The estimated median values and 95% credible intervals for the parameters are as follow:

```
for(i in 1:length(ALL[1,])) print(round(unname(
  quantile(ALL[,i], probs=c(0.025, 0.5, 0.975))
  ),3))
```

Parameter	Credible Lower 95%	Median	Credible Upper 95%
θ_1 (Basmati)	2.810	3.569	4.308
θ_2 (Non-Basmati)	5.790	6.533	7.305
θ_3 (Beverage)	1.869	4.231	6.448
θ_4 (Cakes)	4.486	5.370	6.301
θ_5 (Cereal)	2.705	3.610	4.507
μ	3.193	4.697	6.182
σ^2	5.109	7.027	10.573
τ^2	0.693	2.251	12.920

1.4 Sensitivity analyses

Evaluation of sensitivity to priors: we try three separate scenarios each tuning prior distribution of parameters:

1. Large expected μ (Prior expectation of mad levels of arsenic)
2. Large σ^2 and ν_0 (High variability within products)
3. Large τ^2 and η_0 (High variability between products)

Scenario 1:

Parameter	Credible Lower 95%	Median	Credible Upper 95%
θ_1 (Basmati)	2.706	3.488	4.265
θ_2^2 (Non-Basmati)	5.902	6.671	7.460
$\theta_3(Beverage)$	0.643	3.774	6.932
$\theta_4(Cakes)$	4.511	5.452	6.429
θ_5 (Cereal)	2.548	3.496	4.456
μ	2.548	3.496	4.456
σ^2	88.793	100	111
τ^2	3021.927	8427	37962

Scenario 2:

Parameter	Credible Lower 95%	Median	Credible Upper 95%
θ_1 (Basmati)	1.237	4.396	7.069
θ_2^2 (Non-Basmati)	2.811	5.401	8.559
$\theta_3(Beverage)$	0.278	4.790	9.023
$\theta_4(Cakes)$	1.931	4.960	8.221
θ_5 (Cereal)	1.137	4.531	7.508
μ	2.42	4.84	7.19
σ^2	182	215	256
τ^2	0.399	1.876	22.3

Scenario 3:

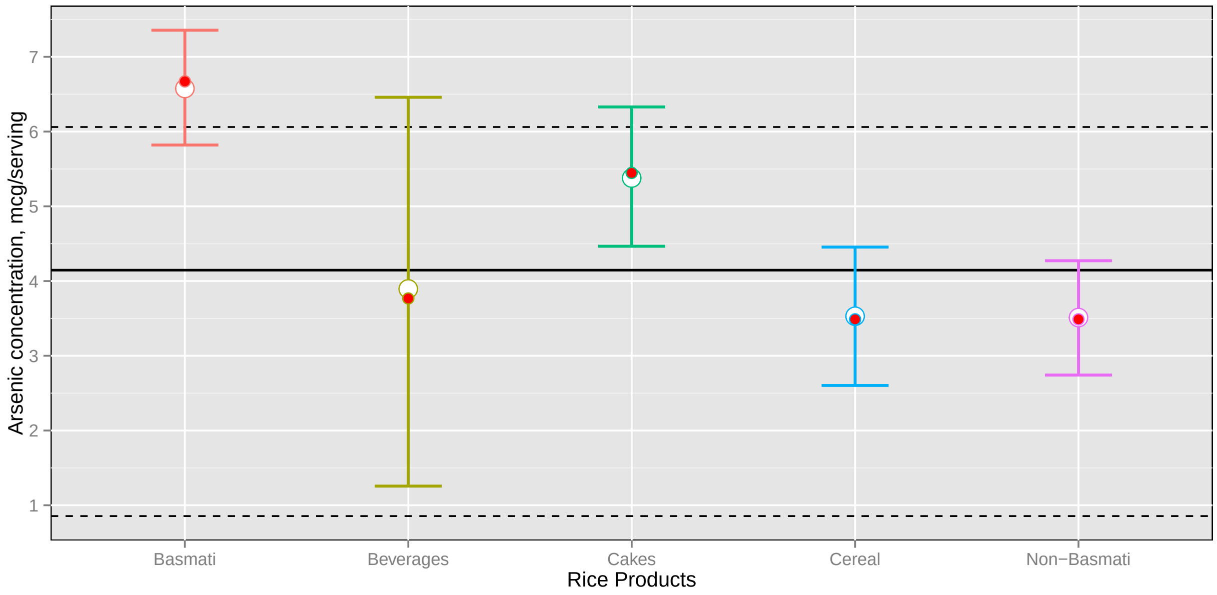
Parameter	Credible Lower 95%	Median	Credible Upper 95%
θ_1 (Basmati)	2.707	3.478	4.268
θ_2^2 (Non-Basmati)	5.907	6.674	7.451
$\theta_3(Beverage)$	0.676	3.739	6.9023
$\theta_4(Cakes)$	4.504	5.447	6.419
θ_5 (Cereal)	2.547	3.494	4.452
μ	-5.302	4.845	5.00
σ^2	5.197	7.327	11.37
τ^2	223	288	384

We observe that excessively large prior expectations of μ will drive up estimates of the within- and between-group variances but will have little effect on the magnitude of the estimates of within-groupmean estimates (although the precision may be negatively affected for groups with realtively few observations). A large prior within-sample variance will bring posterior within-group means closer to μ, as could be expected since the posterior estimates need to become more conservative. Increasing prior between-sample variance appears to drive up uncertainty on μ and bring it closer to 0, without however having a notable impact on the rest of the model.

1.5 Results presentation

Non-Basmati rice had the highest arsenic concentration, at an estimated 6.7 mcg/serving. Rice cakes came second, at 5.4 mcg/serving, and non-Basmati and rice cereal had comparatively low amounts, slightly below 3.5 mcg/serving. There lacked data to reliably evaluate arsenic concentration in rice beverages, whose 3.8 mcg/serving estimate was particulary imprecise (95% CI=0.64, 6.93). Posterior median estimates and observed mean concentrations of arsenic are presented by product type in the following graph. Markers are θ estimates with 95% credible interval lines; horizontal lines are the median estimate of μ (solid) and corresponding 95% credibal interval (dashed, like my hopes and dreams).

```
qmat=apply(THETA[,1:5],2,quantile,probs=c(0.025,.5,0.975))
mu_ci = quantile(OTH[,1], probs=(c(0.025, 0.5, 0.975)))
res <- data.frame("Rice"=c("Non-Basmati", "Basmati", "Beverages", "Cakes", "Cereal"),
                  "l95"=qmat[1,], "median"=qmat[2,], "u95"=qmat[3,], "mean"=ybar)
g <- ggplot(res, aes(x = Rice, group=Rice, colour=Rice)) +
  labs(x="Rice Products", y="Arsenic concentration, mcg/serving") +
  theme(legend.position="none", panel.background =element_rect(colour = "black")) +
  scale_y_continuous(breaks=seq(0, 7.5, 1)) +
  geom_hline(aes(yintercept=c(mu_ci[2])), size=0.7) +
  geom_hline(aes(yintercept=c(mu_ci[1])), linetype="dashed") +
  geom_hline(aes(yintercept=c(mu_ci[3])), linetype="dashed") +
  geom_errorbar(aes(ymin=l95, ymax=u95), width=.3, size=0.8) +
  geom_point(aes(y=median), fill="white", shape=21, size=5)  +
  geom_point(aes(y=mean), fill="red", shape=21, size=3)
g
```



2. Parameter-expanded specification of the hierarchical model

2.1 Model specification

Under this model specification, instead of group means we are interested in differences between groups and the population average μ, which is given by η_j for the group j, so that (under the prior belief that all groups will have equal mean):

$$\begin{aligned} y_{ij} &\sim \mathcal{N}\left(\mu + \xi\eta_j, \sigma_y^2\right) \\ \eta_i &\sim \mathcal{N}\left(0, \sigma_\eta^2\right) \end{aligned}$$

Word's on the street that well-behaved conditionally-conjugate specifications for the distributions of ξ and σ_η^2 are:

$$\begin{aligned} \xi &\sim \mathcal{N}\left(0, 1\right) \\ 1/\sigma_\eta^2 &\sim \text{gamma}\left(\frac{\omega_0}{2}, \frac{\omega_0\sigma_{\eta0}^2}{2}\right) \\ 1/\sigma_y^2 &\sim \text{gamma}\left(\frac{\nu_0}{2}, \frac{\nu_0\sigma_{y0}^2}{2}\right) \end{aligned}$$

The prior distribution of the population mean is still $\mu \sim \mathcal{N}\left(\mu_0, \gamma_0^2\right)$. We set out to find full conditionals: Reparametrizing the full conditionals in exercise 1 easily yields what we need:

Note that each of the N observation in the data supports, $\mu = y_{ij} - xi\eta_j$ given known ξ and η_j's, so that, summing over this expression and weighting it against the prior yields:

$$p\left(\mu, \sigma_y^2, \sigma_\eta^2, \xi, \eta_1, \ldots, \eta_m \,|\, \mathbf{y}\right) \,\alpha\, \mathcal{N}\left(\frac{\frac{\sum\limits_j\sum\limits_i y_{ij}-\xi\eta_j}{\sigma_y^2} + \frac{\mu_0}{\gamma_0^2}}{\frac{N}{\sigma_y^2} + \frac{1}{\gamma_0^2}}, \frac{1}{\frac{N}{\sigma_y^2} + \frac{1}{\gamma_0^2}}\right)$$

Posterior on the variances can be similarly rewritten. The sum of squares for $\mathbf{y}$ given μ, η_j and ξ is, of course, $\sum\limits_j\sum\limits_i y_{ij} - \mu - \xi\eta_j$, and the sum of squares of $\boldsymbol{\eta}$ given its $\mathbb{E} = 0$ is simply $\sum\limits_j \eta_j^2$

$$\begin{aligned} p\left(\sigma_y^2 \,|\, \mathbf{y}, \mu, \sigma_\eta^2, \xi, \eta_1, \ldots, \eta_m\right) \,\alpha\, \mathcal{N}\left(\frac{\nu_0 + N}{2}, \frac{1}{2}\left(\nu_0\sigma_{y0}^2 + \sum_j\sum_i y_{ij} - \mu - \xi\eta_j\right)^2\right) \\ p\left(\sigma_\eta^2 \,|\, \mathbf{y}, \mu, \sigma_y^2, \xi, \eta_1, \ldots, \eta_m\right) \,\alpha\, \mathcal{N}\left(\frac{\omega_0 + m}{2}, \frac{1}{2}\left(\omega_0\sigma_{\eta_0}^2 + \sum_j \eta_j^2\right)\right) \end{aligned}$$

Getting full conditionals is trickier and requires recognizing that $\mathbb{E}\left(aX\right) = a\mathbb{E}\left(X\right)$ and $Var\left(aX\right) = a^2 Var\left(X\right)$, so that:

$$p\left(\eta_j \,|\, \mathbf{y}, \mu, \sigma_y^2, \sigma_\eta^2, \xi\right) \,\alpha\, \mathcal{N}\left(\frac{\frac{\xi \sum\limits_i y_{ij} - \mu}{\sigma_y^2}}{\frac{n_j \xi^2}{\sigma_y^2} + \frac{1}{\sigma_\eta^2}}, \frac{1}{\frac{n_j \xi^2}{\sigma_y^2} + \frac{1}{\sigma_\eta^2}}\right)$$

and

$$p\left(\xi \,|\, \mathbf{y}, \mu, \sigma_y^2, \sigma_\eta^2, \eta_1, \ldots, \eta_m\right) \,\alpha\, \mathcal{N}\left(\frac{\frac{\sum\limits_j \eta_j \sum\limits_i (y_{ij} - \mu)}{\sigma_y^2}}{\frac{\sum\limits_j n_j \eta_j^2}{\sigma_y^2} + 1}, \frac{1}{\frac{\sum\limits_j n_j \eta_j^2}{\sigma_y^2} + 1}\right)$$

2.2 Analyses

Similarly to exercise 1, we pick the priors $\sigma_{y0}^2 = 10, \nu_0 = 1, \omega_0 = 1, \sigma_{0\eta}^2 = 10$ and $\gamma_0^2 = 10$, and proceed with Gibbs sampling:

We read the data

```
library(foreign)
library(hdrcde)
```

```
## Loading required package: mvtnorm
## hdrcde 3.1 loaded
```

```
data <- read.dta(file="arsenicrice2.dta")
Y <- read.dta(file="arsenicrice2.dta")
```

We set prior values

```
n <-  nrow(Y)
nu0 <- 1; omega0 <- 1;
s2_eta0 <- 10; s2_y0 <- 10
mu0 <- mean(Y$arsenic);
g20 <- var(Y$arsenic)
```

We setup the MCMC

```
#Setup starting values
m <- length(unique(Y$food_num)) #number of groups
n <- sv_y <- ybar <- rep(NA,m) #create empty vectors for group descriptions
for (i in 1:m)
{
  n[i] <- sum(Y$food_num==i)
  sv_y[i] <- var(Y$arsenic[which(Y$food_num==i)])
  ybar[i] <-  mean(Y$arsenic[which(Y$food_num==i)])
}
eta <- ybar - mean(Y$arsenic); s2_y <- mean(sv_y)
```

```
mu <- mean(Y$arsenic); s2_eta <- var(eta)
xi <- 0

#Setup MCMC
set.seed(0808)
S <- 10000
THETA <- matrix(nrow=S, ncol=m)
RES <- matrix(nrow=S, ncol=4)
ALL <- matrix(nrow=S, ncol=4+m)

#Setup MCMC
set.seed(0808)
S <- 10000
ETA <- matrix(nrow=S, ncol=m)
RES <- matrix(nrow=S, ncol=4)
ALL <- matrix(nrow=S, ncol=4+m)
```

Our updating functions correspond to the full conditionals derived above:

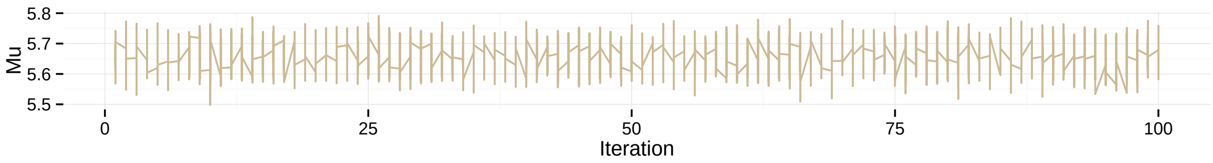
```
#FUNCTIONS
newS2eta <- function(m, s2_eta, s2_eta0, eta, omega0)
{
  nomega = omega0 + m
  ss <- s2_eta0*omega0 + sum( eta^2 )
  s2_eta <- 1/rgamma(1, nomega/2, ss/2)
  return(s2_eta)
}
newS2y <- function(nu0, n, s2_y0, Y, m, eta, xi, mu)
{
  nun = nu0 + sum(n)
  ss = nu0 * s2_y0
  for(i in 1:m) ss = ss+sum((Y$arsenic[which(Y$food_num==i)] - mu - xi*eta[j])^2)
  s2_y <- 1/rgamma(1, nun/2, ss/2)
  return(s2_y)
}
newMu <- function(Y, xi, eta, s2_y, g20, mu0, n)
{
  v = 1/(sum(n)/s2_y + 1/g20)
  sumd = 0
  for(i in 1:m) sumd = sumd + sum(Y$arsenic[which(Y$food_num==i)] - xi*eta[i])
  e = v* sumd/s2_y + mu0/g20
  mu = rnorm(1, e, v)
  return(mu)
}
newEta <- function(xi, mu, Y, n, s2_y, s2_eta)
{
  v = 1/(n*xi^2/s2_y + 1/s2_eta)
  e = v * xi/s2_y*sum(Y$arsenic[which(Y$food_num==i)] - mu)
  eta = rnorm(1, e, v)
  return(eta)
}
newXi <- function(eta, m, Y, n, mu, s2_y)
{
  v = 1/(sum(n*eta^2)/s2_y + 1)
  sumd = 0
  for(i in 1:m) sumd = sumd + sum((Y$arsenic[which(Y$food_num==i)] - mu)*eta[i])
  e = v * sumd/s2_y
  xi = rnorm(1, e, v)
  return(xi)
}
```

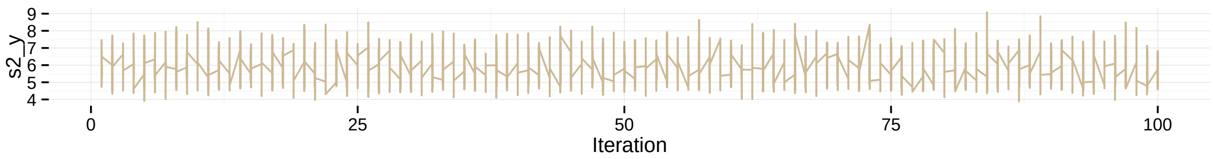
We run the MCMC algorithm:

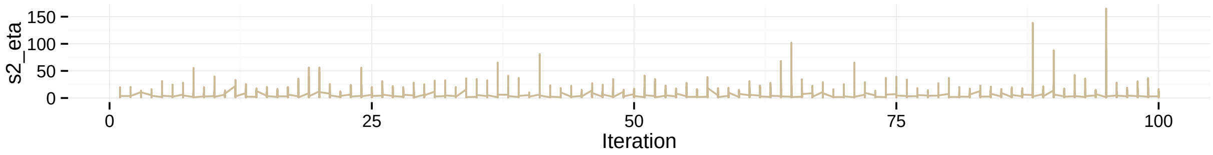
```
#RUN MCMC
for(i in 1:S)
{
  #Get new values for parameters
  for(j in 1:m) theta[j] <- newEta(xi, mu, Y, n, s2_y, s2_eta)
  s2_y <- newS2y(nu0, n, s2_y0, Y, m, eta, xi, mu)
  s2_eta <- newS2eta(m, s2_eta, s2_eta0, eta, omega0)
  mu <- newMu(Y, xi, eta, s2_y, g20, mu0, n)
  xi <- newXi(eta, m, Y, n, mu, s2_y)

  #Store in chain
  ETA[i,] <- eta
  RES[i,] <- c(mu,s2_y, s2_eta, xi)
  ALL[i,] <- c(eta, mu,s2_y, s2_eta, xi)
}
```

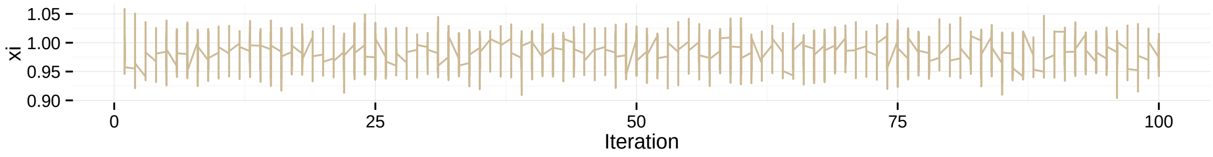
Again, before we get too psyched about results, we check MCMC convergence criteria:

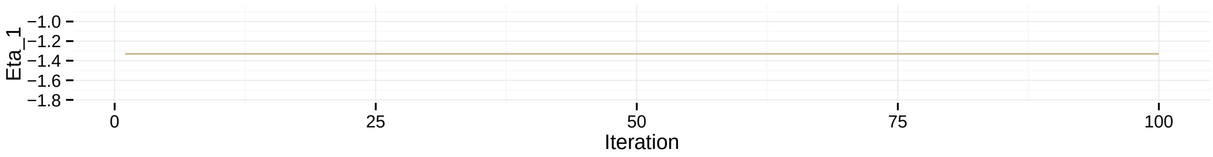
```
##            [,1]     [,2]      [,3]      [,4]      [,5]     [,6]     [,7]
##  [1,] -1.330427 1.853247 -1.051515 0.6286932 -1.330682 5.667968 4.903008
##  [2,] -1.330427 1.853247 -1.051515 0.6286932 -1.330682 5.683804 6.006653
##  [3,] -1.330427 1.853247 -1.051515 0.6286932 -1.330682 5.652164 6.628165
##  [4,] -1.330427 1.853247 -1.051515 0.6286932 -1.330682 5.645594 6.068511
##  [5,] -1.330427 1.853247 -1.051515 0.6286932 -1.330682 5.620257 5.474539
##  [6,] -1.330427 1.853247 -1.051515 0.6286932 -1.330682 5.641334 6.338744
##  [7,] -1.330427 1.853247 -1.051515 0.6286932 -1.330682 5.642847 6.180044
##  [8,] -1.330427 1.853247 -1.051515 0.6286932 -1.330682 5.688390 5.774271
##  [9,] -1.330427 1.853247 -1.051515 0.6286932 -1.330682 5.718387 5.871248
## [10,] -1.330427 1.853247 -1.051515 0.6286932 -1.330682 5.613030 6.048364
## [11,] -1.330427 1.853247 -1.051515 0.6286932 -1.330682 5.589698 5.280218
## [12,] -1.330427 1.853247 -1.051515 0.6286932 -1.330682 5.621325 5.720858
## [13,] -1.330427 1.853247 -1.051515 0.6286932 -1.330682 5.691593 5.509906
## [14,] -1.330427 1.853247 -1.051515 0.6286932 -1.330682 5.590404 6.907127
## [15,] -1.330427 1.853247 -1.051515 0.6286932 -1.330682 5.657124 5.480785
## [16,] -1.330427 1.853247 -1.051515 0.6286932 -1.330682 5.678040 6.070337
## [17,] -1.330427 1.853247 -1.051515 0.6286932 -1.330682 5.711129 5.536881
## [18,] -1.330427 1.853247 -1.051515 0.6286932 -1.330682 5.708383 5.813949
## [19,] -1.330427 1.853247 -1.051515 0.6286932 -1.330682 5.647771 6.878466
## [20,] -1.330427 1.853247 -1.051515 0.6286932 -1.330682 5.605647 6.191938
##            [,8]      [,9]
##  [1,]  4.473581 0.9799366
##  [2,]  3.738244 0.9552652
##  [3,] 11.004107 0.9412060
##  [4,]  5.327483 0.9675979
##  [5,]  1.867751 0.9843936
##  [6,]  3.104652 0.9590547
##  [7,]  6.200617 0.9802011
##  [8,]  2.002595 1.0020117
##  [9,]  3.204125 0.9694308
## [10,]  3.417037 0.9828693
## [11,]  6.280930 0.9810677
## [12,] 22.359658 0.9988978
## [13,]  8.994325 0.9852938
## [14,]  2.123587 0.9948622
## [15,]  3.909016 0.9885898
## [16,]  2.068103 0.9843450
## [17,]  3.012525 0.9837987
## [18,]  3.080198 0.9811986
## [19,]  8.931096 1.0101091
## [20,] 12.275305 0.9794007
```

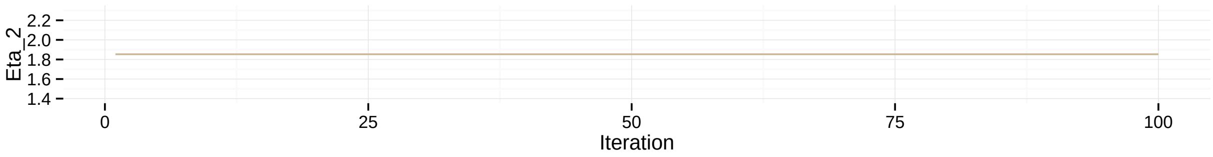


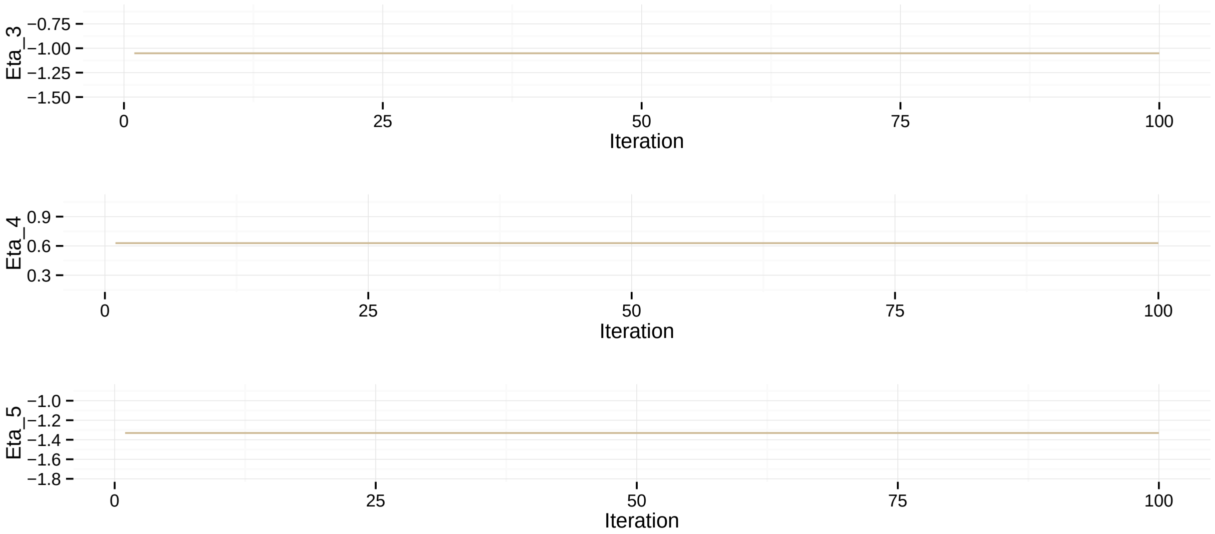












2.3 Algorithm output

2.4 Sensitivity analyses

3. Conditionally-conjugate specification of the hierarchical model with group-specific variances

3.1 Model specification

3.2 Analyses

3.3 Algorithm output

3.4 Sensitivity analyses